

Auto dataset two regimes: Pre-oilshock and Post-oilshock

Table of contents

We can also test if there are two regimes that contribute to the heteroskedasticity by running separate regressions for pre-oilshock and post-oilshock.	1
Imports for python objects and libraries	1
Data Cleaning and exploratory data analysis	2
Create two datasets based on whether the car models have been exposed to the 1973 oil shock or not	3
Analysis for pre-oil shock model	6
Analysis for post Oil Shock	13
Pre-oilshock model	22
Explanatory power of preoilshock model	23
Explanatory power of postoilshock model	23
Post oil shock model with intercept (Corollary)	23
Explanatory power of postoilshock model with intercept	23
Finished	23

We can also test if there are two regimes that contribute to the heteroskedasticity by running separate regressions for pre-oilshock and post-oilshock.

Imports for python objects and libraries

Set up IPython libraries for customizing notebook display

```
from notebookfuncs import *
```

Import standard libraries

```
import numpy as np
import pandas as pd

pd.set_option("display.max_rows", 1000)
pd.set_option("display.max_columns", 1000)
```

```
pd.set_option("display.width", 1000)
pd.set_option("display.max.colwidth", None)
import matplotlib.pyplot as plt
import seaborn as sns
import itertools
```

Statsmodels imports

```
import statsmodels.api as sm
```

Import statsmodels.objects

```
from statsmodels.stats.outliers_influence import summary_table
```

Import ISLP objects

```
import ISLP
from ISLP import models
from ISLP import load_data
from ISLP.models import ModelSpec as MS, summarize, poly
```

Import user functions

```
from userfuncs import display_residuals_plot
from userfuncs import identify_least_significant_feature
from userfuncs import calculate_VIFs
from userfuncs import identify_highest_VIF_feature
from userfuncs import standardize
from userfuncs import perform_analysis
```

Set level of significance (alpha)

```
LOS_Alpha = 0.01
```

0.01

Data Cleaning and exploratory data analysis

```
Auto = load_data("Auto")
Auto = Auto.sort_values(by=["year"], ascending=True)
Auto.head()
Auto.columns
```

```
Auto = Auto.dropna()
Auto.shape
Auto.describe()
```

	mpg	cylinders	displacement	horsepower	weight	acceleration	year	origin
count	392.000000	392.000000	392.000000	392.000000	392.000000	392.000000	392.000000	392.000000
mean	23.445918	5.471939	194.411990	104.469388	2977.584184	15.541327	75.979592	1.576531
std	7.805007	1.705783	104.644004	38.491160	849.402560	2.758864	3.683737	0.805518
min	9.000000	3.000000	68.000000	46.000000	1613.000000	8.000000	70.000000	1.000000
25%	17.000000	4.000000	105.000000	75.000000	2225.250000	13.775000	73.000000	1.000000
50%	22.750000	4.000000	151.000000	93.500000	2803.500000	15.500000	76.000000	1.000000
75%	29.000000	8.000000	275.750000	126.000000	3614.750000	17.025000	79.000000	2.000000
max	46.600000	8.000000	455.000000	230.000000	5140.000000	24.800000	82.000000	3.000000

Convert origin to categorical type

```
Auto["origin"] = Auto["origin"].astype("category")
Auto["origin"] = Auto["origin"].cat.rename_categories(
    {1: "America", 2: "Europe", 3: "Japan"}
)
Auto.describe()
```

	mpg	cylinders	displacement	horsepower	weight	acceleration	year
count	392.000000	392.000000	392.000000	392.000000	392.000000	392.000000	392.000000
mean	23.445918	5.471939	194.411990	104.469388	2977.584184	15.541327	75.979592
std	7.805007	1.705783	104.644004	38.491160	849.402560	2.758864	3.683737
min	9.000000	3.000000	68.000000	46.000000	1613.000000	8.000000	70.000000
25%	17.000000	4.000000	105.000000	75.000000	2225.250000	13.775000	73.000000
50%	22.750000	4.000000	151.000000	93.500000	2803.500000	15.500000	76.000000
75%	29.000000	8.000000	275.750000	126.000000	3614.750000	17.025000	79.000000
max	46.600000	8.000000	455.000000	230.000000	5140.000000	24.800000	82.000000

Create two datasets based on whether the car models have been exposed to the 1973 oil shock or not

```
Auto_preos = Auto[Auto["year"] <= 76]
Auto_preos.shape
Auto_preos.describe()
Auto_preos.corr(numeric_only=True)
```

	mpg	cylinders	displacement	horsepower	weight	acceleration	year
mpg	1.000000	-0.863133	-0.878385	-0.812052	-0.903557	0.494406	0.172135
cylinders	-0.863133	1.000000	0.955270	0.852144	0.906436	-0.616635	-0.157796
displacement	-0.878385	0.955270	1.000000	0.900549	0.926890	-0.653019	-0.195140
horsepower	-0.812052	0.852144	0.900549	1.000000	0.861309	-0.748969	-0.294137
weight	-0.903557	0.906436	0.926890	0.861309	1.000000	-0.522137	-0.073366
acceleration	0.494406	-0.616635	-0.653019	-0.748969	-0.522137	1.000000	0.298412
year	0.172135	-0.157796	-0.195140	-0.294137	-0.073366	0.298412	1.000000

```
Auto_postos = Auto[Auto["year"] > 76]
Auto_postos.shape
Auto_postos.describe()
```

	mpg	cylinders	displacement	horsepower	weight	acceleration	year
count	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000
mean	27.900562	4.960674	160.426966	91.410112	2726.679775	16.133146	79.455056
std	7.504963	1.466624	80.477444	27.144212	670.417952	2.504227	1.714248
min	15.000000	3.000000	70.000000	48.000000	1755.000000	11.100000	77.000000
25%	20.875000	4.000000	98.000000	70.000000	2144.250000	14.500000	78.000000
50%	28.000000	4.000000	134.500000	87.000000	2630.000000	15.800000	79.000000
75%	33.650000	6.000000	200.000000	105.000000	3208.750000	17.600000	81.000000
max	46.600000	8.000000	400.000000	190.000000	4360.000000	24.800000	82.000000

```
display(
    "If you look at the two datasets as displayed above, it's evident that the oil
    ↪ shock had a major impact on the models produced since."
)
display(Auto_preos.mean(numeric_only=True), Auto_postos.mean(numeric_only=True))
display(
    "Mileage increased, number of cylinders decreased, displacement decreased,
    ↪ horsepower decreased, weight decreased and time to acceleration increased
    ↪ thus indicating that less powerful and less performant cars were produced in
    ↪ the immediate period after the oil shock of 1973."
)
```

"If you look at the two datasets as displayed above, it's evident that the oil shock had a major impact on the models produced since."

```
mpg          19.740654
cylinders     5.897196
displacement  222.679907
horsepower    115.331776
weight       3186.280374
```

```

acceleration    15.049065
year            73.088785
dtype: float64

mpg            27.900562
cylinders       4.960674
displacement    160.426966
horsepower      91.410112
weight          2726.679775
acceleration    16.133146
year            79.455056
dtype: float64

```

'Mileage increased, number of cylinders decreased, displacement decreased, horsepower decreased, weight decreased and time to acceleration increased thus indicating that less powerful and less performant cars were produced in the immediate period after the oil shock of 1973.'

Standardize numeric variables in the model

```

Auto_preos = Auto_preos.apply(standardize)
Auto_preos.describe()

```

	mpg	cylinders	displacement	horsepower	weight	acceleration	year
count	2.140000e+02	2.140000e+02	2.140000e+02	2.140000e+02	214.000000	214.000000	2.140000e+02
mean	-4.150366e-17	-2.490220e-17	2.490220e-17	-1.494132e-16	0.000000	0.000000	-5.312469e-16
std	1.002345e+00	1.002345e+00	1.002345e+00	1.002345e+00	1.002345	1.002345	1.002345e+00
min	-1.829062e+00	-1.635252e+00	-1.362364e+00	-1.617309e+00	-1.705900	-2.463723	-1.552289e+00
25%	-8.073018e-01	-1.070826e+00	-9.550106e-01	-6.842252e-01	-0.944725	-0.698694	-1.049733e+00
50%	-1.261285e-01	5.802508e-02	4.685742e-02	-3.576458e-01	-0.081084	-0.017149	-4.461951e-02
75%	7.891982e-01	1.186877e+00	8.395442e-01	8.087090e-01	0.913215	0.629446	9.604936e-01
max	2.598565e+00	1.186877e+00	2.046190e+00	2.674877e+00	2.118409	2.953691	1.463050e+00

```

Auto_postos = Auto_postos.apply(standardize)
Auto_postos.describe()

```

	mpg	cylinders	displacement	horsepower	weight	acceleration	year
count	1.780000e+02	1.780000e+02	1.780000e+02	1.780000e+02	178.000000	1.780000e+02	1.780000e+02
mean	-3.193450e-16	2.794269e-16	-7.983626e-17	-1.796316e-16	0.000000	-1.237462e-15	-1.516889e-15
std	1.002821e+00	1.002821e+00	1.002821e+00	1.002821e+00	1.002821	1.002821e+00	1.002821e+00
min	-1.723786e+00	-1.340633e+00	-1.126801e+00	-1.603751e+00	-1.453453	-2.015529e+00	-1.436187e+00
25%	-9.387629e-01	-6.568717e-01	-7.778958e-01	-7.909792e-01	-0.871207	-6.539953e-01	-8.511958e-01
50%	1.328704e-02	-6.568717e-01	-3.230732e-01	-1.629280e-01	-0.144615	-1.334087e-01	-2.662041e-01
75%	7.682459e-01	7.106507e-01	4.931154e-01	5.020674e-01	0.721088	5.874034e-01	9.037793e-01

	mpg	cylinders	displacement	horsepower	weight	acceleration	year
max	2.498638e+00	2.078173e+00	2.985294e+00	3.642323e+00	2.443144	3.470652e+00	1.488771e+00

Encode categorical variables as dummy variables dropping the first to remove multicollinearity.

```
Auto_preos = pd.get_dummies(
    Auto_preos, columns=list(["origin"]), drop_first=True, dtype=np.uint8
)
Auto_preos.columns
```

```
Index(['mpg', 'cylinders', 'displacement', 'horsepower', 'weight', 'acceleration',
      'year', 'origin_Europe', 'origin_Japan'], dtype='object')
```

```
Auto_postos = pd.get_dummies(
    Auto_postos, columns=list(["origin"]), drop_first=True, dtype=np.uint8
)
Auto_postos.columns
```

```
Index(['mpg', 'cylinders', 'displacement', 'horsepower', 'weight', 'acceleration',
      'year', 'origin_Europe', 'origin_Japan'], dtype='object')
```

Analysis for pre-oil shock model

Test for multicollinearity using correlation matrix and variance inflation factors

```
Auto_preos.corr(numeric_only=True)
```

	mpg	cylinders	displacement	horsepower	weight	acceleration	year	origin_Europe	origin_Japan
mpg	1.000000	-0.863133	-0.878385	-0.812052	-0.903557	0.494406	0.172135	0.429946	-0.192745
cylinders	-0.863133	1.000000	0.955270	0.852144	0.906436	-0.616635	-0.157796	-0.507897	-0.408555
displacement	-0.878385	0.955270	1.000000	0.900549	0.926890	-0.653019	-0.195140	-0.499456	-0.428045
horsepower	-0.812052	0.852144	0.900549	1.000000	0.861309	-0.748969	-0.294137	-0.373257	-0.292877
weight	-0.903557	0.906436	0.926890	0.861309	1.000000	-0.522137	-0.073366	-0.420078	-0.424328
acceleration	0.494406	-0.616635	-0.653019	-0.748969	-0.522137	1.000000	0.298412	0.215335	0.164038
year	0.172135	-0.157796	-0.195140	-0.294137	-0.073366	0.298412	1.000000	0.061819	0.030362
origin_Europe	0.429946	-0.507897	-0.499456	-0.373257	-0.420078	0.215335	0.061819	1.000000	
origin_Japan	0.454576	-0.408555	-0.428045	-0.292877	-0.424328	0.164038	0.030362		1.000000

```
vifdf = calculate_VIFs("mpg ~ " + " + ".join(Auto_preos.columns) + " - mpg",
    ↪ Auto_preos)
vifdf
```

Feature	VIF
cylinders	12.409093
displacement	23.483690
horsepower	9.924721
weight	10.993223
acceleration	2.965117
year	1.296707
origin_Europe	2.286473
origin_Japan	2.062780

```
identify_highest_VIF_feature(vifdf)
```

We find the highest VIF in this model is displacement with a VIF of 23.483689524756567

Hence, we drop displacement from the model to be fitted.

```
('displacement', 23.483689524756567)
```

```
vifdf = calculate_VIFs(
    "mpg ~ " + " + ".join(Auto_preos.columns) + " - mpg - displacement", Auto_preos
)
vifdf
```

Feature	VIF
cylinders	8.727646
horsepower	8.845099
weight	9.513189
acceleration	2.856231
year	1.287027
origin_Europe	1.960903
origin_Japan	1.789531

```
identify_highest_VIF_feature(vifdf)
```

No variables are significantly collinear.

Linear Regression for mpg ~ horsepower + acceleration + weight + cylinders + year + origin_Europe + origin_Japan

```
cols = list(Auto_preos.columns)
cols.remove("mpg")
```



```
cols.remove("displacement")
formula = " + ".join(cols)
results = perform_analysis("mpg", formula, Auto_preos)
```

OLS Regression Results

Dep. Variable:	mpg	R-squared:	0.848
Model:	OLS	Adj. R-squared:	0.842
Method:	Least Squares	F-statistic:	163.8
Date:	Tue, 25 Feb 2025	Prob (F-statistic):	1.51e-80
Time:	14:37:55	Log-Likelihood:	-102.32
No. Observations:	214	AIC:	220.6
Df Residuals:	206	BIC:	247.6
Df Model:	7		
Covariance Type:	nonrobust		

	coef	std err	t	P> t	[0.025	0.975]
Intercept	-0.1025	0.040	-2.583	0.010	-0.181	-0.024
cylinders	-0.1149	0.080	-1.430	0.154	-0.273	0.043
horsepower	-0.1394	0.081	-1.724	0.086	-0.299	0.020
weight	-0.6079	0.084	-7.248	0.000	-0.773	-0.443
acceleration	-0.0653	0.046	-1.421	0.157	-0.156	0.025
year	0.0776	0.031	2.514	0.013	0.017	0.138
origin_Europe	0.2534	0.097	2.618	0.009	0.063	0.444
origin_Japan	0.3985	0.106	3.749	0.000	0.189	0.608

Omnibus:	12.372	Durbin-Watson:	1.407
Prob(Omnibus):	0.002	Jarque-Bera (JB):	16.578
Skew:	-0.403	Prob(JB):	0.000251
Kurtosis:	4.099	Cond. No.	9.30

Notes:

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

	df	sum_sq	mean_sq	F	PR(>F)
cylinders	1.0	159.429574	159.429574	1007.436126	2.877624e-81
horsepower	1.0	4.577852	4.577852	28.927463	2.030739e-07
weight	1.0	13.283446	13.283446	83.938147	5.242535e-17
acceleration	1.0	0.533174	0.533174	3.369126	6.787066e-02
year	1.0	1.267919	1.267919	8.011985	5.107121e-03
origin_Europe	1.0	0.083174	0.083174	0.525577	4.692948e-01
origin_Japan	1.0	2.224788	2.224788	14.058446	2.302318e-04
Residual	206.0	32.600074	0.158253	NaN	NaN

```
<statsmodels.regression.linear_model.RegressionResultsWrapper at 0x76450324c890>
```

```
identify_least_significant_feature(results, alpha=LOS_Alpha)
```

We find the least significant variable in this model is acceleration with a p-value of 0.15682628665346462 and a coefficient of -0.06530735672959463

Using the backward methodology, we suggest dropping acceleration from the new model

Linear Regression after dropping acceleration in pre-oil shock. The model now is mpg ~ horsepower + weight + cylinder + year + origin_Europe + origin_Japan

```
cols.remove("acceleration")
formula = " + ".join(cols)
results = perform_analysis("mpg", formula, Auto_preos)
```

OLS Regression Results

```
=====
Dep. Variable:          mpg      R-squared:                0.846
Model:                  OLS      Adj. R-squared:           0.842
Method:                 Least Squares      F-statistic:        189.8
Date:                  Tue, 25 Feb 2025     Prob (F-statistic):    2.86e-81
Time:                  14:37:55      Log-Likelihood:       -103.36
No. Observations:      214          AIC:                  220.7
Df Residuals:          207          BIC:                  244.3
Df Model:               6
Covariance Type:       nonrobust
=====
```

	coef	std err	t	P> t	[0.025	0.975]
Intercept	-0.1073	0.040	-2.705	0.007	-0.185	-0.029
cylinders	-0.0832	0.077	-1.075	0.284	-0.236	0.069
horsepower	-0.0718	0.066	-1.095	0.275	-0.201	0.057
weight	-0.6564	0.077	-8.546	0.000	-0.808	-0.505
year	0.0789	0.031	2.552	0.011	0.018	0.140
origin_Europe	0.2722	0.096	2.832	0.005	0.083	0.462
origin_Japan	0.4069	0.106	3.825	0.000	0.197	0.617

```
=====
Omnibus:                 9.704      Durbin-Watson:           1.384
Prob(Omnibus):            0.008      Jarque-Bera (JB):        10.825
Skew:                    -0.398      Prob(JB):                0.00446
Kurtosis:                 3.763      Cond. No.                 8.52
=====
```

Notes:

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

	df	sum_sq	mean_sq	F	PR(>F)
cylinders	1.0	159.429574	159.429574	1002.499833	2.735220e-81
horsepower	1.0	4.577852	4.577852	28.785723	2.155884e-07
weight	1.0	13.283446	13.283446	83.526863	5.919063e-17
year	1.0	1.323199	1.323199	8.320328	4.335077e-03
origin_Europe	1.0	0.139721	0.139721	0.878569	3.496863e-01
origin_Japan	1.0	2.326581	2.326581	14.629642	1.731691e-04
Residual	207.0	32.919628	0.159032	NaN	NaN

<statsmodels.regression.linear_model.RegressionResultsWrapper at 0x764503279fd0>

```
identify_least_significant_feature(results, alpha=LOS_Alpha)
```

We find the least significant variable in this model is cylinders with a p-value of 0.28351001934768794 and a coefficient of -0.08318186983127318

Using the backward methodology, we suggest dropping cylinders from the new model

```
cols.remove("cylinders")
formula = " + ".join(cols)
results = perform_analysis("mpg", formula, Auto_preos)
```

OLS Regression Results

```
=====
Dep. Variable:          mpg      R-squared:                0.845
Model:                  OLS      Adj. R-squared:           0.842
Method:                 Least Squares      F-statistic:          227.3
Date:                  Tue, 25 Feb 2025     Prob (F-statistic):    3.20e-82
Time:                  14:37:55             Log-Likelihood:       -103.95
No. Observations:      214                AIC:                 219.9
Df Residuals:          208                BIC:                 240.1
Df Model:              5
Covariance Type:       nonrobust
=====
```

	coef	std err	t	P> t	[0.025	0.975]
Intercept	-0.1213	0.037	-3.235	0.001	-0.195	-0.047
horsepower	-0.0964	0.061	-1.569	0.118	-0.218	0.025
weight	-0.6974	0.067	-10.455	0.000	-0.829	-0.566
year	0.0802	0.031	2.597	0.010	0.019	0.141
origin_Europe	0.3185	0.086	3.708	0.000	0.149	0.488
origin_Japan	0.4445	0.101	4.422	0.000	0.246	0.643

```
=====
Omnibus:              7.861      Durbin-Watson:          1.406
Prob(Omnibus):        0.020      Jarque-Bera (JB):       8.096
Skew:                 -0.371     Prob(JB):               0.0175
Kurtosis:             3.598      Cond. No.               6.43
=====
```

Notes:

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

	df	sum_sq	mean_sq	F	PR(>F)
horsepower	1.0	141.117636	141.117636	886.687803	6.026431e-77
weight	1.0	34.542884	34.542884	217.044124	4.053391e-34
year	1.0	1.552002	1.552002	9.751732	2.046623e-03
origin_Europe	1.0	0.572100	0.572100	3.594690	5.935071e-02
origin_Japan	1.0	3.111879	3.111879	19.552944	1.576086e-05
Residual	208.0	33.103499	0.159151	NaN	NaN

<statsmodels.regression.linear_model.RegressionResultsWrapper at 0x7645032a3e00>

```
identify_least_significant_feature(results, alpha=LOS_Alpha)
```

We find the least significant variable in this model is horsepower with a p-value of 0.11823063227848224 and a coefficient of -0.09641477003432276

Using the backward methodology, we suggest dropping horsepower from the new model

```
cols.remove("horsepower")
formula = " + ".join(cols)
results = perform_analysis("mpg", formula, Auto_preos)
```

OLS Regression Results

```
=====
Dep. Variable:          mpg      R-squared:                0.843
Model:                  OLS      Adj. R-squared:           0.840
Method:                 Least Squares      F-statistic:         281.6
Date:                  Tue, 25 Feb 2025     Prob (F-statistic):    6.06e-83
Time:                  14:37:55             Log-Likelihood:       -105.21
No. Observations:      214                AIC:                220.4
Df Residuals:          209                BIC:                237.3
Df Model:              4
Covariance Type:       nonrobust
=====
```

	coef	std err	t	P> t	[0.025	0.975]
Intercept	-0.1151	0.037	-3.077	0.002	-0.189	-0.041
weight	-0.7850	0.037	-21.422	0.000	-0.857	-0.713
year	0.1028	0.027	3.742	0.000	0.049	0.157
origin_Europe	0.3078	0.086	3.582	0.000	0.138	0.477
origin_Japan	0.4140	0.099	4.183	0.000	0.219	0.609

```
=====
Omnibus:                 10.672      Durbin-Watson:           1.398
Prob(Omnibus):            0.005      Jarque-Bera (JB):        11.650
Skew:                    -0.443      Prob(JB):                0.00295
Kurtosis:                 3.722      Cond. No.                 4.59
=====
```

Notes:

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

	df	sum_sq	mean_sq	F	PR(>F)
weight	1.0	174.712905	174.712905	1090.157519	7.188986e-85
year	1.0	2.410414	2.410414	15.040281	1.409621e-04
origin_Europe	1.0	0.576721	0.576721	3.598570	5.920817e-02
origin_Japan	1.0	2.804802	2.804802	17.501148	4.226183e-05
Residual	209.0	33.495157	0.160264	NaN	NaN

<statsmodels.regression.linear_model.RegressionResultsWrapper at 0x76450324d730>

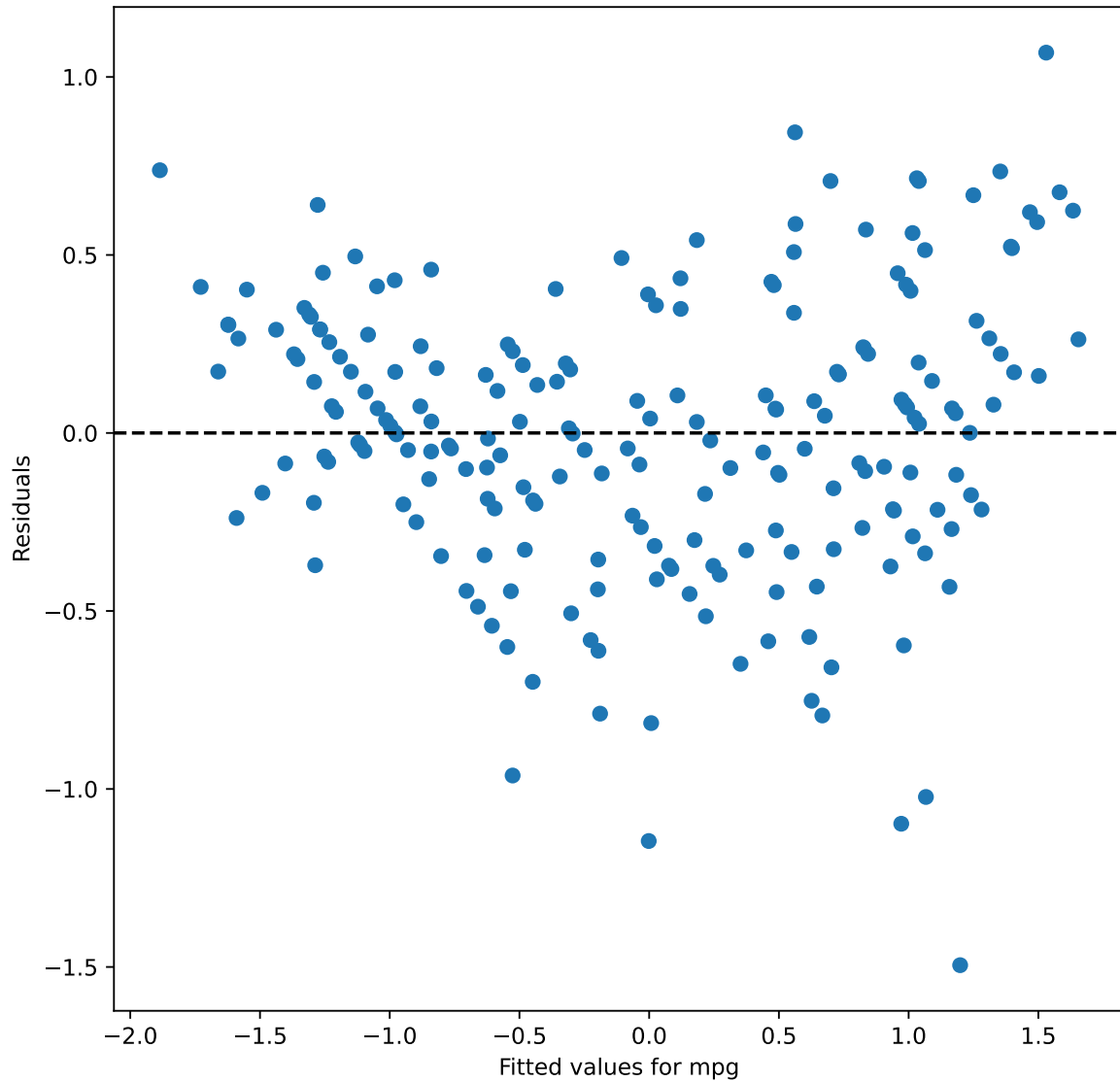
```
identify_least_significant_feature(results, alpha=LOS_Alpha)
```

No variables are statistically insignificant.

The model `mpg ~ weight + year + origin_Europe + origin_Japan` cannot be pruned further.

Residual plot for model for pre-oil shock

```
display_residuals_plot(results)
```



```
preoilshock_model = results
```

```
<statsmodels.regression.linear_model.RegressionResultsWrapper at 0x76450324d730>
```

Analysis for post Oil Shock

Test for multicollinearity using correlation matrix and variance inflation factors

```
Auto_postos.corr(numeric_only=True)
```

	mpg	cylinders	displacement	horsepower	weight	acceleration	year	origin_Eur
mpg	1.000000	-0.710232	-0.771039	-0.796617	-0.837333	0.278650	0.460200	0.212795
cylinders	-0.710232	1.000000	0.936943	0.796697	0.860088	-0.247767	-0.338905	-0.181385
displacement	-0.771039	0.936943	1.000000	0.854454	0.929346	-0.264374	-0.319411	-0.240143
horsepower	-0.796617	0.796697	0.854454	1.000000	0.837067	-0.535033	-0.353954	-0.214702
weight	-0.837333	0.860088	0.929346	0.837067	1.000000	-0.130152	-0.319783	-0.144152
acceleration	0.278650	-0.247767	-0.264374	-0.535033	-0.130152	1.000000	0.157159	0.235217
year	0.460200	-0.338905	-0.319411	-0.353954	-0.319783	0.157159	1.000000	-0.057596
origin_Europe	0.212795	-0.181385	-0.240143	-0.214702	-0.144152	0.235217	-0.057596	1.000000
origin_Japan	0.405159	-0.359263	-0.436964	-0.317954	-0.459869	0.000714	0.155368	-0.264286

```
vifdf = calculate_VIFs(
    "mpg ~ " + " + ".join(Auto_postos.columns) + " - mpg", Auto_postos
)
vifdf
```

	VIF
Feature	
cylinders	9.017020
displacement	20.423355
horsepower	9.245687
weight	12.693737
acceleration	2.788052
year	1.185236
origin_Europe	1.452328
origin_Japan	1.651675

```
identify_highest_VIF_feature(vifdf)
```

We find the highest VIF in this model is displacement with a VIF of 20.423354692792778

Hence, we drop displacement from the model to be fitted.

```
('displacement', 20.423354692792778)
```

```
vifdf = calculate_VIFs(
    "mpg ~ " + " + ".join(Auto_postos.columns) + " - mpg - displacement",
    ↪ Auto_postos
)
vifdf
```

	VIF
Feature	
cylinders	4.251590
horsepower	9.104343
weight	9.540921
acceleration	2.770794
year	1.182561
origin_Europe	1.278261
origin_Japan	1.512852

```
identify_highest_VIF_feature(vifdf)
```

No variables are significantly collinear.

Linear Regression Analysis for post oil shock dropping feature displacement

```
cols = list(Auto_postos.columns)
cols.remove("mpg")
cols.remove("displacement")
formula = " + ".join(cols)
results = perform_analysis("mpg", formula, Auto_postos)
```

OLS Regression Results

Dep. Variable:	mpg	R-squared:	0.788			
Model:	OLS	Adj. R-squared:	0.779			
Method:	Least Squares	F-statistic:	90.11			
Date:	Tue, 25 Feb 2025	Prob (F-statistic):	7.20e-54			
Time:	14:37:56	Log-Likelihood:	-114.64			
No. Observations:	178	AIC:	245.3			
Df Residuals:	170	BIC:	270.7			
Df Model:	7					
Covariance Type:	nonrobust					
=====						
	coef	std err	t	P> t	[0.025	0.975]

Intercept	-0.1072	0.051	-2.096	0.038	-0.208	-0.006
cylinders	0.1988	0.073	2.728	0.007	0.055	0.343
horsepower	-0.1879	0.107	-1.762	0.080	-0.398	0.023
weight	-0.7149	0.109	-6.550	0.000	-0.930	-0.499
acceleration	0.0713	0.059	1.212	0.227	-0.045	0.187
year	0.2148	0.038	5.589	0.000	0.139	0.291
origin_Europe	0.3461	0.111	3.108	0.002	0.126	0.566
origin_Japan	0.1946	0.097	2.012	0.046	0.004	0.385

Omnibus:	6.408	Durbin-Watson:	1.583
Prob(Omnibus):	0.041	Jarque-Bera (JB):	6.069
Skew:	0.398	Prob(JB):	0.0481
Kurtosis:	3.431	Cond. No.	7.71

=====

Notes:

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

	df	sum_sq	mean_sq	F	PR(>F)
cylinders	1.0	89.788355	89.788355	403.941459	8.824859e-47
horsepower	1.0	25.953062	25.953062	116.758098	4.752402e-21
weight	1.0	15.387223	15.387223	69.224316	2.748274e-14
acceleration	1.0	0.660414	0.660414	2.971082	8.658318e-02
year	1.0	6.087213	6.087213	27.385264	4.863030e-07
origin_Europe	1.0	1.436421	1.436421	6.462195	1.191261e-02
origin_Japan	1.0	0.899608	0.899608	4.047172	4.582475e-02
Residual	170.0	37.787704	0.222281	NaN	NaN

<statsmodels.regression.linear_model.RegressionResultsWrapper at 0x764500f83ef0>

```
identify_least_significant_feature(results, alpha=LOS_Alpha)
```

We find the least significant variable in this model is acceleration with a p-value of 0.22719512270297804 and a coefficient of 0.07129263347256862

Using the backward methodology, we suggest dropping acceleration from the new model

```
cols.remove("acceleration")
formula = " + ".join(cols)
results = perform_analysis("mpg", formula, Auto_postos)
```

OLS Regression Results

```
=====
Dep. Variable:          mpg      R-squared:          0.786
Model:                  OLS      Adj. R-squared:      0.778
Method:                 Least Squares      F-statistic:      104.6
Date:                   Tue, 25 Feb 2025    Prob (F-statistic): 1.39e-54
Time:                   14:37:56          Log-Likelihood:    -115.40
No. Observations:      178              AIC:              244.8
Df Residuals:          171              BIC:              267.1
Df Model:               6
Covariance Type:       nonrobust
=====
```

	coef	std err	t	P> t	[0.025	0.975]
Intercept	-0.1148	0.051	-2.261	0.025	-0.215	-0.015
cylinders	0.1915	0.073	2.633	0.009	0.048	0.335
horsepower	-0.2864	0.069	-4.148	0.000	-0.423	-0.150

weight	-0.6311	0.085	-7.462	0.000	-0.798	-0.464
year	0.2149	0.038	5.584	0.000	0.139	0.291
origin_Europe	0.3689	0.110	3.355	0.001	0.152	0.586
origin_Japan	0.2096	0.096	2.183	0.030	0.020	0.399

```
=====
Omnibus:                6.875   Durbin-Watson:                1.555
Prob(Omnibus):          0.032   Jarque-Bera (JB):            6.653
Skew:                   0.400   Prob(JB):                    0.0359
Kurtosis:               3.507   Cond. No.                    6.16
=====
```

Notes:

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

	df	sum_sq	mean_sq	F	PR(>F)
cylinders	1.0	89.788355	89.788355	402.836748	8.041561e-47
horsepower	1.0	25.953062	25.953062	116.438785	4.931248e-21
weight	1.0	15.387223	15.387223	69.034999	2.864764e-14
year	1.0	6.001042	6.001042	26.923762	5.941221e-07
origin_Europe	1.0	1.693907	1.693907	7.599738	6.471569e-03
origin_Japan	1.0	1.062190	1.062190	4.765532	3.039795e-02
Residual	171.0	38.114221	0.222890	NaN	NaN

<statsmodels.regression.linear_model.RegressionResultsWrapper at 0x764500f85a30>

```
identify_least_significant_feature(results, alpha=LOS_Alpha)
```

We find the least significant variable in this model is origin_Japan with a p-value of 0.030397952937226073 and a coefficient of 0.20964740213594493

Using the backward methodology, we suggest dropping origin_Japan from the new model

- However, origin_Japan is one of three levels with origin_Europe significant. So we do not drop it from the model.
- We can check what will happen with dropping the Intercept with it also insignificant especially since we have standardized the variables.
- <https://stats.stackexchange.com/questions/197923/difference-between-centered-and-uncentered-r2>

```
postoilshock_model_intercept = results
formula = " + ".join(cols)
formula += " - 1"
results = perform_analysis("mpg", formula, Auto_postos)
```

OLS Regression Results

```
=====
Dep. Variable:          mpg   R-squared (uncentered):
0.779
```

```

Model:                                OLS    Adj. R-squared (uncentered):
0.772
Method:                               Least Squares    F-statistic:
101.3
Date:                                 Tue, 25 Feb 2025    Prob (F-statistic):
8.07e-54
Time:                                 14:37:57    Log-Likelihood:
-118.03
No. Observations:                     178    AIC:
248.1
Df Residuals:                         172    BIC:
267.1
Df Model:                             6
Covariance Type:                      nonrobust
=====
              coef      std err          t      P>|t|      [0.025      0.975]
-----
cylinders      0.1892      0.074      2.572      0.011      0.044      0.334
horsepower    -0.2877      0.070     -4.117      0.000     -0.426     -0.150
weight       -0.6656      0.084     -7.905      0.000     -0.832     -0.499
year          0.2098      0.039      5.398      0.000      0.133      0.287
origin_Europe 0.2400      0.095      2.523      0.013      0.052      0.428
origin_Japan   0.0688      0.074      0.930      0.353     -0.077      0.215
=====
Omnibus:                9.950    Durbin-Watson:                1.526
Prob(Omnibus):          0.007    Jarque-Bera (JB):              10.241
Skew:                   0.498    Prob(JB):                      0.00597
Kurtosis:               3.622    Cond. No.                      5.06
=====

```

Notes:

[1] R^2 is computed without centering (uncentered) since the model does not contain a constant.

[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.

```

              df      sum_sq      mean_sq          F      PR(>F)
cylinders      1.0  89.788355  89.788355  393.430947  2.582371e-46
horsepower      1.0  25.953062  25.953062  113.720066  1.057893e-20
weight          1.0  15.387223  15.387223   67.423107  5.008832e-14
year            1.0   6.001042   6.001042   26.295121  7.831453e-07
origin_Europe   1.0   1.419140   1.419140    6.218329  1.358813e-02
origin_Japan     1.0   0.197537   0.197537    0.865561  3.534910e-01
Residual      172.0  39.253641   0.228219         NaN         NaN

```

<statsmodels.regression.linear_model.RegressionResultsWrapper at 0x7644feefacf0>

```
identify_least_significant_feature(results, alpha=LOS_Alpha)
```

We find the least significant variable in this model is origin_Japan with a p-value of 0.35349096552578385 and a coefficient of 0.06881297255714136

Using the backward methodology, we suggest dropping origin_Japan from the new model

- We drop both origin_Europe and origin_Japan from the model.

```
cols.remove("origin_Europe")
cols.remove("origin_Japan")
formula = " + ".join(cols)
formula += " - 1"
results = perform_analysis("mpg", formula, Auto_postos)
```

OLS Regression Results						
=====						
Dep. Variable:	mpg	R-squared (uncentered):				
0.770						
Model:	OLS	Adj. R-squared (uncentered):				
0.765						
Method:	Least Squares	F-statistic:				
146.0						
Date:	Tue, 25 Feb 2025	Prob (F-statistic):				
1.73e-54						
Time:	14:37:57	Log-Likelihood:				
-121.62						
No. Observations:	178	AIC:				
251.2						
Df Residuals:	174	BIC:				
264.0						
Df Model:	4					
Covariance Type:	nonrobust					
=====						
	coef	std err	t	P> t	[0.025	0.975]

cylinders	0.1776	0.074	2.388	0.018	0.031	0.324
horsepower	-0.3084	0.070	-4.424	0.000	-0.446	-0.171
weight	-0.6688	0.082	-8.173	0.000	-0.830	-0.507
year	0.1974	0.039	5.055	0.000	0.120	0.274
=====						
Omnibus:	13.678	Durbin-Watson:		1.582		
Prob(Omnibus):	0.001	Jarque-Bera (JB):		14.628		
Skew:	0.630	Prob(JB):		0.000666		
Kurtosis:	3.619	Cond. No.		4.67		
=====						

Notes:

[1] R^2 is computed without centering (uncentered) since the model does not contain a constant.

[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.

	df	sum_sq	mean_sq	F	PR(>F)
cylinders	1.0	89.788355	89.788355	382.262105	8.909529e-46
horsepower	1.0	25.953062	25.953062	110.491745	2.548773e-20
weight	1.0	15.387223	15.387223	65.509078	9.616073e-14
year	1.0	6.001042	6.001042	25.548647	1.085130e-06
Residual	174.0	40.870318	0.234887	NaN	NaN

<statsmodels.regression.linear_model.RegressionResultsWrapper at 0x76450324dc40>

```
identify_least_significant_feature(results, alpha=LOS_Alpha)
```

We find the least significant variable in this model is cylinders with a p-value of 0.018006259822639592 and a coefficient of 0.17760733234778334

Using the backward methodology, we suggest dropping cylinders from the new model

```
cols.remove("cylinders")
formula = " + ".join(cols)
formula += " - 1"
results = perform_analysis("mpg", formula, Auto_postos)
```

```

=====
                        OLS Regression Results
=====
Dep. Variable:          mpg      R-squared (uncentered):
0.763
Model:                  OLS      Adj. R-squared (uncentered):
0.759
Method:                 Least Squares      F-statistic:
187.7
Date:                   Tue, 25 Feb 2025      Prob (F-statistic):
1.90e-54
Time:                   14:37:57      Log-Likelihood:
-124.49
No. Observations:       178      AIC:
255.0
Df Residuals:           175      BIC:
264.5
Df Model:                3
Covariance Type:        nonrobust
=====

```

	coef	std err	t	P> t	[0.025	0.975]
horsepower	-0.2653	0.068	-3.888	0.000	-0.400	-0.131
weight	-0.5548	0.067	-8.238	0.000	-0.688	-0.422
year	0.1889	0.039	4.793	0.000	0.111	0.267

```
=====
Omnibus:                15.435    Durbin-Watson:                1.592
Prob(Omnibus):          0.000    Jarque-Bera (JB):          16.821
Skew:                   0.690    Prob(JB):                  0.000223
Kurtosis:               3.601    Cond. No.                  3.56
=====
```

Notes:

[1] R^2 is computed without centering (uncentered) since the model does not contain a constant.

[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.

	df	sum_sq	mean_sq	F	PR(>F)
horsepower	1.0	112.958534	112.958534	468.320143	2.378397e-51
weight	1.0	17.289976	17.289976	71.683331	9.907388e-15
year	1.0	5.541596	5.541596	22.975165	3.490133e-06
Residual	175.0	42.209894	0.241199	NaN	NaN

```
<statsmodels.regression.linear_model.RegressionResultsWrapper at 0x7645032a19d0>
```

```
identify_least_significant_feature(results, alpha=LOS_Alpha)
```

No variables are statistically insignificant.

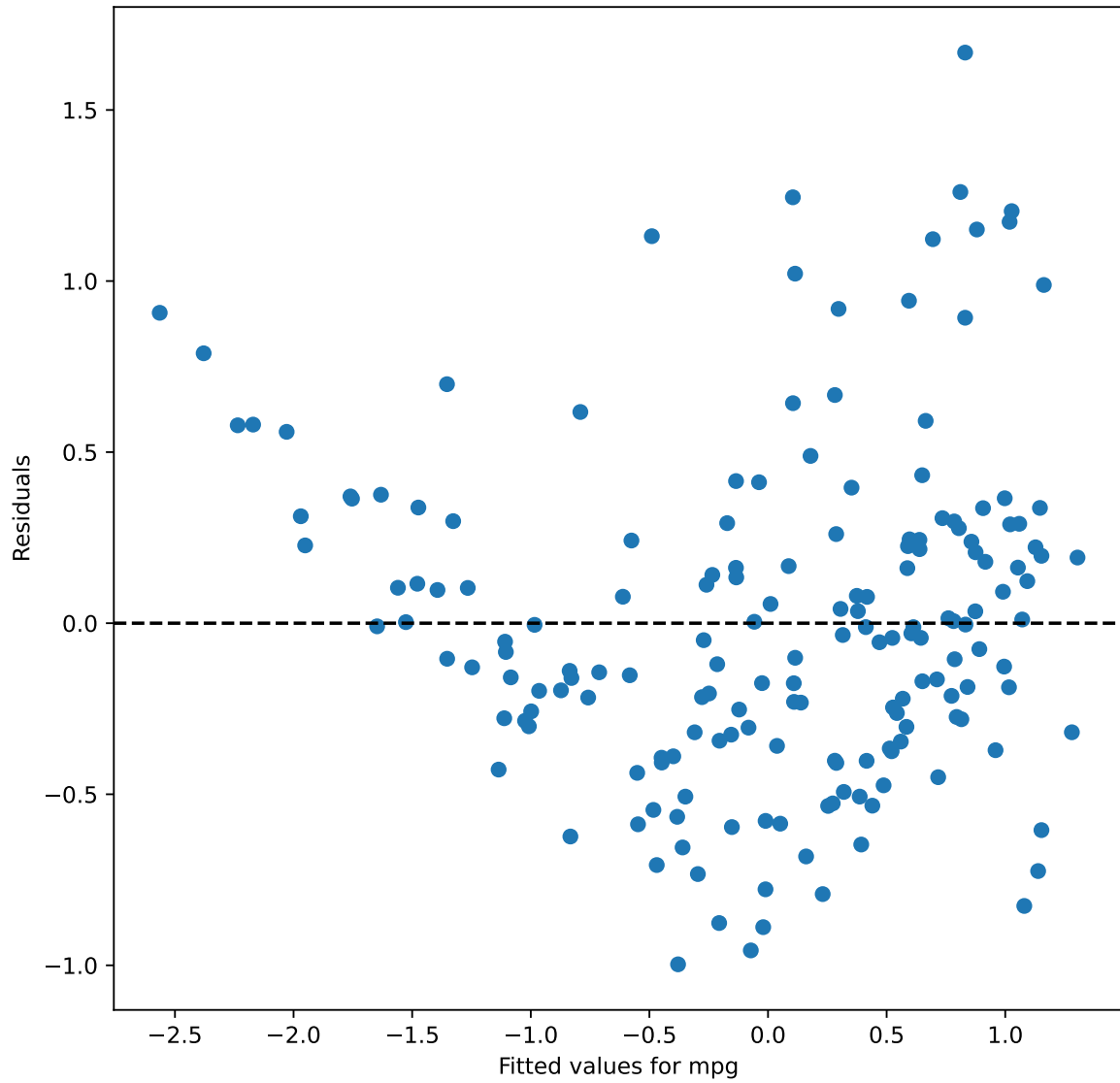
The model `mpg ~ horsepower + weight + year - 1` cannot be pruned further.

```
postoilshock_model = results
```

```
<statsmodels.regression.linear_model.RegressionResultsWrapper at 0x7645032a19d0>
```

Residual plot for model for post-oil shock

```
display_residuals_plot(results)
```



Pre-oilshock model

```
preoilshock_model.model.formula
```

```
'mpg ~ weight + year + origin_Europe + origin_Japan'
```

Explanatory power of preoilshock model

```
preoilshock_model.rsquared_adj
```

```
0.8404849876892488
```

```
### Post-oil shock model without intercept
```

```
postoilshock_model.model.formula
```

```
'mpg ~ horsepower + weight + year - 1'
```

Explanatory power of postoilshock model

```
postoilshock_model.rsquared_adj
```

```
0.7588006068263029
```

- Thus, we can conclude that prior to the oil shock of 1973, mileage was determined mostly by weight, year and origin.
- Post the oil shock of 1973, mileage was determined by horsepower, weight and year. Origin no longer played an important role as before.

Post oil shock model with intercept (Corollary)

```
postoilshock_model_intercept.model.formula
```

```
'mpg ~ cylinders + horsepower + weight + year + origin_Europe + origin_Japan'
```

Explanatory power of postoilshock model with intercept

```
postoilshock_model_intercept.rsquared_adj
```

```
0.7783620129852484
```

Finished

```
allDone()
```

```
<IPython.lib.display.Audio object>
```