

BOOTCAMP DATA ANALYTICS

Squads: Mae C. Jemison & Marie Curie


Desafio Final: Reclamações do Consumidor





SQUAD: MAE C. JAMISON





Ana Caroline de Souza




Ana Luisa Silveira



Eduarda Martins



Glauce Gomes Galvino



Raquel Mitie Harano



SQUAD: MARIE CURIE



Carla Marra



Edwiges Bárbara Oliveira



Grazielle Henrique



Juliana Portela



Leli Araújo



Sandra Correia de Resende

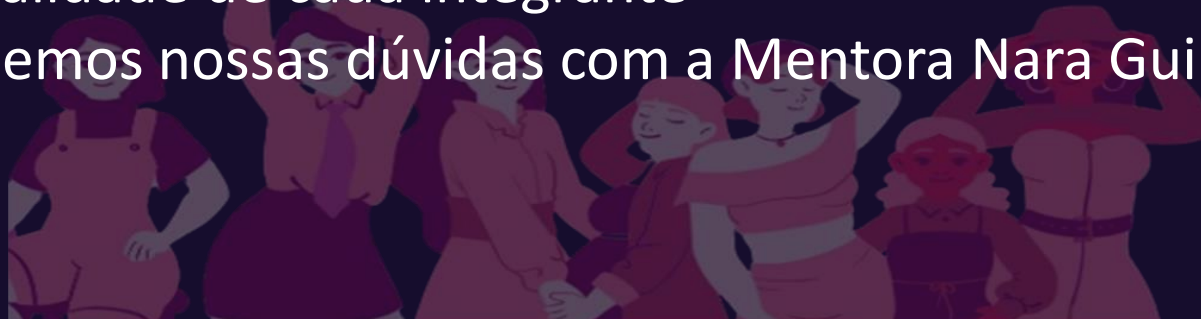




ORGANIZAÇÃO DAS SQUADS

Como foi realizado o projeto:

- O projeto foi realizado no Google Colaboratory;
- As etapas do desafio foram divididas e executadas conforme avançávamos o desenvolvimento;
- Nos comunicamos pelo Discord para discutir os pontos do nosso projeto e realizamos reuniões também;
- Houve colaboração entre as equipes, respeitando a individualidade de cada integrante
- Esclarecemos nossas dúvidas com a Mentora Nara Guimarães.





CONTEXTO

- Importância da análise de reclamações
- Impacto na satisfação do consumidor e na imagem da empresa

OBJETIVO

- Entender o comportamento das reclamações do consumidor
- Identificar padrões e variáveis relevantes
- Construir um modelo preditivo eficiente





OS DADOS

Os dados retirados do Kaggle são do Procon, que monitora e resolve reclamações de consumidores entre os anos de 2017 e 2021.

Nosso objetivo é analisar esses dados e criar um modelo preditivo para estimar o tempo médio de resolução das reclamações.



Data columns (total 23 columns):

#	Column	Non-Null Count	Dtype
---	-----	-----	----
0	AnoCalendario	115921 non-null	int64
1	DataArquivamento	115896 non-null	object
2	DataAbertura	115909 non-null	object
3	CodigoRegiao	115921 non-null	int64
4	Regiao	115921 non-null	object
5	UF	115921 non-null	object
6	strRazaoSocial	115918 non-null	object
7	strNomeFantasia	95800 non-null	object
8	Tipo	115921 non-null	int64
9	NumeroCNPJ	109662 non-null	float64
10	RadicalCNPJ	109425 non-null	float64
11	RazaoSocialRFB	100313 non-null	object
12	NomeFantasiaRFB	49180 non-null	object
13	CNAEPrincipal	100313 non-null	float64
14	DescCNAEPrincipal	99749 non-null	object
15	Atendida	115921 non-null	object
16	CodigoAssunto	115907 non-null	float64
17	DescricaoAssunto	115907 non-null	object
18	CodigoProblema	50481 non-null	float64
19	DescricaoProblema	50481 non-null	object
20	SexoConsumidor	115884 non-null	object
21	FaixaEtariaConsumidor	115921 non-null	object
22	CEPConsumidor	103611 non-null	float64





OS DADOS

Adicionamos ao projeto dados do Censo de 2022 para obtermos uma análise e resultados mais precisos.



<https://sidra.ibge.gov.br/tabela/4709>

	UF	Populacao	Regiao	Reclamacoes	MediaPonderadaReclamacoes
0	RO	1581196	Norte	6195	51.887503
1	PA	8120131	Norte	987	42.453739
2	TO	1511460	Norte	567	4.539578
3	MA	6776699	Nordeste	216	7.753677
4	PI	3271199	Nordeste	2153	37.306693
5	CE	8794957	Nordeste	1982	92.336440
6	RN	3302729	Nordeste	8128	142.197641
7	PB	3974687	Nordeste	1486	31.286537
8	PE	9058931	Nordeste	211	10.125003
9	BA	14141626	Nordeste	25	1.872730
10	MG	20539989	Sudeste	8550	930.255140
11	ES	3833712	Sudeste	2781	56.474999
12	RJ	16055174	Sudeste	3612	307.183946
13	SP	44411238	Sudeste	40998	9644.757396
14	PR	11444380	Sul	875	53.043977
15	SC	7610361	Sul	4920	198.338086
16	RS	10882965	Sul	182	10.491906
17	MS	2757013	Centro-oeste	6972	101.819732
18	MT	3658649	Centro-oeste	7822	151.591313
19	GO	7056495	Centro-oeste	17259	645.119878

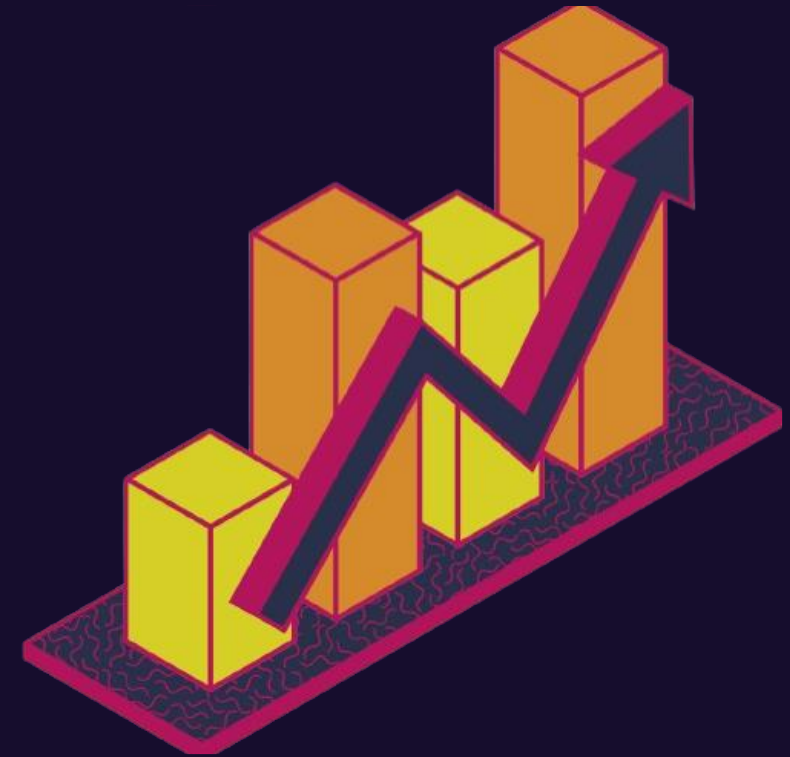




PERGUNTAS DO TIME DE NEGÓCIOS

Análise dos dados:

- Existe sazonalidade na abertura de reclamações?
- Qual o tempo médio de uma reclamação ativa?
- O número de reclamações varia por região e estado?
- Quais empresas receberam mais reclamações?





PERGUNTAS DO TIME DE NEGÓCIOS

Modelagem dos dados:

- Quais variáveis estão correlacionadas com o tempo de uma reclamação ativa?
- Construa variáveis correlacionadas.
- Analise a correlação das variáveis.
- Construa um modelo de regressão linear.





BIBLIOTECAS UTILIZADAS

Pandas



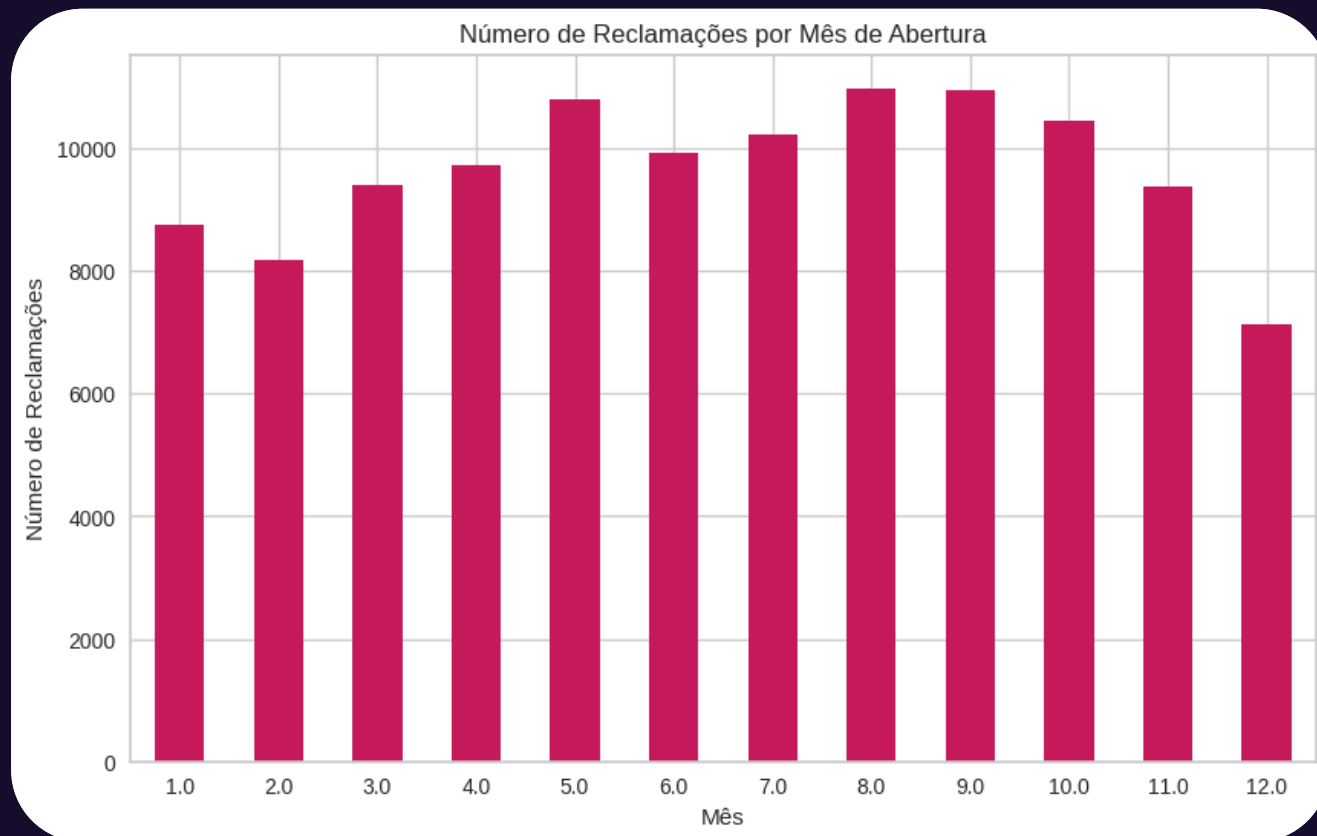
matplotlib





SAZONALIDADE NAS RECLAMAÇÕES

Existe sazonalidade na abertura de reclamações?



Análise de forma mensal para todos os anos presentes, a fim de verificar quais meses possuem maior concentração de abertura de reclamações.

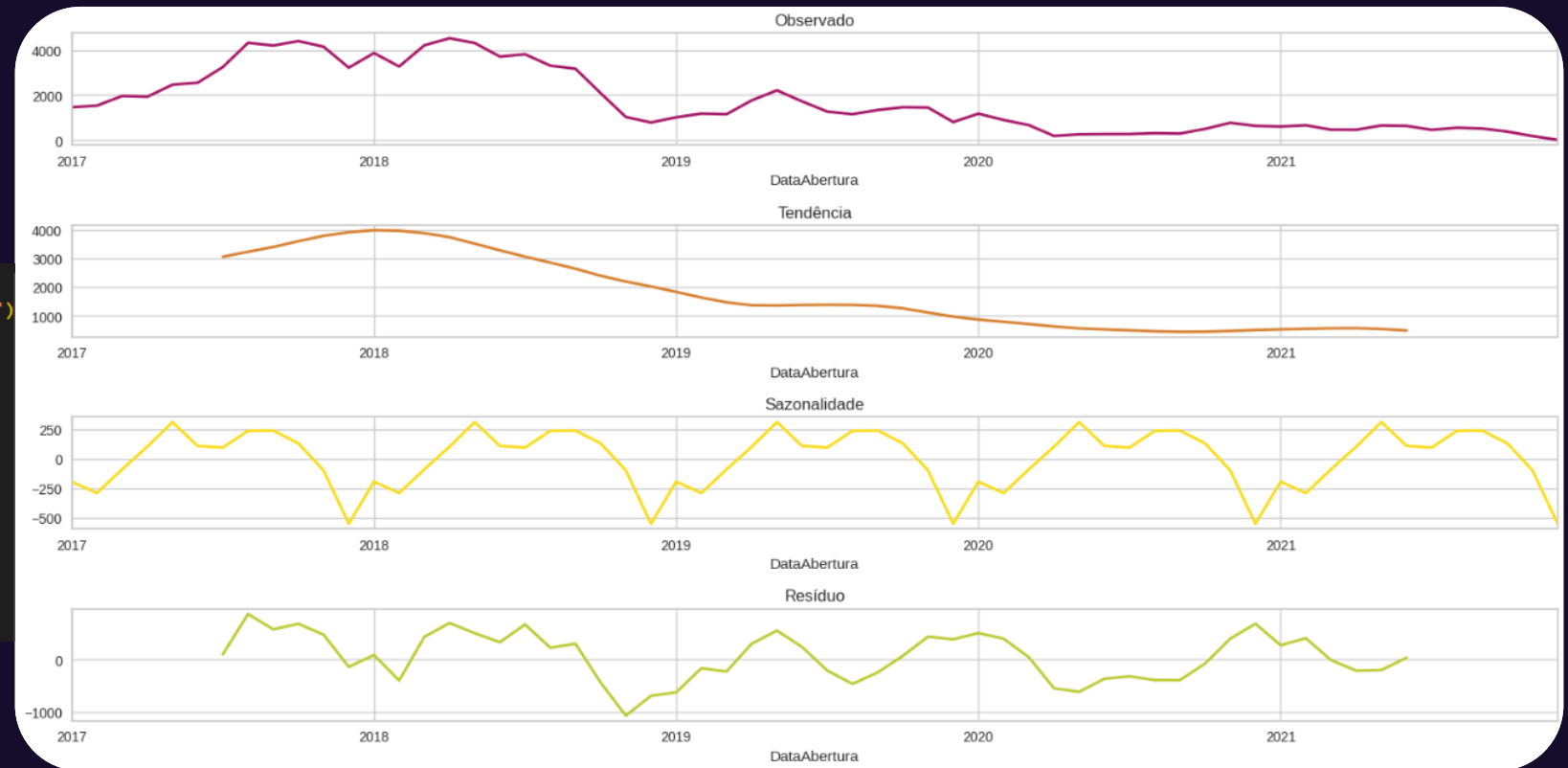




SAZONALIDADE NAS RECLAMAÇÕES

```
# Realizando a decomposição sazonal
decomposition = seasonal_decompose(monthly_counts['Contagem'], model='additive')

# Plotando os resultados da decomposição
fig, (ax1, ax2, ax3, ax4) = plt.subplots(4, 1, figsize=(16, 8))
decomposition.observed.plot(ax=ax1, color= '#a31063', linewidth=2)
ax1.set_title('Observado')
decomposition.trend.plot(ax=ax2, color= '#d77e2b', linewidth=2)
ax2.set_title('Tendência')
decomposition.seasonal.plot(ax=ax3, color= '#f9dd15', linewidth=2)
ax3.set_title('Sazonalidade')
decomposition.resid.plot(ax=ax4, color= '#bbcb30', linewidth=2)
ax4.set_title('Resíduo')
plt.tight_layout()
plt.show()
```



Observamos a presença de sazonalidade nas reclamações pelo período de 2017 a 2021.





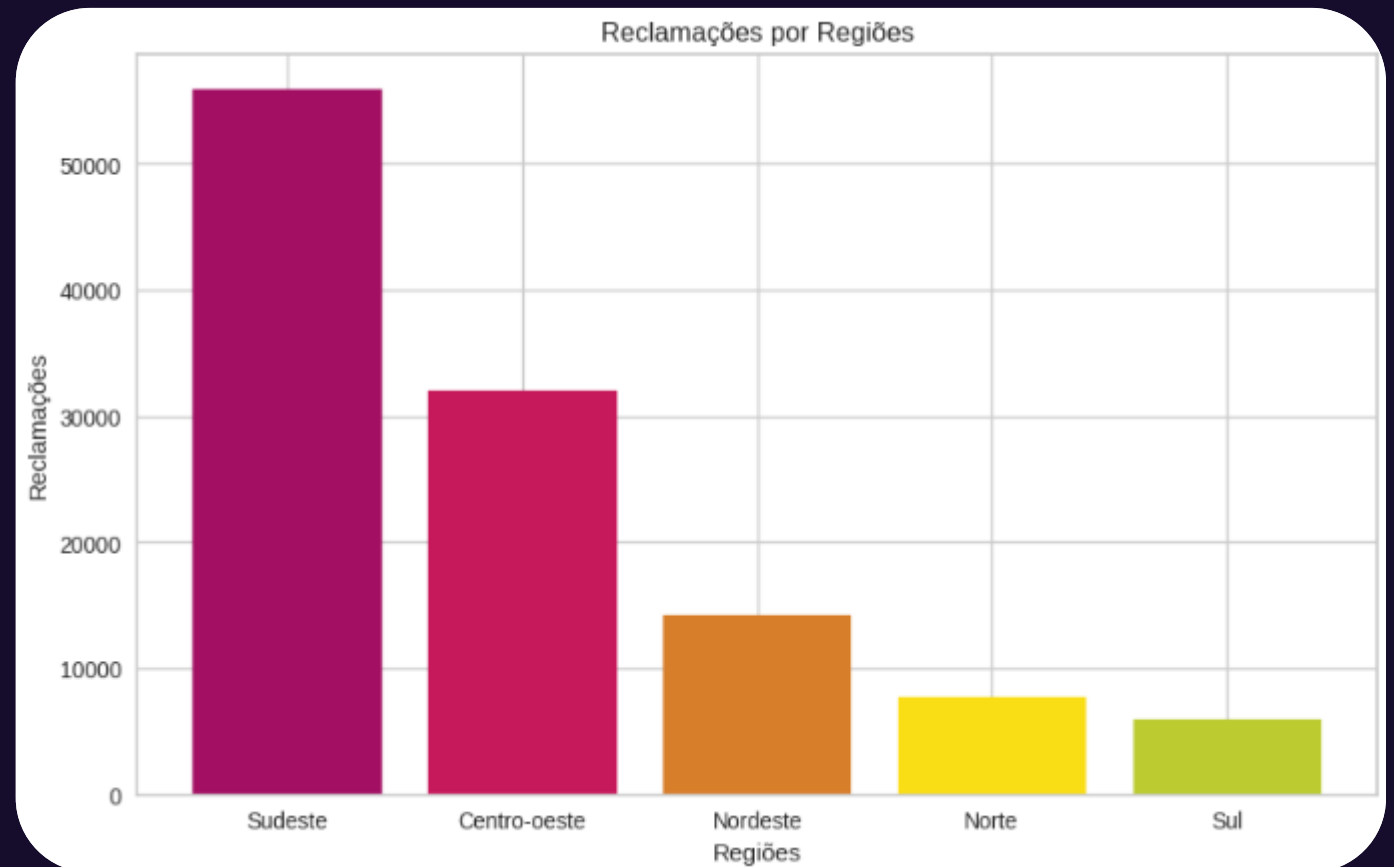
VARIAÇÃO DE RECLAMAÇÕES POR REGIÃO E ESTADO

O número de reclamações varia por região e estado?

```
# Plot do gráfico de barras por região, quantidade de reclamações
plt.figure(figsize=(10, 6))

region_custom_colors = {
    'Sudeste': '#a31063',
    'Centro-oeste': '#c6195b',
    'Nordeste': '#d77e2b',
    'Norte': '#f9dd15',
    'Sul': '#bbcb30',
}

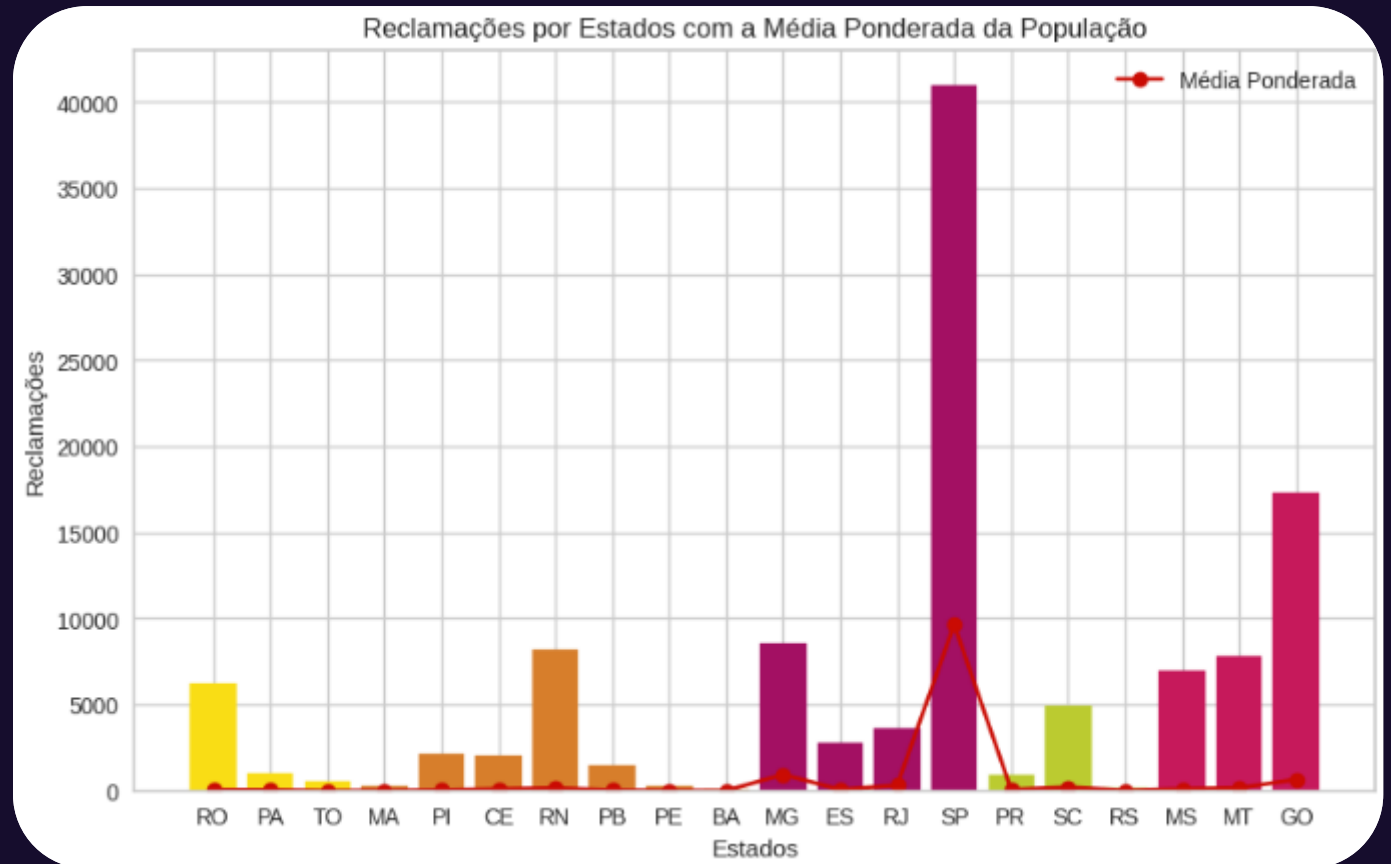
axs = plt.bar(df_reclamacoes_regiao['Regiao'],
              df_reclamacoes_regiao['Reclamacoes'])
plt.xlabel('Regiões')
plt.ylabel('Reclamações')
plt.title('Reclamações por Regiões')
for i, bar in enumerate(axs):
    bar.set_color(region_custom_colors[df_reclamacoes_regiao['Regiao'][i]])
plt.show()
```





VARIAÇÃO DE RECLAMAÇÕES POR REGIÃO E ESTADO

```
#Plotagem de gráfico de barras para verificar as reclamações por estados em  
#comparação a média ponderada da população obtida pelo censo 2022  
  
plt.figure(figsize=(10, 6))  
plt.bar(df_reclamacoes_estado_censo['UF'],  
        df_reclamacoes_estado_censo['Reclamacoes'])  
axs = plt.bar(df_reclamacoes_estado_censo['UF'],  
              df_reclamacoes_estado_censo['Reclamacoes'])  
for i, bar in enumerate(axs):  
    bar.set_color(region_custom_colors[df_reclamacoes_estado_censo['Regiao']][i])  
plt.plot(df_reclamacoes_estado_censo['UF'],  
         df_reclamacoes_estado_censo['MediaPonderadaReclamacoes'],  
         marker='o', color='r', label = 'Média Ponderada')  
plt.xlabel('Estados')  
plt.ylabel('Reclamações')  
plt.title('Reclamações por Estados com a Média Ponderada da População')  
plt.legend()  
plt.show()
```



Estados como São Paulo e Goiás tiveram um número significativamente maior de reclamações, mesmo quando ajustado pela população.





EMPRESAS COM MAIS RECLAMAÇÕES

Quais empresas receberam mais reclamações?

```
# Agrupamento por Empresas (Razão Social), regiões, com a quantidade de reclamações e ordenado por reclamações
df_reclamacoes_empresas_regiao = combined_df.groupby(['strRazaoSocial', 'Regiao'])['strRazaoSocial'].count().sort_values(ascending=False)
df_reclamacoes_empresas_regiao

# Conversão para DataFrame para facilitar a manipulação
df_reclamacoes_empresas_regiao = df_reclamacoes_empresas_regiao.reset_index(name='counts')

# Seleção das 5 empresas com mais reclamações por região
df_reclamacoes_empresas_regiao_top5 = df_reclamacoes_empresas_regiao.groupby('Regiao').apply(lambda x: x.nlargest(5, 'counts')).reset_index(drop=True)

# Definindo a paleta de cores personalizada com base no dicionário
region_custom_colors = {
    'Sudeste': '#a31063',
    'Centro-oeste': '#c6195b',
    'Nordeste': '#d77e2b',
    'Norte': '#f9dd15',
    'Sul': '#bbcb30',
}

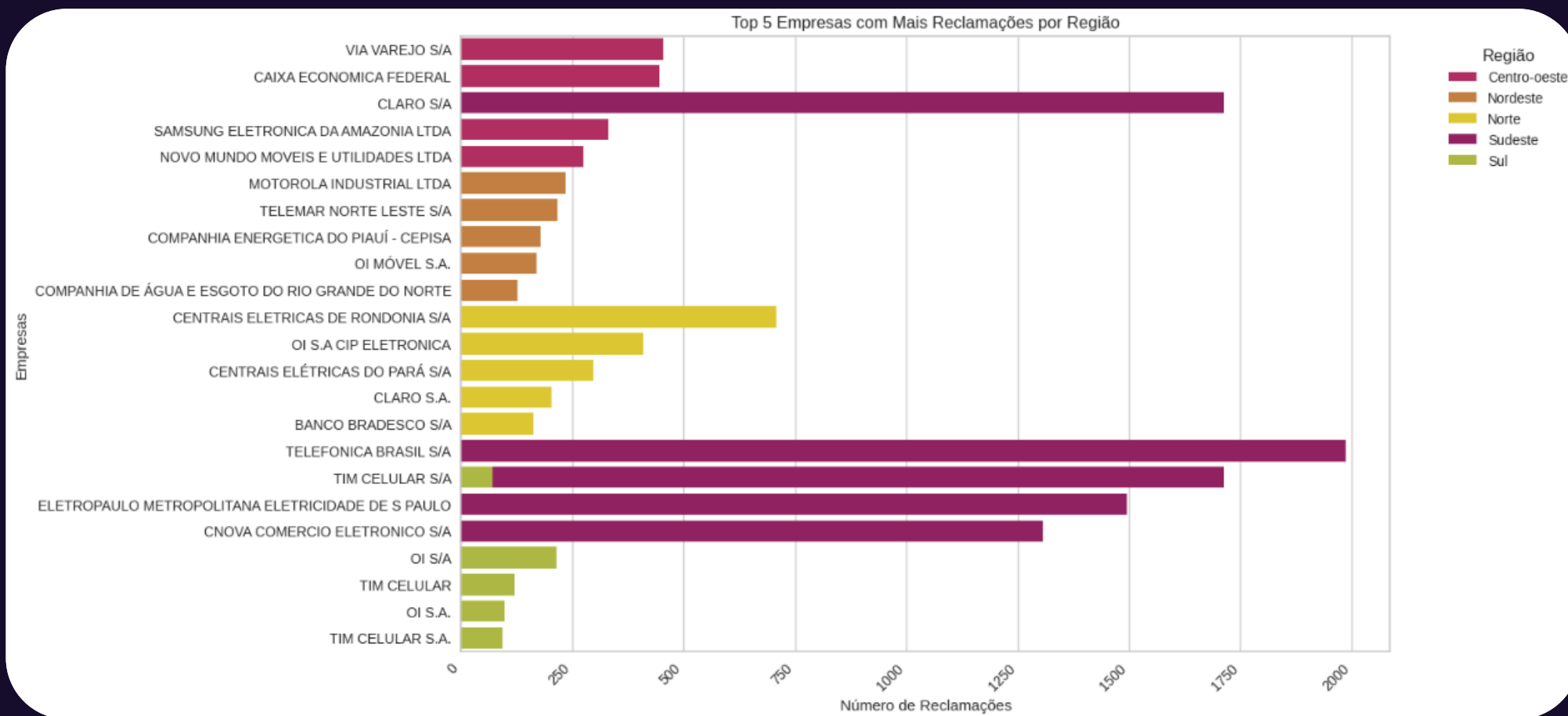
# Criando uma paleta de cores para Seaborn a partir do dicionário
custom_palette = [region_custom_colors[regiao] for regiao in df_reclamacoes_empresas_regiao_top5['Regiao'].unique()]

# Gráfico de barras das empresas com mais reclamações por região
plt.figure(figsize=(12, 8))
sns.barplot(data=df_reclamacoes_empresas_regiao_top5, x='counts', y='strRazaoSocial', hue='Regiao', dodge=False, palette=custom_palette)
plt.title('Top 5 Empresas com Mais Reclamações por Região')
plt.xlabel('Número de Reclamações')
plt.ylabel('Empresas')
plt.legend(title='Região', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.xticks(rotation=45, ha='right')
plt.show()
```





EMPRESAS COM MAIS RECLAMAÇÕES



Empresas como Telefônica e Claro lideraram o número de reclamações, especialmente na região sudeste.





TEMPO MÉDIO DE UMA RECLAMAÇÃO ATIVA

Qual o tempo médio de uma reclamação ativa?

```
# Calculando os valores de outliers do tempo de reclamação ativa
# Calcular Q1 (primeiro quartil) e Q3 (terceiro quartil)
Q1 = combined_df['DiferencaTempoNum'].quantile(0.25)
Q3 = combined_df['DiferencaTempoNum'].quantile(0.75)

# Calcular o intervalo interquartil (IQR)
IQR = Q3 - Q1

# Calcular o limite superior
limite_superior = Q3 + 1.5 * IQR

# Filtra os outliers do dataset
df_sem_outlier = combined_df[(combined_df['DiferencaTempoNum'] < limite_superior)]
media_sem_outlier = df_sem_outlier['DiferencaTempoNum'].mean().round(2)
print('A média de tempo de reclamação ativa sem outliers é de {}'.format(media_sem_outlier)) # Média sem outliers
print('A mediana de tempo de reclamação ativa com outliers é de {}'.format(combined_df['DiferencaTempoNum'].median()))

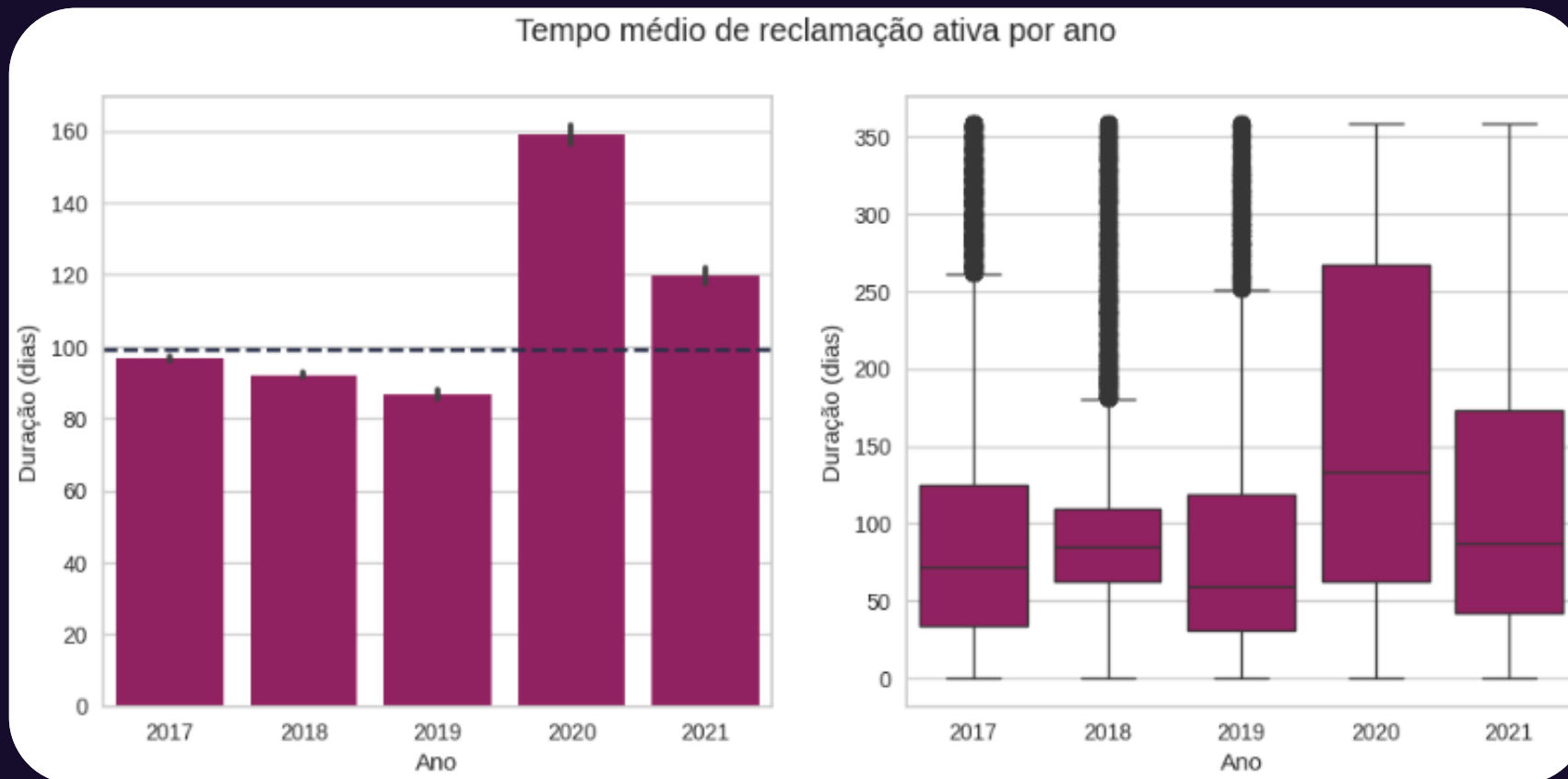
A média de tempo de reclamação ativa sem outliers é de 99.06.
A mediana de tempo de reclamação ativa com outliers é de 91.0.
```

Observa-se que a proximidade da mediana com a média dos dados tratados, sendo a primeira contendo os outliers e a segunda não. Assim, a nível de manipulação, é preferível utilizar a mediana se considerar o dataset sem tratamento dos outliers.





TEMPO MÉDIO DE UMA RECLAMAÇÃO ATIVA



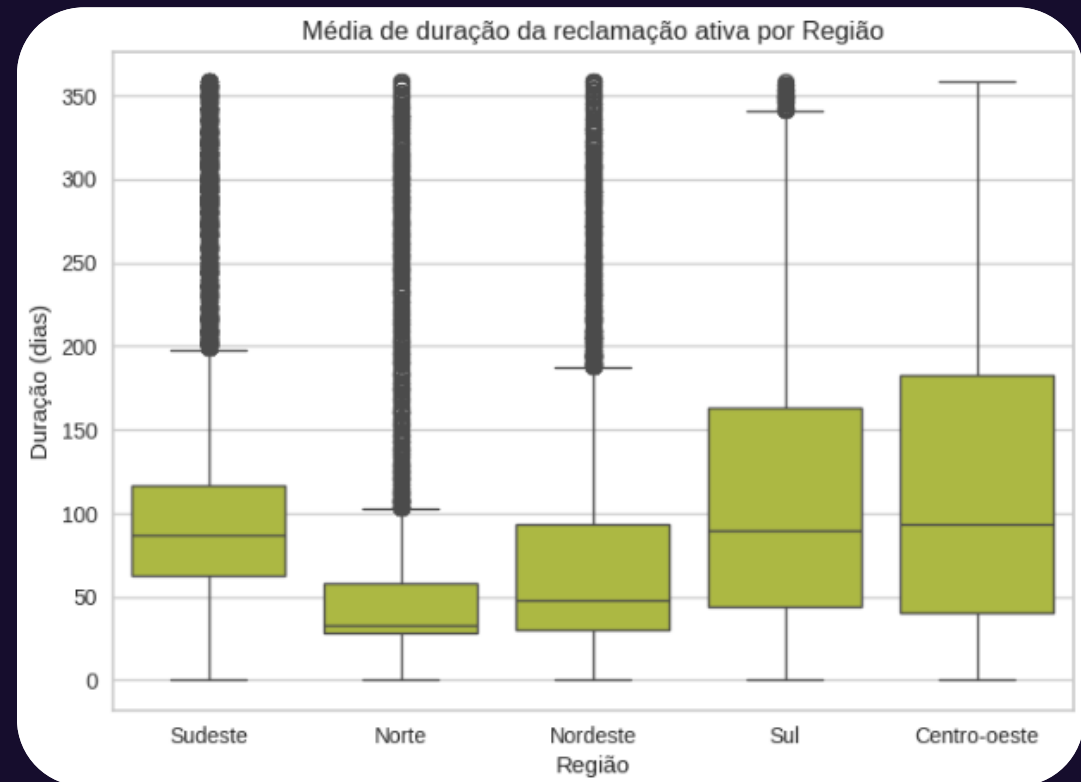


TEMPO MÉDIO DE UMA RECLAMAÇÃO ATIVA

```
sns.boxplot(df_sem_outlier, x = 'Regiao', y = 'DiferencaTempoNum', color= '#bbcb30')

plt.title('Média de duração da reclamação ativa por Região')
plt.xlabel('Região')
plt.ylabel('Duração (dias)')
plt.xticks(rotation = 0)
plt.show();
```

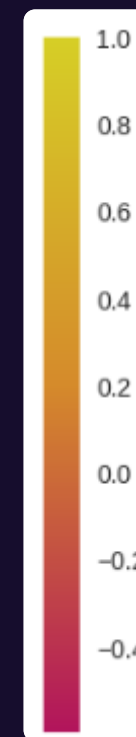
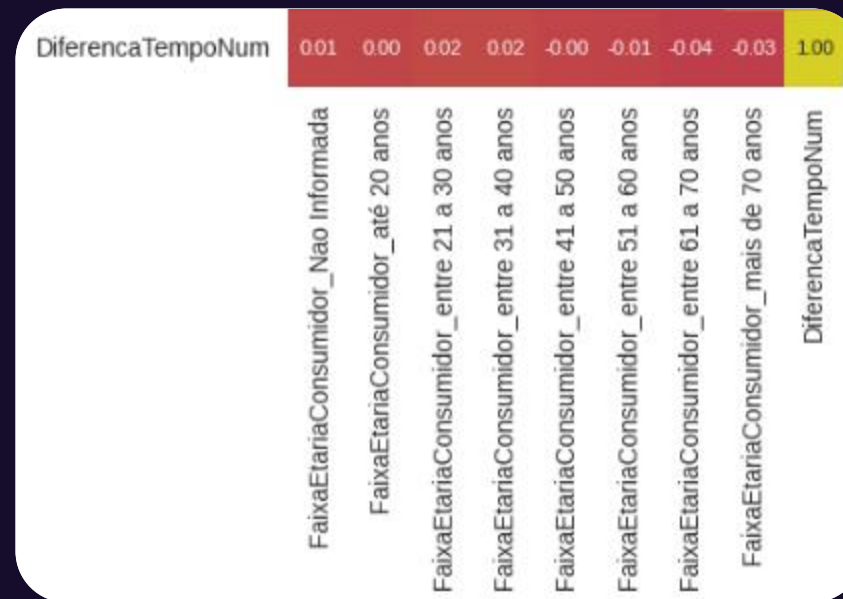
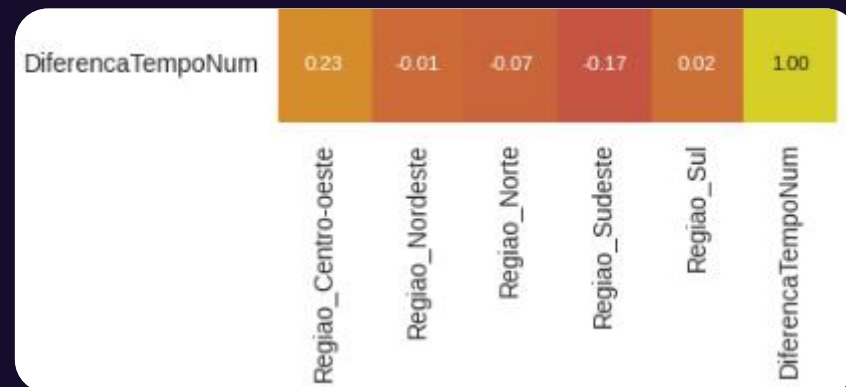
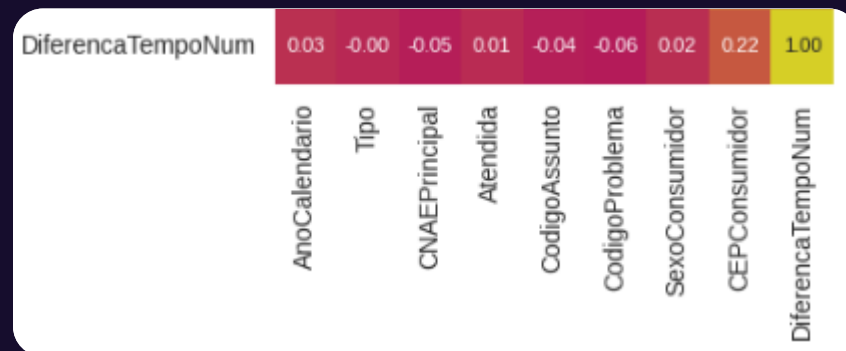
Visualização da dispersão dos dados do tempo de reclamação ativa de acordo com suas respectivas regiões.





VARIÁVEIS CORRELACIONADAS COM O TEMPO DE RECLAMAÇÃO

Quais variáveis estão correlacionadas com o tempo de uma reclamação ativa?





CONSTRUÇÃO DE VARIÁVEIS E ANÁLISE DE CORRELAÇÃO

Construa variáveis correlacionadas.

```
#A coluna ATENDIDA retornava S e N, portanto gerará a dummy.
def coluna_dummy(coluna, df_list):
    for dataframe in df_list:
        dataframe[coluna] = dataframe[coluna].map({'S': 1, 'N': 0})
        print(f"Coluna dummy criada com sucesso nos anos de {dataframe['AnoCalendario'].unique()}")

# Aplica a função
coluna_dummy('Atendida', [df_tratado])
```

Coluna dummy criada com sucesso nos anos de [2017 2018 2019 2020 2021]

```
# Alterando a coluna SexoConsumidor para valor binário 0/1
df_tratado['SexoConsumidor'] = df_tratado['SexoConsumidor'].replace({'M': 1, 'F': 0}).astype(int)

# Aplicando One-hot-Encoding para a coluna de faixa etaria
df_tratado = pd.get_dummies(df_tratado, columns = ['FaixaEtariaConsumidor'], dtype = float)

# Aplicando One-hot-Encoding para a coluna de região
df_tratado = pd.get_dummies(df_tratado, columns = ['Regiao'], dtype = float)

# Aplicando One-hot-Encoding para a coluna estado
df_tratado = pd.get_dummies(df_tratado, columns = ['UF'], dtype = float)
```





CONSTRUÇÃO DE VARIÁVEIS E ANÁLISE DE CORRELAÇÃO

Data columns (total 43 columns):

#	Column	Non-Null Count	Dtype
0	AnoCalendario	114763 non-null	int64
1	Tipo	114763 non-null	int64
2	CNAEPrincipal	99357 non-null	float64
3	Atendida	114763 non-null	int64
4	CodigoAssunto	114749 non-null	float64
5	CodigoProblema	49738 non-null	float64
6	SexoConsumidor	114763 non-null	int64
7	CEPConsumidor	102488 non-null	float64
8	DiferencaTempoNum	114750 non-null	float64
9	MesAbertura	114763 non-null	float64
10	FaixaEtariaConsumidor_Nao Informada	114763 non-null	float64
11	FaixaEtariaConsumidor_até 20 anos	114763 non-null	float64
12	FaixaEtariaConsumidor_entre 21 a 30 anos	114763 non-null	float64
13	FaixaEtariaConsumidor_entre 31 a 40 anos	114763 non-null	float64
14	FaixaEtariaConsumidor_entre 41 a 50 anos	114763 non-null	float64
15	FaixaEtariaConsumidor_entre 51 a 60 anos	114763 non-null	float64
16	FaixaEtariaConsumidor_entre 61 a 70 anos	114763 non-null	float64
17	FaixaEtariaConsumidor_mais de 70 anos	114763 non-null	float64
18	Regiao_Centro-oeste	114763 non-null	float64
19	Regiao_Nordeste	114763 non-null	float64

20	Regiao_Norte	114763 non-null	float64
21	Regiao_Sudeste	114763 non-null	float64
22	Regiao_Sul	114763 non-null	float64
23	UF_BA	114763 non-null	float64
24	UF_CE	114763 non-null	float64
25	UF_ES	114763 non-null	float64
26	UF_GO	114763 non-null	float64
27	UF_MA	114763 non-null	float64
28	UF_MG	114763 non-null	float64
29	UF_MS	114763 non-null	float64
30	UF_MT	114763 non-null	float64
31	UF_PA	114763 non-null	float64
32	UF_PB	114763 non-null	float64
33	UF_PE	114763 non-null	float64
34	UF_PI	114763 non-null	float64
35	UF_PR	114763 non-null	float64
36	UF_RJ	114763 non-null	float64
37	UF_RN	114763 non-null	float64
38	UF_RO	114763 non-null	float64
39	UF_RS	114763 non-null	float64
40	UF_SC	114763 non-null	float64
41	UF_SP	114763 non-null	float64
42	UF_TO	114763 non-null	float64

As variáveis construídas, como dummies regiões, são relevantes para o modelo.





MODELO DE REGRESSÃO LINEAR

Construa um modelo de regressão linear.

```
# Salvamos os dados dos resíduos na variável residuos
residuos = model.resid

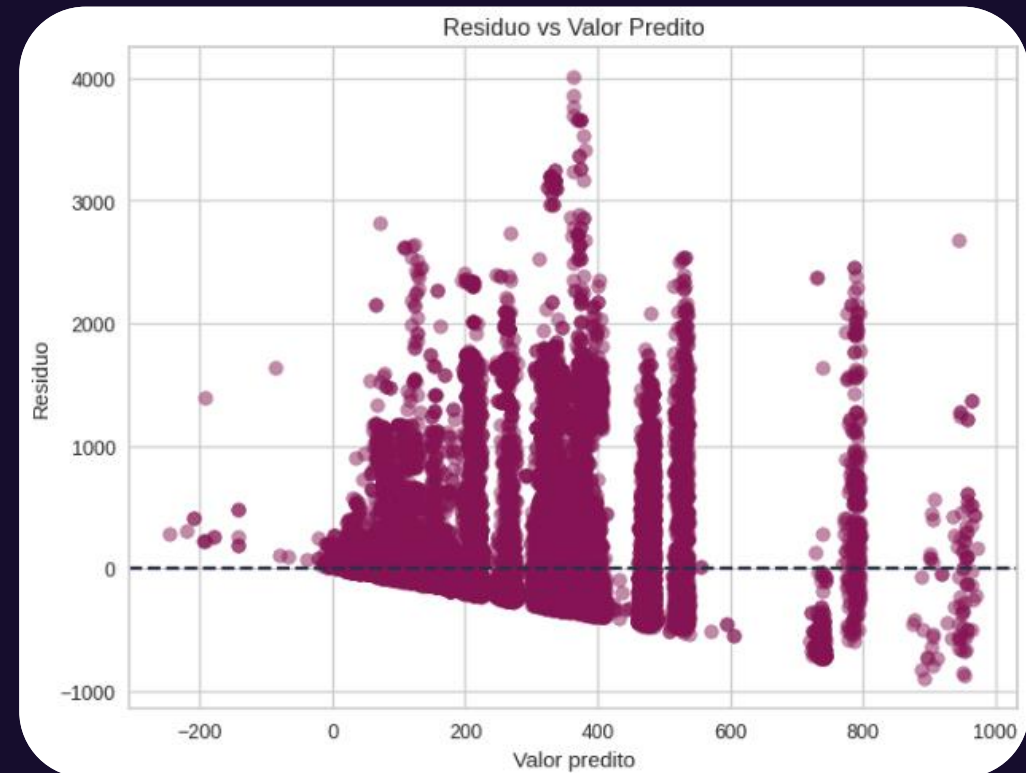
# Calculamos os valores preditos
predicted_values = model.fittedvalues

#Verificação através de scatterplot
plt.figure(figsize=(8, 6))
plt.scatter(predicted_values, residuos, color='#861454', alpha=0.5)

# adiciona linha
plt.axhline(y=0, color='#293049', linestyle='--')

# Títulos
plt.title('Residuo vs Valor Predito')
plt.xlabel('Valor predito')
plt.ylabel('Residuo')

plt.show()
```



O modelo de regressão linear não se ajusta bem para previsão do tempo de reclamação ativa levando em consideração os dados disponíveis.





MODELO DE REGRESSÃO LINEAR

```
#Exibir o sumário detalhado do modelo de regressão linear múltipla
import statsmodels.api as sm

#adicionando um constante de intercepto ao modelo
X = sm.add_constant(X)

#criar o modelo de regressão linear
model = sm.OLS(y, X).fit()

#imprimir o sumário do modelo
print(model.summary())
```

OLS Regression Results			
=====			
Dep. Variable:	DiferencaTempoNum	R-squared:	0.161
Model:	OLS	Adj. R-squared:	0.161
Method:	Least Squares	F-statistic:	852.7
Date:	Fri, 26 Jul 2024	Prob (F-statistic):	0.00
Time:	20:48:28	Log-Likelihood:	-7.3244e+05
No. Observations:	102461	AIC:	1.465e+06
Df Residuals:	102437	BIC:	1.465e+06
Df Model:	23		
Covariance Type:	nonrobust		





MODELO DE REGRESSÃO LINEAR

Rodamos um modelo de RandomForestRegressor:

```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor

X = df_lr.copy()
X.drop(columns = ['DiferencaTempoNum'], inplace = True)
y = df_lr['DiferencaTempoNum']

#Dividir conjuntos de treinamento e teste
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
model = RandomForestRegressor()
model.fit(X_train, y_train)

# Realizando previsões
y_pred = model.predict(X_test)
```

```
# Métricas de avaliação do modelo
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
import numpy as np
```

```
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
mae = mean_absolute_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print(f'MSE: {mse:.2f}')
print(f'RMSE: {rmse:.2f}')
print(f'MAE: {mae:.2f}')
print(f'R²: {r2:.2f}')
```

```
MSE: 31071.33
RMSE: 176.27
MAE: 71.22
R²: 0.72
```

O modelo RandomForestRegressor demonstrou melhor ajuste para previsão do tempo de reclamação ativa levando em consideração o R^2 em comparação a Regressão Linear Múltipla.



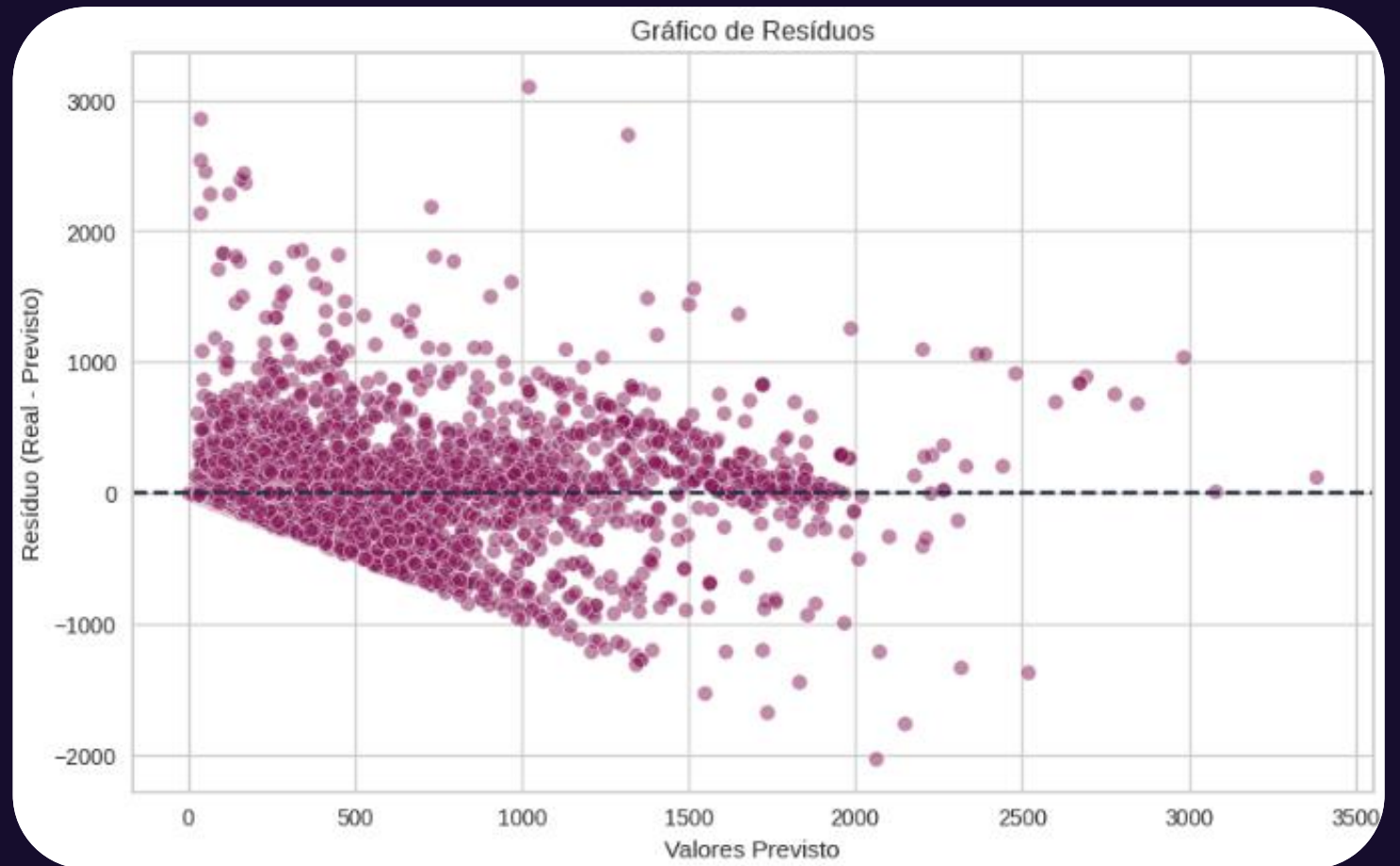


MODELO DE REGRESSÃO LINEAR

```
# Calculando a diferença (resíduos)
residuals = y_test - y_pred

# Criando um DataFrame para o gráfico
results = pd.DataFrame({'Real': y_test, 'Previsto': y_pred,
                        'Resíduo': residuals})

# Plotando os resíduos
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Previsto', y='Resíduo', data=results,
               color='#861454', alpha=0.5)
plt.axhline(0, color='#293049', linestyle='--')
plt.xlabel('Valores Previsto')
plt.ylabel('Resíduo (Real - Previsto)')
plt.title('Gráfico de Resíduos')
plt.show()
```



CONCLUSÃO FINAL

Foi possível detectar padrões sazonais com o dataset analisado com o decorrer dos anos.

Observou-se a majoritariedade das reclamações provenientes da Região Sudeste, resultado esperado visto que a maior concentração de companhias no Brasil encontram-se em São Paulo e região, visto que e por estado, bem como empresas telefônicas possuem grande quantidade de reclamações.

O modelo preditivo de regressão linear não obteve bons valores estatísticos, porém existem outros modelos que podem sanar a complexidade necessária.



AGRADECIMENTOS

"Agradeço a atenção de todos e a oportunidade de apresentar este projeto. Um especial agradecimento aos organizadores e patrocinadores do bootcamp pelo suporte e dedicação. Muito obrigado!"



S&P Global Foundation