NOTEBOOK DESTINADO A REGISTRAR AS TAREFAS DA DISCIPLINA DE AEDI - 1º/2025

PROFESSOR: JOÃO GABRIEL DE MORAES SOUZA

ALUNO: GUSTAVO PARREIRA LIMA CUNHA

TAREFA 3

QUESTÃO A - COMPARAÇÃO DE PREÇOS ENTRE CARACTERÍSTICAS

In [31]:
```
import kagglehub
import pandas as pd
import os


# Baixar dataset
path = kagglehub.dataset_download("prevek18/ames-housing-dataset")
print("Path to dataset files:", path)

# leitura do arquivo
csv_path = os.path.join(path, "AmesHousing.csv")
df = pd.read_csv(csv_path)

# Análise exploratória
print(df.info())
display(df)

print(df['Lot Area'].agg(['min', 'mean', 'max']))
print()

print(df['Gr Liv Area'].agg(['min', 'mean', 'max']))
print()

print(df['Year Remod/Add'].agg(['min', 'mean', 'max']))
print()

print(df['SalePrice'].agg(['min', 'mean', 'max']))
print()
```

```
Path to dataset files: /kaggle/input/ames-housing-dataset
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2930 entries, 0 to 2929
Data columns (total 82 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Order            2930 non-null   int64
 1   PID              2930 non-null   int64
 2   MS SubClass      2930 non-null   int64
 3   MS Zoning        2930 non-null   object
 4   Lot Frontage     2440 non-null   float64
 5   Lot Area         2930 non-null   int64
 6   Street           2930 non-null   object
 7   Alley            198 non-null    object
 8   Lot Shape        2930 non-null   object
 9   Land Contour     2930 non-null   object
 10  Utilities        2930 non-null   object
 11  Lot Config       2930 non-null   object
 12  Land Slope       2930 non-null   object
 13  Neighborhood     2930 non-null   object
 14  Condition 1      2930 non-null   object
 15  Condition 2      2930 non-null   object
 16  Bldg Type        2930 non-null   object
 17  House Style      2930 non-null   object
 18  Overall Qual     2930 non-null   int64
 19  Overall Cond     2930 non-null   int64
 20  Year Built       2930 non-null   int64
 21  Year Remod/Add   2930 non-null   int64
 22  Roof Style       2930 non-null   object
 23  Roof Matl        2930 non-null   object
 24  Exterior 1st     2930 non-null   object
 25  Exterior 2nd     2930 non-null   object
 26  Mas Vnr Type     1155 non-null   object
 27  Mas Vnr Area     2907 non-null   float64
 28  Exter Qual       2930 non-null   object
 29  Exter Cond       2930 non-null   object
 30  Foundation       2930 non-null   object
 31  Bsmt Qual        2850 non-null   object
 32  Bsmt Cond        2850 non-null   object
 33  Bsmt Exposure    2847 non-null   object
 34  BsmtFin Type 1   2850 non-null   object
 35  BsmtFin SF 1     2929 non-null   float64
 36  BsmtFin Type 2   2849 non-null   object
 37  BsmtFin SF 2     2929 non-null   float64
 38  Bsmt Unf SF      2929 non-null   float64
 39  Total Bsmt SF    2929 non-null   float64
 40  Heating          2930 non-null   object
 41  Heating QC       2930 non-null   object
 42  Central Air      2930 non-null   object
 43  Electrical       2929 non-null   object
 44  1st Flr SF       2930 non-null   int64
 45  2nd Flr SF       2930 non-null   int64
 46  Low Qual Fin SF  2930 non-null   int64
 47  Gr Liv Area      2930 non-null   int64
 48  Bsmt Full Bath   2928 non-null   float64
 49  Bsmt Half Bath   2928 non-null   float64
 50  Full Bath        2930 non-null   int64
 51  Half Bath        2930 non-null   int64
 52  Bedroom AbvGr    2930 non-null   int64
 53  Kitchen AbvGr    2930 non-null   int64
```

```
 54  Kitchen Qual     2930 non-null   object
 55  TotRms AbvGrd    2930 non-null   int64
 56  Functional       2930 non-null   object
 57  Fireplaces       2930 non-null   int64
 58  Fireplace Qu     1508 non-null   object
 59  Garage Type      2773 non-null   object
 60  Garage Yr Blt    2771 non-null   float64
 61  Garage Finish    2771 non-null   object
 62  Garage Cars      2929 non-null   float64
 63  Garage Area      2929 non-null   float64
 64  Garage Qual      2771 non-null   object
 65  Garage Cond      2771 non-null   object
 66  Paved Drive      2930 non-null   object
 67  Wood Deck SF     2930 non-null   int64
 68  Open Porch SF    2930 non-null   int64
 69  Enclosed Porch   2930 non-null   int64
 70  3Ssn Porch       2930 non-null   int64
 71  Screen Porch     2930 non-null   int64
 72  Pool Area        2930 non-null   int64
 73  Pool QC          13 non-null     object
 74  Fence            572 non-null    object
 75  Misc Feature     106 non-null    object
 76  Misc Val         2930 non-null   int64
 77  Mo Sold          2930 non-null   int64
 78  Yr Sold          2930 non-null   int64
 79  Sale Type        2930 non-null   object
 80  Sale Condition   2930 non-null   object
 81  SalePrice        2930 non-null   int64
dtypes: float64(11), int64(28), object(43)
memory usage: 1.8+ MB
None
```

	Order	PID	MS SubClass	MS Zoning	Lot Frontage	Lot Area	Street	Alley	Lot Shape	La Contc
0	1	526301100	20	RL	141.0	31770	Pave	NaN	IR1	
1	2	526350040	20	RH	80.0	11622	Pave	NaN	Reg	
2	3	526351010	20	RL	81.0	14267	Pave	NaN	IR1	
3	4	526353030	20	RL	93.0	11160	Pave	NaN	Reg	
4	5	527105010	60	RL	74.0	13830	Pave	NaN	IR1	
...	...	...	...	...	...	...	...	...	...	
2925	2926	923275080	80	RL	37.0	7937	Pave	NaN	IR1	
2926	2927	923276100	20	RL	NaN	8885	Pave	NaN	IR1	L
2927	2928	923400125	85	RL	62.0	10441	Pave	NaN	Reg	
2928	2929	924100070	20	RL	77.0	10010	Pave	NaN	Reg	
2929	2930	924151050	60	RL	74.0	9627	Pave	NaN	Reg	

2930 rows × 82 columns

```
min       1300.000000
mean     10147.921843
max     215245.000000
Name: Lot Area, dtype: float64

min      334.000000
mean    1499.690444
max     5642.000000
Name: Gr Liv Area, dtype: float64

min     1950.000000
mean    1984.266553
max     2010.000000
Name: Year Remod/Add, dtype: float64

min      12789.000000
mean    180796.060068
max     755000.000000
Name: SalePrice, dtype: float64
```

Escolha das características da propriedade:

1. Lot Area: área total do terreno
2. Gr Liv Area: Ground Living Area: área habitável acima do nível do solo
3. Year Remod/Add: Ano em que o imóvel foi reformado ou teve alguma ampliação adicionada.

Essas características supostamente impactam diretamente no valor da casa, pois indicam a área construída, tamanho do terreno e o quão nova é a casa ou parte dela.

Para utilizar a ANOVA, é necessário analisar a distribuição dos resíduos.

In [32]:
```
#filtrando dados

dados_filtrados = df[['Lot Area', 'Gr Liv Area', 'Year Remod/Add', 'SalePrice']]
display(dados_filtrados)
```

	Lot Area	Gr Liv Area	Year Remod/Add	SalePrice
0	31770	1656	1960	215000
1	11622	896	1961	105000
2	14267	1329	1958	172000
3	11160	2110	1968	244000
4	13830	1629	1998	189900
...	...	...	...	...
2925	7937	1003	1984	142500
2926	8885	902	1983	131000
2927	10441	970	1992	132000
2928	10010	1389	1975	170000
2929	9627	2000	1994	188000

2930 rows × 4 columns

Serão comparados os preços de venda médio para diferentes níveis das 3 características. Como elas são contínuas, serão criadas categorias para cada uma delas.

Para criação das categorias, será usada a função qcut(), que divide o conjunto de dados de acordo com o número de observações, ou seja, percentis.

qcut() não divide pelos valores, mas pela posição no ranking (percentis).

In [33]:
```
df['LotArea_cat'] = pd.qcut(df['Lot Area'], q=4, labels=['Pequeno', 'Médio', 'Gran

df['GrLivArea_cat'] = pd.qcut(df['Gr Liv Area'], q=4, labels=['Pequeno', 'Médio',

df['YearRemod_Add_cat'] = pd.qcut(df['Year Remod/Add'], q=4, labels=['Antiga', 'De

from statsmodels.formula.api import ols
import statsmodels.api as sm

modelo_lot = ols('SalePrice ~ C(LotArea_cat)', data=df).fit()
anova_lot = sm.stats.anova_lm(modelo_lot, typ=2)
print(anova_lot)

modelo_gr = ols('SalePrice ~ C(GrLivArea_cat)', data=df).fit()
anova_gr = sm.stats.anova_lm(modelo_gr, typ=2)
print(anova_gr)

modelo_yr = ols('SalePrice ~ C(YearRemod_Add_cat)', data=df).fit()
anova_yr = sm.stats.anova_lm(modelo_yr, typ=2)
print(anova_yr)
```

```
                     sum_sq      df           F         PR(>F)
C(LotArea_cat)  3.491763e+12     3.0  224.043355  7.769813e-131
Residual        1.520077e+13  2926.0         NaN            NaN
                      sum_sq      df           F  PR(>F)
C(GrLivArea_cat)  7.845476e+12     3.0  705.440377     0.0
Residual          1.084706e+13  2926.0         NaN     NaN
                           sum_sq      df           F         PR(>F)
C(YearRemod_Add_cat)  5.222605e+12     3.0  378.159349  1.477807e-207
Residual              1.346993e+13  2926.0         NaN            NaN
```

É necessário aplicar ANOVA, mesmo sem confirmar os requisitos, para avaliar posteriormente os resíduos.

A aplicação do ANOVA mostrou uma forte relação entre o preço de venda do imóvel com a área do lote, a área construída da casa e também com a data da última reforma. Essas conclusões se basearam nos p-valores extremamente baixos obtidos nas três análises.

Um p-valor baixo indica que é improvável observar diferenças tão grandes entre as médias dos grupos se a hipótese nula fosse verdadeira. No contexto da ANOVA, a hipótese nula afirma que todas as médias populacionais entre os grupos são iguais, enquanto a hipótese alternativa sustenta que ao menos uma das médias difere das demais.

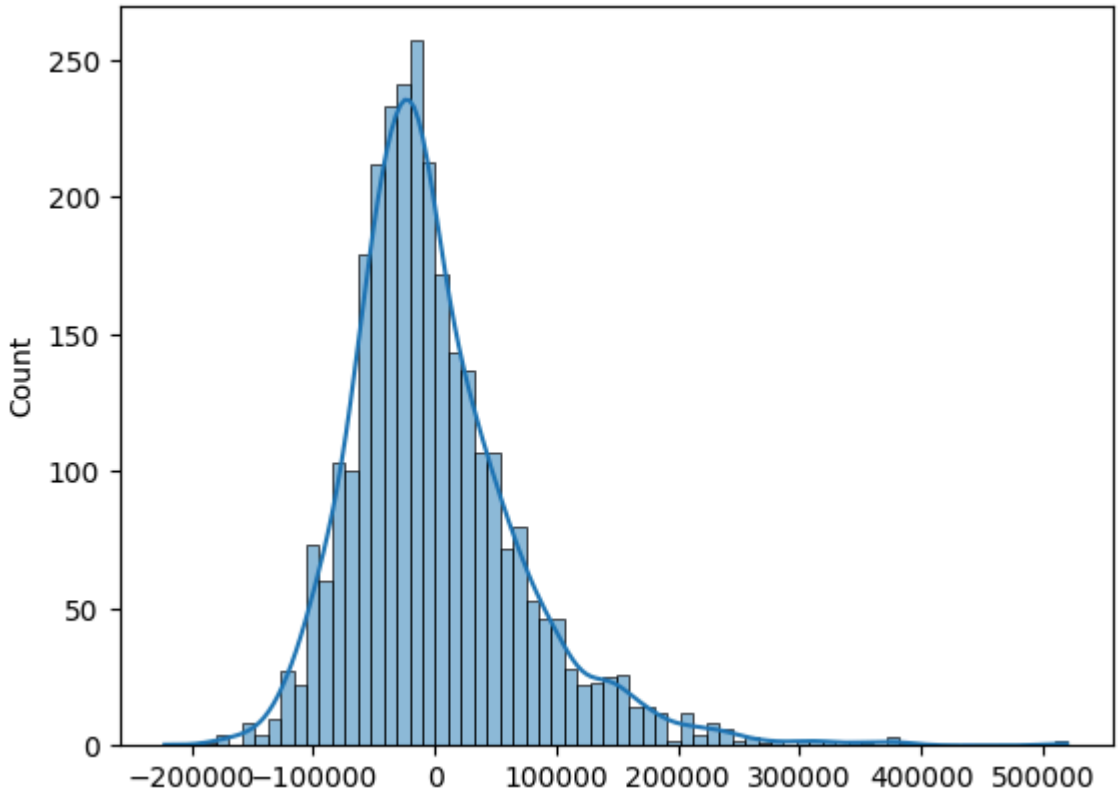
Como os p-valores foram próximos de zero, há forte evidência estatística contra a hipótese nula, o que leva à sua rejeição. Isso implica que as médias de preço de venda diferem significativamente entre os grupos definidos pelas variáveis categorizadas (LotArea, GrLivArea e YearRemod/Add).

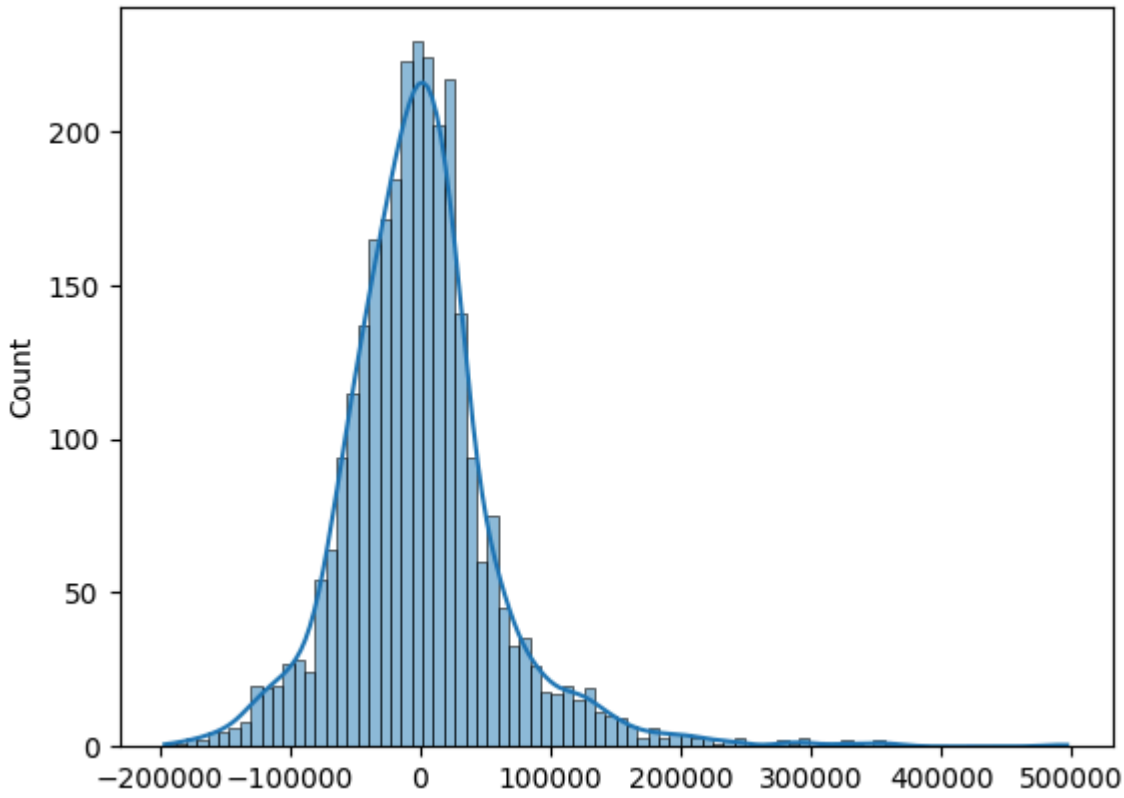
Importante destacar que essa conclusão inicial parte da suposição de que os pressupostos do teste ANOVA são atendidos — especialmente a normalidade dos resíduos e a homogeneidade das variâncias. Por isso, é necessário, após o ajuste do modelo, avaliar se esses pressupostos se confirmam. Caso sejam violados, recomenda-se o uso de testes não paramétricos, como o de Kruskal-Wallis, que não dependem dessas condições.

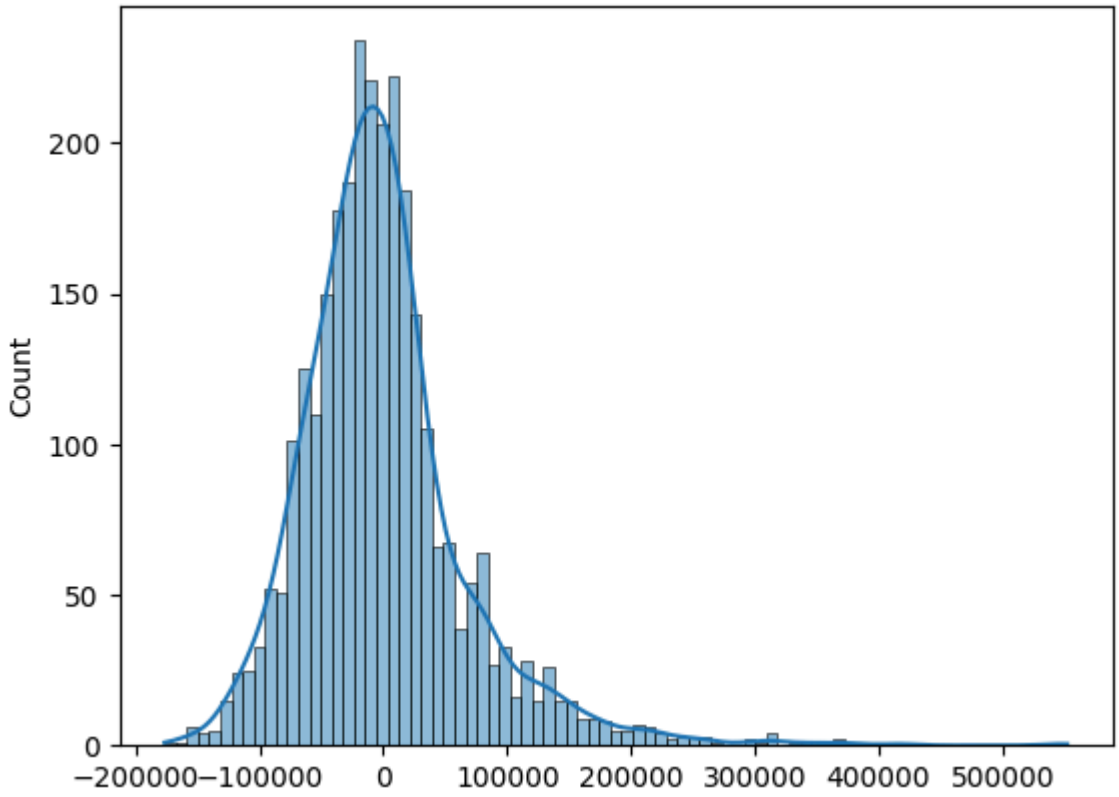
Verificação dos resíduos

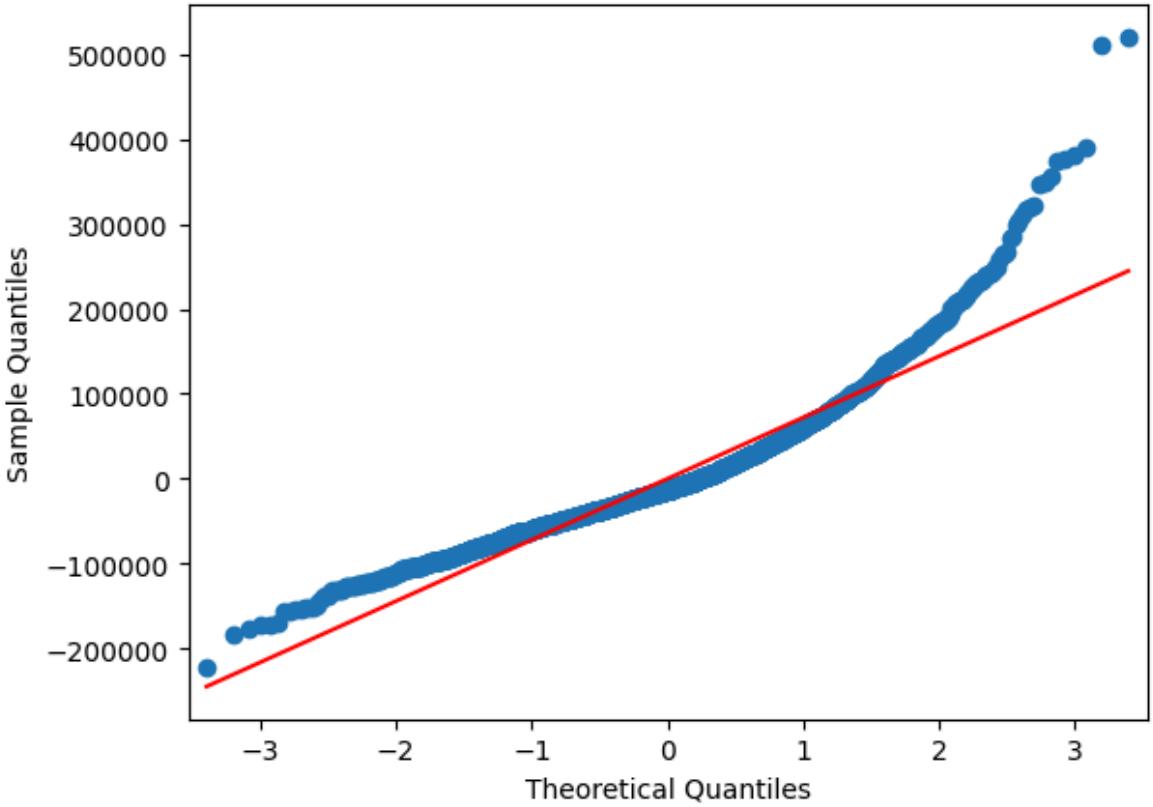
In [34]:
```
residuos_lot = modelo_lot.resid

residuos_gr = modelo_gr.resid

residuos_yr = modelo_yr.resid

#HISTOGRAMA

import seaborn as sns
import matplotlib.pyplot as plt

sns.histplot(residuos_lot, kde=True)
plt.show()
print()
sns.histplot(residuos_gr, kde=True)
plt.show()
```

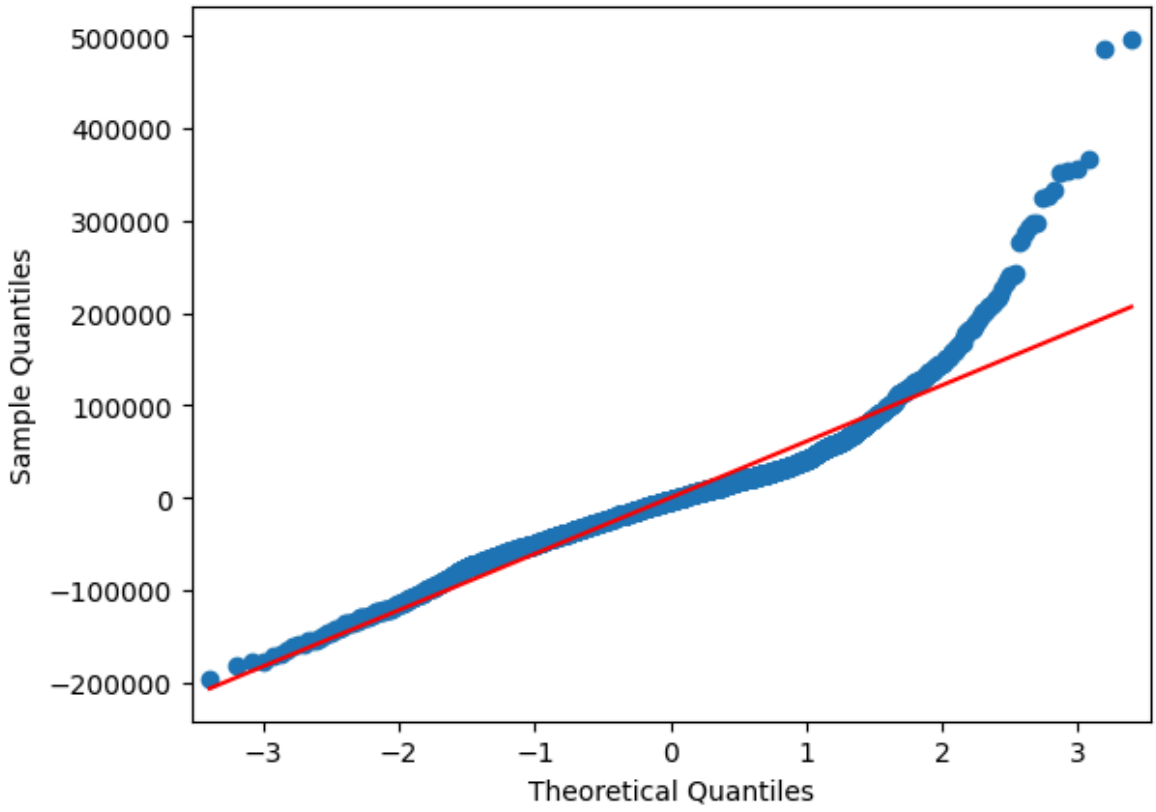
```
print()
sns.histplot(residuos_yr, kde=True)
plt.show()
print()

# QQ-plot

sm.qqplot(residuos_lot, line='s')
plt.show()
print()

sm.qqplot(residuos_gr, line='s')
plt.show()
print()

sm.qqplot(residuos_yr, line='s')
plt.show()
print()

from scipy.stats import shapiro

est_lot, p_valor_lot = shapiro(residuos_lot)
print()

est_gr, p_valor_gr = shapiro(residuos_gr)
print()

est_yr, p_valor_yr = shapiro(residuos_yr)
print()


# Exibindo o resultado
print(f'Estatística de Shapiro-Wilk: {est_lot}')
print(f'Valor-p: {p_valor_lot}')
print()

print(f'Estatística de Shapiro-Wilk: {est_gr}')
print(f'Valor-p: {p_valor_gr}')
print()

print(f'Estatística de Shapiro-Wilk: {est_yr}')
print(f'Valor-p: {p_valor_yr}')
print()
```

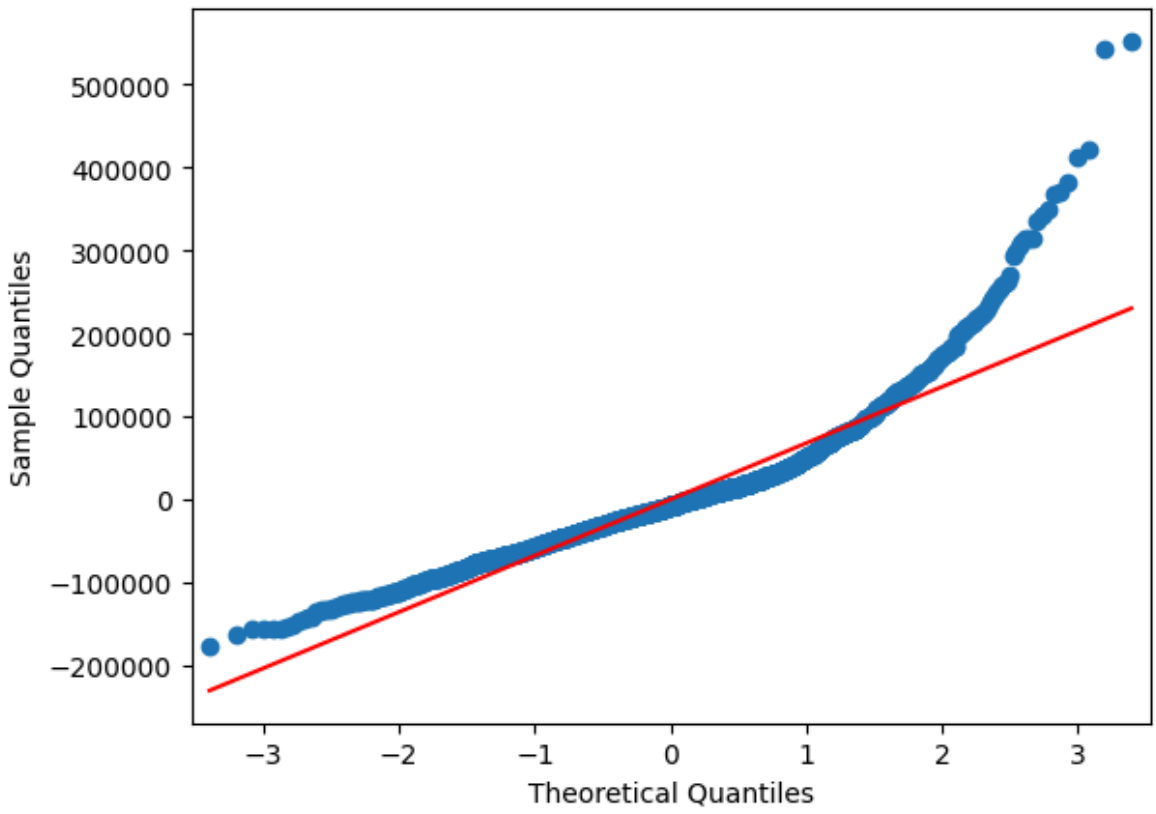

Count
250
200
150
100
50
0
−200000
−100000
0
100000
200000
300000
400000
500000


Count
200
150
100
50
0
−200000
−100000
0
100000
200000
300000
400000
500000


Count
200
150
100
50
0
−200000
−100000
0
100000
200000
300000
400000
500000


Sample Quantiles
500000
400000
300000
200000
100000
0
−100000
−200000
−3
−2
−1
0
1
2
3
Theoretical Quantiles


500000
400000
300000
200000
100000
0
−100000
−200000
Sample Quantiles
−3
−2
−1
0
1
2
3
Theoretical Quantiles


500000
400000
300000
200000
100000
0
−100000
−200000
Sample Quantiles
−3
−2
−1
0
1
2
3
Theoretical Quantiles

```
Estatística de Shapiro-Wilk: 0.9189826745718804
Valor-p: 4.664381685064501e-37

Estatística de Shapiro-Wilk: 0.9153699669493505
Valor-p: 1.1055244200720357e-37

Estatística de Shapiro-Wilk: 0.9022796008269076
Valor-p: 8.784409182961244e-40
```

Apesar de, aparentemente, o histograma dos resíduos sugerir uma distribuição normal, o QQ-Plot evidencia a não normalidade dos erros. Serão a seguir executados mais testes para confirmar esse achado.

Para confirmar a não normalidade dos resíduos, será realizado um teste de Kolmogorov-Smirnov (K-S)

In [35]:
```
from scipy.stats import kstest
from scipy.stats import zscore

# Padroniza os resíduos (média = 0, desvio padrão = 1)
residuos_padronizados_lot = zscore(residuos_lot)

# Aplica o teste K-S contra uma normal padrão
estatistica_lot_ks, p_valor_lot_ks = kstest(residuos_padronizados_lot, 'norm')

# Mostra os resultados
print(f'Estatística de Kolmogorov-Smirnov: {estatistica_lot_ks}')
print(f'Valor-p: {p_valor_lot_ks}')
print()


# Padroniza os resíduos (média = 0, desvio padrão = 1)
residuos_padronizados_gr = zscore(residuos_gr)

# Aplica o teste K-S contra uma normal padrão
estatistica_gr_ks, p_valor_gr_ks = kstest(residuos_padronizados_gr, 'norm')

# Mostra os resultados
print(f'Estatística de Kolmogorov-Smirnov: {estatistica_gr_ks}')
print(f'Valor-p: {p_valor_gr_ks}')
print()


# Padroniza os resíduos (média = 0, desvio padrão = 1)
residuos_padronizados_yr = zscore(residuos_yr)

# Aplica o teste K-S contra uma normal padrão
estatistica_yr_ks, p_valor_yr_ks = kstest(residuos_padronizados_yr, 'norm')

# Mostra os resultados
print(f'Estatística de Kolmogorov-Smirnov: {estatistica_yr_ks}')
print(f'Valor-p: {p_valor_yr_ks}')
print()
```

```
Estatística de Kolmogorov-Smirnov: 0.10008942199161042
Valor-p: 5.268047736762113e-26

Estatística de Kolmogorov-Smirnov: 0.10186218515306078
Valor-p: 6.396008018140674e-27

Estatística de Kolmogorov-Smirnov: 0.11007159197385363
Valor-p: 2.25763795648472e-31
```

O teste K-S confirmou a não normalidade dos resíduos, portanto será aplicado o teste de Kruskal-Wallis, que não assume normalidade dos resíduos.

In [36]:
```
from scipy.stats import kruskal

# Lote
grupos_lot = [df[df['LotArea_cat'] == cat]['SalePrice'] for cat in df['LotArea_cat
estat_lot, pval_lot = kruskal(*grupos_lot)
print(f"LotArea_cat — Estatística: {estat_lot:.2f} | p-valor: {pval_lot:.4e}")
print()

# Área construída
grupos_gr = [df[df['GrLivArea_cat'] == cat]['SalePrice'] for cat in df['GrLivArea_
estat_gr, pval_gr = kruskal(*grupos_gr)
print(f"GrLivArea_cat — Estatística: {estat_gr:.2f} | p-valor: {pval_gr:.4e}")
print()

# Ano da reforma
grupos_yr = [df[df['YearRemod_Add_cat'] == cat]['SalePrice'] for cat in df['YearRe
estat_yr, pval_yr = kruskal(*grupos_yr)
print(f"YearRemod_Add_cat — Estatística: {estat_yr:.2f} | p-valor: {pval_yr:.4e}")
print()
```

```
LotArea_cat — Estatística: 530.55 | p-valor: 1.1402e-114

GrLivArea_cat — Estatística: 1427.97 | p-valor: 2.5113e-309

YearRemod_Add_cat — Estatística: 1003.27 | p-valor: 3.5069e-217
```

Interpretação dos resultados.

Após a verificação da normalidade dos resíduos do modelo ajustado com ANOVA, constatou-se, por meio dos testes de Shapiro-Wilk e Kolmogorov-Smirnov, que os resíduos não seguem uma distribuição normal. Diante disso, optou-se pela aplicação do teste não paramétrico de Kruskal-Wallis, que não exige a suposição de normalidade nem de homogeneidade das variâncias entre os grupos.

Os resultados do teste de Kruskal-Wallis para as três variáveis categorizadas foram os seguintes:

LotArea_cat — Estatística H: 530,55 | p-valor: $1{,}14 \times 10^{-114}$

GrLivArea_cat — Estatística H: 1427,97 | p-valor: $2{,}51 \times 10^{-309}$

YearRemod_Add_cat — Estatística H: 1003,27 | p-valor: $3{,}51 \times 10^{-217}$

Em todos os casos, os p-valores obtidos foram extremamente baixos, permitindo rejeitar a hipótese nula de que as distribuições dos preços de venda são iguais entre os grupos definidos por cada variável categorizada. Isso indica que as diferenças observadas entre os grupos são estatisticamente significativas, ou seja, os valores medianos (ou, de forma mais ampla, a distribuição dos dados) de preço de venda variaram de forma consistente entre os diferentes níveis de área do lote, área construída e ano da última reforma.

Esses resultados reforçam os achados da ANOVA e confirmam que, mesmo sem os pressupostos paramétricos, há evidências robustas de associação entre essas variáveis explicativas e o preço de venda dos imóveis.

BIBLIOGRAFIA

1- Azizi, Fateme, Rasoul Ghasemi, and Maryam Ardalan. "Two common mistakes in applying anova test: guide for biological researchers." Preprint 10 (2022).

2- Larson, Martin G. "Analysis of variance." Circulation 117.1 (2008): 115-121.