Actividad Integradora 2

Fernanda Pérez Ruiz A01742102

In [ ]:
```
import pandas as pd
```

In [ ]:
```
path_csv = 'd:\Downloads\precios_autos.csv'
path_diccionario = 'd:\Downloads\precios_autos_Diccionario.xlsx'


df_autos = pd.read_csv(path_csv)
diccionario = pd.read_excel(path_diccionario)

df_autos.head()
```

Out[ ]:

	symboling	CarName	fueltype	carbody	drivewheel	enginelocation	wheelbase	c
0	3	alfa-romero giulia	gas	convertible	rwd	front	88.6	
1	3	alfa-romero stelvio	gas	convertible	rwd	front	88.6	
2	1	alfa-romero Quadrifoglio	gas	hatchback	rwd	front	94.5	
3	2	audi 100 ls	gas	sedan	fwd	front	99.8	
4	2	audi 100ls	gas	sedan	4wd	front	99.4	

5 rows × 21 columns

In [ ]:
```
diccionario.head()
```

Out[ ]:

	Symboling	Its assigned insurance risk rating, A value of +3 indicates that the auto is risky, -3 that it is probably pretty safe.(Categorical)
0	CarName	Name of car company (Categorical)
1	fueltype	Car fuel type i.e gas or diesel (Categorical)
2	carbody	body of car (Categorical)
3	drivewheel	type of drive wheel (Categorical)
4	enginelocation	Location of car engine (Categorical)

Con el grupo de variables seleccionadas realiza el siguiente procesamiento de los datos:

Exploración de la base de datos

Exploración de la base de datos

Calcula medidas estadísticas apropiadas para las variables:

cuantitativas (media, desviación estándar, cuantiles, etc)

In [ ]:
```
variables_grupo_3 = ['enginesize', 'stroke', 'price']

df_grupo_3 = df_autos[variables_grupo_3]

medidas_estadisticas = df_grupo_3.describe()
print(medidas_estadisticas)
```

```
       enginesize      stroke         price
count  205.000000  205.000000    205.000000
mean   126.907317    3.255415  13276.710571
std     41.642693    0.313597   7988.852332
min     61.000000    2.070000   5118.000000
25%     97.000000    3.110000   7788.000000
50%    120.000000    3.290000  10295.000000
75%    141.000000    3.410000  16503.000000
max    326.000000    4.170000  45400.000000
```

cualitativas: cuantiles, frecuencias (puedes usar el comando table o prop.table)

In [ ]:
```
frecuencia_enginelocation = df_autos['enginelocation'].value_counts()
print(frecuencia_enginelocation)
```

```
front    202
rear       3
Name: enginelocation, dtype: int64
```

In [ ]:
```
cuantiles_enginesize = df_autos['enginesize'].quantile([0.25, 0.5, 0.75])
cuantiles_stroke = df_autos['stroke'].quantile([0.25, 0.5, 0.75])

print("Cuantiles de enginesize:\n", cuantiles_enginesize)
print("Cuantiles de stroke:\n", cuantiles_stroke)
```

```
Cuantiles de enginesize:
 0.25     97.0
0.50    120.0
0.75    141.0
Name: enginesize, dtype: float64
Cuantiles de stroke:
 0.25    3.11
0.50    3.29
0.75    3.41
Name: stroke, dtype: float64
```

Analiza la correlación entre las variables (analiza posible colinealidad entre las variables)

matriz de correlación

In [ ]:
```
correlacion_grupo_3 = df_grupo_3.corr()
print(correlacion_grupo_3)
```

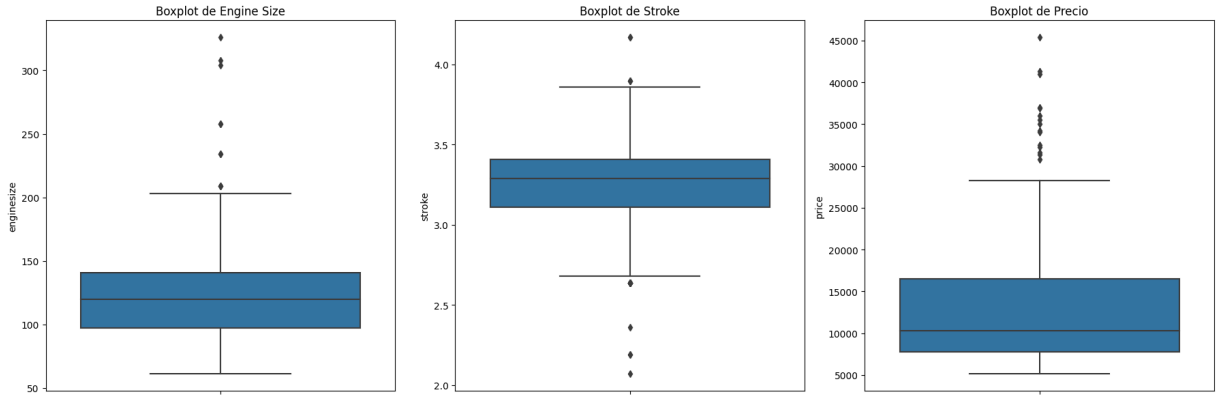
```
            enginesize    stroke     price
enginesize    1.000000  0.203129  0.874145
stroke        0.203129  1.000000  0.079443
price         0.874145  0.079443  1.000000
```

Explora los datos usando herramientas de visualización (si lo consideras necesario):

Variables cuantitativas:

Boxplots

In [ ]:
```
import matplotlib.pyplot as plt
import seaborn as sns

fig, axes = plt.subplots(1, 3, figsize=(18, 6))

sns.boxplot(y=df_grupo_3['enginesize'], ax=axes[0])
axes[0].set_title('Boxplot de Engine Size')

sns.boxplot(y=df_grupo_3['stroke'], ax=axes[1])
axes[1].set_title('Boxplot de Stroke')

sns.boxplot(y=df_grupo_3['price'], ax=axes[2])
axes[2].set_title('Boxplot de Precio')

plt.tight_layout()
plt.show()
```



Histogramas

In [ ]:
```
fig, axes = plt.subplots(1, 3, figsize=(18, 6))

sns.histplot(df_grupo_3['enginesize'], bins=10, kde=True, ax=axes[0])
axes[0].set_title('Histograma de Engine Size')

sns.histplot(df_grupo_3['stroke'], bins=10, kde=True, ax=axes[1])
axes[1].set_title('Histograma de Stroke')

sns.histplot(df_grupo_3['price'], bins=10, kde=True, ax=axes[2])
axes[2].set_title('Histograma de Precio')

plt.tight_layout()
plt.show()
```

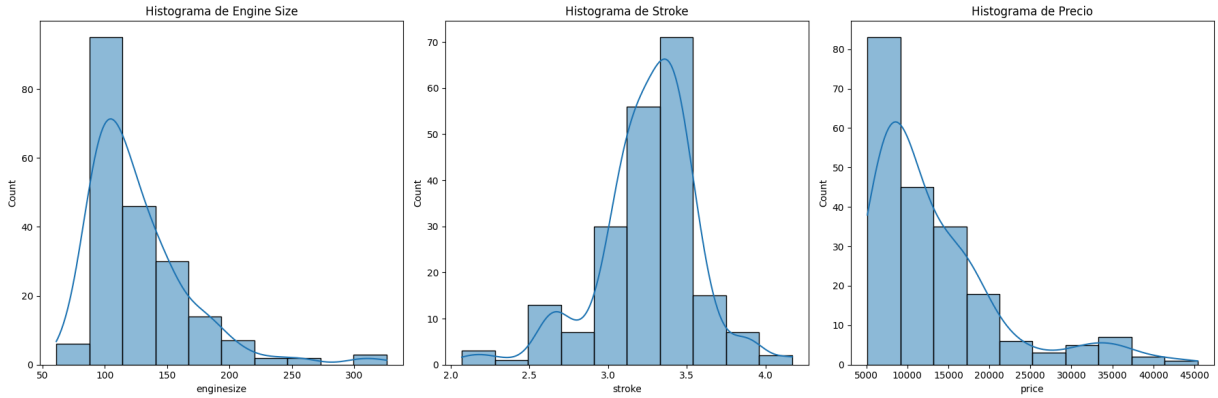
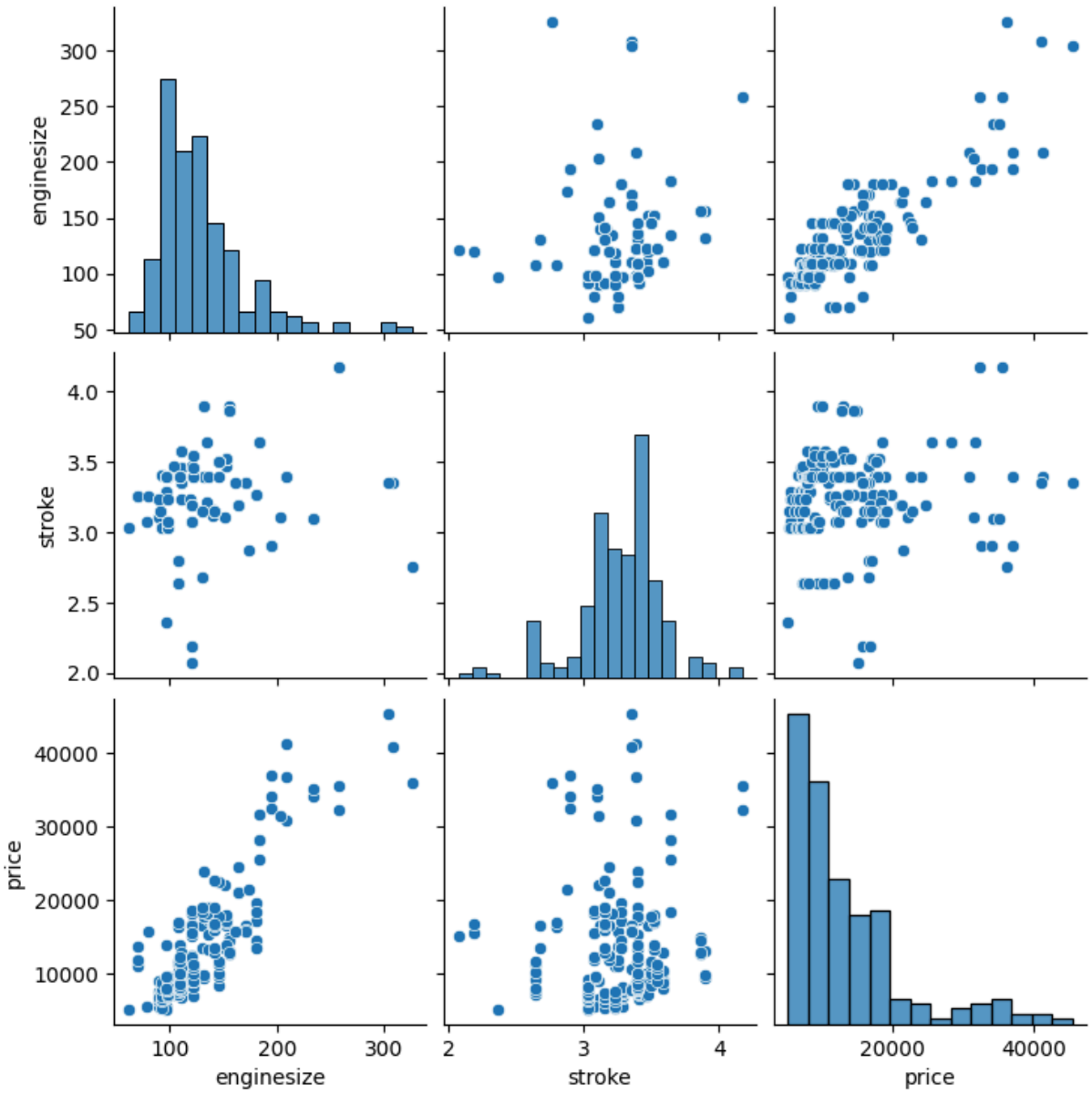


diagrama de dispersión para ver la correlación por pares

In [ ]:
```
sns.pairplot(df_grupo_3[['enginesize', 'stroke', 'price']])
plt.show()
```

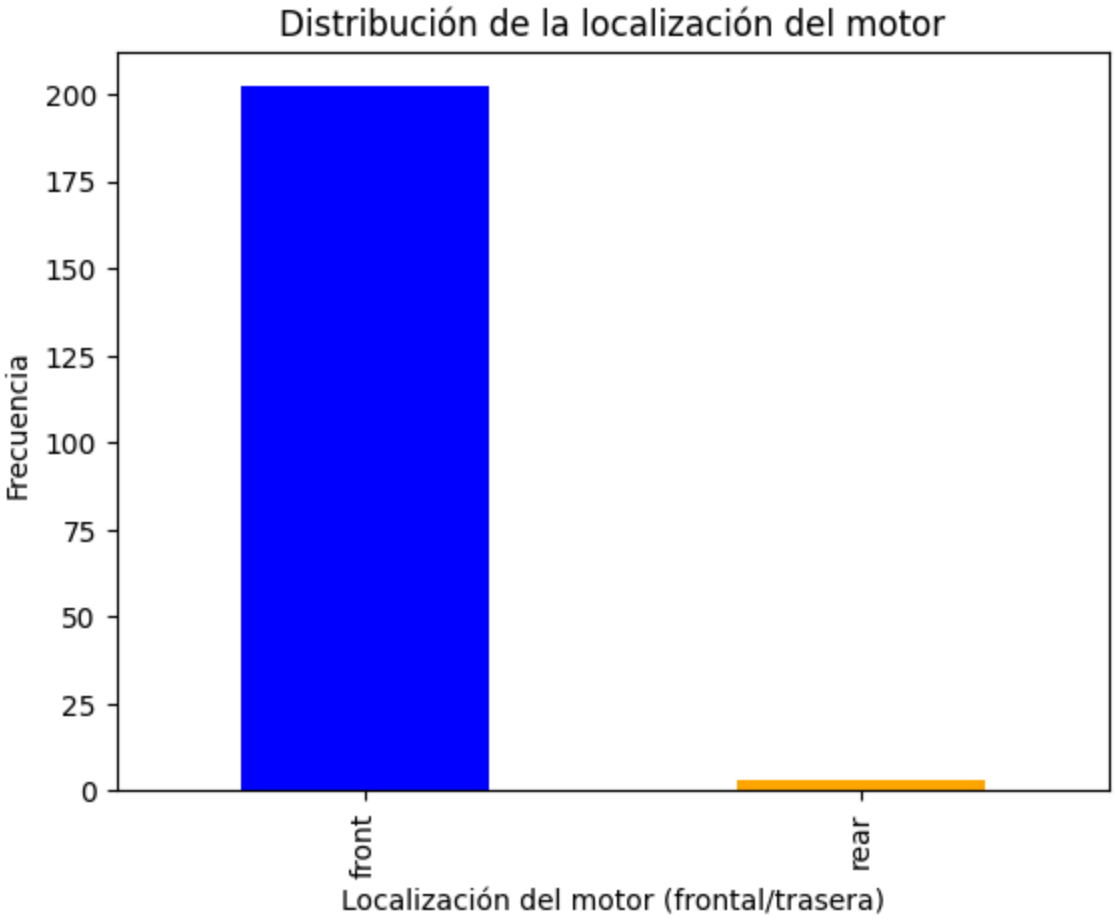


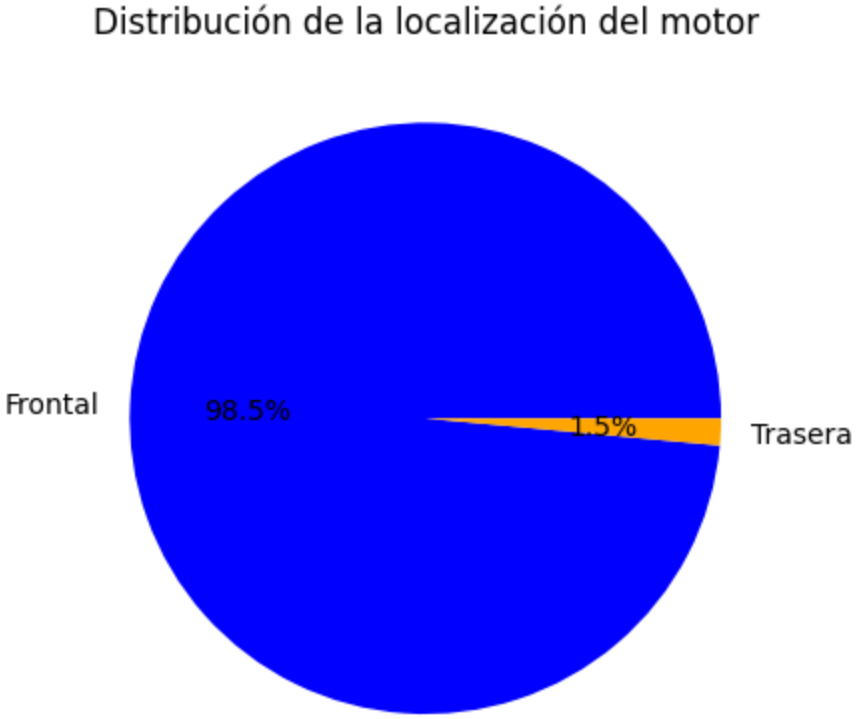
Variables categóricas

Distribución de los datos (diagramas de barras, diagramas de pastel)

In [ ]:
```
import matplotlib.pyplot as plt

frecuencia_enginelocation = df_autos['enginelocation'].value_counts()
frecuencia_enginelocation.plot(kind='bar', color=['blue', 'orange'])
plt.title('Distribución de la localización del motor')
plt.xlabel('Localización del motor (frontal/trasera)')
plt.ylabel('Frecuencia')
plt.show()


frecuencia_enginelocation.plot(kind='pie', autopct='%1.1f%%', colors=['blue', 'oran
plt.title('Distribución de la localización del motor')
```

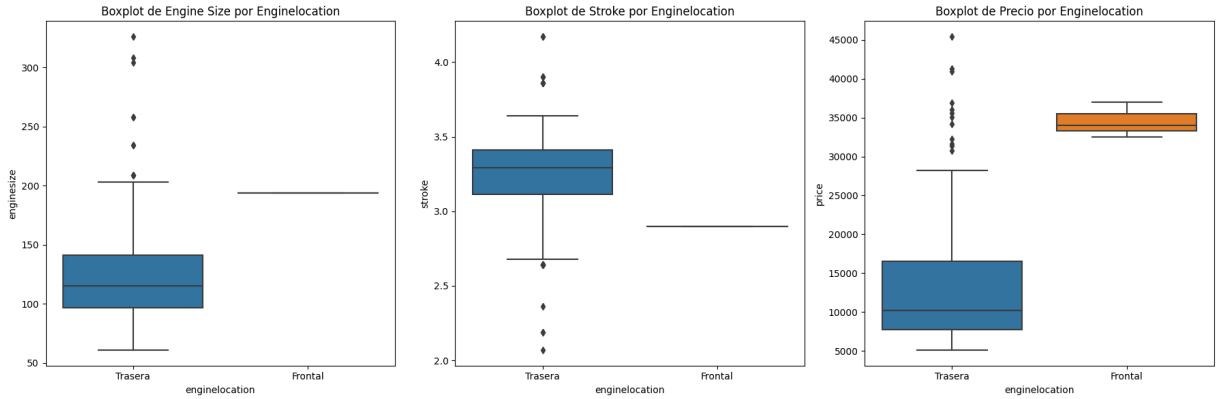
```
plt.ylabel('')
plt.show()
```





Boxplot por categoría de las variables cuantitativas

In [ ]:
```
import seaborn as sns

fig, axes = plt.subplots(1, 3, figsize=(18, 6))

sns.boxplot(x=df_autos['enginelocation'], y=df_grupo_3['enginesize'], ax=axes[0])
axes[0].set_title('Boxplot de Engine Size por Enginelocation')
axes[0].set_xticklabels(['Trasera', 'Frontal'])

sns.boxplot(x=df_autos['enginelocation'], y=df_grupo_3['stroke'], ax=axes[1])
axes[1].set_title('Boxplot de Stroke por Enginelocation')
axes[1].set_xticklabels(['Trasera', 'Frontal'])

sns.boxplot(x=df_autos['enginelocation'], y=df_grupo_3['price'], ax=axes[2])
axes[2].set_title('Boxplot de Precio por Enginelocation')
axes[2].set_xticklabels(['Trasera', 'Frontal'])

plt.tight_layout()
plt.show()
```



2 Modelación y verificación del modelo

1. Encuentra la ecuación de regresión de mejor ajuste. Propón al menos 2 modelos de ajuste para encontrar la mejor forma de ajustar la variable precio.

In [ ]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
import statsmodels.api as sm
import numpy as np

# Modelo 1: solamente enginesize
X1 = df_autos[['enginesize']]
# Modelo 2: enginesize y stroke
X2 = df_autos[['enginesize', 'stroke']]
y = df_autos['price']
```

```
X1_train, X1_test, y_train, y_test = train_test_split(X1, y, test_size=0.2, random_
X2_train, X2_test, _, _ = train_test_split(X2, y, test_size=0.2, random_state=42)

modelo1 = LinearRegression()
modelo1.fit(X1_train, y_train)

modelo2 = LinearRegression()
modelo2.fit(X2_train, y_train)

print("Coeficientes del Modelo 1 (enginesize):", modelo1.coef_, "Intercepto:", mode
print("Coeficientes del Modelo 2 (enginesize + stroke):", modelo2.coef_, "Intercept
```

```
Coeficientes del Modelo 1 (enginesize): [165.84456256] Intercepto: -7741.76506716659
4
Coeficientes del Modelo 2 (enginesize + stroke): [  169.58201546 -2926.4976967 ] Int
ercepto: 1311.248527660011
```

2. Para cada uno de los modelos propuestos:

Realiza la regresión entre las variables involucradas

Analiza la significancia del modelo: Valida la significancia del modelo con un alfa de 0.04 (incluye las hipótesis que pruebas y el valor frontera)

In [ ]:
```
X1_train_sm = sm.add_constant(X1_train)
X2_train_sm = sm.add_constant(X2_train)

modelo1_sm = sm.OLS(y_train, X1_train_sm).fit()
print(modelo1_sm.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.751
Model:                            OLS   Adj. R-squared:                  0.749
Method:                 Least Squares   F-statistic:                     487.8
Date:                Fri, 06 Sep 2024   Prob (F-statistic):           9.81e-51
Time:                        17:43:23   Log-Likelihood:                -1586.9
No. Observations:                 164   AIC:                             3178.
Df Residuals:                     162   BIC:                             3184.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const      -7741.7651    996.381     -7.770      0.000   -9709.334   -5774.196
enginesize   165.8446      7.509     22.087      0.000     151.017     180.672
==============================================================================
Omnibus:                       15.749   Durbin-Watson:                   2.029
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               18.073
Skew:                           0.676   Prob(JB):                     0.000119
Kurtosis:                       3.903   Cond. No.                         436.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly spe
cified.
```

Interpretación del modelo 1:

Al tener un R-squared: 0.751, nos dice que el modelo explica el 75.1% de la variabilidad en el precio basado en enginesize.

Al ser el coeficiente de enginesize: 165.8446, significa qu e en promedio, un aumento de una unidad en el tamaño del motor está asociado con un aumento de $165.84 en el precio del auto.

Al tener un P-valor de enginesize: 0.000 (es muy significativo), nos indica que enginesize es un predictor significativo en el modelo con un alfa de 0.04.

In [ ]:
```
modelo2_sm = sm.OLS(y_train, X2_train_sm).fit()
print(modelo2_sm.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.765
Model:                            OLS   Adj. R-squared:                  0.762
Method:                 Least Squares   F-statistic:                     261.7
Date:                Fri, 06 Sep 2024   Prob (F-statistic):           2.56e-51
Time:                        17:43:23   Log-Likelihood:                -1582.2
No. Observations:                 164   AIC:                             3170.
Df Residuals:                     161   BIC:                             3180.
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       1311.2485   3077.455      0.426      0.671   -4766.135    7388.632
enginesize   169.5820      7.415     22.869      0.000     154.938     184.226
stroke     -2926.4977    944.018     -3.100      0.002   -4790.751   -1062.244
==============================================================================
Omnibus:                       15.284   Durbin-Watson:                   1.986
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               21.203
Skew:                           0.555   Prob(JB):                     2.49e-05
Kurtosis:                       4.368   Cond. No.                     1.44e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly spe
cified.
[2] The condition number is large, 1.44e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

Interpretación del modelo 2:

Al tener un R-squared: 0.765, nos dice que el modelo explica el 76.5% de la variabilidad en el precio basado en enginesize.

Al ser el coeficiente de enginesize: 169.5820, significa un aumento en el tamaño del motor está asociado con un incremento de $169.58 en el precio, manteniendo constante el valor de stroke.

Como nuestro coeficiente de stroke: -2926.4977, es negativo, nos dice que que por cada unidad adicional en stroke, el precio del auto disminuye en promedio $2926.49.

Al ser el P-valor de enginesize: 0.000 (es tambien muy significativo), y P-valor de stroke: 0.002 (también significativo), nos dice que ambos predictores son significativos con un alfa de 0.04.

Valida la significancia de βi con un alfa de 0.04 (incluye las hipótesis que pruebas y el valor frontera de cada una de ellas)

H0: El coeficiente es igual a cero .

H1: El coeficiente no es igual a cero

Modelo 1: P-valor: 0.000 < 0.04.

Por lo cual se rechaza H0.

El coeficiente de enginesize es significativo y no es igual a cero.

Modelo 2:

Enginisize: P-valor: 0.000 < 0.04, por lo cual rechazamos H0

El coeficiente de enginesize es significativo.

stroke: P-valor: 0.002 < 0.04 , por lo que tambien se rechaza H0

El coeficiente de stroke es significativo.

Ambos modelos son significativos por los resultados que obtuvimos, sin embargo el modelo 2 presenta un R^2 más alto, por lo cual podria considerarse un poco mejor, hasta el momento.

Indica cuál es el porcentaje de variación explicada por el modelo.

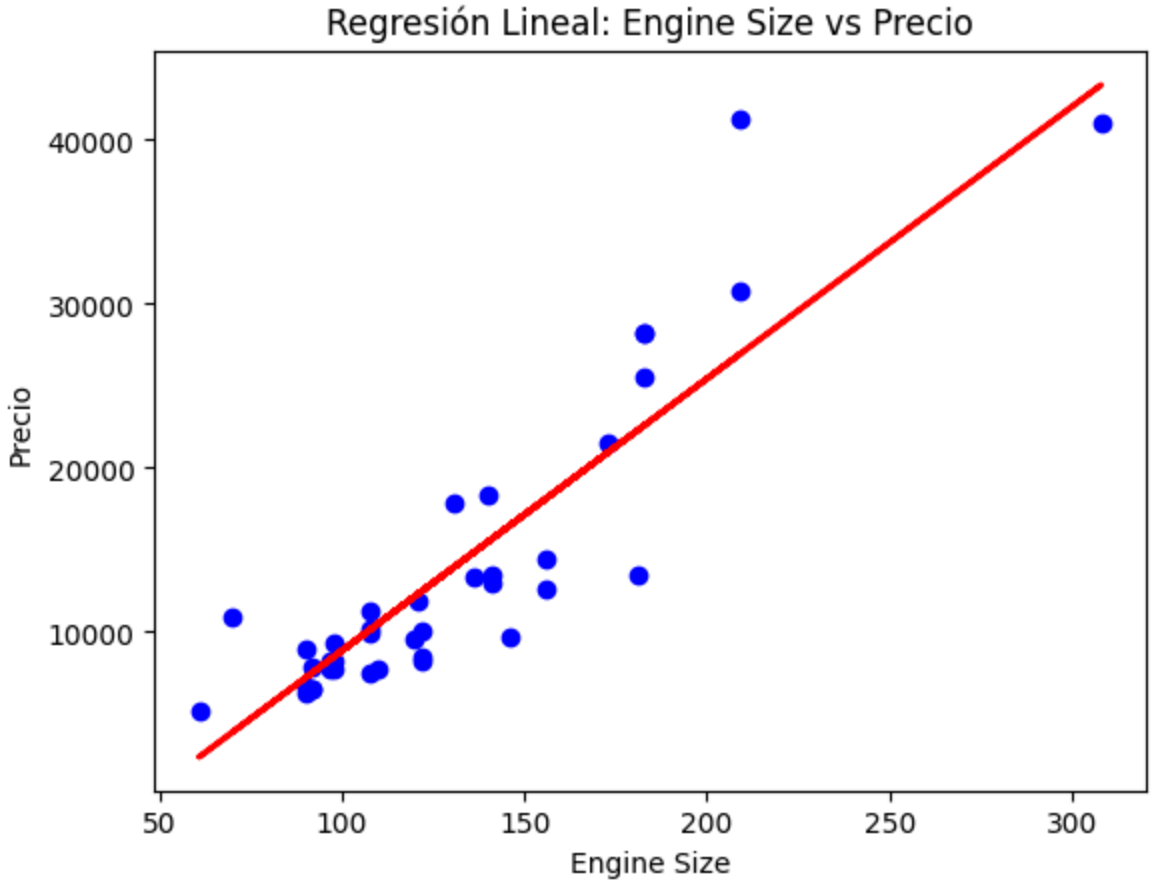
In [ ]:
```
print("R-squared del Modelo 1:", modelo1_sm.rsquared)
print("R-squared del Modelo 2:", modelo2_sm.rsquared)
```

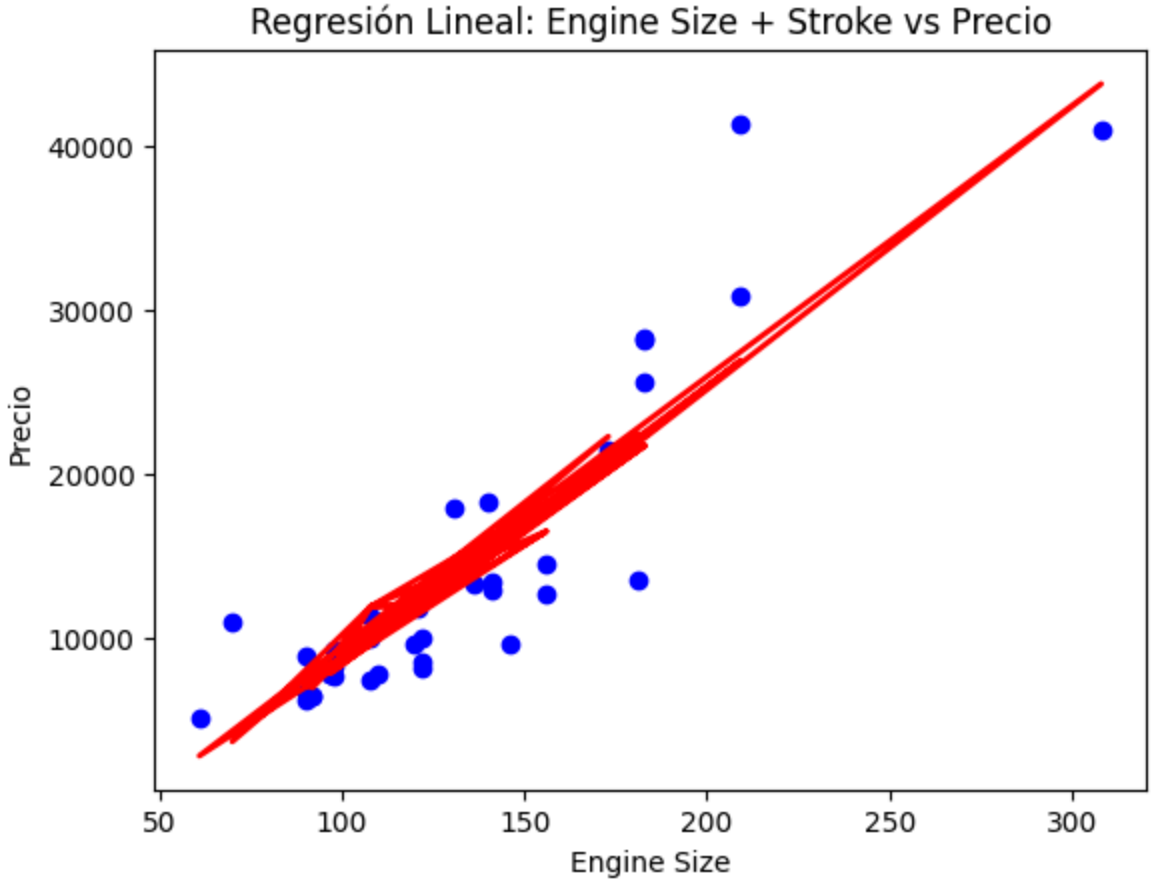
```
R-squared del Modelo 1: 0.7507055061579612
R-squared del Modelo 2: 0.7647479788644422
```

el modelo 2 presenta un R^2 más alto, por lo cual podria considerarse un poco mejor, hasta el momento.

Dibuja el diagrama de dispersión de los datos por pares y la recta de mejor ajuste.

In [ ]:
```
import matplotlib.pyplot as plt

plt.scatter(X1_test, y_test, color='blue')
plt.plot(X1_test, modelo1.predict(X1_test), color='red', linewidth=2)
plt.title('Regresión Lineal: Engine Size vs Precio')
plt.xlabel('Engine Size')
plt.ylabel('Precio')
plt.show()

plt.scatter(X2_test['enginesize'], y_test, color='blue')
plt.plot(X2_test['enginesize'], modelo2.predict(X2_test), color='red', linewidth=2)
plt.title('Regresión Lineal: Engine Size + Stroke vs Precio')
plt.xlabel('Engine Size')
plt.ylabel('Precio')
plt.show()
```


Regresión Lineal: Engine Size vs Precio
Precio
Engine Size
40000
30000
20000
10000
50
100
150
200
250
300


Regresión Lineal: Engine Size + Stroke vs Precio
Precio
Engine Size
40000
30000
20000
10000
50
100
150
200
250
300

Interpreta en el contexto del problema cada uno de los análisis que hiciste. Debajo de cada análisis viene la interpretación

Analiza la validez de los modelos propuestos:

Normalidad de los residuos

(Normalidad de los residuos de los modelos con test de Shapiro-Wilk)

In [ ]:
```
import scipy.stats as stats


residuos_modelo1 = y_train - modelo1.predict(X1_train)
print("Normalidad de los residuos del Modelo 1:", stats.shapiro(residuos_modelo1))

residuos_modelo2 = y_train - modelo2.predict(X2_train)
print("Normalidad de los residuos del Modelo 2:", stats.shapiro(residuos_modelo2))
```

```
Normalidad de los residuos del Modelo 1: ShapiroResult(statistic=0.9540757536888123,
pvalue=3.2809548429213464e-05)
Normalidad de los residuos del Modelo 2: ShapiroResult(statistic=0.9496104717254639,
pvalue=1.3171607861295342e-05)
```

En los 2 modelos, el p-valor es mucho menor que 0.05 (modelo 1 pvalue=3.280620088188724e-05 y modelo 2 pvalue=1.3170437670233928e-05), lo que indica que no se puede asumir la normalidad de los residuos.

Verificación de media cero

In [ ]:
```
print("Media de los residuos del Modelo 1:", np.mean(residuos_modelo1))
print("Media de los residuos del Modelo 2:", np.mean(residuos_modelo2))
```

```
Media de los residuos del Modelo 1: 1.3309678562530656e-13
Media de los residuos del Modelo 2: 1.1756882730235414e-12
```

Los 2 residuos tienen una media muy muy cercana a cero

Homocedasticidad, linealidad e independencia

In [ ]:
```
plt.scatter(modelo1.predict(X1_train), residuos_modelo1)
plt.title('Gráfico de Residuos vs Predicciones (Modelo 1)')
plt.xlabel('Predicciones')
plt.ylabel('Residuos')
plt.show()

plt.scatter(modelo2.predict(X2_train), residuos_modelo2)
plt.title('Gráfico de Residuos vs Predicciones (Modelo 2)')
plt.xlabel('Predicciones')
plt.ylabel('Residuos')
plt.show()
```

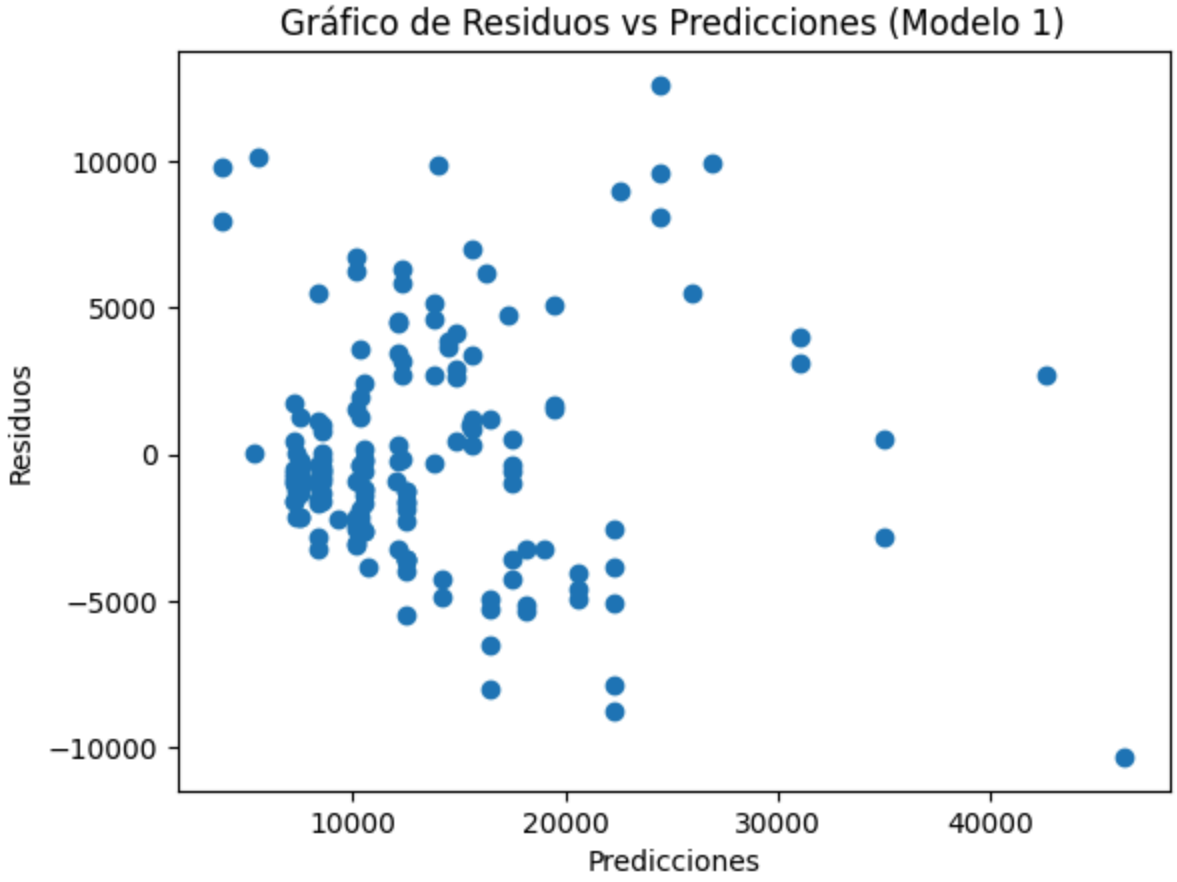


Gráfico de Residuos vs Predicciones (Modelo 1)
Residuos
Predicciones
10000
5000
0
−5000
−10000
10000
20000
30000
40000

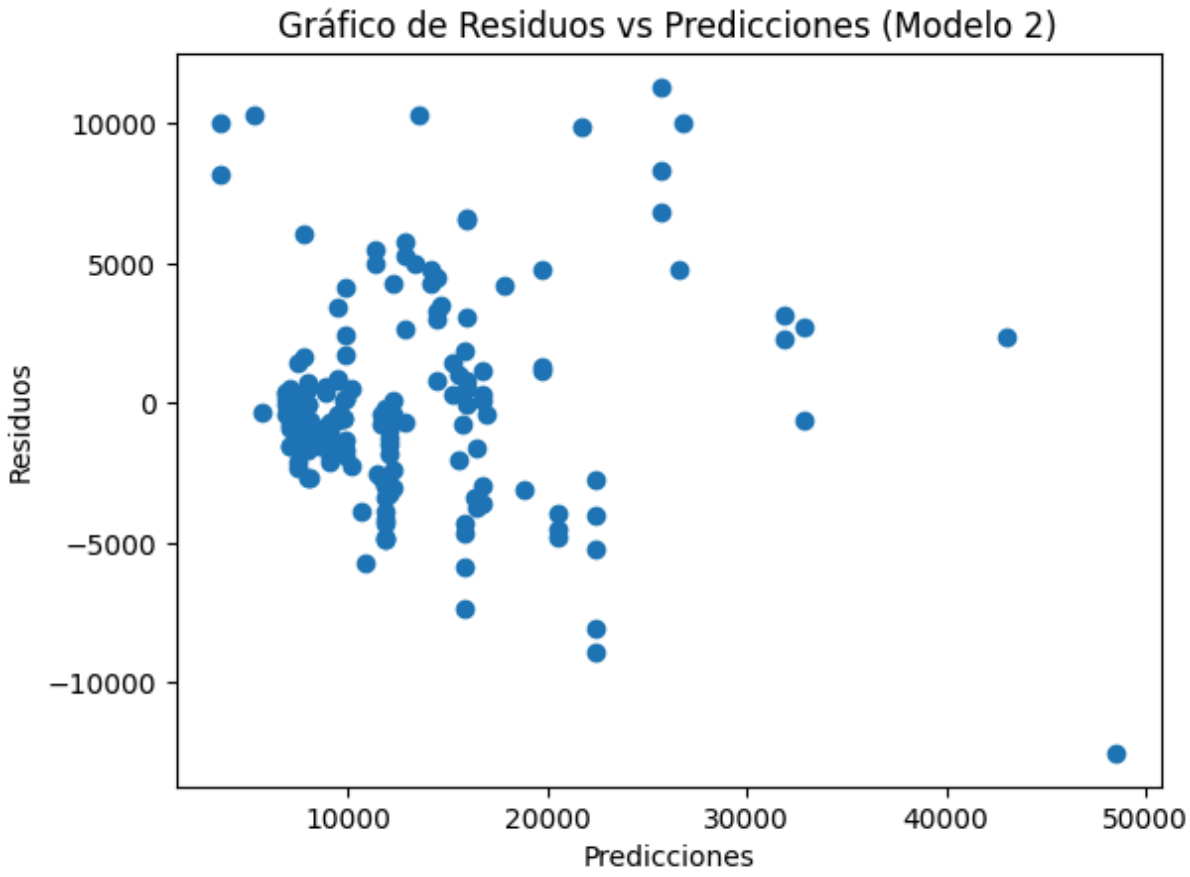


Gráfico de Residuos vs Predicciones (Modelo 2)
Residuos
Predicciones
10000
5000
0
−5000
−10000
10000
20000
30000
40000
50000

En las 2 graficas vemos que hay un poco de dispersión en los residuos, y vemos que la mayoría se agrupa cerca de cero, pero hay variabilidad en los extremos, lo que indica posibles problemas de homoscedasticidad,

Interpreta cada uno de los analisis que realizaste.

Normalidad de los residuos: Vemos que no se llega a cumplir en ninguno de los 2 modelos.

Media de los residuos: Cumple en los 2 modelos.

Homoscedasticidad: Por las visualizaciones obtenidas vemos que hay heteroscedasticidad en ambos modelos.

Emite una conclusión final sobre el mejor modelo de regresión lineal y contesta la pregunta central:

Concluye sobre el mejor modelo que encontraste y argumenta por qué es el mejor ¿Cuáles de las variables asignadas influyen en el precio del auto? ¿de qué manera lo hacen?

Considero que el mejor modelo es el 2, ya que es el que presenta un mejor ajuste, ya que:

El R^2 es mayor (0.765> 0.751), por lo cual que explica un poco más de la variabilidad en el precio del automóvil.

Coeficientes significativos: En el Modelo 2, tanto enginesize como stroke resultan ser predictores significativos con un alfa de 0.04, por loc ual es un modelo más comleto a comparación del modelo 1, ya que en el modelo 2 incluye más de un predictor importante.

A pesar de que los 2 modelos presentan problemas con la normalidad de los residuos y una posible heteroscedasticidad, el modelo 2 sigue siendo mejor explicando.

Asi que el modelo 2 es el mejor en este caso porque incluye tanto enginesize como stroke, lo que permite una mayor capacidad explicativa del precio del automóvil y ambas variables influyen significativamente en el precio.

Intervalos de predicción y confianza

Con los datos de las variables asignadas construye la gráfica de los intervalos de confianza y predicción para la estimación y predicción del precio para el mejor modelo seleccionado:

Calcula los intervalos para la variable Y

In [ ]:
```
import numpy as np
import statsmodels.api as sm
```

```
X2_train_sm = sm.add_constant(X2_train)
modelo2_sm = sm.OLS(y_train, X2_train_sm).fit()

intervalos_confianza = modelo2_sm.conf_int(alpha=0.05)
print("Intervalos de confianza para los coeficientes del Modelo 2:\n", intervalos_c

predicciones = modelo2_sm.get_prediction(sm.add_constant(X2_test))
intervalos_prediccion = predicciones.summary_frame(alpha=0.05)
print(intervalos_prediccion.head())
```

```
Intervalos de confianza para los coeficientes del Modelo 2:
                      0           1
const      -4766.135368  7388.632423
enginesize   154.937936   184.226095
stroke     -4790.751484 -1062.243909
             mean     mean_se  mean_ci_lower  mean_ci_upper  obs_ci_lower  \
15   26833.062566  673.092475   25503.834100   28162.291033  19250.107453
9    13576.400384  324.876360   12934.831919   14217.968848   6083.338925
100  11506.143375  365.752635   10783.852112   12228.434637   4005.739079
132  12846.324469  341.973204   12170.993019   13521.655920   5350.296688
68   21692.305740  588.019678   20531.079722   22853.531759  14136.989347

     obs_ci_upper
15   34416.017679
9    21069.461843
100  19006.547671
132  20342.352250
68   29247.622133
```

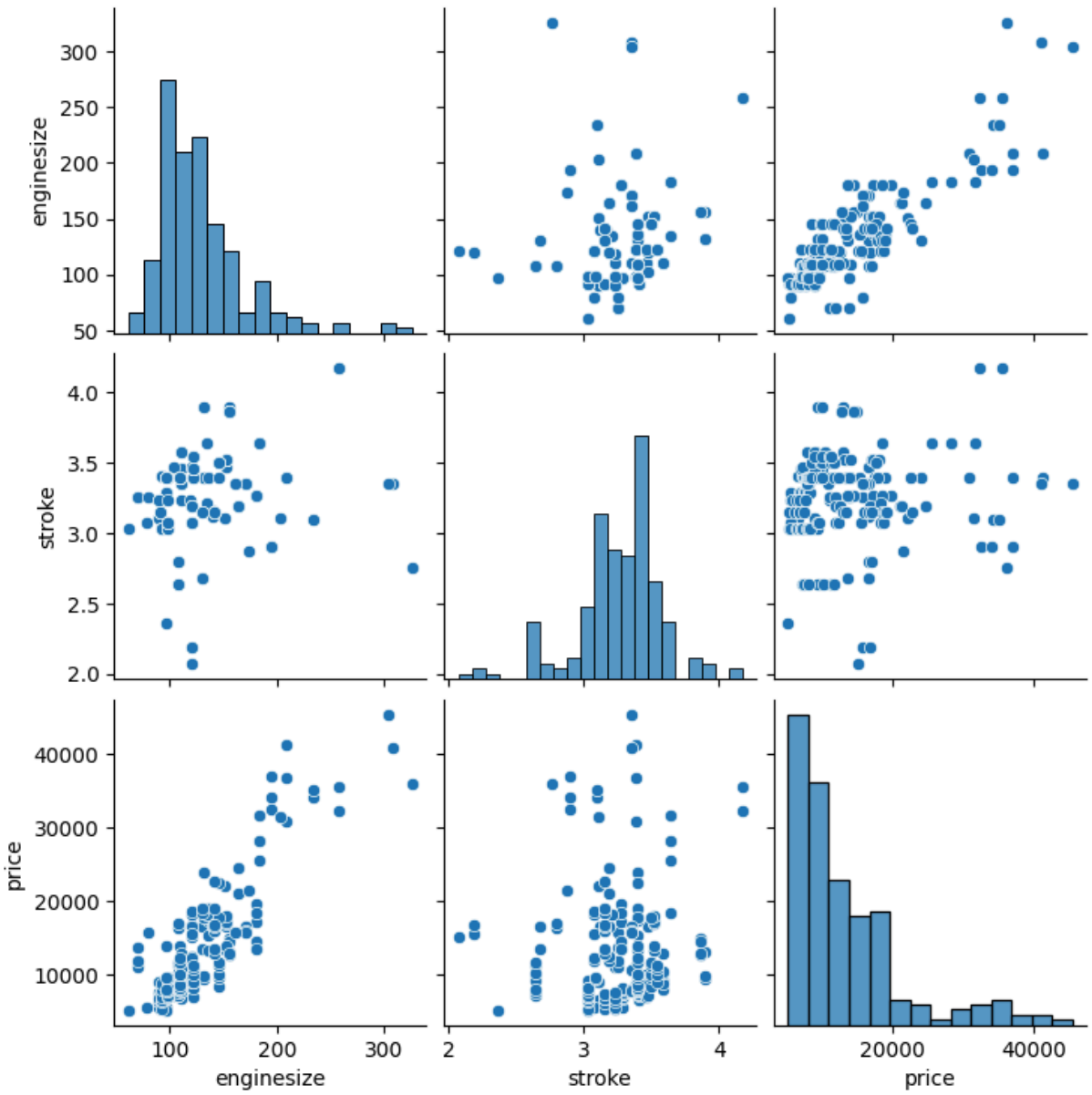
Selecciona la categoría de la variable cualitativa que, de acuerdo a tu análisis resulte la más importante, y separa la base de datos por esa variable categórica.

In [ ]:
```
df_front = df_autos[df_autos['enginelocation'] == 'front']
df_rear = df_autos[df_autos['enginelocation'] == 'rear']

print("Autos con motor en la parte delantera:\n", df_front.head())
print("Autos con motor en la parte trasera:\n", df_rear.head())
```

```
Autos con motor en la parte delantera:
    symboling                  CarName fueltype      carbody drivewheel  \
0           3        alfa-romero giulia      gas  convertible        rwd
1           3       alfa-romero stelvio      gas  convertible        rwd
2           1  alfa-romero Quadrifoglio      gas    hatchback        rwd
3           2               audi 100 ls      gas        sedan        fwd
4           2                audi 100ls      gas        sedan        4wd

  enginelocation  wheelbase  carlength  carwidth  carheight  ...  enginetype  \
0          front       88.6      168.8      64.1       48.8  ...        dohc
1          front       88.6      168.8      64.1       48.8  ...        dohc
2          front       94.5      171.2      65.5       52.4  ...        ohcv
3          front       99.8      176.6      66.2       54.3  ...         ohc
4          front       99.4      176.6      66.4       54.3  ...         ohc

  cylindernumber enginesize  stroke  compressionratio  horsepower  peakrpm  \
0           four        130    2.68               9.0         111     5000
1           four        130    2.68               9.0         111     5000
2            six        152    3.47               9.0         154     5000
3           four        109    3.40              10.0         102     5500
4           five        136    3.40               8.0         115     5500

   citympg  highwaympg    price
0       21          27  13495.0
1       21          27  16500.0
2       19          26  16500.0
3       24          30  13950.0
4       18          22  17450.0

[5 rows x 21 columns]
Autos con motor en la parte trasera:
      symboling           CarName fueltype      carbody drivewheel  \
126           3  porcshce panamera      gas      hardtop        rwd
127           3    porsche cayenne      gas      hardtop        rwd
128           3     porsche boxter      gas  convertible        rwd

    enginelocation  wheelbase  carlength  carwidth  carheight  ...  \
126           rear       89.5      168.9      65.0       51.6  ...
127           rear       89.5      168.9      65.0       51.6  ...
128           rear       89.5      168.9      65.0       51.6  ...

     enginetype cylindernumber enginesize  stroke  compressionratio  \
126        ohcf            six        194     2.9               9.5
127        ohcf            six        194     2.9               9.5
128        ohcf            six        194     2.9               9.5

     horsepower  peakrpm  citympg  highwaympg    price
126         207     5900       17          25  32528.0
127         207     5900       17          25  34028.0
128         207     5900       17          25  37028.0

[3 rows x 21 columns]
```

Grafica por pares de variables numéricas

In [ ]:
```
import seaborn as sns
import matplotlib.pyplot as plt

sns.pairplot(df_autos[['enginesize', 'stroke', 'price']])
plt.show()
```



Interpreta en el contexto del problema:

El enginesize tiene un impacto positivo significativo en el precio, y el stroke afecta negativamente.

Los autos con motor delantero son más comunes y más económicos, y por el contrariolos autos con motor trasero son más caros.

Y en la grafica vemos que hay una correlación clara y positiva entre el tamaño del motor y el precio. La relación entre el stroke y el precio es más dispersa y negativa.

Más allá:

Contesta la pregunta referida a la agrupación de variables que propuso la empresa para el análisis: ¿propondrías una nueva agrupación de las variables a la empresa automovilísitica?

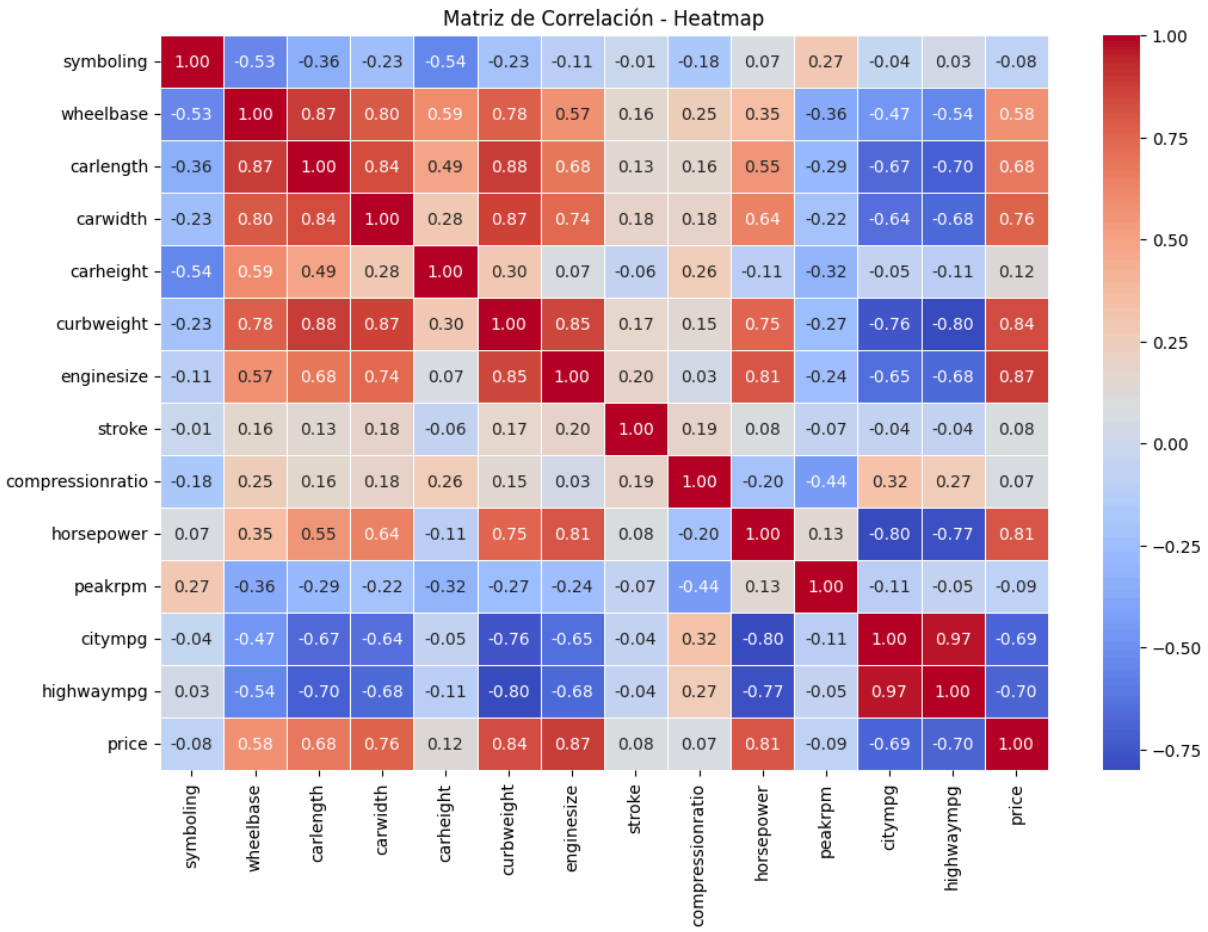
Si se podrian sugerir nuevas agrupaciones de variables, si tratamos el lado del rendimiento del auto podriamos enfocarnos en el motor con las siguientes variables: enginesize, stroke y horsepower ya que estas variables influyen directamente en el rendimiento del motor y por ende en el proecio final del auto.

Si nos enfocaramos en las dimensiones del vehiculo podriamos elegir las siguientes variables: wheelbase, carlength , carwidth y carheight ya que esta variables se encargan de la representación fisica del auto es decir si es carró, camioneta, usv ,etc.

Y el conjunto de ese grupo de varibles nos ayudaria a enfocarnos en esas áreas para optener mejores resultados.

Retoma todas las variables y haz un análisis estadístico muy leve (medias y correlación) de cómo crees que se deberían agrupar para analizarlas.

In [ ]:
```
import seaborn as sns
df_numerico = df_autos.select_dtypes(include=['float64', 'int64'])

correlacion_completa = df_numerico.corr()

plt.figure(figsize=(12, 8))
sns.heatmap(correlacion_completa, annot=True, cmap='coolwarm', fmt='.2f', linewidth
plt.title('Matriz de Correlación - Heatmap')
plt.show()
```



Como vemos hay una alta correlación entre las varaibles wheelbase, carlenght , carwidth y curbweight ya que reflejan el tamaño y peso del automovil.

Y la vraiables: enginesize , horsepower , stroke y compressionratio se correlacionan con el rendimiento del montos

si tenemos a la variables ; price, curbweight, enginesize y horsepower vemos qye habra una alata correlación en el precio lo cual determina el valor del vehículo en el mercado.

Y con estas agrupaciones sirven para facilitar el análisis de cada aspecto clave del vehículo.