Relatório de Análise de Dados: Presença e Influência dos Dados no Setor Imobiliário de São Paulo

Pedro Miller, Pedro Bolzan, Hálison P Q Souza, Luis Martins, Daniel Muradas Rodrigues

25 de junho de 2025

Conteúdo

1 Introdução

Este relatório detalha a aplicação do ciclo de ciência de dados para explorar a presença e a influência dos dados no setor imobiliário da cidade de São Paulo. O objetivo principal é conduzir uma análise passo a passo, desde a coleta e organização dos dados até a sua exploração e análise, culminando na apresentação de conclusões através de visualizações. O trabalho busca responder a questões fundamentais sobre os fatores que impactam o preço dos imóveis e a relevância da localização nesse contexto.

Questões de Pesquisa:

- Como as características dos imóveis (área, número de quartos/banheiros, vagas de estacionamento) influenciam seu preço de venda?
- Qual o impacto da localização (bairro) na variação do preço por metro quadrado dos apartamentos em São Paulo?
- Existem outros fatores, como o tempo de anúncio, que correlacionam com o preço dos imóveis?

2 Coleta e Visão Geral dos Dados

A etapa inicial envolveu a aquisição de um dataset relevante para o setor imobiliário de São Paulo, focado em apartamentos. Após a obtenção dos dados, foi realizada uma inspeção preliminar para entender sua estrutura, o tipo de informações contidas e a presença de valores ausentes, o que é fundamental para planejar as próximas fases de preparação.

2.1 Carregamento do Dataset

O dataset foi carregado utilizando a biblioteca Pandas no Python.

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

# Configura  es para melhorar a visualiza  o de gr ficos (opcional)
sns.set_style("whitegrid")
plt.rcParams['figure.figsize'] = (10, 6)
plt.rcParams['figure.dpi'] = 100

# Carregar o dataset imobili rio de apartamentos de S o Paulo
df = pd.read_csv('SaoPaulo_OnlyAppartments.csv')

print("Dataset carregado com sucesso!")
```

Listing 1: Carregamento do Dataset

2.2 Inspeção Inicial do Dataset

Para uma visão geral, as primeiras linhas, informações sobre as colunas (tipos de dados e valores não-nulos) e estatísticas descritivas das colunas numéricas foram inspecionadas.

```
# Visualizar as primeiras linhas do DataFrame
print("### Primeiras 5 linhas do Dataset:")
print(df.head())

# Obter informa  es sobre as colunas e tipos de dados
print("\n### Informa  es Detalhadas do Dataset:")
df.info()

# Estat sticas descritivas das colunas num ricas
# Isso nos d  uma ideia da distribui  o, m dia, desvio padr o e valores m nimo/
    m ximo.
print("\n### Estat sticas Descritivas das Colunas Num ricas (Dados Originais):")
print(df.describe())
```

Listing 2: Inspeção Inicial do Dataset

Saída da Inspeção Inicial (Exemplo)

```
 ID created_date Price below_price Area
0 2742871875 2018-05-24T15:02:18Z 2000000 False 62
1 2670612721 2018-12-18T14:40:10Z 1015680 False 42
...

<class 'pandas.core.frame.DataFrame'>
RangeIndex:  27828 entries, 0 to 27827
Data columns (total 10 columns):
# Column Non-Null Count Dtype
-- --- ------- ---
0 ID 27828 non-null int64
1 created_date 27828 non-null object
2 Price 27828 non-null int64
...
dtypes:  bool(1), int64(6), object(3)
memory usage:  1.9+ MB

Estatísticas Descritivas:
ID Price Area Bedrooms Bathrooms
count 2.782800e+04 2.782800e+04 27828.000000 27828.000000 27828.000000
mean 2.685599e+09 2.037172e+06 136.288163 2.740477 2.831177
...
max 2.758322e+09 1.220290e+08 3367.000000 20.000000 20.000000
Parking_Spaces
count 27828.000000
mean 2.153730
std 2.927744
min 0.000000
25% 1.000000
50% 2.000000
75% 3.000000
max 350.000000
```

3 Preparação e Limpeza de Dados

A qualidade da análise de dados depende diretamente da qualidade dos dados brutos. Nesta seção, o dataset foi preparado e limpo para garantir consistência e confiabilidade, construindo o dataset de trabalho final.

3.1 Extração da Informação do Bairro

A coluna 'Adress' continha o endereço completo. Para facilitar a análise baseada em localização, o nome do bairro foi extraído para uma nova coluna 'Bairro'.

```
# Extrair o nome do Bairro da coluna 'Adress'
# A l gica tenta dividir a string pelo primeiro ',', depois pelo primeiro '-' e remove
    espa os extras.
# Se a extra  o falhar, o bairro ser  classificado como 'Desconhecido'.
df['Bairro'] = df['Adress'].apply(lambda x: x.split(',')[1].split('-')[0].strip() if pd.
    notnull(x) and ',' in x and '-' in x.split(',')[1] else 'Desconhecido')

# Verificar a contagem dos bairros mais frequentes, incluindo 'Desconhecido'
print("### Top 10 Bairros (antes da remo  o de 'Desconhecido'):")
print(df['Bairro'].value_counts().head(10))
```

Listing 3: Extração da Coluna Bairro

3.2 Tratamento de Colunas de Data

As colunas 'created$_{d}ate$'e'$extract_{d}ate$' $foramconvertidasparaotipo$'$datetime$'$parapermitiroperaestemporais.Almdisso, f$

```python
# Converter colunas de data para o tipo datetime
print("Tipos de dados antes da convers o de datas:")
print(df[['created_date', 'extract_date']].dtypes)

# Converter created_date e remover a informa  o de fuso hor rio (tz-localize(None))
df['created_date'] = pd.to_datetime(df['created_date']).dt.tz_localize(None)

# Convert extract_date
df['extract_date'] = pd.to_datetime(df['extract_date'])

print("\nTipos de dados ap s a convers o de datas:")
print(df[['created_date', 'extract_date']].dtypes)

# Criar a coluna 'age_of_listing_days' (idade do an ncio em dias)
# Esta coluna representa h  quanto tempo o im vel estava anunciado na data da
    extra  o do dado.
df['age_of_listing_days'] = (df['extract_date'] - df['created_date']).dt.days

print("\n### Primeiras linhas com a nova coluna 'age_of_listing_days':")
print(df[['created_date', 'extract_date', 'age_of_listing_days']].head())

# Verificar se h  idades negativas (inconsist ncia de dados)
print("\n### N mero de an ncios com idade negativa (data de cria  o > data de
    extra  o):")
print(df[df['age_of_listing_days'] < 0].shape[0])
```

Listing 4: Conversão e Cálculo de Datas

3.3 Tratamento de Outliers em Variáveis Numéricas

A análise inicial revelou a presença de outliers significativos em colunas como 'Price', 'Area', 'Bedrooms', 'Bathrooms' e 'Parking$_spaces$'.$Paramitigaroimpactodessesvaloresextremos, foiaplicadaumaestratgiade*capping*(limitandoosvaloresaumpercentilde1\%e99\%)e, emcasosespecficos, removodevaloresclaramenteimplausveis.Av$

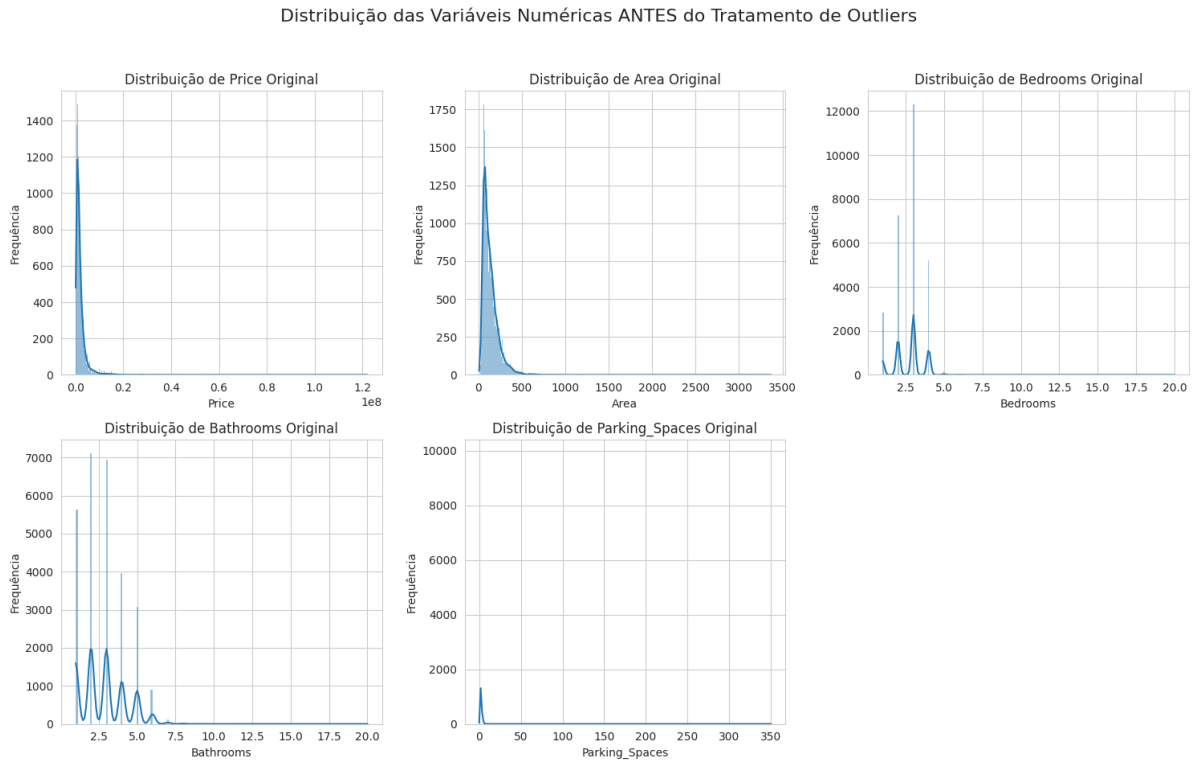

Figura 1: Distribuição das Variáveis Numéricas ANTES do Tratamento de Outliers

```python
# Aplicar Capping e Remo  es para Outliers
```

```
print("\n### In cio do Tratamento de Outliers ###")

cols_to_treat = ['Price', 'Area', 'Bedrooms', 'Bathrooms', 'Parking_Spaces']

for col in cols_to_treat:
    print(f"\nTratando outliers para a coluna: **{col}**")

    # Calcular percentis para capping (1% e 99%)
    lower_perc = df[col].quantile(0.01)
    upper_perc = df[col].quantile(0.99)

    # Antes do tratamento (valores originais do df atual)
    print(f"Valores ANTES do capping (min/max): {df[col].min():.2f} / {df[col].max():.2f
    }")

    # Aplicar capping: valores abaixo do percentil 1% s o ajustados para o valor do
    percentil 1%
    # e valores acima do percentil 99% s o ajustados para o valor do percentil 99%.
    df[col] = np.where(df[col] < lower_perc, lower_perc, df[col])
    df[col] = np.where(df[col] > upper_perc, upper_perc, df[col])

    print(f"Valores AP S capping (min/max): {df[col].min():.2f} / {df[col].max():.2f}")
    print(f"Outliers de {col} contidos via capping.")

# Remo  es espec ficas para valores fisicamente improv veis (p s-capping)
initial_rows_before_specific_removal = df.shape[0]

# Remover apartamentos com  rea  muito pequena (ex: menos de 20 m  pode ser erro ou
    est dio extremo)
df = df[df['Area'] >= 20]
removed_area_rows = initial_rows_before_specific_removal - df.shape[0]
print(f"\nRemovidos {removed_area_rows} linhas com  rea  < 20 m  .")

# Remover apartamentos com um n mero irreal de vagas de estacionamento (ex: mais de 10)
initial_rows_after_area_removal = df.shape[0]
df = df[df['Parking_Spaces'] <= 10]
removed_parking_rows = initial_rows_after_area_removal - df.shape[0]
print(f"Removidos {removed_parking_rows} linhas com Parking_Spaces > 10.")

print(f"\n### N mero final de linhas no dataset ap s remo  es espec ficas: {df.
    shape[0]} ###")
```

Listing 5: Tratamento de Outliers

```
# Re-verificar as estat sticas descritivas ap s o tratamento de outliers
print("\n### Estat sticas Descritivas Ap s Tratamento de Outliers: ###")
print(df[cols_to_treat].describe())
```

Listing 6: Estatísticas Descritivas Após Tratamento de Outliers

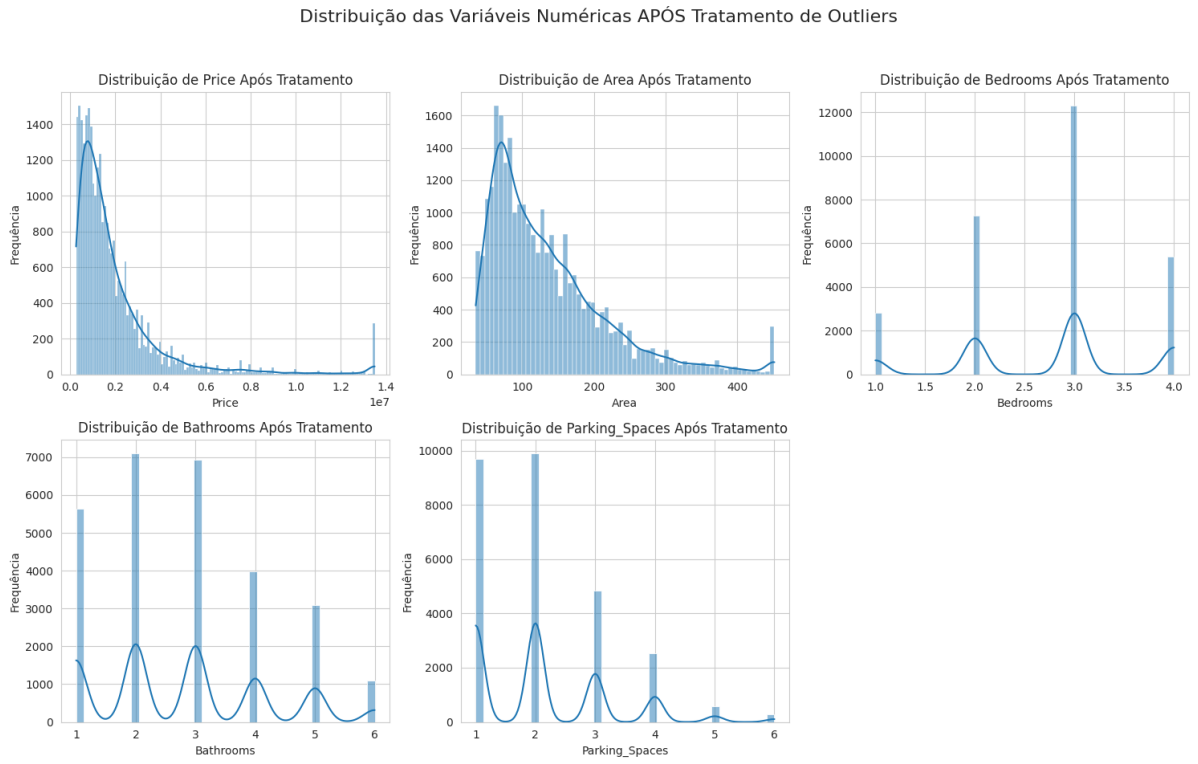

Figura 2: Distribuição das Variáveis Numéricas APÓS Tratamento de Outliers

3.4 Limpeza da Categoria 'Desconhecido' na Coluna Bairro

A extração do bairro resultou em alguns endereços não identificados, categorizados como 'Desconhecido'. Para assegurar a precisão da análise de localização, essas linhas foram removidas do dataset.

```
# Remover linhas onde a coluna 'Bairro'    'Desconhecido'
print("\n### Tratamento de Bairros 'Desconhecido' ###")

initial_rows_bairro = df.shape[0]
print(f"N mero de linhas antes da remo  o de Bairro 'Desconhecido': {
    initial_rows_bairro}")
print(f"Contagem original de 'Desconhecido': {df['Bairro'].value_counts().get('
    Desconhecido', 0)}")

df = df[df['Bairro'] != 'Desconhecido']
removed_rows_bairro = initial_rows_bairro - df.shape[0]

print(f"N mero de linhas removidas com Bairro 'Desconhecido': {removed_rows_bairro}")
print(f"Novo n mero de linhas no dataset: {df.shape[0]}")

# Verificar os 10 bairros mais frequentes ap s a remo  o
print("\n### Top 10 Bairros ap s remo  o de 'Desconhecido':")
print(df['Bairro'].value_counts().head(10))
print(f"Nova contagem de 'Desconhecido' (deve ser 0): {df['Bairro'].value_counts().get('
    Desconhecido', 0)}")
```

Listing 7: Limpeza de Bairros 'Desconhecido'

3.5 Criação da Métrica 'Preço por Metro Quadrado' (Price_per_sqm)

Para permitir uma comparação justa do valor de imóveis de diferentes tamanhos e aprofundar a análise da influência da localização, foi criada a métrica '$\text{Price}_{p}er_{s}qm$'.

```
# Calcular a coluna 'Price_per_sqm'
# Adicionar um pequeno valor (1e-6)     rea  para evitar divis o por zero.
df['Price_per_sqm'] = df['Price'] / (df['Area'] + 1e-6)
```

```
4
5  print("\n### Estat sticas Descritivas para 'Price_per_sqm':")
6  print(df['Price_per_sqm'].describe())
7
8  # Visualizar a distribui  o do Pre o por Metro Quadrado
9  plt.figure(figsize=(10, 6))
10 sns.histplot(df['Price_per_sqm'], kde=True, bins=50)
11 plt.title('Distribui  o do Pre o por Metro Quadrado')
12 plt.xlabel('Pre o por m  (R$)')
13 plt.ylabel('Frequ ncia')
14 plt.ticklabel_format(style='plain', axis='x')
15 plt.savefig('distribuicao_preco_por_m2.png')
16 plt.show()
17 print("Distribui  o do Pre o por m  salva como 'distribuicao_preco_por_m2.png'.")
```

Listing 8: Criação de Preço por Metro Quadrado

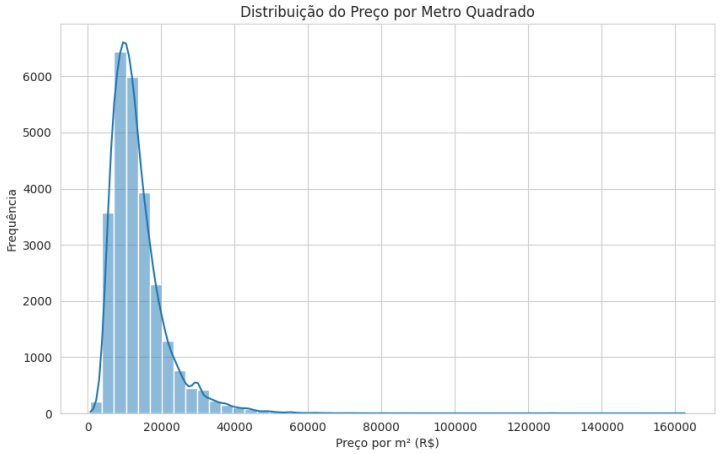

Figura 3: Distribuição do Preço por Metro Quadrado

4 Análise Exploratória de Dados (EDA) e Insights

Com o dataset limpo e preparado, a fase de Análise Exploratória de Dados (EDA) foi conduzida para identificar padrões, tendências e relações entre as variáveis, buscando responder às questões de pesquisa.

4.1 Correlação entre Variáveis Numéricas

A matriz de correlação foi utilizada para quantificar a relação linear entre as variáveis numéricas, permitindo identificar quais características exercem maior influência sobre o preço.

```
1  # Calcular e visualizar a matriz de correla  o
2  # Selecionar apenas as colunas num ricas relevantes para an lise de correla  o
3  numerical_cols_for_corr = ['Price', 'Area', 'Bedrooms', 'Bathrooms', 'Parking_Spaces', '
       age_of_listing_days', 'Price_per_sqm']
4
5  # Calcular a matriz de correla  o
6  correlation_matrix = df[numerical_cols_for_corr].corr()
7  print("### Matriz de Correla  o das Vari veis Num ricas:")
8  print(correlation_matrix)
9
10 # Visualizar a matriz de correla  o com um heatmap para melhor interpreta  o
11 plt.figure(figsize=(10, 8))
12 sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=.5)
13 plt.title('Matriz de Correla  o das Vari veis Num ricas')
14 plt.savefig('matriz_correlacao.png')
15 plt.show()
16 print("Matriz de Correla  o salva como 'matriz_correlacao.png'.")
```

Listing 9: Matriz de Correlação

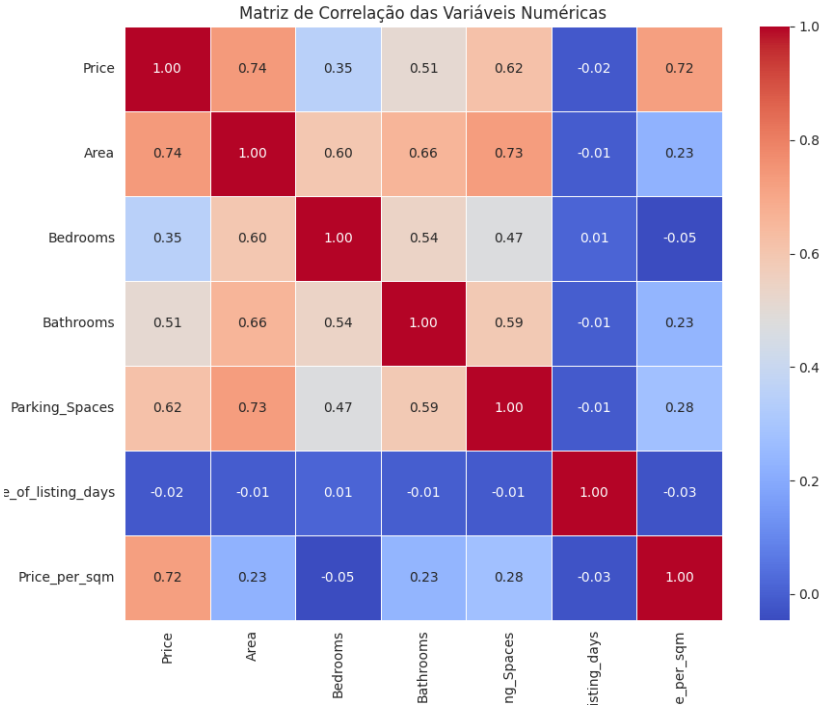

Figura 4: Matriz de Correlação das Variáveis Numéricas

4.2 Impacto das Características do Imóvel no Preço

Gráficos de dispersão foram gerados para visualizar a relação direta entre as características mais correlacionadas e o preço dos apartamentos.

```
# Gr ficos de dispers o (scatter plots) para as vari veis mais correlacionadas com o Pre o
plt.figure(figsize=(18, 6))

# Pre o vs.  rea : Espera-se uma rela  o positiva forte.
plt.subplot(1, 3, 1)
sns.scatterplot(x='Area', y='Price', data=df, alpha=0.6, s=20)
plt.title('Pre o vs.  rea ')
plt.xlabel(' rea  (m )')
plt.ylabel('Pre o (R$)')
plt.ticklabel_format(style='plain', axis='y')

# Pre o vs. Banheiros: Espera-se uma rela  o positiva.
plt.subplot(1, 3, 2)
sns.scatterplot(x='Bathrooms', y='Price', data=df, alpha=0.6, s=20)
plt.title('Pre o vs. Banheiros')
plt.xlabel('N mero de Banheiros')
plt.ylabel('Pre o (R$)')
plt.ticklabel_format(style='plain', axis='y')

# Pre o vs. Quartos: Espera-se uma rela  o positiva.
plt.subplot(1, 3, 3)
sns.scatterplot(x='Bedrooms', y='Price', data=df, alpha=0.6, s=20)
plt.title('Pre o vs. Quartos')
plt.xlabel('N mero de Quartos')
plt.ylabel('Pre o (R$)')
plt.ticklabel_format(style='plain', axis='y')

plt.tight_layout()
plt.savefig('preco_vs_caracteristicas.png')
plt.show()
```

```
print("Gr ficos de Pre o vs. Caracter sticas salvos como 'preco_vs_caracteristicas.
    png'.")
```

Listing 10: Preço vs. Características

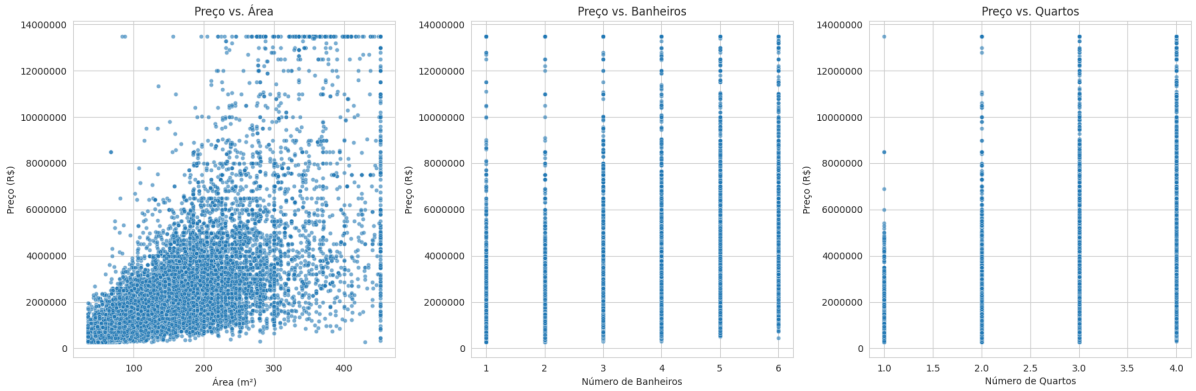

Figura 5: Preço vs. Características do Imóvel

4.3 A Influência da Localização (Bairro) no Preço

A localização é um dos fatores mais críticos no mercado imobiliário. Esta seção investiga como o preço por metro quadrado varia significativamente entre os diferentes bairros de São Paulo.

```
# Top 10 Bairros com maior e menor pre o m dio por m
# Calcular o pre o m dio por metro quadrado para cada bairro
price_per_sqm_by_bairro = df.groupby('Bairro')['Price_per_sqm'].mean().sort_values(
    ascending=False)

print("\n### Top 10 Bairros com maior pre o m dio por m :")
print(price_per_sqm_by_bairro.head(10))

print("\n### Top 10 Bairros com menor pre o m dio por m :")
print(price_per_sqm_by_bairro.tail(10))

# Visualizar os 10 bairros mais caros e os 10 mais baratos em termos de pre o por m
plt.figure(figsize=(15, 7))

% Top 10 Bairros mais caros
plt.subplot(1, 2, 1)
sns.barplot(x=price_per_sqm_by_bairro.head(10).values, y=price_per_sqm_by_bairro.head
    (10).index,
            palette='viridis', hue=price_per_sqm_by_bairro.head(10).index, legend=False)
plt.title('Top 10 Bairros Mais Caros (Pre o M dio por m )')
plt.xlabel('Pre o M dio por m (R$)')
plt.ylabel('Bairro')
plt.ticklabel_format(style='plain', axis='x')

% Top 10 Bairros mais baratos
plt.subplot(1, 2, 2)
sns.barplot(x=price_per_sqm_by_bairro.tail(10).values, y=price_per_sqm_by_bairro.tail
    (10).index,
            palette='magma', hue=price_per_sqm_by_bairro.tail(10).index, legend=False)
plt.title('Top 10 Bairros Mais Baratos (Pre o M dio por m )')
plt.xlabel('Pre o M dio por m (R$)')
plt.ylabel('Bairro')
plt.ticklabel_format(style='plain', axis='x')

plt.tight_layout()
plt.savefig('top_bairros_preco_m2.png')
plt.show()
print("Gr ficos dos Top Bairros por Pre o por m salvos como 'top_bairros_preco_m2.
    png'.")
```

Listing 11: Análise de Preço por m^2 por Bairro

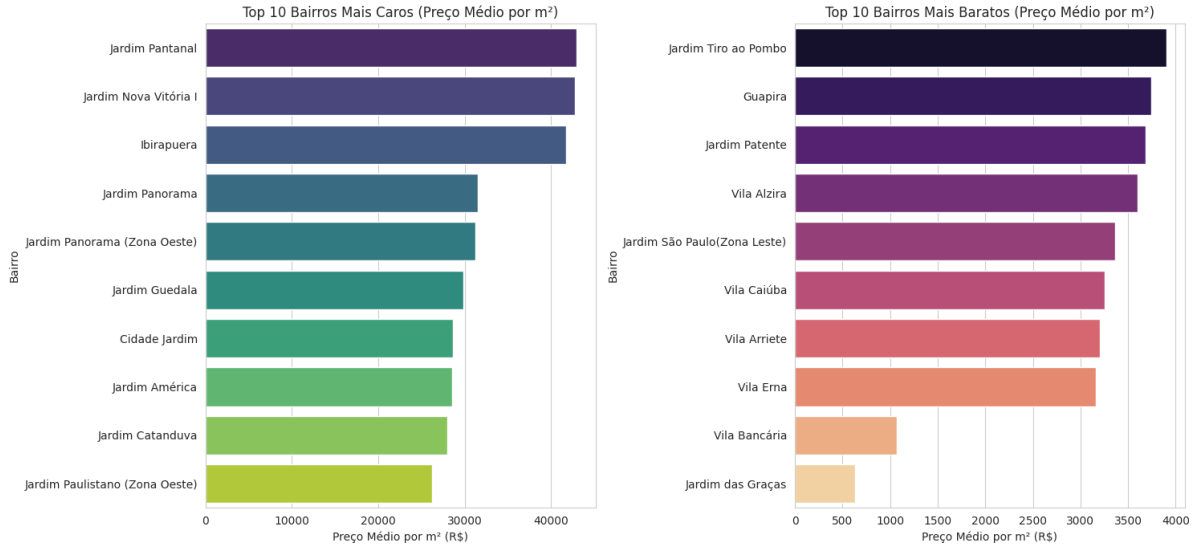

Figura 6: Top 10 Bairros Mais Caros e Mais Baratos (Preço Médio por m²)

```
# Box plot da distribui  o do Pre o por m  nos Top 20 Bairros (mais detalhado)
# Selecionar os 20 bairros com mais apartamentos no dataset para o box plot
top_20_bairros = df['Bairro'].value_counts().head(20).index
df_top_bairros = df[df['Bairro'].isin(top_20_bairros)]

plt.figure(figsize=(14, 8))
sns.boxplot(x='Price_per_sqm', y='Bairro', data=df_top_bairros.sort_values('
    Price_per_sqm', ascending=False),
            palette='coolwarm', hue='Bairro', legend=False)
plt.title('Distribui  o do Pre o por m  nos Top 20 Bairros de S o Paulo')
plt.xlabel('Pre o por m  (R$)')
plt.ylabel('Bairro')
plt.ticklabel_format(style='plain', axis='x')
plt.grid(axis='x', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.savefig('boxplot_preco_m2_top_bairros.png')
plt.show()
print("Box plot do Pre o por m  nos Top 20 Bairros salvo como '
    boxplot_preco_m2_top_bairros.png'.")
```

Listing 12: Box Plot de Preço por m² nos Top Bairros

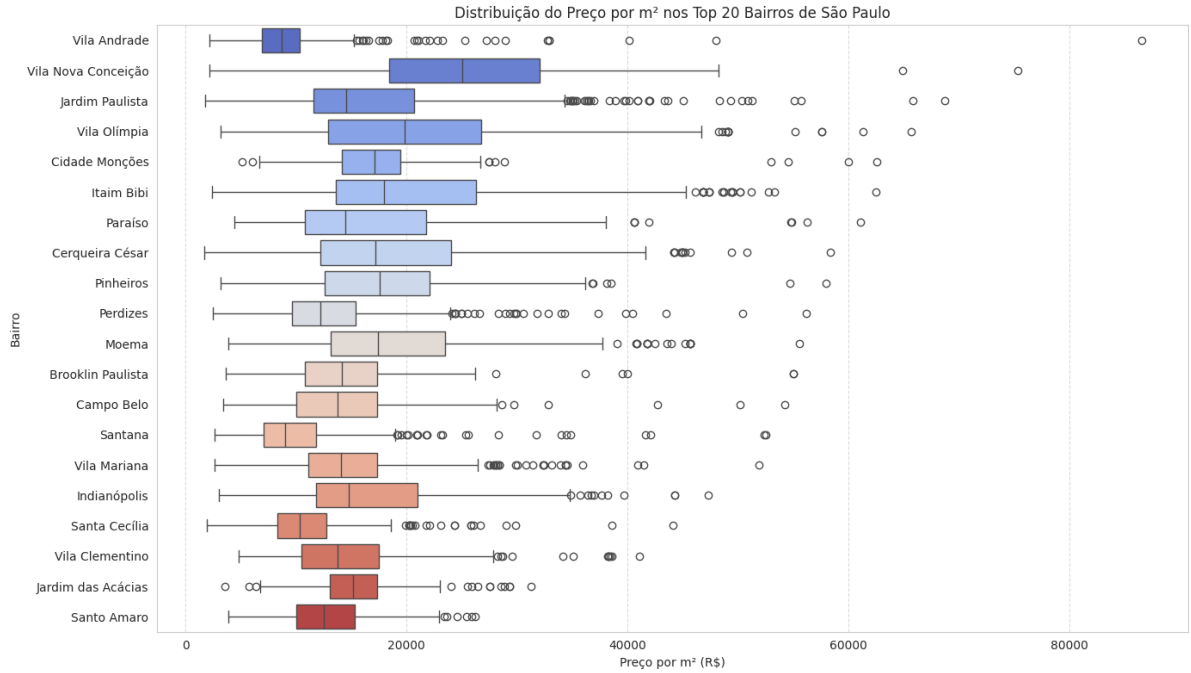

Figura 7: Distribuição do Preço por m² nos Top 20 Bairros de São Paulo

5 Conclusões Finais

Este estudo demonstrou a aplicação de um ciclo completo de ciência de dados para analisar a influência dos dados no setor imobiliário de São Paulo. As principais conclusões são:

- **A Importância Crítica da Preparação de Dados:** A fase de limpeza e organização (conversão de datas, extração de texto e, principalmente, tratamento de outliers) foi fundamental. As visualizações "antes"e "depois"do tratamento de outliers evidenciaram como essa etapa é crucial para obter análises precisas e evitar distorções, transformando dados brutos em um dataset confiável.

- **Influência Determinante das Características do Imóvel no Preço:** A análise de correlação e os gráficos de dispersão confirmaram que a **área do imóvel** é o fator mais impactante no preço de venda. O número de **banheiros**, **quartos** e **vagas de estacionamento** também demonstraram forte correlação positiva, refletindo que maior espaço e mais comodidades elevam o valor da propriedade.

- **Localização como Fator Primordial de Precificação:** A análise do preço por metro quadrado revelou uma **disparidade extrema** entre os bairros de São Paulo. Bairros de alto padrão como Jardim Pantanal e Ibirapuera mostram um valor médio por m² que pode ser dezenas de vezes maior do que em regiões mais acessíveis como Jardim das Graças e Vila Bancária. O Box Plot detalhou essa variação, mostrando não apenas as diferenças de medianas entre os bairros, mas também a vasta **variabilidade e assimetria na distribuição de preços** dentro de um mesmo bairro, sublinhando a complexidade e a micro-segmentação do mercado imobiliário da cidade.

- **Outros Fatores:** Curiosamente, a coluna `age_of_listing_days` (idade do anúncio) mostrou uma correlação linear muito baixa com o preço, sugerindo que, em um primeiro momento, o tempo de permanência de um anúncio online não é um fator linearmente determinante para a variação do valor final do imóvel.

- **A Essência dos Dados no Setor Imobiliário:** Em suma, este trabalho reitera que os dados são um ativo indispensável no setor imobiliário. Eles permitem uma compreensão aprofundada das dinâmicas de mercado, embasando avaliações precisas de propriedades, identificação de tendências de valorização/desvalorização, otimização de estratégias de venda e aluguel, e uma tomada de decisão de investimento muito mais informada e estratégica. A capacidade de extrair inteligência a partir desses dados confere uma vantagem competitiva inestimável.

Este estudo demonstra como a aplicação sistemática de um ciclo de ciência de dados pode transformar informações brutas em insights valiosos para o mercado.