Lấy dữ liệu từ File Excel

```
import pandas as pd
file_path = "C:/Users/ADMIN KH/Downloads/TEST_Ad_table.csv"
df = pd.read_csv(file_path)
df.head()
```

```
C:\Users\ADMIN KH\AppData\Local\Temp\ipykernel_17248\2055285933.py:3:
DtypeWarning: Columns (13) have mixed types. Specify dtype option on
import or set low_memory=False.
  df = pd.read_csv(file_path)
```

```
     Maker Genmodel Genmodel_ID   Adv_ID  Adv_year  Adv_month   Color
\
0  Bentley   Arnage        10_1  10_1$$1      2018          4  Silver

1  Bentley   Arnage        10_1  10_1$$2      2018          6    Grey

2  Bentley   Arnage        10_1  10_1$$3      2017         11    Blue

3  Bentley   Arnage        10_1  10_1$$4      2018          4   Green

4  Bentley   Arnage        10_1  10_1$$5      2017         11    Grey


   Reg_year Bodytype Runned_Miles Engin_size    Gearbox Fuel_type
Price  \
0    2000.0   Saloon        60000       6.8L  Automatic    Petrol
21500
1    2002.0   Saloon        44000       6.8L  Automatic    Petrol
28750
2    2002.0   Saloon        55000       6.8L  Automatic    Petrol
29999
3    2003.0   Saloon        14000       6.8L  Automatic    Petrol
34948
4    2003.0   Saloon        61652       6.8L  Automatic    Petrol
26555

   Seat_num  Door_num
0       5.0       4.0
1       5.0       4.0
2       5.0       4.0
3       5.0       4.0
4       5.0       4.0
```

Tên trường	Ý nghĩa
`Maker`	Hãng sản xuất xe, ví dụ: *Bentley*
`Genmodel`	Dòng xe, ví dụ: *Arnage*
`Genmodel_ID`	Mã dòng xe (mã hóa dòng xe, không phải tên lặp)

Tên trường	Ý nghĩa
`Adv_ID`	ID quảng cáo xe, ví dụ: *10_1$$1*
`Adv_year`	Năm đăng quảng cáo, ví dụ: *2018*
`Adv_month`	Tháng đăng quảng cáo, ví dụ: *4*
`Color`	Màu xe, ví dụ: *Silver*, *Grey*, *Blue*
`Reg_year`	Năm đăng ký xe lần đầu, ví dụ: *2000*
`Bodytype`	Kiểu thân xe, ví dụ: *Saloon* (xe sedan), *Hatchback*, *SUV*
`Runned_Miles`	Số dặm đã chạy, ví dụ: *60000*
`Engin_size`	Dung tích động cơ, ví dụ: *6.8L*
`Gearbox`	Hộp số, ví dụ: *Automatic*, *Manual*
`Fuel_type`	Loại nhiên liệu, ví dụ: *Petrol*, *Diesel*
`Price`	Giá bán xe, ví dụ: *21500*, đơn vị có thể là USD
`Seat_num`	Số ghế ngồi, ví dụ: *5*
`Door_num`	Số cửa xe, ví dụ: *4*

```
df.describe()
```

```
            Adv_year      Adv_month       Reg_year       Seat_num  \
count  268255.000000  268255.000000  268248.000000  261781.000000
mean     2018.127778       5.626143    2012.708430       4.904306
std         0.747476       2.091577       4.465705       0.877934
min      2012.000000       1.000000    1900.000000       1.000000
25%      2018.000000       4.000000    2010.000000       5.000000
50%      2018.000000       5.000000    2014.000000       5.000000
75%      2018.000000       7.000000    2016.000000       5.000000
max      2021.000000      33.000000    2019.000000      17.000000

            Door_num
count  263702.000000
mean        4.371594
std         1.009339
min         0.000000
25%         4.000000
50%         5.000000
75%         5.000000
max         7.000000
```

LÀM SẠCH DỮ LIỆU

```
# Đếm số giá trị null theo từng cột
df_null_counts = df.isnull().sum()

# Kết hợp kiểu dữ liệu và số giá trị null thành một bảng
df_overview = pd.DataFrame({
    'Dtype': df.dtypes,
    'Null Count': df_null_counts,
    'Null Percent (%)': (df_null_counts / len(df)) * 100
```

```
}).sort_values(by='Null Count', ascending=False)

display(df_overview)
```

	Dtype	Null Count	Null Percent (%)
Color	object	21875	8.154554
Seat_num	float64	6474	2.413375
Door_num	float64	4553	1.697266
Engin_size	object	2064	0.769417
Runned_Miles	object	1055	0.393283
Bodytype	object	954	0.355632
Fuel_type	object	409	0.152467
Gearbox	object	167	0.062254
Reg_year	float64	7	0.002609
Adv_month	int64	0	0.000000
Adv_year	int64	0	0.000000
Adv_ID	object	0	0.000000
Genmodel	object	0	0.000000
Maker	object	0	0.000000
Genmodel_ID	object	0	0.000000
Price	object	0	0.000000

CẦN LÀM SẠCH DỮ LIỆU

1. Chuyển đổi kiểu dữ liệu (Data Type Cleaning)

Trường	Hiện tại	Cần chuyển thành
`Runned_Miles`	Dạng chuỗi (`object`), ví dụ: `"60,000"`	Số nguyên (`int`) hoặc `float`
`Price`	Dạng chuỗi, ví dụ: `"21,500"`	Số nguyên (`int`) hoặc `float`
`Engin_size`	Dạng chuỗi, ví dụ: `"6.8L"`	Tách `"L"` → chuyển về `float`: `6.8`

2. Xử lý dữ liệu thiếu (Missing Values)

Trường	Tỷ lệ thiếu	Chiến lược xử lý
`Color`	~8.15%	Thay bằng giá trị phổ biến nhất (mode)
`Runned_Miles`	~0.39%	Bỏ các bản ghi này (rất ít, không ảnh hưởng nhiều)
`Engin_size`	~0.77%	Tách phần số và điền giá trị thiếu bằng trung vị (median)
`Seat_num`	~2.41%	Điền giá trị trung bình theo từng `Genmodel`
`Door_num`	~1.70%	Điền giá trị trung bình theo `Genmodel` hoặc `Bodytype`

3. Loại bỏ giá trị bất thường

- Adv_Month có giá trị 13-17-33 => cần phải loại bỏ

```
import numpy as np
```

```
# 1. Làm sạch kiểu dữ liệu

df['Runned_Miles'] = (
    df['Runned_Miles']
    .replace(',', '', regex=True)
    .replace(r'[^\d.]', '', regex=True)
    .replace('', np.nan)
    .astype(float)
)

df['Price'] = (
    df['Price']
    .replace(',', '', regex=True)
    .replace(r'[^\d.]', '', regex=True)
    .replace('', np.nan)
    .astype(float)
)

df['Engin_size'] = (
    df['Engin_size']
    .replace(r'[^\d.]', '', regex=True)
    .replace('', np.nan)
    .astype(float)
)

# 2. Xử lý dữ liệu thiếu

most_common_color = df['Color'].mode()[0]
df['Color'] = df['Color'].fillna(most_common_color)

df = df[df['Runned_Miles'].notnull()]

df['Engin_size'] = df['Engin_size'].fillna(df['Engin_size'].median())

df['Seat_num'] = df.groupby('Genmodel')['Seat_num'].transform(lambda
x: x.fillna(x.mean()))

df['Door_num'] = df.groupby('Genmodel')['Door_num'].transform(lambda
x: x.fillna(x.mean()))
df['Door_num'] = df.groupby('Bodytype')['Door_num'].transform(lambda
x: x.fillna(x.mean()))

df['Price'] = df['Price'].fillna(df['Price'].median())

df['Bodytype'] = df['Bodytype'].fillna(df['Bodytype'].mode()[0])

df['Door_num'] = df['Door_num'].fillna(df['Door_num'].mean())

df['Fuel_type'] = df['Fuel_type'].fillna(df['Fuel_type'].mode()[0])
df['Gearbox'] = df['Gearbox'].fillna(df['Gearbox'].mode()[0])
```

```
df['Seat_num'] = df['Seat_num'].fillna(df['Seat_num'].mean())


# Kiểm tra kết quả
cleaning_result = df[['Runned_Miles', 'Price', 'Engin_size', 'Color',
'Seat_num', 'Door_num']].info()

# Đếm số giá trị null theo từng cột
df_null_counts = df.isnull().sum()

# Kết hợp kiểu dữ liệu và số giá trị null thành một bảng
df_overview = pd.DataFrame({
    'Dtype': df.dtypes,
    'Null Count': df_null_counts,
    'Null Percent (%)': (df_null_counts / len(df)) * 100
}).sort_values(by='Null Count', ascending=False)

display(df_overview)
```

	Dtype	Null Count	Null Percent (%)
Maker	object	0	0.0
Genmodel	object	0	0.0
Genmodel_ID	object	0	0.0
Adv_ID	object	0	0.0
Adv_year	int64	0	0.0
Adv_month	int64	0	0.0
Color	object	0	0.0
Reg_year	float64	0	0.0
Bodytype	object	0	0.0
Runned_Miles	float64	0	0.0
Engin_size	float64	0	0.0
Gearbox	object	0	0.0
Fuel_type	object	0	0.0
Price	float64	0	0.0
Seat_num	float64	0	0.0
Door_num	float64	0	0.0

KHÁM PHÁ DỮ LIỆU

Phân tích dữ liệu khám phá (EDA) được sử dụng để phân tích và điều tra các tập dữ liệu và tóm tắt các đặc điểm chính của chúng, thường sử dụng các phương pháp trực quan hóa dữ liệu

```
df.describe()
```

	Adv_year	Adv_month	Reg_year	Runned_Miles	\
count	267200.000000	267200.000000	267200.000000	2.672000e+05	
mean	2018.128084	5.625647	2012.705341	4.812342e+04	
std	0.747649	2.090895	4.460922	4.185477e+04	
min	2012.000000	1.000000	1900.000000	0.000000e+00	

```
25%      2018.000000       4.000000    2010.000000  1.410100e+04
50%      2018.000000       5.000000    2014.000000  3.920200e+04
75%      2018.000000       7.000000    2016.000000  7.500000e+04
max      2021.000000      33.000000    2019.000000  6.363342e+06
```

```
          Engin_size         Price       Seat_num       Door_num
count  267200.000000  2.672000e+05  267200.000000  267200.000000
mean        1.962738  1.472051e+04       4.903363       4.369246
std         9.344411  3.212282e+04       0.885365       1.009185
min         0.100000  1.000000e+02       1.000000       0.000000
25%         1.400000  4.995000e+03       5.000000       4.000000
50%         1.800000  9.298000e+03       5.000000       5.000000
75%         2.000000  1.700000e+04       5.000000       5.000000
max      3500.000000  9.999999e+06      17.000000       7.000000
```

Bộ dữ liệu được thu thập từ năm 2012- 2021 với 267200 bản ghi

Các hãng xe

```
df['Maker'] = df.Maker.apply(lambda x : ' '.join(x.split(' ')[:1]))
df['Maker'].unique()
```

```
array(['Bentley', 'Brooke', 'Bugatti', 'Cadillac', 'Caterham',
       'Chevrolet', 'Chrysler', 'Citroen', 'Corvette', 'DAX', 'DS',
       'Dacia', 'Daewoo', 'Daihatsu', 'Daimler', 'Dodge', 'Ferrari',
       'Fiat', 'Ford', 'Abarth', 'GMC', 'Ginetta', 'Great',
'Grinnall',
       'Honda', 'Hummer', 'Hyundai', 'Infiniti', 'Isuzu', 'Jaguar',
       'Jeep', 'Jensen', 'KTM', 'Kia', 'Koenigsegg', 'Lamborghini',
       'Land', 'Lexus', 'Lincoln', 'Alfa', 'London', 'Lotus', 'MEV',
'MG',
       'MINI', 'Maserati', 'Maybach', 'Mazda', 'McLaren', 'Mercedes-
Benz',
       'Mitsubishi', 'Morgan', 'Nissan', 'Noble', 'Opel', 'Pagani',
       'Perodua', 'Peugeot', 'Aston', 'Pilgrim', 'Porsche', 'Proton',
       'Raw', 'Renault', 'Reva', 'Rolls-Royce', 'Rover', 'SEAT',
'Audi',
       'SKODA', 'Saab', 'Santana', 'Sebring', 'Smart', 'Ssangyong',
       'Subaru', 'Suzuki', 'TVR', 'BMW', 'Tesla', 'Tiger', 'Toyota',
       'Vauxhall', 'Volkswagen', 'Volvo', 'Westfield', 'Zenos'],
      dtype=object)
```

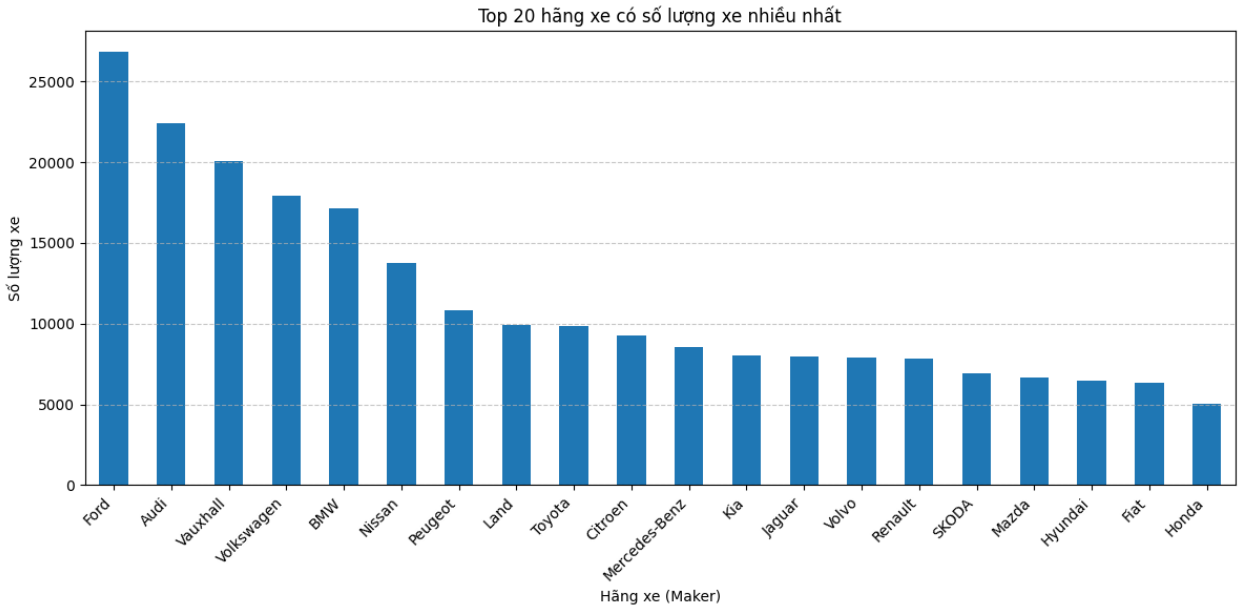
```
df['Maker'].value_counts()
```

```
Maker
Ford          26815
Audi          22441
Vauxhall      20062
Volkswagen    17913
BMW           17164
```

```
              ...
Raw              1
Reva             1
Pagani           1
Sebring          1
Santana          1
Name: count, Length: 87, dtype: int64
```

Top 20 hãng xe có số lượng xe nhiều nhất'

```
import matplotlib.pyplot as plt

# Đếm số lượng xe theo hãng sản xuất
maker_counts = df['Maker'].value_counts().head(20)  # top 20 hãng phổ biến

# Vẽ biểu đồ cột
plt.figure(figsize=(12, 6))
maker_counts.plot(kind='bar')
plt.title('Top 20 hãng xe có số lượng xe nhiều nhất')
plt.xlabel('Hãng xe (Maker)')
plt.ylabel('Số lượng xe')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```



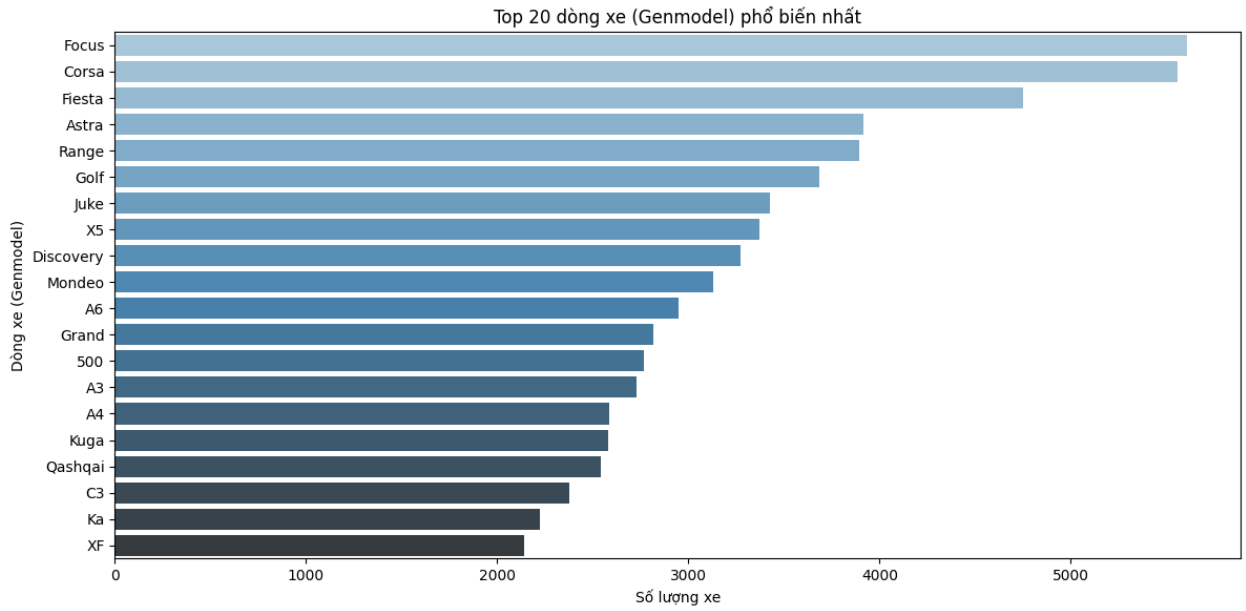
 Top 20 Hãng Xe Được Rao Bán Nhiều Nhất (2012–2021)

Biểu đồ trên thể hiện 20 hãng xe (Maker) có số lượng xe được rao bán nhiều nhất trong giai đoạn 2012–2021.

 Các hãng phổ thông như Ford, Audi, Vauxhall, Volkswagen, BMW, Nissan, Peugeot, Toyota chiếm tỷ trọng cao, cho thấy mức độ phổ biến và thanh khoản tốt trên thị trường.

```
import matplotlib.pyplot as plt
import seaborn as sns

top_genmodels = df['Genmodel'].value_counts().nlargest(20)

plt.figure(figsize=(12,6))
sns.barplot(x=top_genmodels.values, y=top_genmodels.index,
palette="Blues_d")
plt.title("Top 20 dòng xe (Genmodel) phổ biến nhất")
plt.xlabel("Số lượng xe")
plt.ylabel("Dòng xe (Genmodel)")
plt.tight_layout()
plt.show()
```

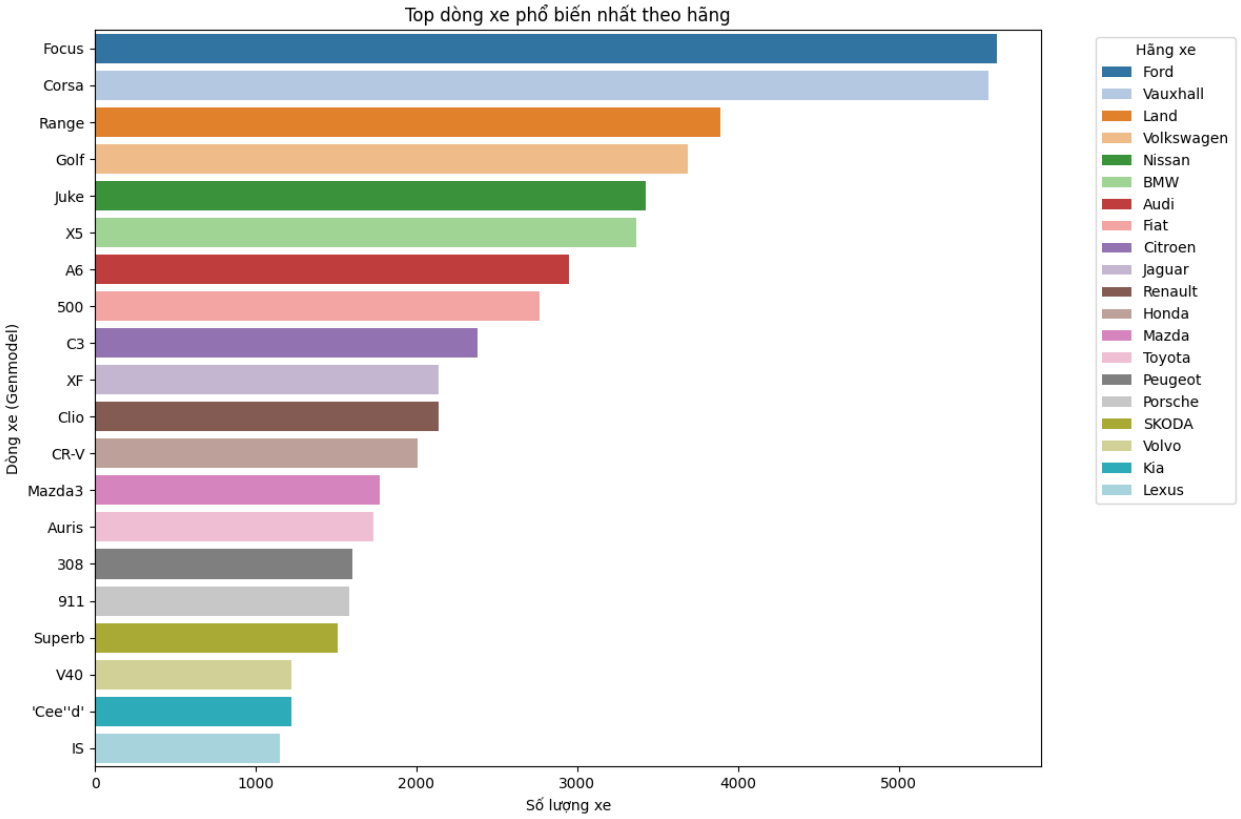
```
C:\Users\ADMIN KH\AppData\Local\Temp\ipykernel_20396\1407195017.py:7:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `y` variable to `hue` and set
`legend=False` for the same effect.

  sns.barplot(x=top_genmodels.values, y=top_genmodels.index,
palette="Blues_d")
```



TOP CÁC DÒNG XE PHỔ BIẾN NHẤT CỦA CÁC HÃNG XE

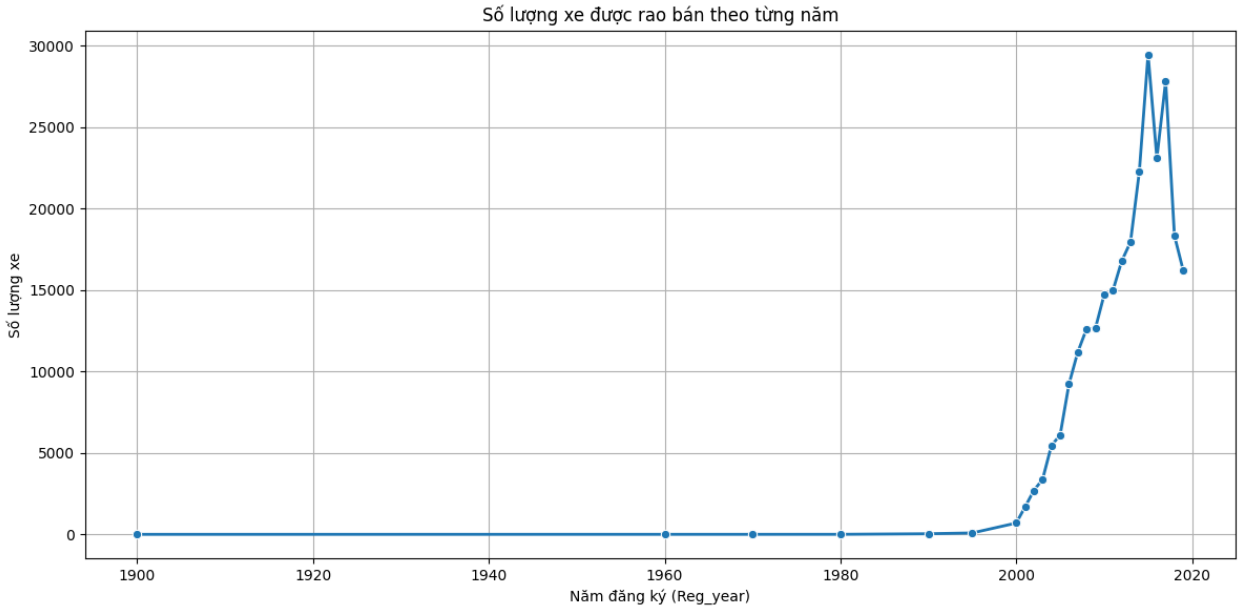
```
import matplotlib.pyplot as plt
import seaborn as sns
```

```
plt.figure(figsize=(12, 8))
sns.barplot(
    data=most_popular_genmodel_by_maker.head(20),
    x='count', y='Genmodel', hue='Maker', dodge=False, palette='tab20'
)
plt.title("Top dòng xe phổ biến nhất theo hãng")
plt.xlabel("Số lượng xe")
plt.ylabel("Dòng xe (Genmodel)")
plt.legend(title="Hãng xe", bbox_to_anchor=(1.05, 1), loc='upper
left')
plt.tight_layout()
plt.show()
```




```
import matplotlib.pyplot as plt
import seaborn as sns

# Đếm số lượng xe theo từng năm
yearly_counts = df['Reg_year'].value_counts().sort_index()

# Vẽ biểu đồ
plt.figure(figsize=(12, 6))
sns.lineplot(x=yearly_counts.index, y=yearly_counts.values,
marker='o', linewidth=2)
plt.title("Số lượng xe được rao bán theo từng năm")
```

```
plt.xlabel("Năm đăng ký (Reg_year)")
plt.ylabel("Số´lượng xe")
plt.grid(True)
plt.tight_layout()
plt.show()
```



 1. Giai đoạn trước năm 2000: Số lượng xe rất thấp, gần như bằng 0.

Có thể là:

Dữ liệu rao bán xe cổ không được thu thập đầy đủ.

Xe đời quá cũ ít được bán lại hoặc đã bị loại bỏ khỏi thị trường.

 2. Giai đoạn 2000–2011: Số lượng xe bắt đầu tăng dần theo thời gian.

Đây là dấu hiệu cho thấy thị trường xe cũ bắt đầu hình thành và mở rộng, đặc biệt với các xe sản xuất sau năm 2005.

 3. Giai đoạn 2012–2019: Tăng trưởng mạnh mẽ, lên đến đỉnh vào khoảng 2018–2019.

Đây là những đời xe được rao bán lại nhiều nhất, cho thấy:

Chu kỳ sử dụng xe thường là 4–8 năm trước khi được bán lại.

Các xe sản xuất giai đoạn này vẫn còn giá trị sử dụng tốt và dễ tiêu thụ.

 4. Giai đoạn 2020–2021: Sự giảm nhẹ đáng chú ý.

Có thể do:

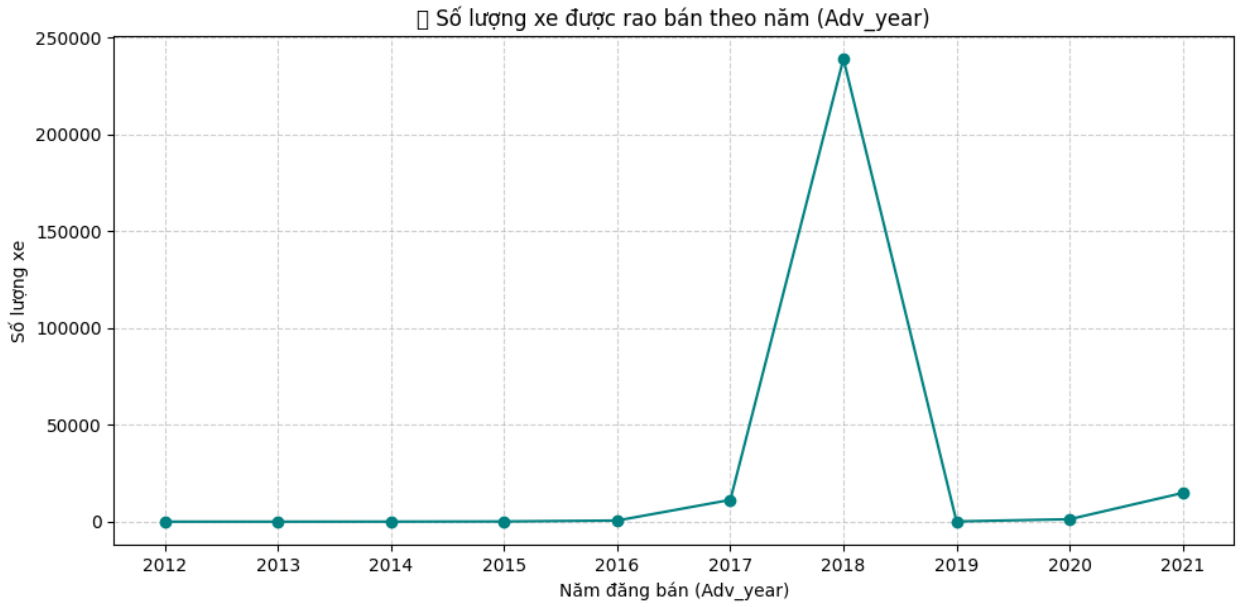
Xe sản xuất quá mới nên chưa đến chu kỳ bán lại.

Tác động từ đại dịch COVID-19, khiến nhu cầu mua – bán xe giảm.

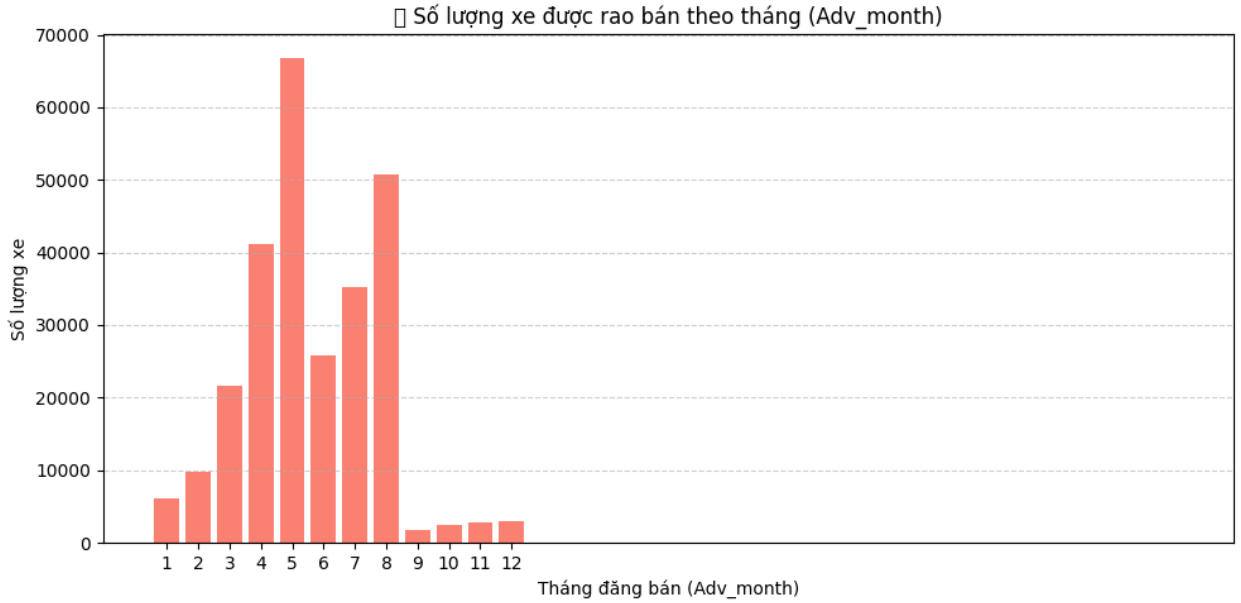
biểu đồ theo Adv_year hoặc Adv_month để biết mùa cao điểm rao bán

```
import matplotlib.pyplot as plt

# Đếm số lượng xe được đăng bán theo năm và theo tháng
adv_year_counts = df['Adv_year'].value_counts().sort_index()
adv_month_counts = df['Adv_month'].value_counts().sort_index()

# Vẽ biểu đồ số lượng xe rao bán theo năm
plt.figure(figsize=(10, 5))
plt.plot(adv_year_counts.index, adv_year_counts.values, marker='o',
color='teal')
plt.title(' Số lượng xe được rao bán theo năm (Adv_year)')
plt.xlabel('Năm đăng bán (Adv_year)')
plt.ylabel('Số lượng xe')
plt.grid(True, linestyle='--', alpha=0.6)
plt.xticks(adv_year_counts.index)
plt.tight_layout()
plt.show()

# Vẽ biểu đồ số lượng xe rao bán theo tháng
plt.figure(figsize=(10, 5))
plt.bar(adv_month_counts.index, adv_month_counts.values,
color='salmon')
plt.title(' Số lượng xe được rao bán theo tháng (Adv_month)')
plt.xlabel('Tháng đăng bán (Adv_month)')
plt.ylabel('Số lượng xe')
plt.xticks(range(1, 13))
plt.grid(axis='y', linestyle='--', alpha=0.6)
plt.tight_layout()
plt.show()
```

```
C:\Users\ADMIN KH\AppData\Local\Temp\ipykernel_20396\2307863892.py:15:
UserWarning: Glyph 128197 (\N{CALENDAR}) missing from font(s) DejaVu
Sans.
  plt.tight_layout()
C:\Users\ADMIN KH\AppData\Roaming\Python\Python310\site-packages\
IPython\core\pylabtools.py:170: UserWarning: Glyph 128197 (\
N{CALENDAR}) missing from font(s) DejaVu Sans.
  fig.canvas.print_figure(bytes_io, **kw)
```

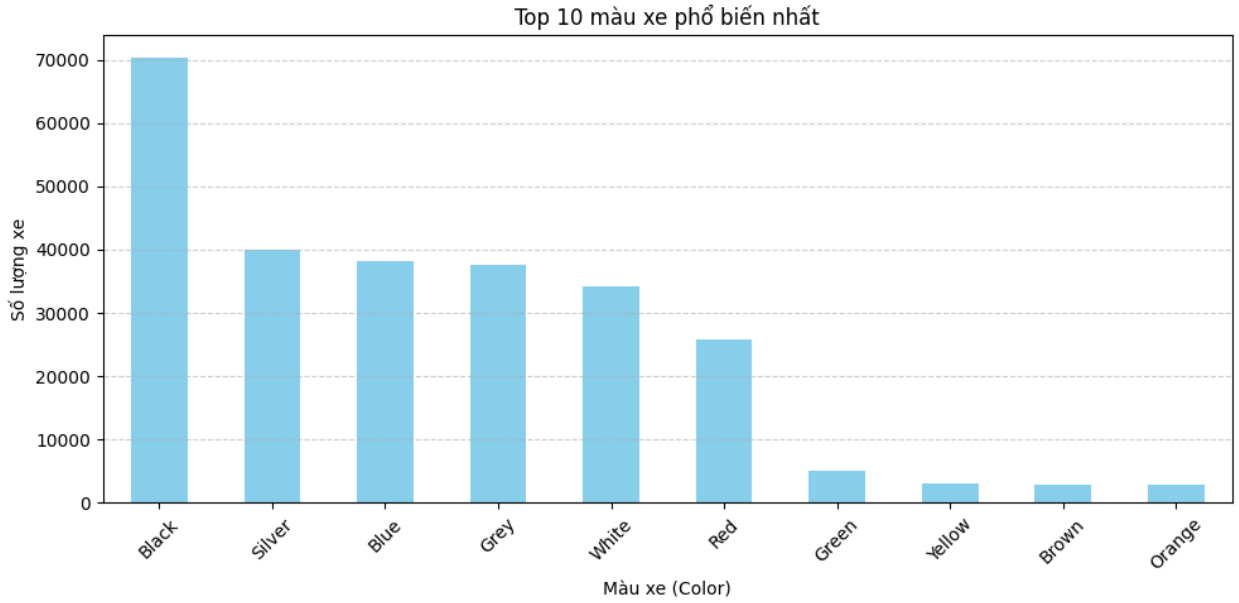



```
C:\Users\ADMIN KH\AppData\Local\Temp\ipykernel_20396\2307863892.py:26:
UserWarning: Glyph 128198 (\N{TEAR-OFF CALENDAR}) missing from font(s)
DejaVu Sans.
  plt.tight_layout()
C:\Users\ADMIN KH\AppData\Roaming\Python\Python310\site-packages\
IPython\core\pylabtools.py:170: UserWarning: Glyph 128198 (\N{TEAR-OFF
CALENDAR}) missing from font(s) DejaVu Sans.
  fig.canvas.print_figure(bytes_io, **kw)
```



Màu xe

```
color_counts = df['Color'].value_counts().head(10)
plt.figure(figsize=(10, 5))
color_counts.plot(kind='bar', color='skyblue')
plt.title('Top 10 màu xe phổ biến nhất')
plt.xlabel('Màu xe (Color)')
plt.ylabel('Số lượng xe')
plt.xticks(rotation=45)
plt.grid(axis='y', linestyle='--', alpha=0.6)
plt.tight_layout()
plt.show()
```



KIỂM TRA MỐI TƯƠNG QUAN

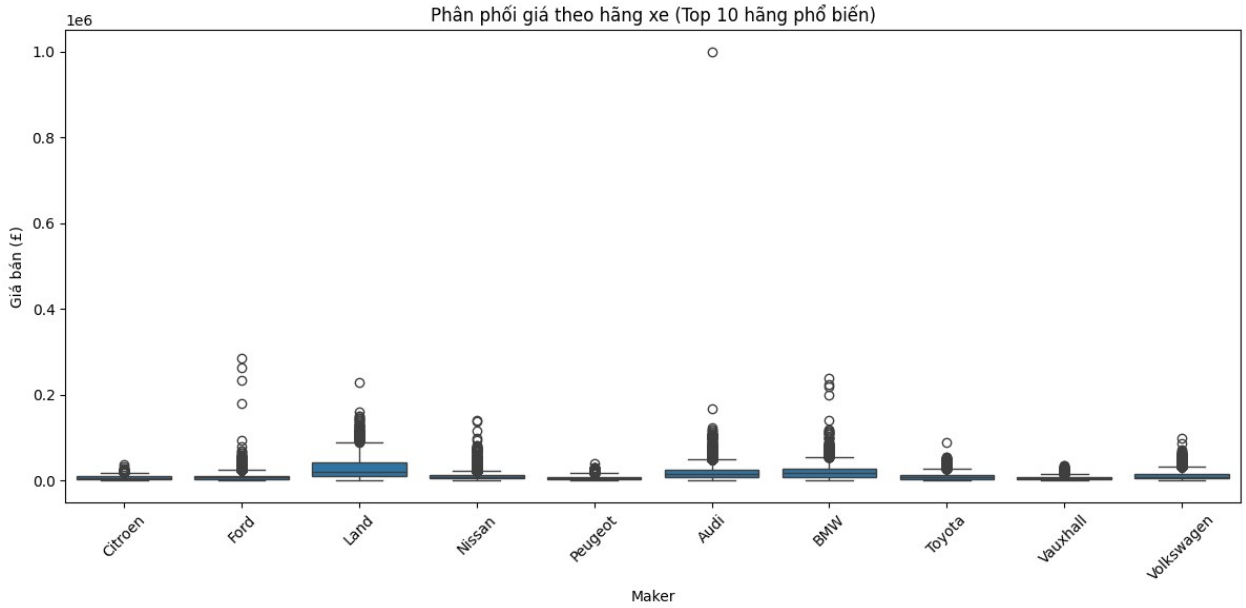
Thống kê mô tả giá theo hãng xe (Maker)

```
price_stats_by_maker = df.groupby('Maker')['Price'].agg(['count',
'mean', 'median', 'min', 'max']).sort_values(by='mean',
ascending=False)
price_stats_by_maker.head(10)
```

Maker	count	mean	median	min	max
Pagani	1	1.650000e+06	1650000.0	1650000.0	1650000.0
Bugatti	3	1.383333e+06	1250000.0	1250000.0	1650000.0
Koenigsegg	1	1.300000e+06	1300000.0	1300000.0	1300000.0
Lamborghini	343	2.329342e+05	214999.0	69990.0	499990.0
McLaren	248	2.247831e+05	138495.0	84995.0	9999999.0
Ferrari	609	1.929391e+05	164995.0	29990.0	2599990.0
Rolls-Royce	96	1.544737e+05	143970.5	24950.0	495000.0
Noble	2	1.537250e+05	153725.0	42500.0	264950.0

```
Maybach          4  1.527448e+05   128490.0    74999.0   279000.0
Aston          117  1.063002e+05   107950.0    25750.0   271015.0
```

```
top_makers = df['Maker'].value_counts().head(10).index
filtered_df = df[df['Maker'].isin(top_makers)]

plt.figure(figsize=(12,6))
sns.boxplot(data=filtered_df, x='Maker', y='Price')
plt.title("Phân phối giá theo hãng xe (Top 10 hãng phổ biến)")
plt.ylabel("Giá bán (£)")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



1. Giá trung vị (median): Các hãng BMW, Audi và Land Rover có giá trung vị cao nhất, cho thấy đây là những hãng thuộc phân khúc xe cao cấp.

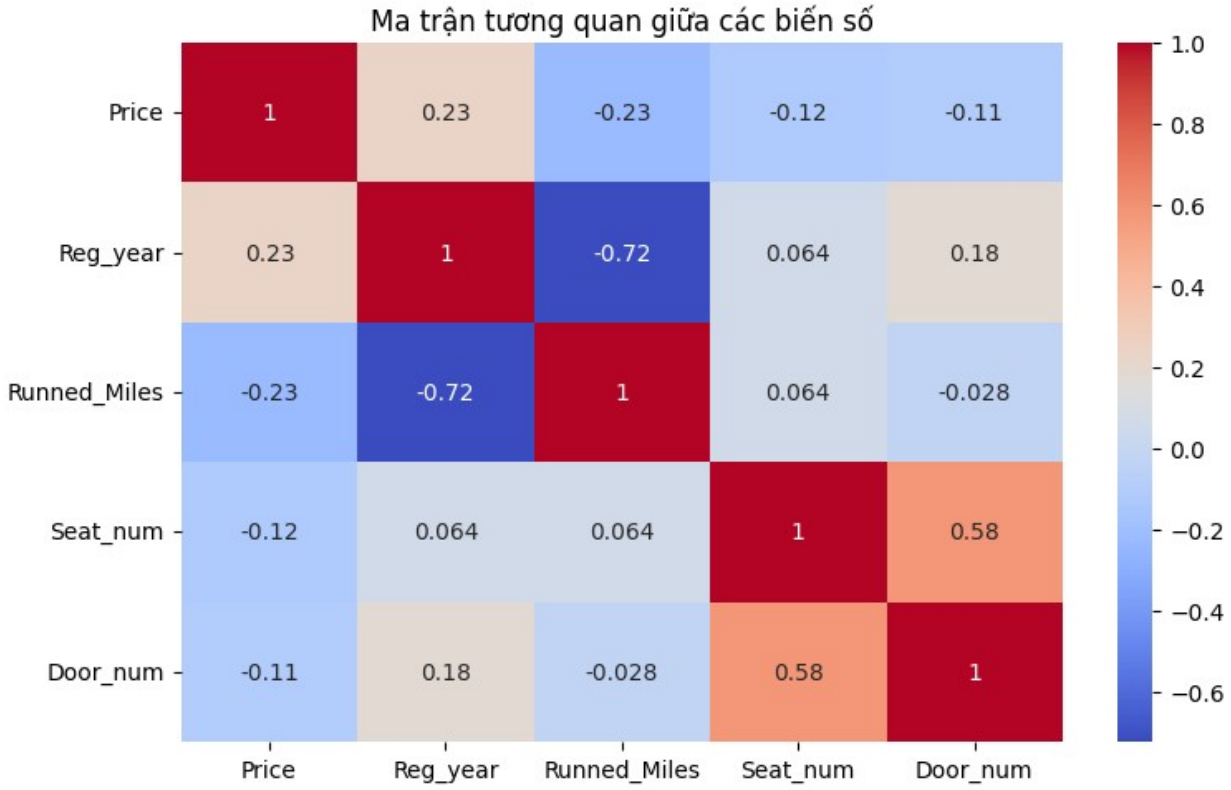
Ngược lại, Ford, Peugeot, Vauxhall, Citroen có giá trung vị thấp hơn, phản ánh đúng vị thế là các hãng xe phổ thông, dễ tiếp cận.

2. Độ phân tán giá: BMW và Land Rover có phân phối giá khá rộng, thể hiện sự đa dạng dòng xe: từ xe phổ thông cho đến xe sang.

Ford, Citroen, Peugeot có phân phối giá tập trung, phù hợp với dòng xe bình dân (ít sự chênh lệch giữa các dòng).

```
numeric_cols = ['Price', 'Reg_year', 'Runned_Miles', 'Seat_num',
'Door_num']

correlation_matrix = df[numeric_cols].corr()
```

```
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(8, 5))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title("Ma trận tương quan giữa các biến số")
plt.tight_layout()
plt.show()
```



NHẬN XÉT TỪ MA TRẬN TƯƠNG QUAN

Biến mục tiêu chính: `Price` (Giá bán)

Biến liên quan	Hệ số tương quan	Giải thích ngắn gọn
`Reg_year`	**+0.23**	Xe **mới hơn** có xu hướng **giá cao hơn** → *tương quan thuận*
`Runned_Miles`	**-0.23**	Xe **chạy nhiều** thì **giá giảm** → *tương quan nghịch*
`Seat_num`	-0.12	Không ảnh hưởng đáng kể, nhưng xe lớn thường có nhiều chỗ ngồi hơn
`Door_num`	-0.11	Rất yếu, gần như **không ảnh hưởng đến giá**

🔍 Những điểm nổi bật:

🔹 1. `Reg_year` và `Runned_Miles` có tương quan mạnh với nhau (`-0.72`)
→ Điều này hợp lý: xe mới thường chạy ít hơn.

🔹 2. Các yếu tố vật lý như `Seat_num`, `Door_num` có ảnh hưởng rất nhỏ
→ Phù hợp với thực tế: chúng ảnh hưởng đến phân khúc, nhưng **không quyết định giá bán**.

**TỪ ĐÓ

🔹 Khi xây dựng mô hình định giá hoặc dashboard hỗ trợ ra quyết định:

- **Ưu tiên các biến quan trọng**:
  `Reg_year`, `Runned_Miles`, `Fuel_type`, `Maker`
- **Có thể loại bỏ hoặc giảm trọng số** cho các biến ít ảnh hưởng như:
  `Seat_num`, `Door_num`

Kiểm định Anova

```
from scipy.stats import f_oneway
color_groups = [group['Price'].dropna() for name, group in
df.groupby('Color')]
f_stat, p_val = f_oneway(*color_groups)
print(f"F-statistic: {f_stat:.2f}, p-value: {p_val:.4f}")

F-statistic: 61.05, p-value: 0.0000
```

🔹 Kết luận: Màu sắc có ảnh hưởng đáng kể đến giá bán xe 🔹 F-statistic = 61.05 cho thấy sự khác biệt giữa các nhóm là rất đáng kể.

🔹 p-value = 0.0000 (nhỏ hơn 0.05) → bác bỏ giả thuyết H_0 (không có sự khác biệt về giá giữa các màu). =>>>>>

TẠO CÁC BẢNG DIM/ FACT

```
import pandas as pd
import os


df = df[df['Adv_month'].between(1, 12)]


df['YearMonth'] = df['Adv_month'].astype(str).str.zfill(2) + '/' +
df['Adv_year'].astype(str)

dim_date = df[['Adv_year', 'Adv_month',
'YearMonth']].drop_duplicates().reset_index(drop=True)
dim_date['Date_ID'] = dim_date.index + 1
```

```
dim_maker = df[['Maker']].drop_duplicates().reset_index(drop=True)
dim_genmodel = df[['Genmodel_ID',
'Genmodel']].drop_duplicates().reset_index(drop=True)
dim_genmodel['Genmodel_Key'] = dim_genmodel.index + 1
dim_bodytype =
df[['Bodytype']].drop_duplicates().reset_index(drop=True)
dim_fuel = df[['Fuel_type']].drop_duplicates().reset_index(drop=True)
dim_gearbox = df[['Gearbox']].drop_duplicates().reset_index(drop=True)
dim_color = df[['Color']].drop_duplicates().reset_index(drop=True)


output_dir = "car_data_etl"
os.makedirs(output_dir, exist_ok=True)

dim_maker.to_csv(os.path.join(output_dir, "dim_maker.csv"),
index=False)
dim_genmodel.to_csv(os.path.join(output_dir, "dim_genmodel.csv"),
index=False)
dim_bodytype.to_csv(os.path.join(output_dir, "dim_bodytype.csv"),
index=False)
dim_fuel.to_csv(os.path.join(output_dir, "dim_fuel_type.csv"),
index=False)
dim_gearbox.to_csv(os.path.join(output_dir, "dim_gearbox.csv"),
index=False)
dim_color.to_csv(os.path.join(output_dir, "dim_color.csv"),
index=False)
dim_date.to_csv(os.path.join(output_dir, "dim_date.csv"), index=False)

print("□ Xuất thành công. Các file trong thư mục:")
print(os.listdir(output_dir))
```

```
□ Xuất thành công. Các file trong thư mục:
['dim_bodytype.csv', 'dim_color.csv', 'dim_date.csv',
'dim_fuel_type.csv', 'dim_gearbox.csv', 'dim_genmodel.csv',
'dim_maker.csv']
```

```
df.describe()
```

```
            Adv_year      Adv_month       Reg_year  Runned_Miles  \
count  267200.000000  267200.000000  267200.000000  2.672000e+05
mean     2018.128084       5.625647    2012.705341  4.812342e+04
std         0.747649       2.090895       4.460922  4.185477e+04
min      2012.000000       1.000000    1900.000000  0.000000e+00
25%      2018.000000       4.000000    2010.000000  1.410100e+04
50%      2018.000000       5.000000    2014.000000  3.920200e+04
75%      2018.000000       7.000000    2016.000000  7.500000e+04
max      2021.000000      33.000000    2019.000000  6.363342e+06

          Engin_size          Price       Seat_num       Door_num
count  267200.000000  2.672000e+05  267200.000000  267200.000000
```

```
mean         1.962738  1.472051e+04       4.903363       4.369246
std          9.344411  3.212282e+04       0.885365       1.009185
min          0.100000  1.000000e+02       1.000000       0.000000
25%          1.400000  4.995000e+03       5.000000       4.000000
50%          1.800000  9.298000e+03       5.000000       5.000000
75%          2.000000  1.700000e+04       5.000000       5.000000
max       3500.000000  9.999999e+06      17.000000       7.000000
```

```
import pandas as pd

df.to_csv("du_lieu_xuat.csv", index=False)
```