```
import pandas as pd
from google.colab import files
uploaded = files.upload()
file_name = list(uploaded.keys())[0]
cars = pd.read_csv(file_name)
```

```
<IPython.core.display.HTML object>
```

```
Saving Cars.csv to Cars.csv
```

```
#vif=varience inflamation factor
rsq_hp=smf.ols('HP~WT+VOL+SP',data=cars).fit().rsquared
vif_hp=1/(1-rsq_hp)
rsq_wt=smf.ols('WT~HP+VOL+SP',data=cars).fit()
vif_wt=1/(1-rsq_wt.rsquared)
rsq_vol=smf.ols('VOL~HP+WT+SP',data=cars).fit()
vif_vol=1/(1-rsq_vol.rsquared)
rsq_sp=smf.ols('SP~HP+VOL+WT',data=cars).fit().rsquared
vif_sp=1/(1-rsq_sp)
d1={'Variables':['HP','WT','VOL','SP'],'VIF':
[vif_hp,vif_wt,vif_vol,vif_sp]}
Vif_frame=pd.DataFrame(d1)
Vif_frame
```

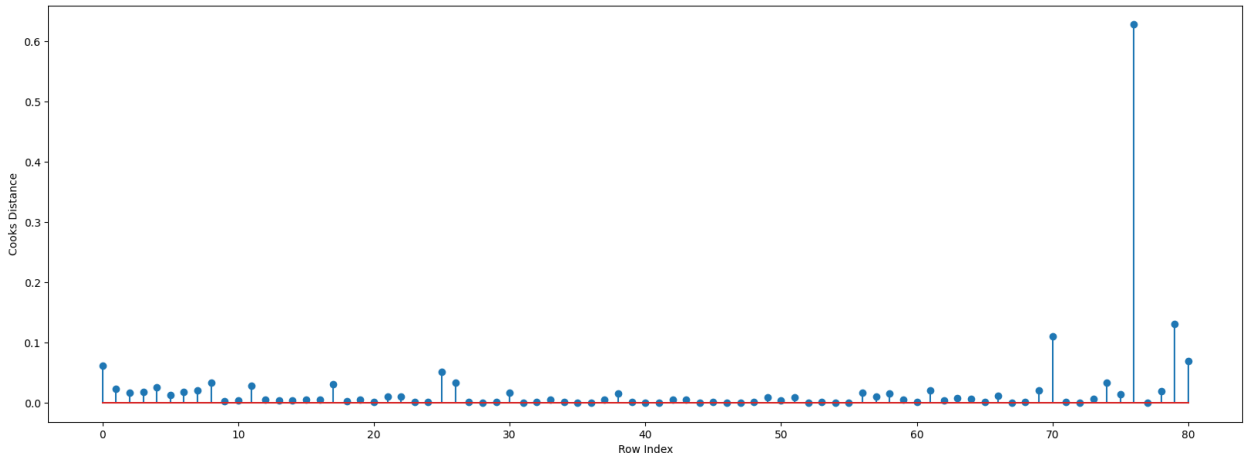
```
{"summary":"{\n  \"name\": \"Vif_frame\",\n  \"rows\": 4,\n
\"fields\": [\n    {\n      \"column\": \"Variables\",\n
\"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 4,\n        \"samples\": [\n          \"WT\",\n
\"SP\",\n          \"HP\"\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"VIF\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 357.49704945001525,\n        \"min\":
19.92658897499852,\n        \"max\": 639.5338175572624,\n
\"num_unique_values\": 4,\n        \"samples\": [\n
639.5338175572624,\n          20.00763878305008,\n
19.92658897499852\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"Vif_frame"}
```

```
ml_v=smf.ols('MPG~VOL',data =cars).fit()
print(ml_v.tvalues, '\n', ml_v.pvalues)
```

```
Intercept    14.106056
VOL          -5.541400
dtype: float64
 Intercept    2.753815e-23
VOL          3.822819e-07
dtype: float64
```

```
import statsmodels.formula.api as smf
```

```
model=smf.ols('MPG~WT+VOL+SP',data =cars).fit()

list(np.where(model.resid>10))

[array([ 0, 76])]

def get_standardized_values(vals):
  return (vals-vals.mean())/vals.std()

model_influence = model.get_influence()
(c, _) = model_influence.cooks_distance

model_influence=model.get_influence()
(c,_)=model_influence.cooks_distance
fig=plt.subplots(figsize=(20,7))
plt.stem(np.arange(len(cars)),np.round(c,3))
plt.xlabel('Row Index')
plt.ylabel('Cooks Distance')
plt.show()
```

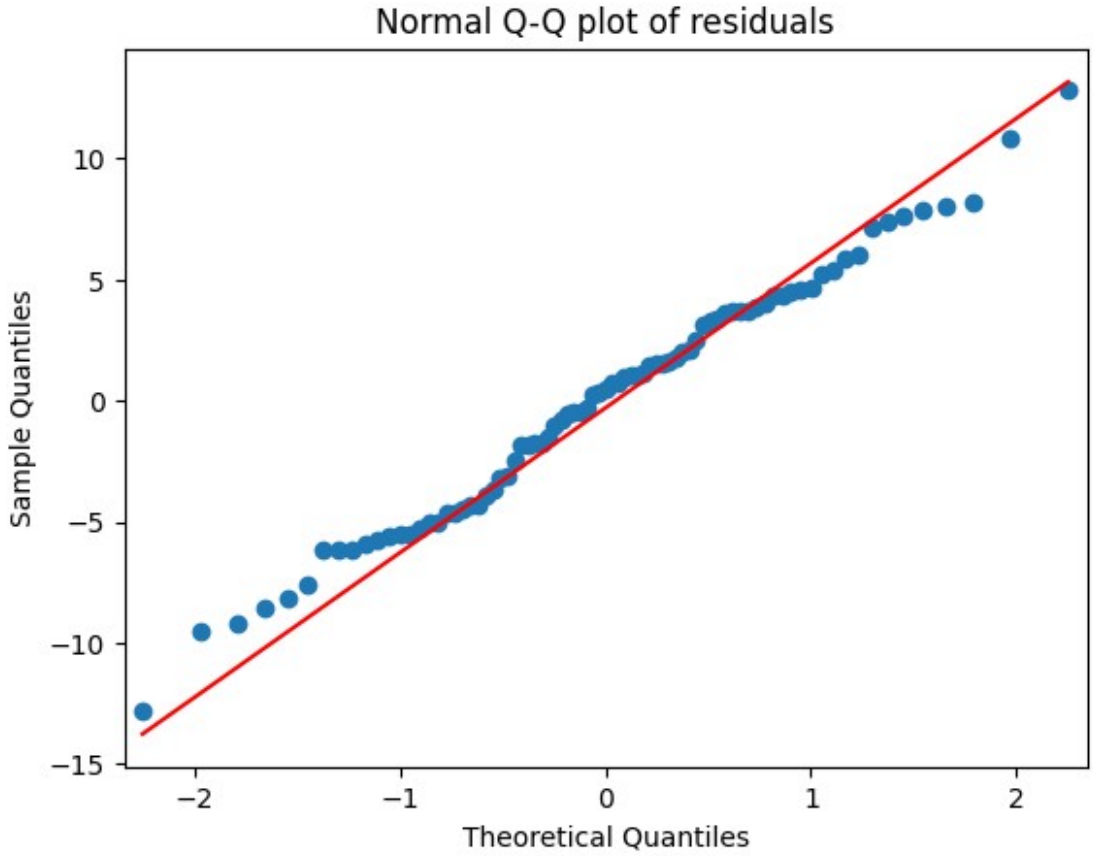



```
import statsmodels.api as sm
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

# Example data (replace with your own dataset)
data = pd.DataFrame({
    'X': np.random.rand(100),
    'Y': np.random.rand(100)
})

x = cars.iloc[:,1:]
y = cars.iloc[:,0]
```

```
x = sm.add_constant(x)

model.fittedvalues
```

```
0     42.832365
1     42.857708
2     42.418722
3     39.825362
4     42.341828
        ...
76    24.015336
77    19.467343
78    30.851867
79    11.800779
80    24.947729
Length: 81, dtype: float64
```

```
sm.qqplot(model.resid, line='q')
plt.title("Normal Q-Q plot of residuals")
plt.show()
```



```
plt.scatter(get_standardized_values(model.fittedvalues),
            get_standardized_values(model.resid))
plt.title('Residual Plot')
plt.xlabel('Standardized Fitted values')
plt.ylabel('Standardized residual values')
plt.show()
```