Modelo OLS para el dataset de emisiones de CO2 en Canada

Importación de librerias

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.api as sm
from scipy.stats import norm, uniform, skewnorm
```

Lectura de los datos

```
df = pd.read_csv('/content/CO2 Emissions_Canada.csv')

df.columns

Index(['Make', 'Model', 'Vehicle Class', 'Engine Size(L)',
'Cylinders',
       'Transmission', 'Fuel Type', 'Fuel Consumption City (L/100
km)',
       'Fuel Consumption Hwy (L/100 km)', 'Fuel Consumption Comb
(L/100 km)',
       'Fuel Consumption Comb (mpg)', 'CO2 Emissions(g/km)'],
      dtype='object')
```

Verificación de valores nulos

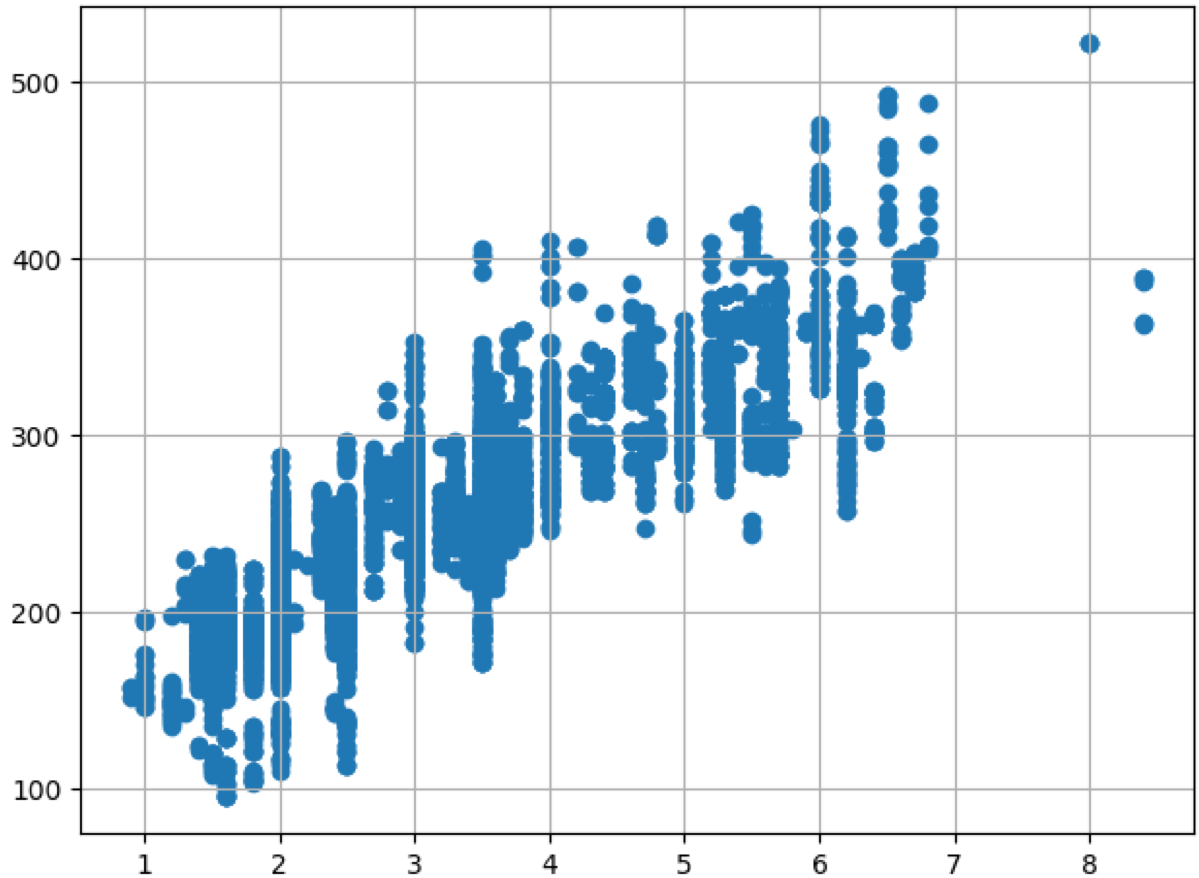
```
df.isnull().sum()

Make                                0
Model                               0
Vehicle Class                       0
Engine Size(L)                      0
Cylinders                           0
Transmission                        0
Fuel Type                           0
Fuel Consumption City (L/100 km)    0
Fuel Consumption Hwy (L/100 km)     0
Fuel Consumption Comb (L/100 km)    0
Fuel Consumption Comb (mpg)         0
CO2 Emissions(g/km)                 0
dtype: int64
```

Asignación de variable dependiente y variables independientes

```
y = df.iloc[:,11]
x = df.iloc[:,3]#[:,0:10]
```

Gráfico de dispersión de la variable independiente

```
plt.scatter(x,y)
plt.grid(True)
plt.tight_layout()
```




```
X = sm.add_constant(x)
```

Creación y ajuste del modelo de regresión lineal

```
model = sm.OLS(y,X)
result = model.fit()
print(result.params)
```

```
const                134.365893
Engine Size(L)        36.777315
dtype: float64
```

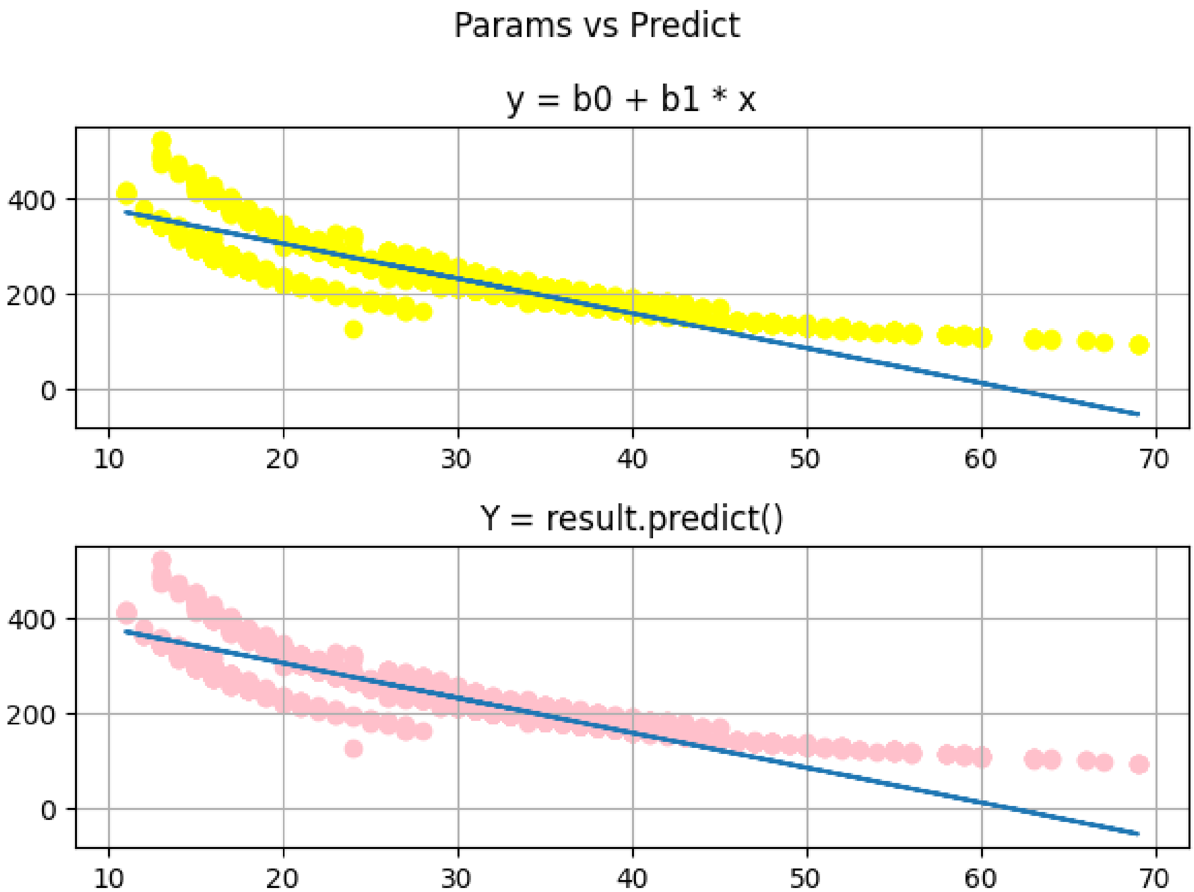
Gráficas de predicción

Se analizaron los diferentes valores de r^2 de las variables independientes y se graficaron las predicciones de los mejores, la gráfica esta dividida en el valor obtenido con la formula, y el valor obtenido con la función

```
X = sm.add_constant(df['Fuel Consumption Comb (mpg)'])
model = sm.OLS(y,X)
result3 = model.fit()
print(result3.params)
print('r^2 =',result.rsquared)
```

```
const                           452.353036
Fuel Consumption Comb (mpg)      -7.341929
dtype: float64
r^2 = 0.6932953649936133
```

```
fig, axs = plt.subplots(2)
fig.suptitle('Params vs Predict')
axs[0].scatter(X['Fuel Consumption Comb (mpg)'],y, color = 'yellow')
b0,b1 = result3.params
Y = b0 + b1*X['Fuel Consumption Comb (mpg)']
axs[0].plot(X['Fuel Consumption Comb (mpg)'],Y)
axs[0].grid(True)

axs[1].scatter(X['Fuel Consumption Comb (mpg)'],y, color = 'pink')
axs[1].plot(X['Fuel Consumption Comb (mpg)'],result3.predict())
axs[1].grid(True)

axs[0].set_title('y = b0 + b1 * x')
axs[1].set_title('Y = result.predict()')

plt.tight_layout()
```

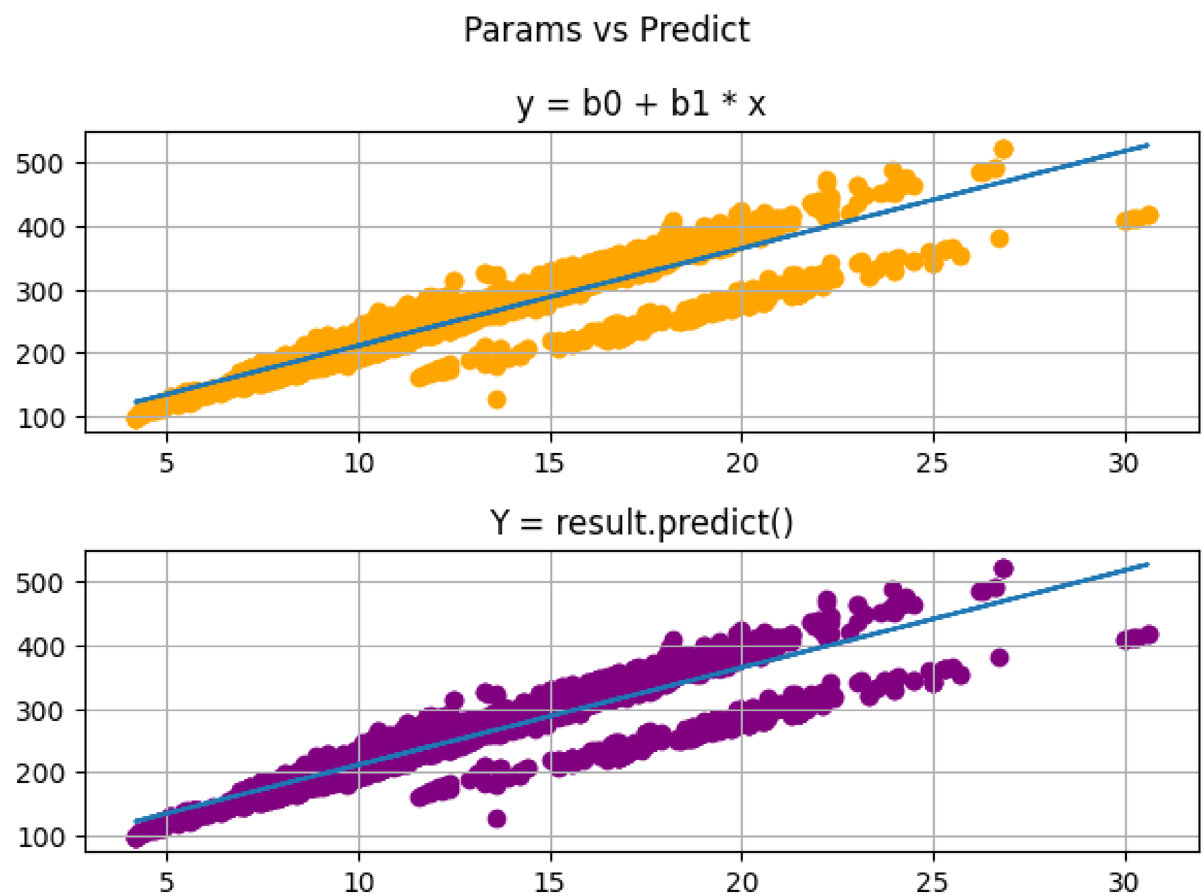



```
X = sm.add_constant(df['Fuel Consumption City (L/100 km)'])
model = sm.OLS(y,X)
result4 = model.fit()
print(result4.params)
print('r^2 =',result.rsquared)
```

```
const                               57.559903
Fuel Consumption City (L/100 km)    15.372459
dtype: float64
r^2 = 0.7244472046524082
```

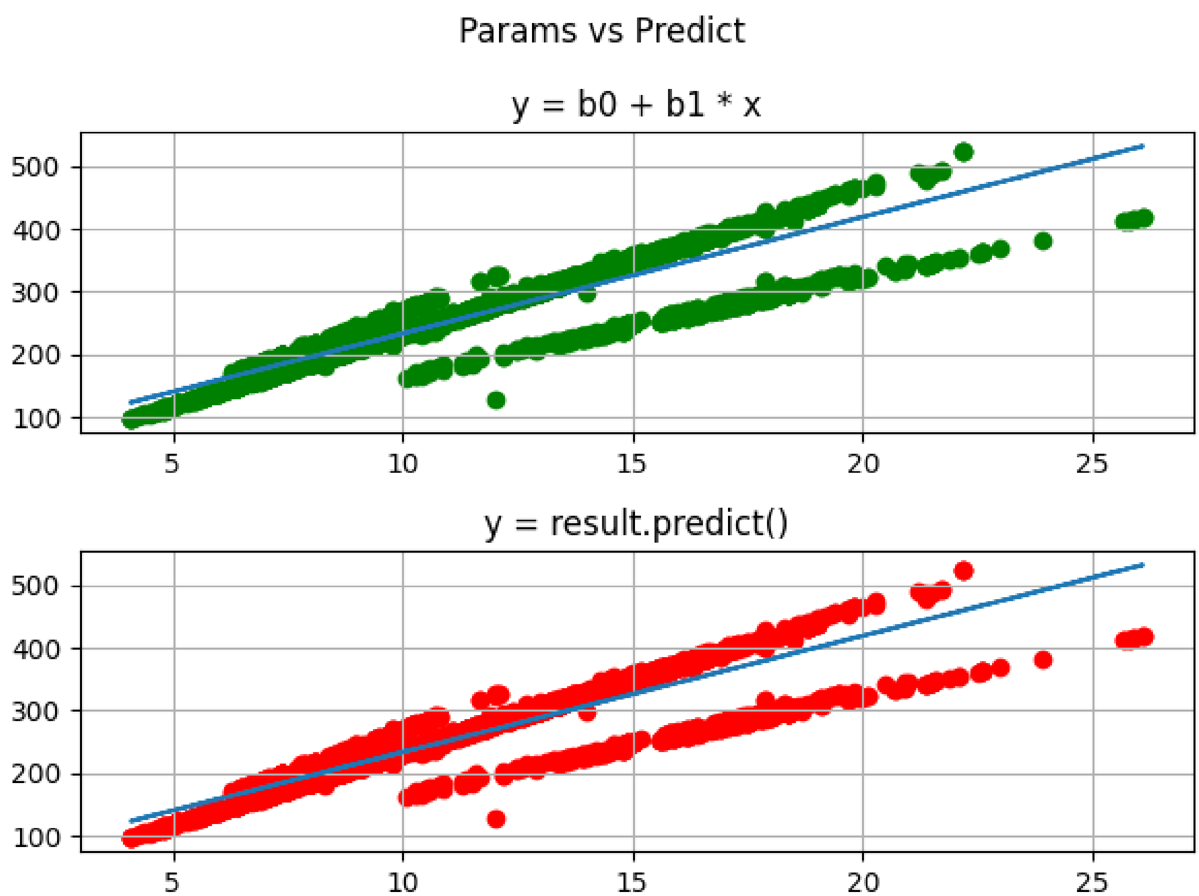
```
fig, axs = plt.subplots(2)
fig.suptitle('Params vs Predict')
axs[0].scatter(X['Fuel Consumption City (L/100 km)'],y, color =
'orange')
b0,b1 = result4.params
Y = b0 + b1*X['Fuel Consumption City (L/100 km)']
axs[0].plot(X['Fuel Consumption City (L/100 km)'],Y)
axs[0].grid(True)

axs[1].scatter(X['Fuel Consumption City (L/100 km)'],y, color =
'purple')
```

```
axs[1].plot(X['Fuel Consumption City (L/100 km)'],result4.predict())
axs[1].grid(True)

axs[0].set_title('y = b0 + b1 * x')
axs[1].set_title('Y = result.predict()')

plt.tight_layout()
```




```
X = sm.add_constant(df['Fuel Consumption Comb (L/100 km)'])
model = sm.OLS(y,X)
result5 = model.fit()
print(result5.params)
print('r^2 =',result.rsquared)

const                               46.763152
Fuel Consumption Comb (L/100 km)    18.571319
dtype: float64
r^2 = 0.7244472046524082

fig, axs = plt.subplots(2)
fig.suptitle('Params vs Predict')
axs[0].scatter(X.iloc[:,1],y, color = 'green')
```

```
b0,b1 = result5.params
Y = b0 + b1*X.iloc[:,1]
axs[0].plot(X.iloc[:,1],Y)
axs[0].grid(True)

axs[1].scatter(X.iloc[:,1],y, color = 'red')
axs[1].plot(X.iloc[:,1],result5.predict())
axs[1].grid(True)

axs[0].set_title('y = b0 + b1 * x')
axs[1].set_title('y = result.predict()')

plt.tight_layout()
```



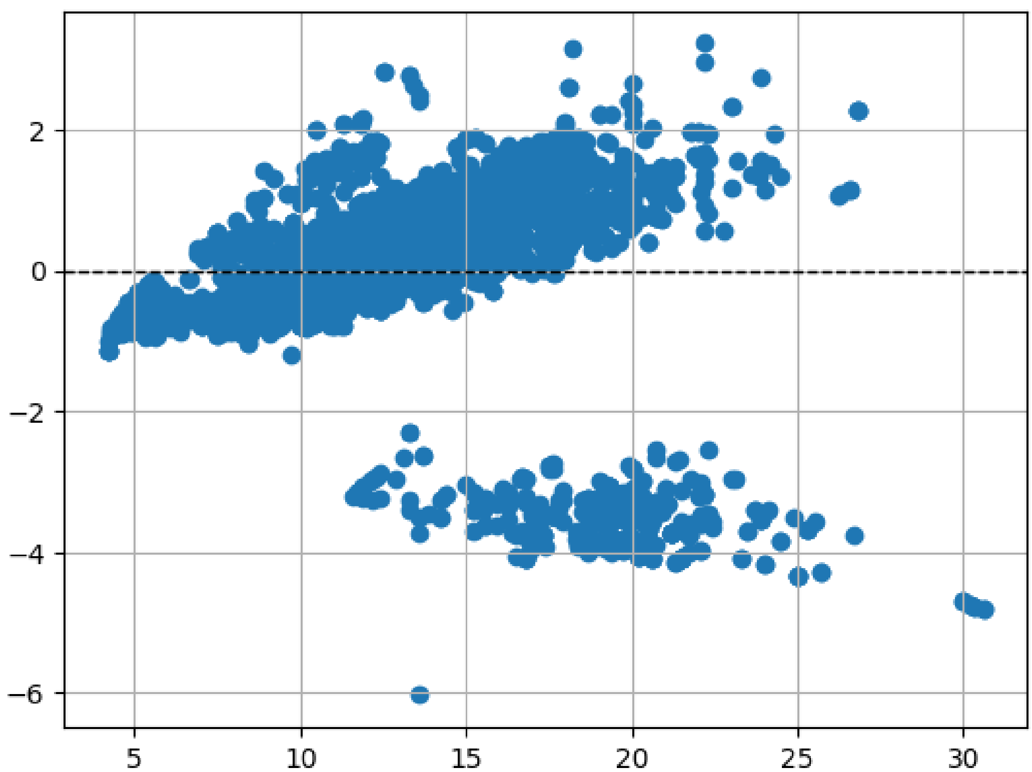
Analizando las gráficas obtenidas se puede determinar que entre las gráficas de consumo en ciudad y consumo combinado por litro son bastante similares, esto se refleja en el r^2 obtenida de cada uno los cuales son casi identicos.

Residuos estandarizados y QQ plots

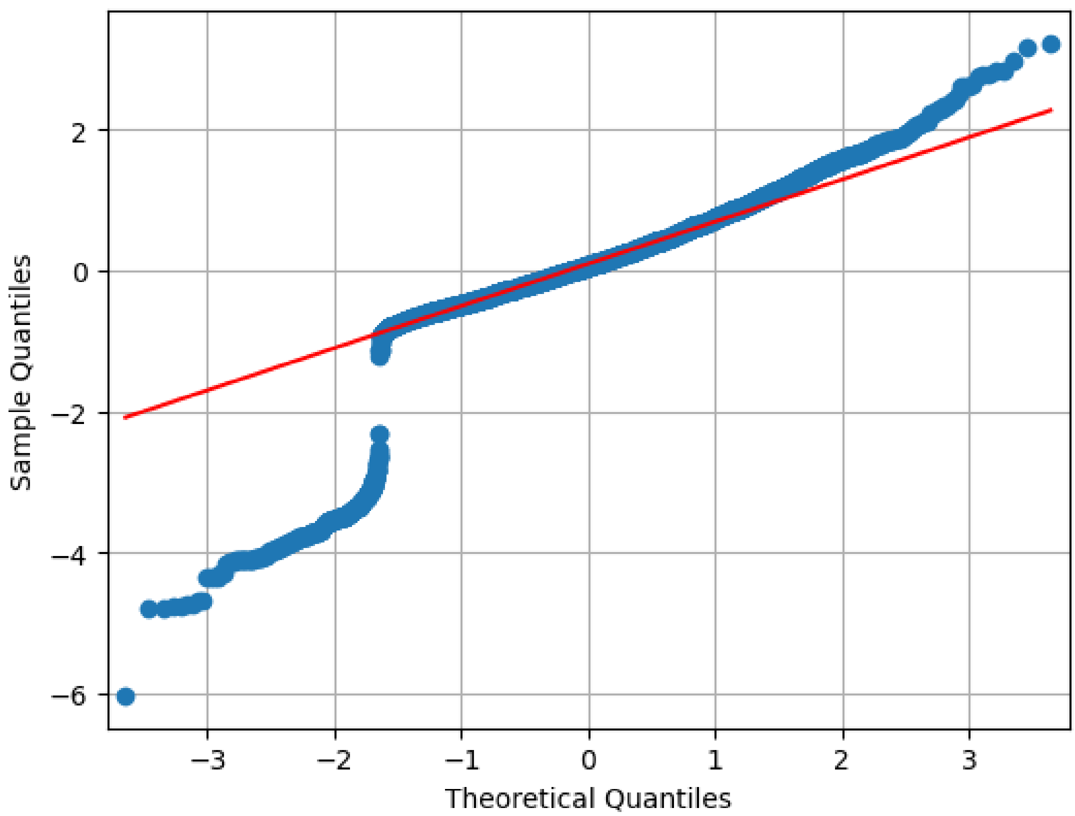
Se analiza la distribución de los residuos de las variables previamente analizadas, y se analiza su gráfico qq-plot

```
X = sm.add_constant(df['Fuel Consumption City (L/100 km)'])
model = sm.OLS(y,X)
result = model.fit()

influence = result.get_influence()
standarized_residuals = influence.resid_studentized_internal

plt.scatter(X.iloc[:,1],standarized_residuals)
plt.axhline(y=0, color='black', linestyle='--', linewidth=1)
print('Distribution graph')
plt.grid()
plt.show()

print("\n")
fig = sm.qqplot(standarized_residuals, dist=norm, line='q')

print('QQ Graph - normal distribution')
plt.y_label=('Standarized residuals quantiles')
plt.grid()
plt.show()
```

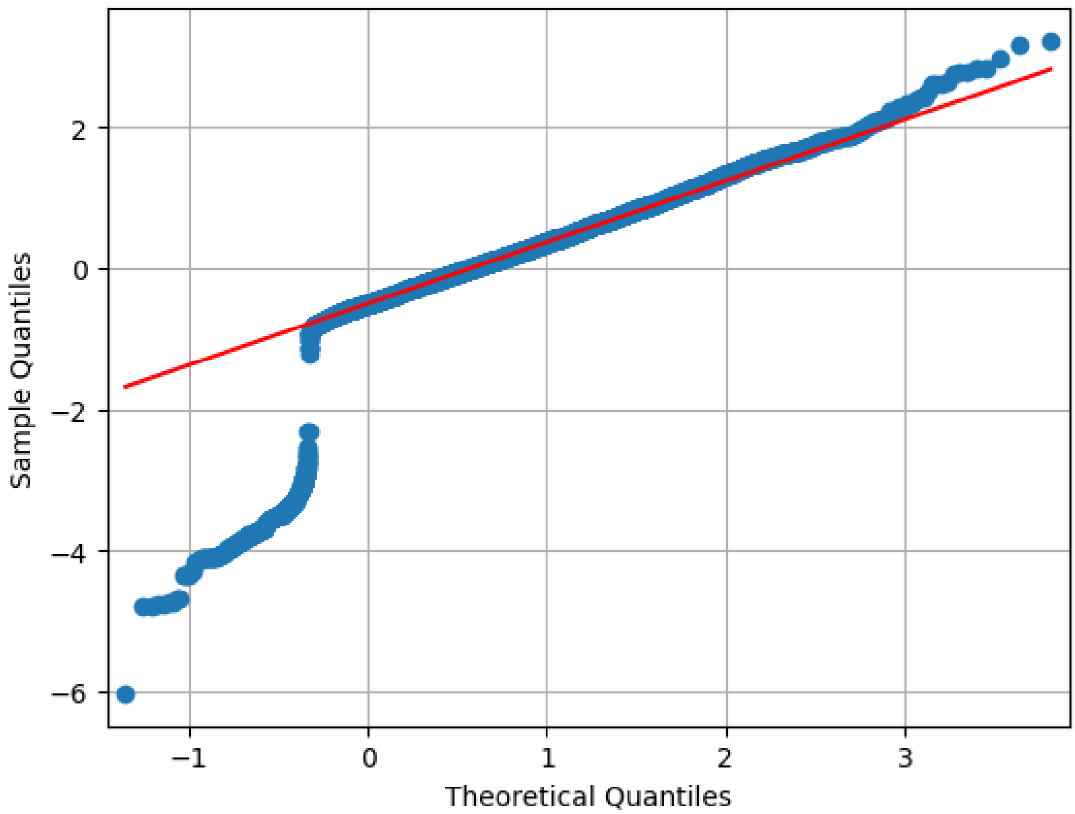
```
Distribution graph
```



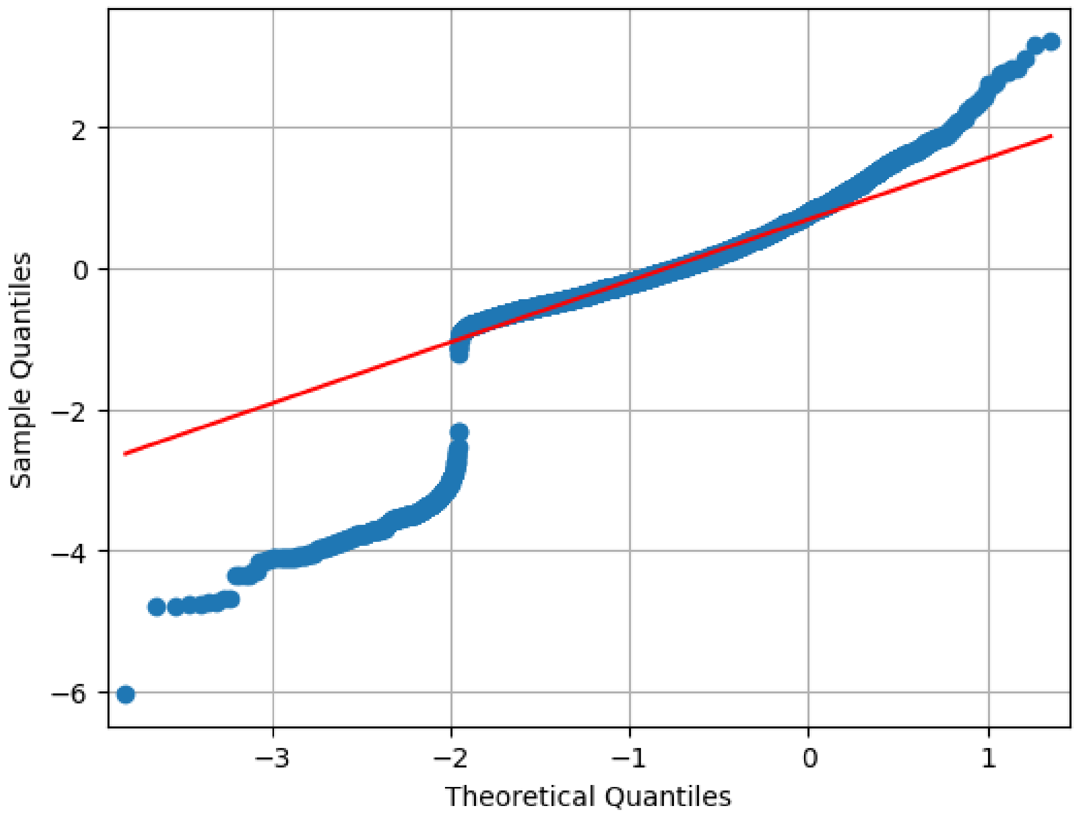
```
QQ Graph - normal distribution
```




```
fig = sm.qqplot(standarized_residuals, dist=skewnorm(2), line='q')

print('QQ Graph - positive skew normal distribution')
plt.y_label=('Standarized residuals quantiles')
plt.grid()
plt.show()

fig = sm.qqplot(standarized_residuals, dist=skewnorm(-2), line='q')

print('QQ Graph - positive skew normal distribution')
plt.y_label=('Standarized residuals quantiles')
plt.grid()
plt.show()

QQ Graph - positive skew normal distribution
```

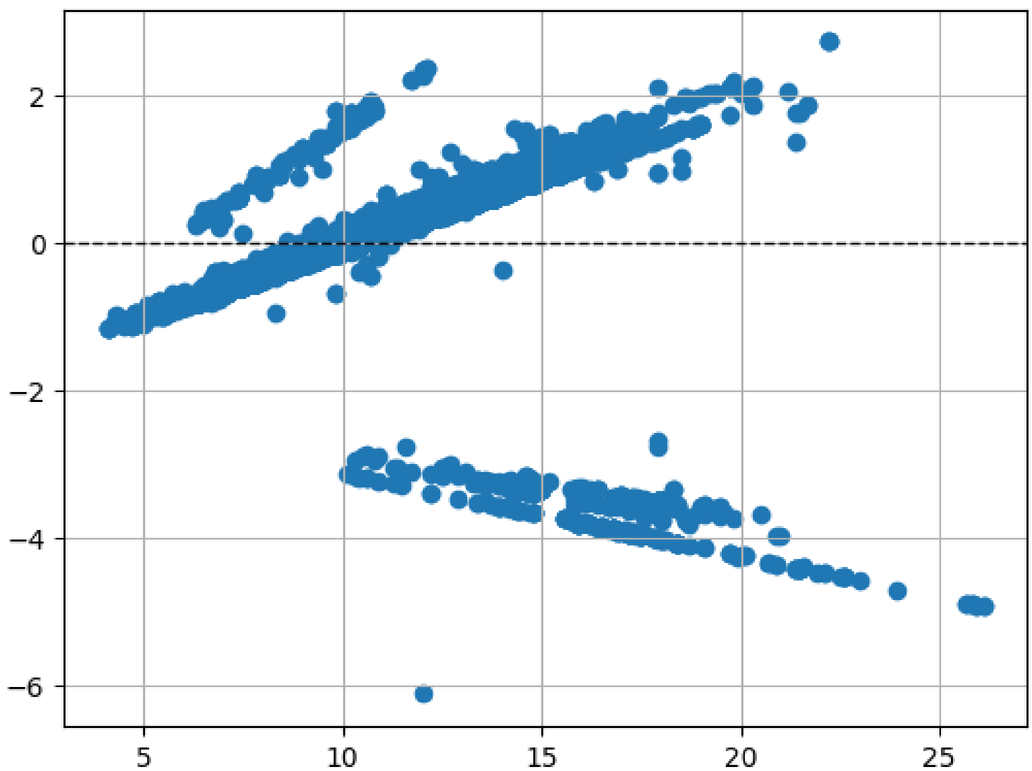



```
QQ Graph - positive skew normal distribution
```

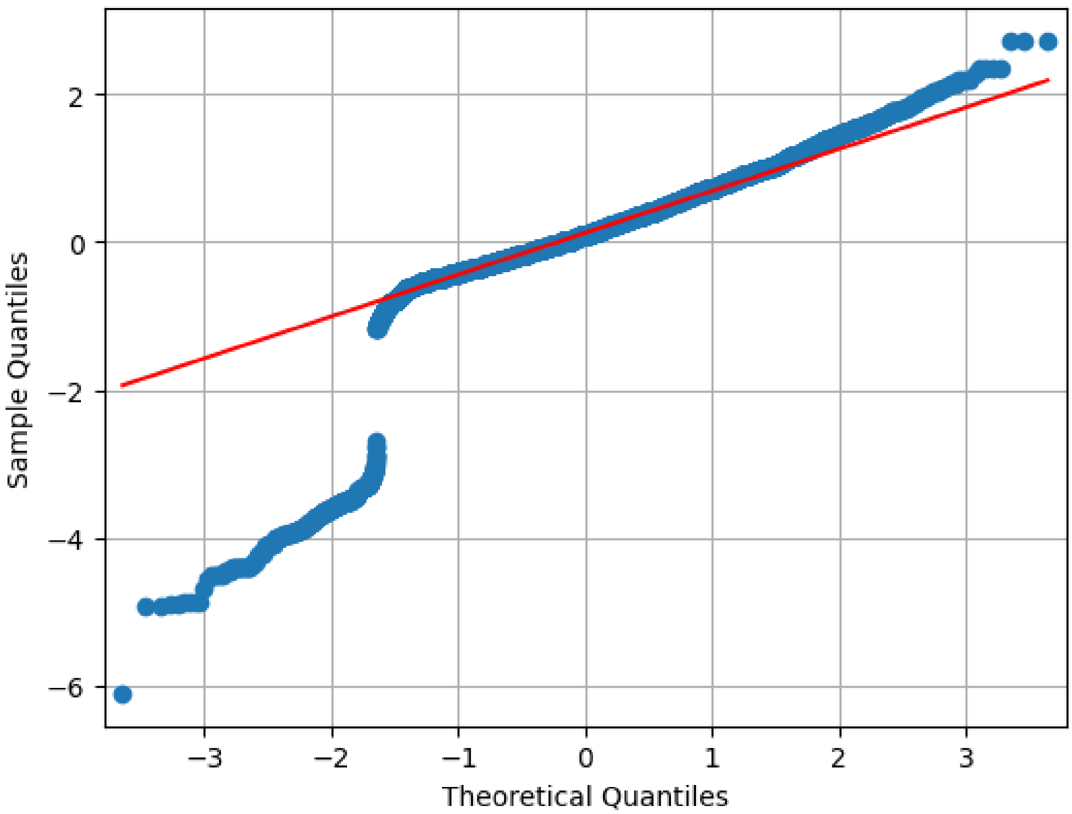



```
X = sm.add_constant(df['Fuel Consumption Comb (L/100 km)'])
model = sm.OLS(y,X)
result = model.fit()

influence = result.get_influence()
standarized_residuals = influence.resid_studentized_internal

plt.scatter(X.iloc[:,1],standarized_residuals)
plt.axhline(y=0, color='black', linestyle='--', linewidth=1)
print('Distribution graph')
plt.grid()
plt.show()

print("\n")
fig = sm.qqplot(standarized_residuals, dist=norm, line='q')

print('QQ Graph - normal distribution')
plt.y_label=('Standarized residuals quantiles')
plt.grid()
plt.show()
```

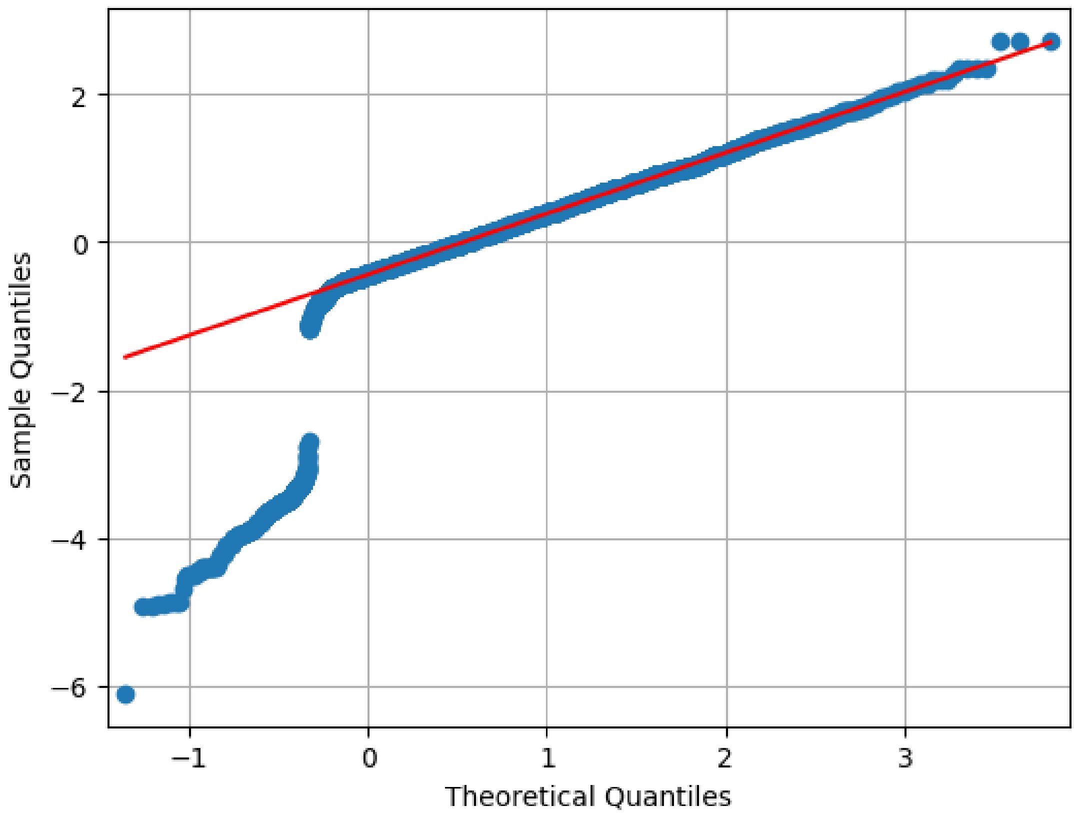
```
Distribution graph
```



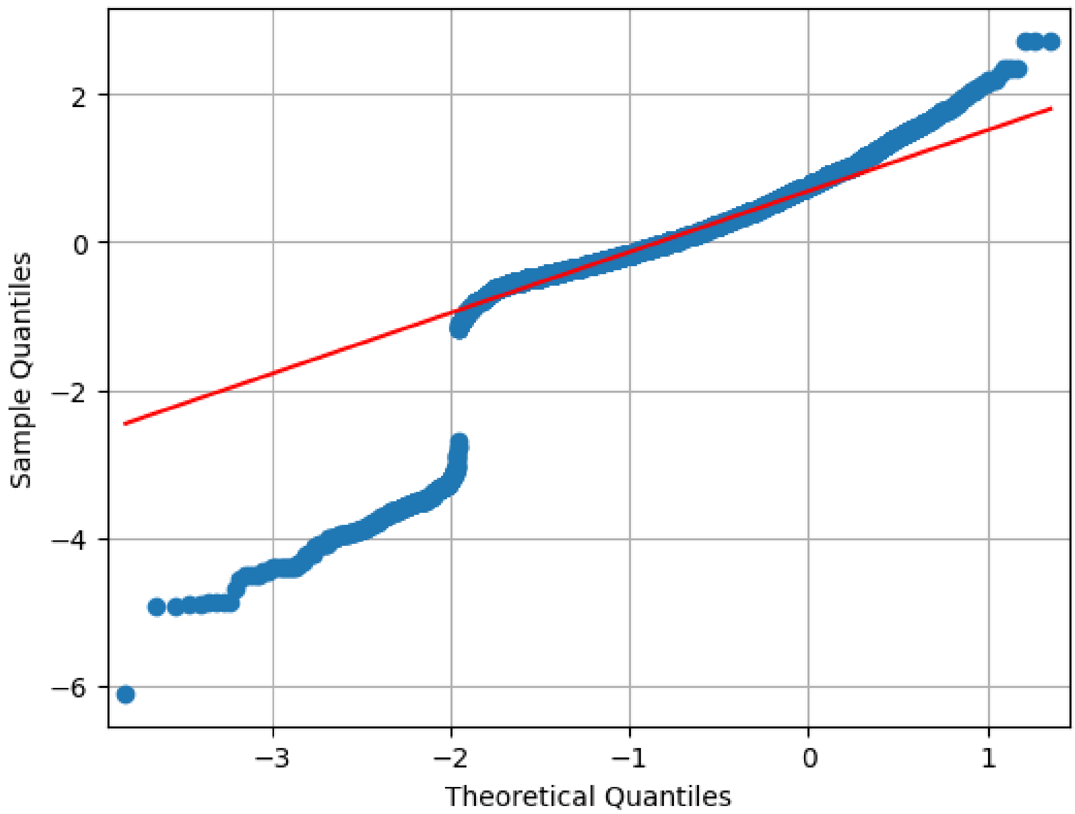

```
QQ Graph - normal distribution
```



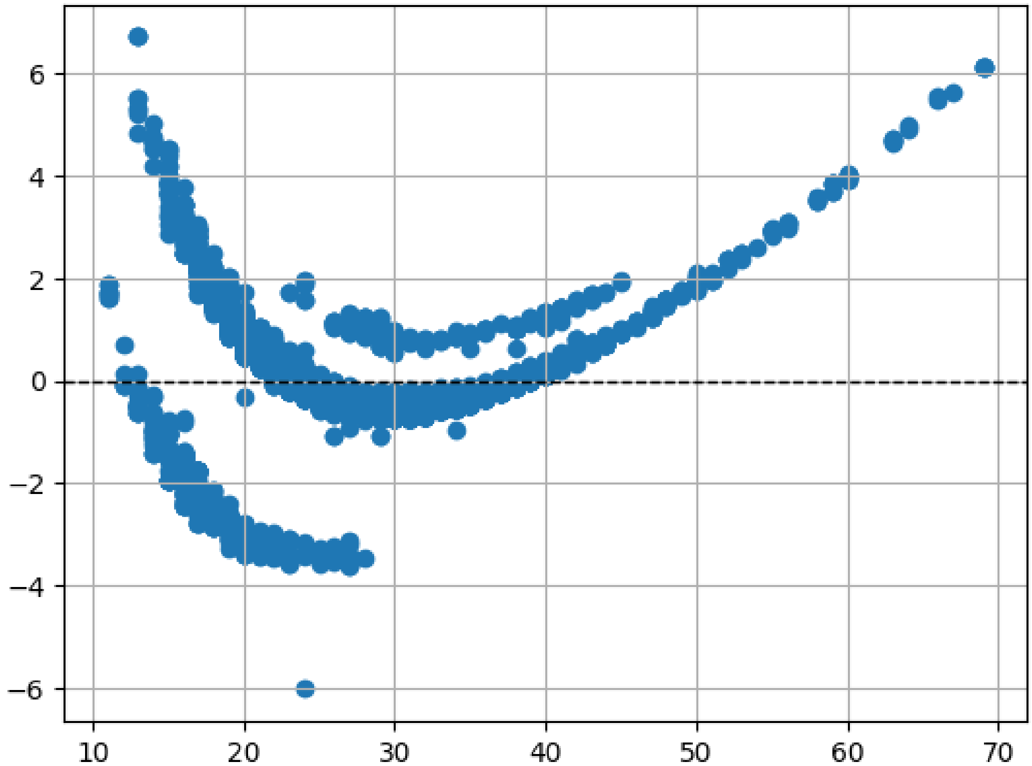

```
fig = sm.qqplot(standarized_residuals, dist=skewnorm(2), line='q')

print('QQ Graph - positive skew normal distribution')
plt.y_label=('Standarized residuals quantiles')
plt.grid()
plt.show()

fig = sm.qqplot(standarized_residuals, dist=skewnorm(-2), line='q')

print('QQ Graph - positive skew normal distribution')
plt.y_label=('Standarized residuals quantiles')
plt.grid()
plt.show()

QQ Graph - positive skew normal distribution
```



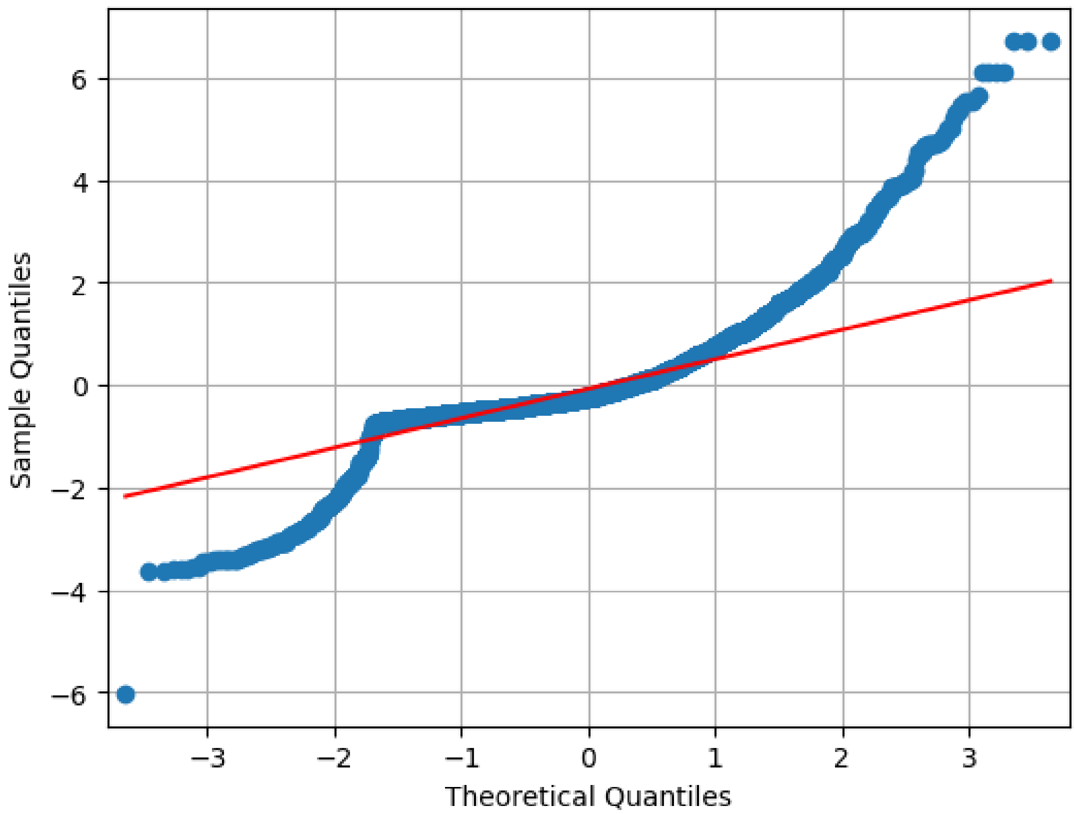

```
QQ Graph - positive skew normal distribution
```




```
X = sm.add_constant(df['Fuel Consumption Comb (mpg)'])
model = sm.OLS(y,X)
result = model.fit()

influence = result.get_influence()
standarized_residuals = influence.resid_studentized_internal

plt.scatter(X.iloc[:,1],standarized_residuals)
plt.axhline(y=0, color='black', linestyle='--', linewidth=1)
print('Distribution graph')
plt.grid()
plt.show()

print("\n")
fig = sm.qqplot(standarized_residuals, dist=norm, line='q')

print('QQ Graph - normal distribution')
plt.y_label=('Standarized residuals quantiles')
plt.grid()
plt.show()

Distribution graph
```

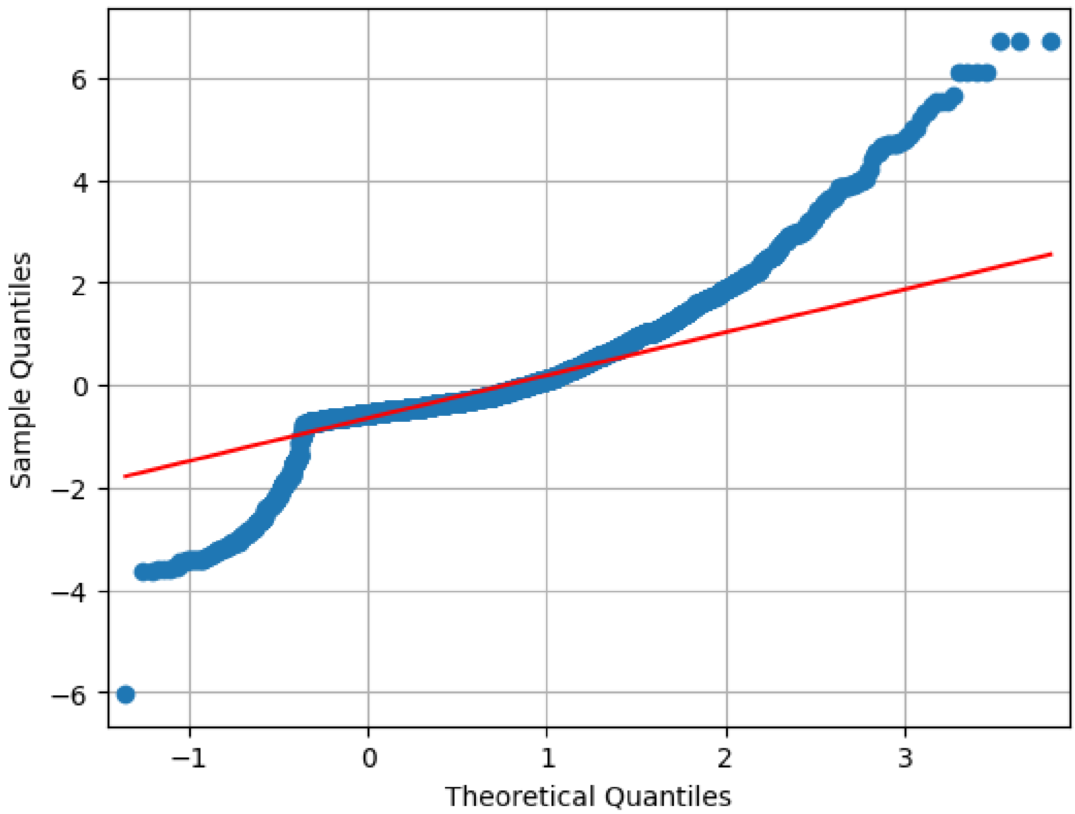



```
QQ Graph - normal distribution
```

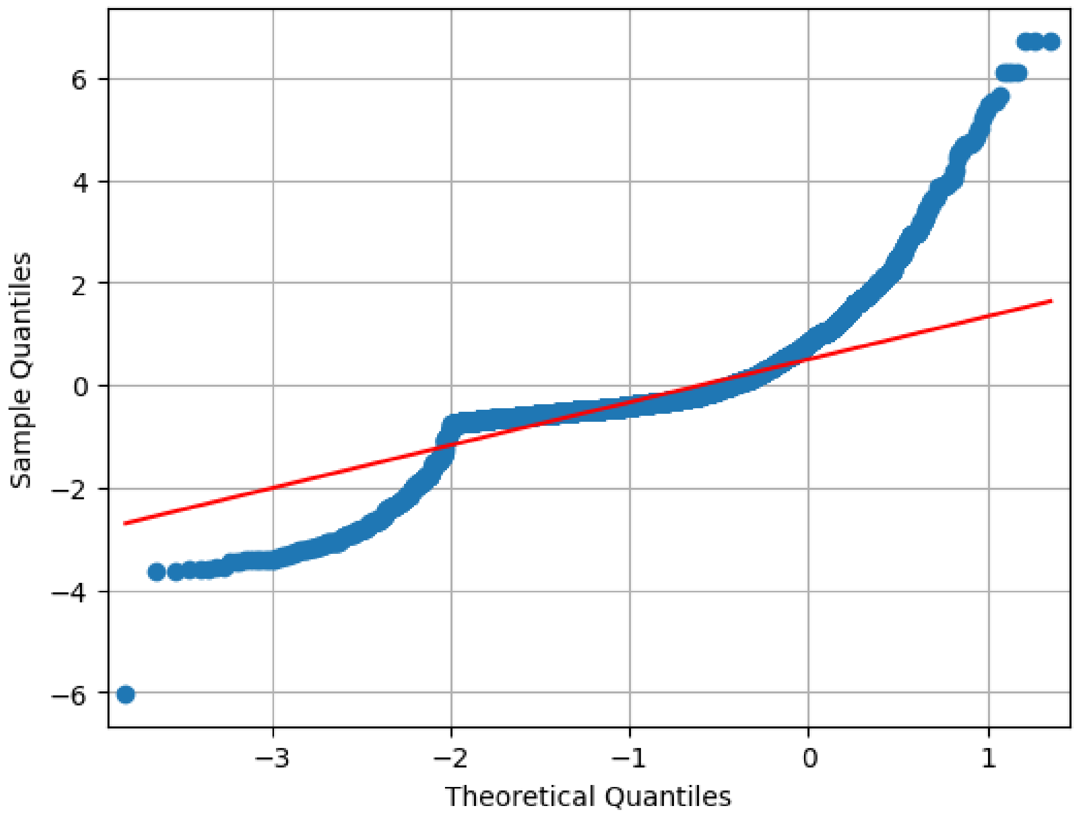



```
fig = sm.qqplot(standarized_residuals, dist=skewnorm(2), line='q')

print('QQ Graph - positive skew normal distribution')
plt.y_label=('Standarized residuals quantiles')
plt.grid()
plt.show()

fig = sm.qqplot(standarized_residuals, dist=skewnorm(-2), line='q')

print('QQ Graph - positive skew normal distribution')
plt.y_label=('Standarized residuals quantiles')
plt.grid()
plt.show()
```

```
QQ Graph - positive skew normal distribution
```




```
QQ Graph - positive skew normal distribution
```



Información de los modelos

Se imprime el resumen de los diferentes modelos que se evaluaron

```
print("\nFuel Consumption Comb (mpg)")
print(result3.summary())
print("\nFuel Consumption City (L/100 km)")
print(result4.summary())
print("\nFuel Consumption Comb (L/100 km)")
print(result5.summary())


Fuel Consumption Comb (mpg)
                            OLS Regression Results

======================================================================
=========
Dep. Variable:     CO2 Emissions(g/km)   R-squared:
0.823
Model:                             OLS   Adj. R-squared:
0.823
Method:                  Least Squares   F-statistic:
3.443e+04
Date:                 Fri, 06 Oct 2023   Prob (F-statistic):
0.00
```

```
Time:                            22:45:49   Log-Likelihood:
-34127.
No. Observations:                    7385   AIC:
6.826e+04
Df Residuals:                        7383   BIC:
6.827e+04
Df Model:                               1

Covariance Type:            nonrobust

======================================================================
=========================
                                  coef    std err          t      P>|
t|      [0.025      0.975]
----------------------------------------------------------------------
-------------------------
const                         452.3530      1.124    402.297
0.000     450.149     454.557
Fuel Consumption Comb (mpg)    -7.3419      0.040   -185.550
0.000      -7.419      -7.264
======================================================================
========
Omnibus:                     1935.010   Durbin-Watson:
1.326
Prob(Omnibus):                  0.000   Jarque-Bera (JB):
13170.162
Skew:                           1.080   Prob(JB):
0.00
Kurtosis:                       9.176   Cond. No.
112.
======================================================================
========

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

Fuel Consumption City (L/100 km)
                            OLS Regression Results

======================================================================
=========
Dep. Variable:     CO2 Emissions(g/km)   R-squared:
0.846
Model:                             OLS   Adj. R-squared:
0.846
Method:                  Least Squares   F-statistic:
4.045e+04
Date:                 Fri, 06 Oct 2023   Prob (F-statistic):
0.00
```

```
Time:                            22:45:49   Log-Likelihood:
-33630.
No. Observations:                    7385   AIC:
6.726e+04
Df Residuals:                        7383   BIC:
6.728e+04
Df Model:                               1

Covariance Type:            nonrobust

======================================================================
==============================
                                       coef    std err          t
P>|t|      [0.025      0.975]
----------------------------------------------------------------------
------------------------------
const                               57.5599      0.996     57.772
0.000      55.607      59.513
Fuel Consumption City (L/100 km)    15.3725      0.076    201.122
0.000      15.223      15.522
======================================================================
========
Omnibus:                     3089.403   Durbin-Watson:
1.913
Prob(Omnibus):                  0.000   Jarque-Bera (JB):
16424.392
Skew:                          -1.963   Prob(JB):
0.00
Kurtosis:                       9.161   Cond. No.
48.8
======================================================================
========

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

Fuel Consumption Comb (L/100 km)
                            OLS Regression Results

======================================================================
=========
Dep. Variable:     CO2 Emissions(g/km)   R-squared:
0.843
Model:                             OLS   Adj. R-squared:
0.843
Method:                  Least Squares   F-statistic:
3.959e+04
Date:                 Fri, 06 Oct 2023   Prob (F-statistic):
0.00
```

```
Time:                            22:45:49   Log-Likelihood:
-33697.
No. Observations:                    7385   AIC:
6.740e+04
Df Residuals:                        7383   BIC:
6.741e+04
Df Model:                               1

Covariance Type:            nonrobust

======================================================================
==============================
                                       coef    std err          t
P>|t|      [0.025      0.975]
----------------------------------------------------------------------
------------------------------
const                               46.7632      1.059     44.142
0.000      44.686      48.840
Fuel Consumption Comb (L/100 km)    18.5713      0.093    198.968
0.000      18.388      18.754
======================================================================
========
Omnibus:                     3592.018   Durbin-Watson:
1.986
Prob(Omnibus):                  0.000   Jarque-Bera (JB):
22309.895
Skew:                          -2.290   Prob(JB):
0.00
Kurtosis:                      10.178   Cond. No.
44.9
======================================================================
========

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

Evaluación con más de una variable independiente

```
Y = df['CO2 Emissions(g/km)']
X = df.loc[:, ['Engine Size(L)', 'Cylinders', 'Fuel Consumption City
(L/100 km)', 'Fuel Consumption Comb (L/100 km)', 'Fuel Consumption
Comb (mpg)']]

X = sm.add_constant(X)
```

Modelo

Se crea el modelo con todas las variables independientes numéricas

```
model = sm.OLS(Y,X)
results = model.fit()
```

Se imprimen los parametros de las variables asi como el r^2

```
print(results.params)
print('\nr^2 =',results.rsquared)
```

```
const                                        227.777330
Engine Size(L)                                 4.984745
Cylinders                                      7.528927
Fuel Consumption City (L/100 km)              -5.336445
Fuel Consumption Comb (L/100 km)              11.456620
Fuel Consumption Comb (mpg)                   -3.418640
dtype: float64

r^2 = 0.9038551538200387
```

Se imprime un resumen del modelo

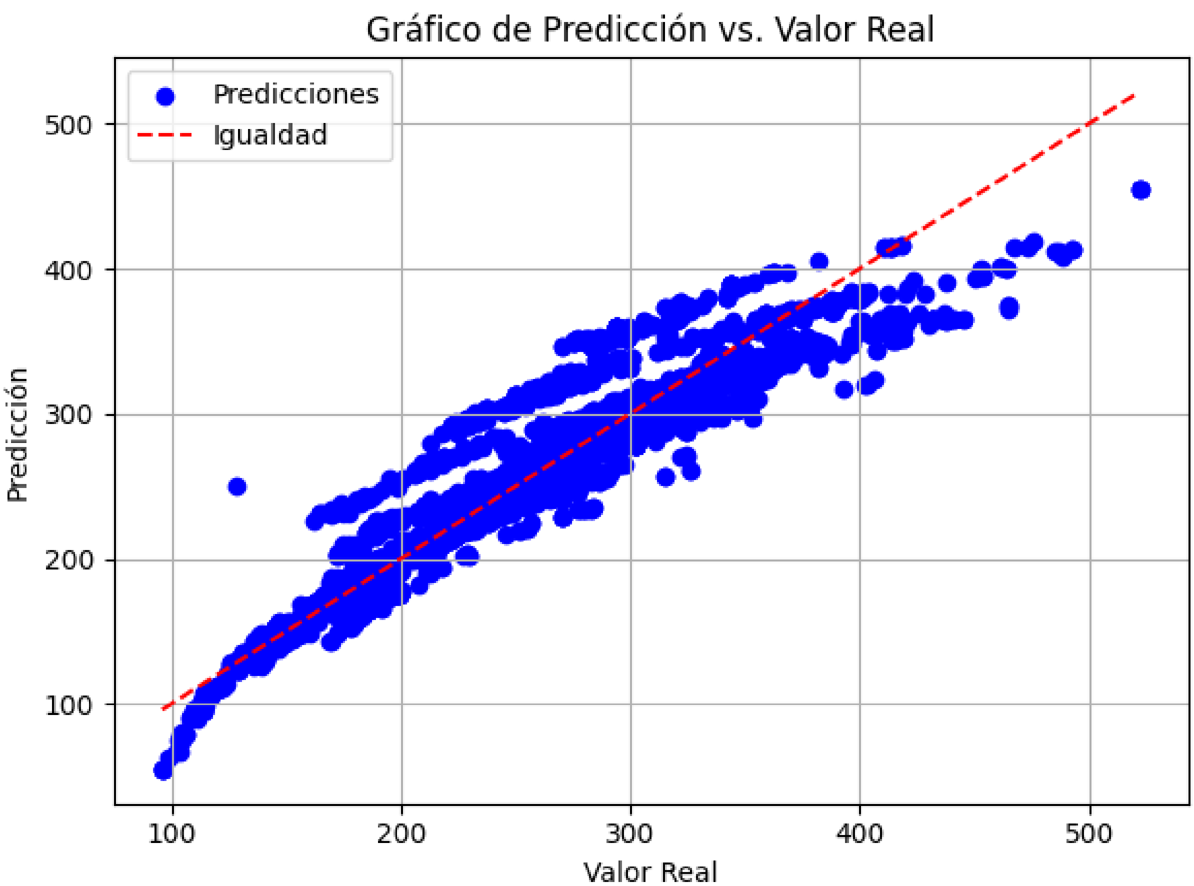
```
print(results.summary())
```

```
                             OLS Regression Results                            
======================================================================
=========
Dep. Variable:     CO2 Emissions(g/km)   R-squared:                       
0.904
Model:                             OLS   Adj. R-squared:                  
0.904
Method:                  Least Squares   F-statistic:                 
1.387e+04
Date:                 Fri, 06 Oct 2023   Prob (F-statistic):              
0.00
Time:                         22:46:11   Log-Likelihood:              
-31882.
No. Observations:                 7385   AIC:                         
6.378e+04
Df Residuals:                     7379   BIC:                         
6.382e+04
Df Model:                            5                                         

Covariance Type:             nonrobust                                         

======================================================================
==============================
                                       coef    std err          t      
P>|t|      [0.025      0.975]
----------------------------------------------------------------------
```

```
------------------------------
const                                    227.7773      4.201     54.222
0.000     219.542     236.012
Engine Size(L)                             4.9847      0.456     10.940
0.000       4.092       5.878
Cylinders                                  7.5289      0.319     23.625
0.000       6.904       8.154
Fuel Consumption City (L/100 km)    -5.3364      0.593     -9.000
0.000      -6.499      -4.174
Fuel Consumption Comb (L/100 km)    11.4566      0.678     16.892
0.000      10.127      12.786
Fuel Consumption Comb (mpg)         -3.4186      0.079    -43.496
0.000      -3.573      -3.265
======================================================================
========
Omnibus:                     1193.043   Durbin-Watson:
1.618
Prob(Omnibus):                  0.000   Jarque-Bera (JB):
7801.545
Skew:                          -0.609   Prob(JB):
0.00
Kurtosis:                       7.886   Cond. No.
657.
======================================================================
========

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

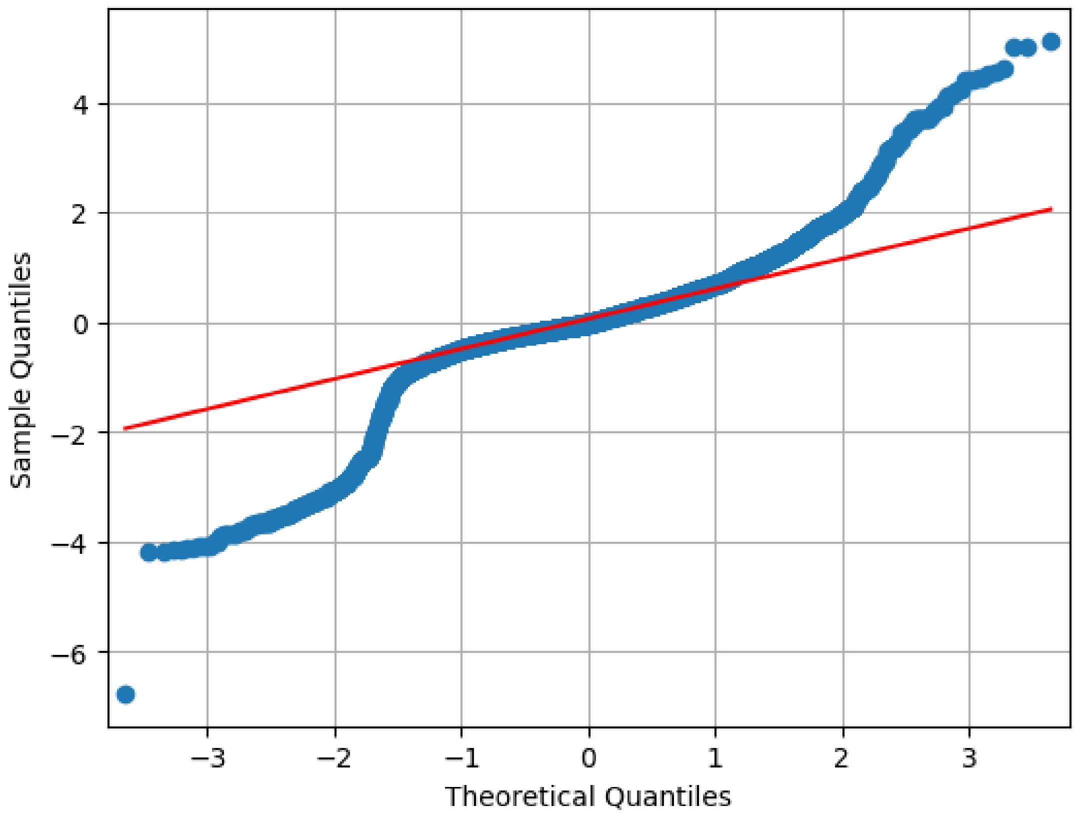
Se gráfica la estimación obtenida con el modelo con respecto al valor real

```
plt.scatter(Y, results.predict(), color='blue', label='Predicciones')
plt.plot([min(Y), max(Y)], [min(Y), max(Y)], linestyle='--',
color='red', label='Igualdad')

plt.xlabel('Valor Real')
plt.ylabel('Predicción')
plt.title('Gráfico de Predicción vs. Valor Real')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```



Se realiza un qq-plot

```
influence = results.get_influence()
standarized_residuals = influence.resid_studentized_internal

fig = sm.qqplot(standarized_residuals, dist=norm, line='q')

print('QQ Graph - normal distribution')
plt.y_label=('Standarized residuals quantiles')
plt.grid()
plt.show()

QQ Graph - normal distribution
```



Preguntas:

- ¿Qué pasa con el fit del modelo y a que se lo atribuye?

El metodo fit, el cual forma parte de la libreria Statsmodels, lleva a cabo el proceso de ajustar el modelo a los datos proporcionados, en el contexto del código, se ajustan varios modelosde regresión lineal para poder visualizar cada variable independiente por si sola, así como un modelo de regresión lineal multiple para evaluar todas las variables.

- ¿Qué sucede con el error y la distribución de este en los datos?

La distribución de los residuos estandarizados es analizada con un análisis QQ plot para compararlos con una distribución normal y con distribuciones con sesgo positivo y negativo utilizando la función skewnorm(). Estas comparaciones ayudan a evaluar si los residuos siguen otras distribuciones, además de la normal.

Este análisis es importante para verificar las suposiciones del modelo de regresión y asegurarse de que los residuos se distribuyan de manera adecuada para realizar inferencias válidas y precisas.

- Describa el impacto de las distintas variables , ¿Que sucede si se omiten las variables con nulo impacto?

La omisión correcta de variables en el modelo puede generar diferentes beneficios como: Simplificar el modelo, evitar sobreajuste, mejoran la eficiencia computacional, entre otros.

Cabe aclararse que esta omisión debe de hacerse con precaución y basarse en un análisis sólido de la relación de cada variable con la variable dependiente.