RELATÓRIO TECH CHALLENGE – FASE 1

Grupo 38 – Gustavo Molina Figueiredo

Bibliotecas Utilizadas Para Desenvolvimento Do Projeto

- ✓ **NumPy:** Usada para operações numéricas e manipulação dos *arrays*.
- ✓ **Pandas:** Utilizada para manipulação dos dados e análise, ou seja, para leitura de arquivos CSV e operações em *Data Frames*.
- ✓ **Matplotlib e Seaborn:** Usadas para visualização de dados, incluindo os gráficos de distribuição, *boxplots* e diagramas de dispersão que foram desenvolvidos.
- ✓ **Scikit-learn:** Utilizada para o pré-processamento dos dados, modelagem e avaliação dos modelos:
    - LabelEncoder: Para codificação das variáveis categóricas.
    - StandardScaler: Para padronização dos dados, ajustando os dados para que tenham média zero e desvio padrão um.
    - train_test_split: Para dividir os dados em conjuntos de treino e teste.
    - RandomForestRegressor e LinearRegression: Para criação de modelos de previsão.
    - cross_val_score e KFold: Para validação cruzada.
- ✓ **SciPy**: Usada para o teste de normalidade Shapiro-Wilk.

Dataset Utilizado

O *dataset* utilizado no desenvolvimento do projeto é da kaggle e pode ser encontrado no *link* a seguir:
https://www.kaggle.com/code/mragpavank/medical-cost-personal-datasets/input

Para desenvolvimento do projeto o arquivo insurance.csv foi traduzido para português gerando dessa forma o arquivo insurance_PT_BR.csv

Imports Utilizados no Projeto

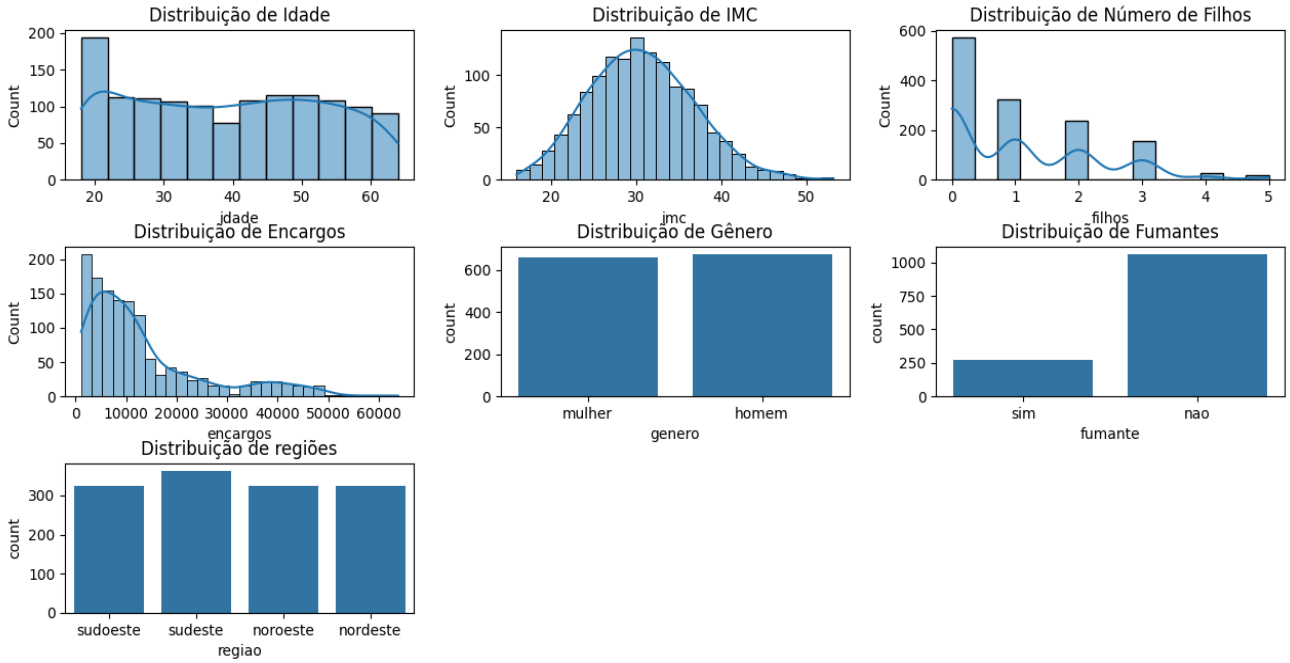
```
1   import numpy as np
2   import seaborn as sb
3   import matplotlib.pyplot as plt
4   import warnings
5   import pandas as pd
6   from sklearn.preprocessing import LabelEncoder
7   from sklearn.preprocessing import StandardScaler
8   from sklearn.model_selection import train_test_split
9   from sklearn.metrics import mean_squared_error, r2_score
10  from sklearn.ensemble import RandomForestRegressor
11  from sklearn.model_selection import cross_val_score
12  from sklearn.model_selection import KFold
13  from sklearn.linear_model import LinearRegression
14  from scipy import stats
```

Observação: A função mean_squared_error aparece “riscada” pois está obsoleta. Porém ela é utilizada no projeto e funciona normalmente 🚀🏆.

Gráficos de Distribuição Geral

```
57  # Retirando FutureWarning dos gráficos
58  warnings.filterwarnings("ignore", category=FutureWarning)
59
60  plt.figure(figsize=(12, 8))
61  plt.subplot(3, 3, 1)
62  sb.histplot(dados['idade'], kde=True)
63  plt.title('Distribuição de Idade')
64
65  plt.subplot(3, 3, 2)
66  sb.histplot(dados['imc'], kde=True)
67  plt.title('Distribuição de IMC')
68
69  plt.subplot(3, 3, 3)
70  sb.histplot(dados['filhos'], kde=True)
71  plt.title('Distribuição de Número de Filhos')
```

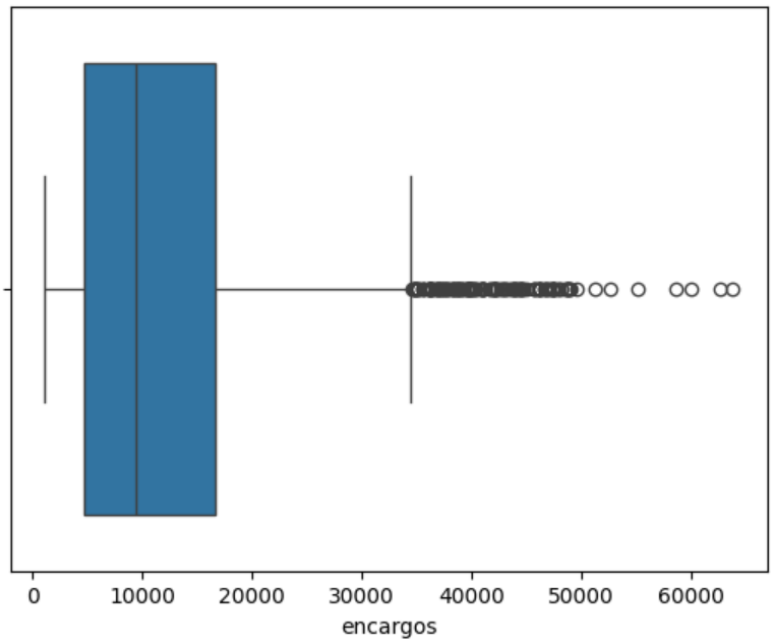
```
73  plt.subplot(3, 3, 4)
74  sb.histplot(dados['encargos'], kde=True)
75  plt.title('Distribuição de Encargos')
76
77  plt.subplot(3, 3, 5)
78  sb.countplot(x='genero', data=dados)
79  plt.title('Distribuição de Gênero')
80
81  plt.subplot(3, 3, 6)
82  sb.countplot(x='fumante', data=dados)
83  plt.title('Distribuição de Fumantes')
84
85  plt.subplot(3, 3, 7)
86  sb.countplot(x='regiao', data=dados)
87  plt.title('Distribuição de regiões')
88
89  plt.tight_layout()
90  plt.show()
```

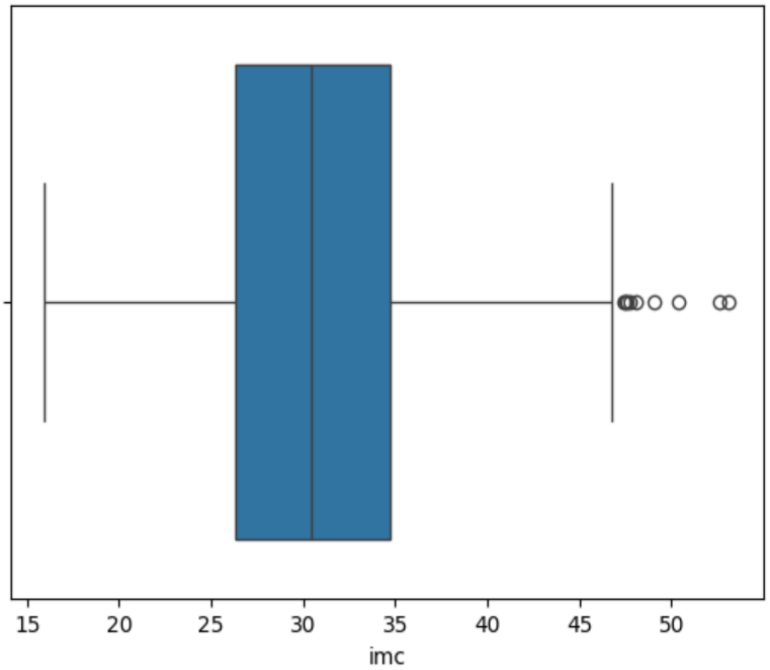


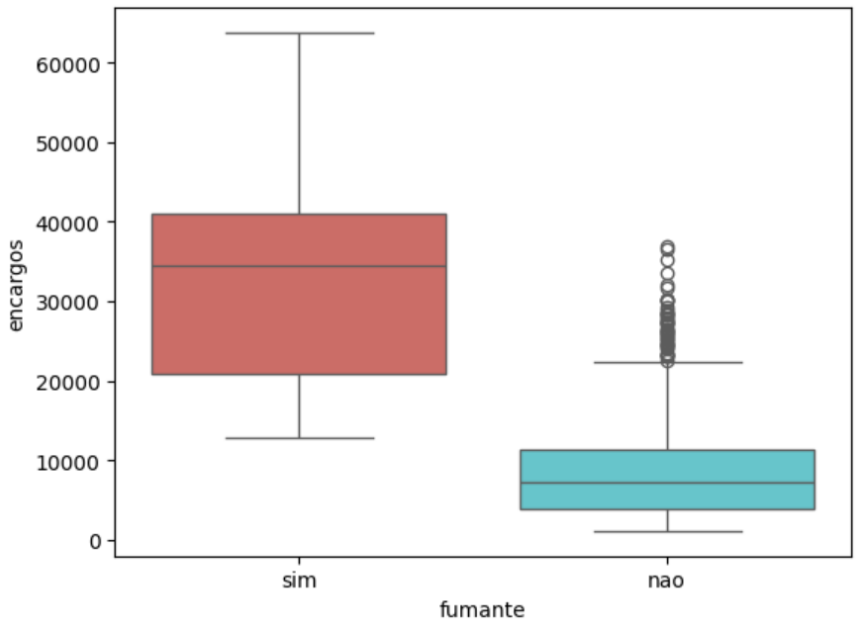
Os gráficos apresentados acima ilustram a frequência dos elementos em relação às características (*features*), mas alguns deles se destacam por suas particularidades, sendo:

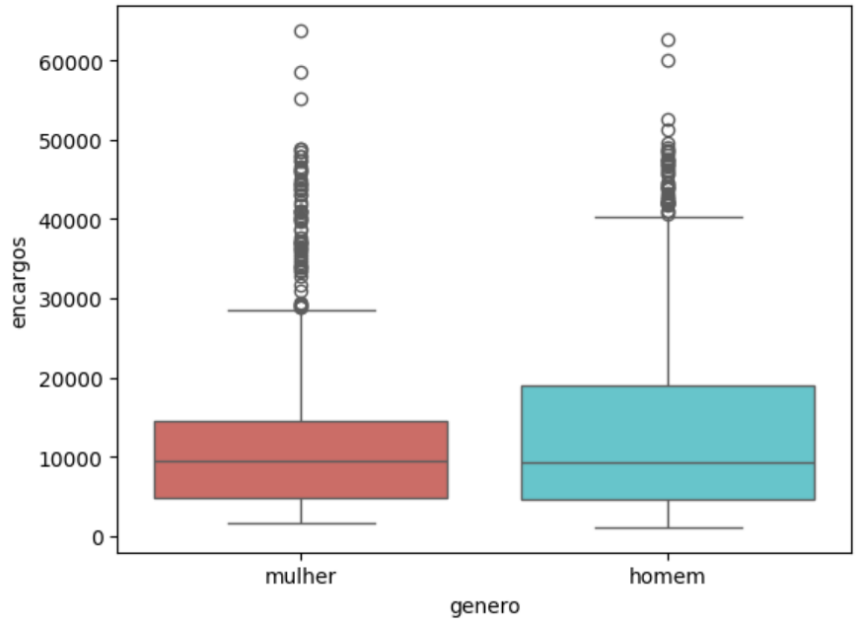
- O gráfico de dispersão em função da IDADE revela uma distribuição aceitável, com as idades apresentando frequências bastante próximas entre si, exceto pela faixa etária de 20 anos, que exibe valores elevados, mas sem comprometer a representatividade geral.
- O gráfico do IMC mostra uma distribuição normal (simétrica), onde os dados estão bem distribuídos ao longo da base.
- Um gráfico que chama a atenção é o de encargos, que apresenta uma distribuição não normal (assimétrica à direita), indicando a presença de potenciais *outliers*.
- Além disso, nota-se que há significativamente menos fumantes do que não fumantes, o que pode influenciar o meu modelo durante o processo de predição.

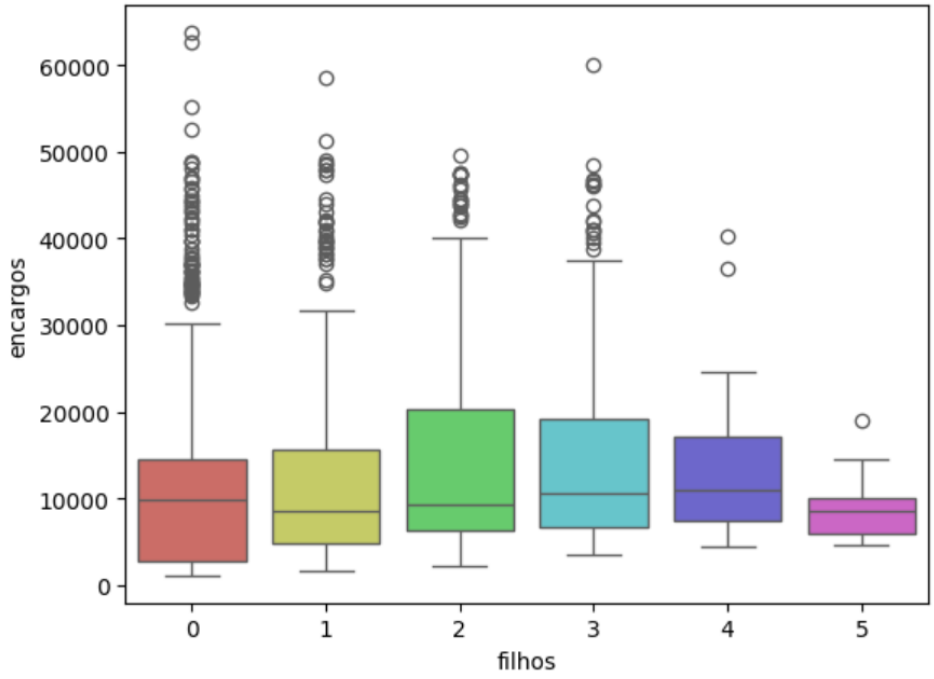
Analisando os *Outliers*

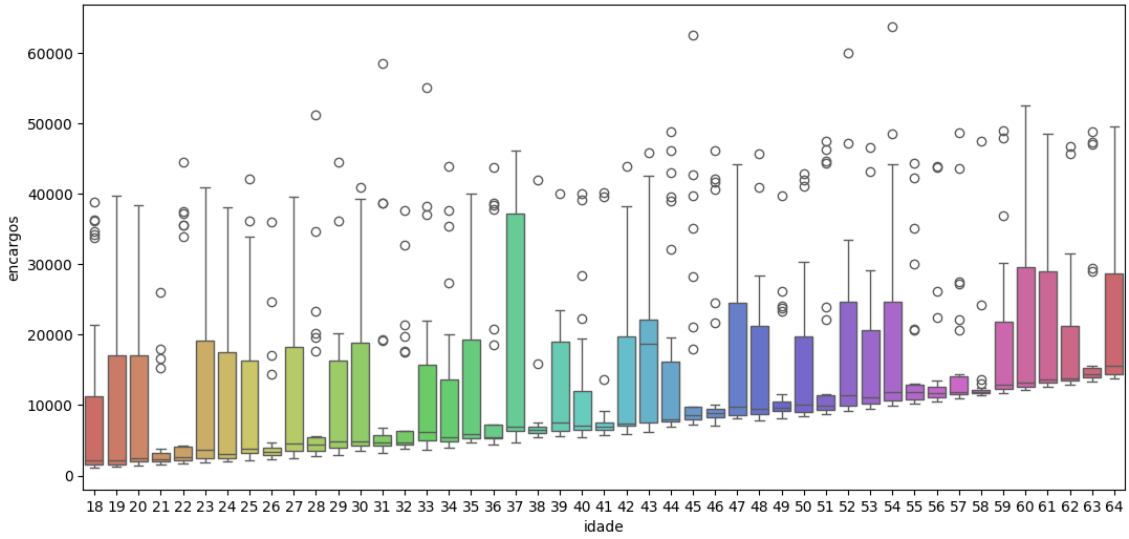





60000
50000
40000
30000
20000
10000
0
encargos
sim
nao
fumante


60000
50000
40000
30000
20000
10000
0
encargos
mulher
homem
genero





Os gráficos de *boxplot* são bastante úteis, pois, oferecem uma visualização clara e objetiva das principais características dos dados, facilitando a detecção de padrões, tendências e anomalias. Nos gráficos apresentados, observam-se diversos padrões, como:

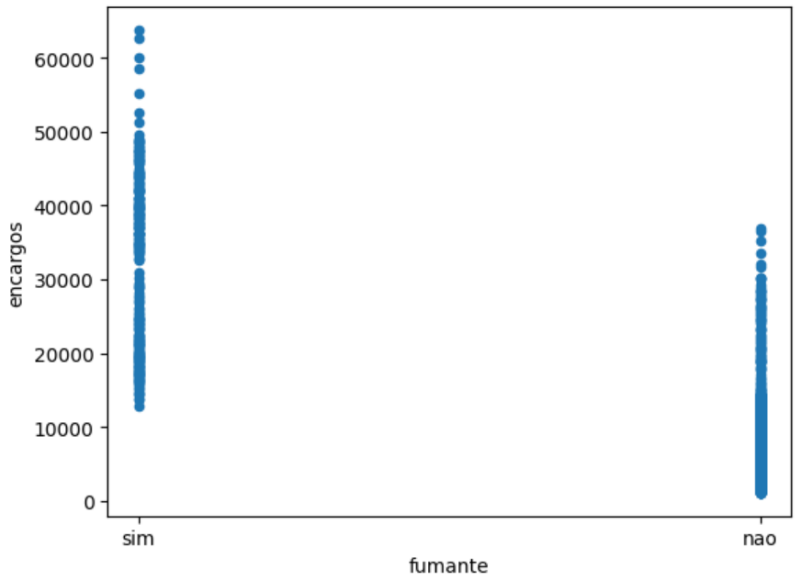
1. Encargos vs Fumantes: Identifiquei a presença de diversos *outliers* entre os não fumantes, cujos encargos elevados não parecem justificáveis, uma vez que este grupo deveria ter planos com custos mais baixos.

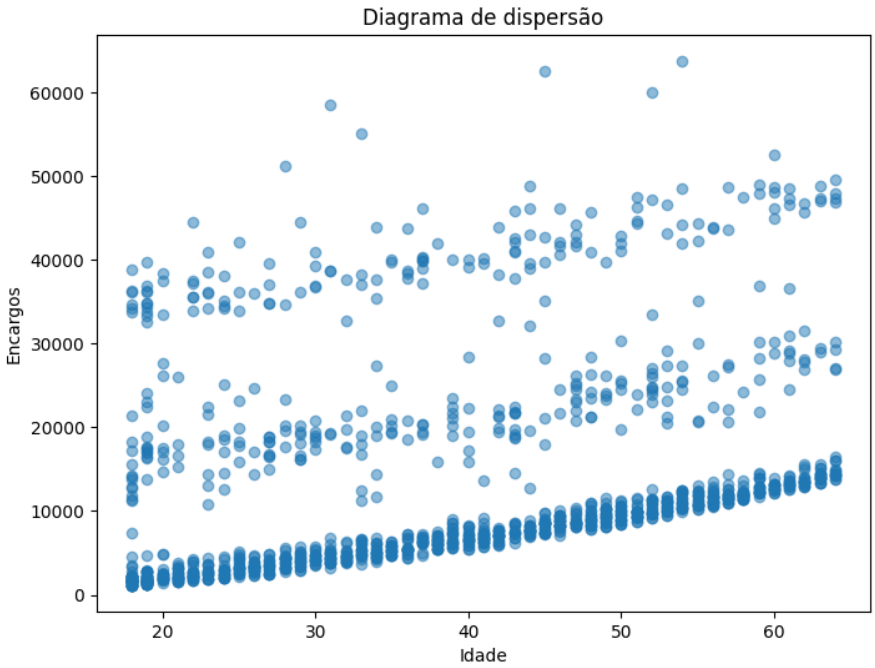
2. Encargos: Este gráfico revela uma grande concentração de *outliers* próximos a $35.000. No entanto, ele não é suficiente para identificar os responsáveis por essas discrepâncias, sendo necessário analisar essa variável em conjunto com outras que possam ser mais relevantes.

3. Encargos vs Gênero: Apresenta uma diferença significativa entre homens e mulheres, mas essa diferença não reflete necessariamente a realidade social. Devido à possibilidade de mulheres serem mães e utilizarem mais os planos de saúde, é esperado que elas tenham despesas maiores. Acredito que os *outliers* neste gráfico possam estar relacionados aos fumantes.

4. Encargos vs Filhos: Um ponto interessante observado aqui é que, ao relacionar o número de filhos com o status de fumante, notei uma grande quantidade de fumantes sem filhos, que pagam cerca de $30.000 em encargos. Isso reforça minha hipótese sobre os *outliers* no gráfico de Encargos vs Gênero.

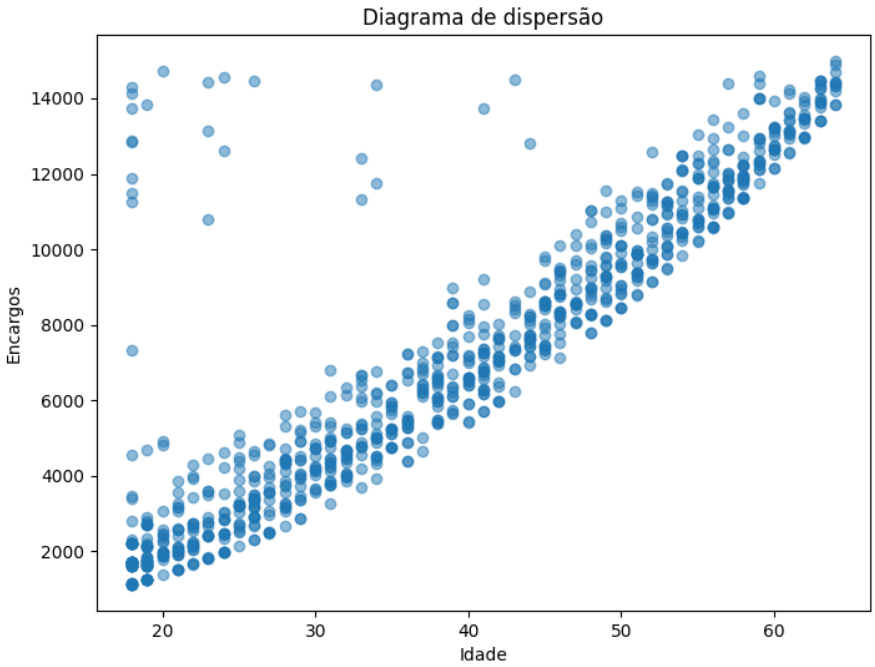
5. Encargos vs Idade: O gráfico que relaciona idade e encargos confirma a ideia de que não há uma justificativa plausível para os *outliers* observados, uma vez que indivíduos da mesma faixa etária apresentam encargos muito diferentes entre si.

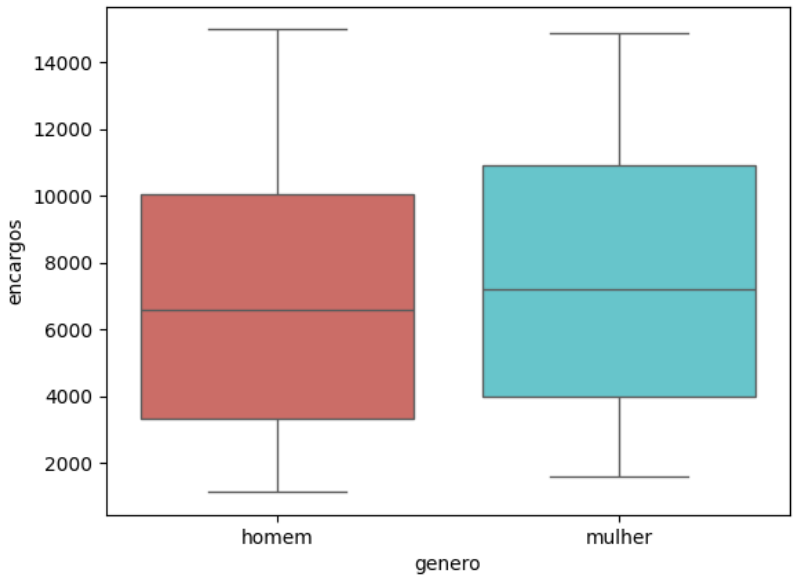


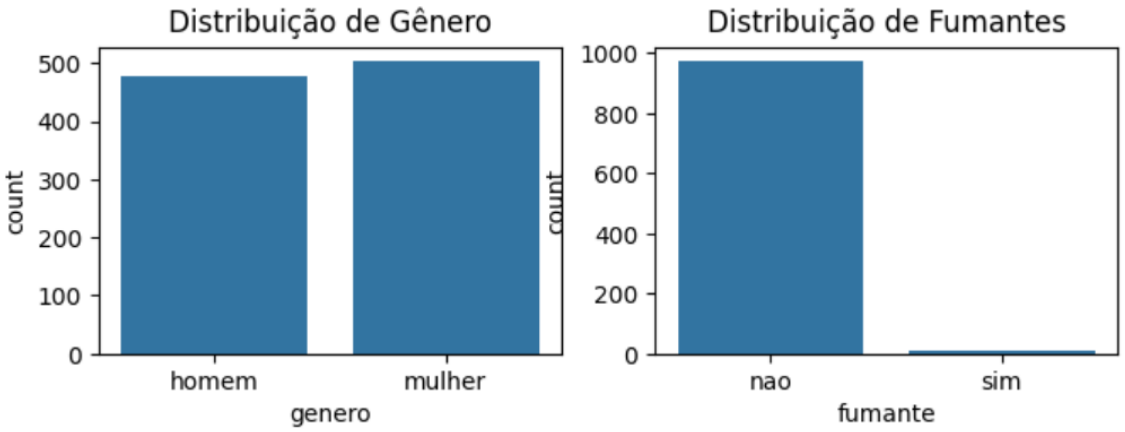


Observa-se que, ao analisar o gráfico de dispersão acima, existem duas camadas de *outliers*, sendo que a primeira camada começa perto da marca de $15.000 em encargos. Como não há variáveis que expliquem esses *outliers*, concluí que é necessário removê-los.

Remoção dos *Outliers*







Os gráficos de distribuição acima revelam que, mesmo após a remoção dos *outliers*, a distribuição entre homens e mulheres permanece equilibrada. No entanto, observa-se uma redução significativa no número de fumantes, refletindo a baixa representatividade desse grupo na base de dados.

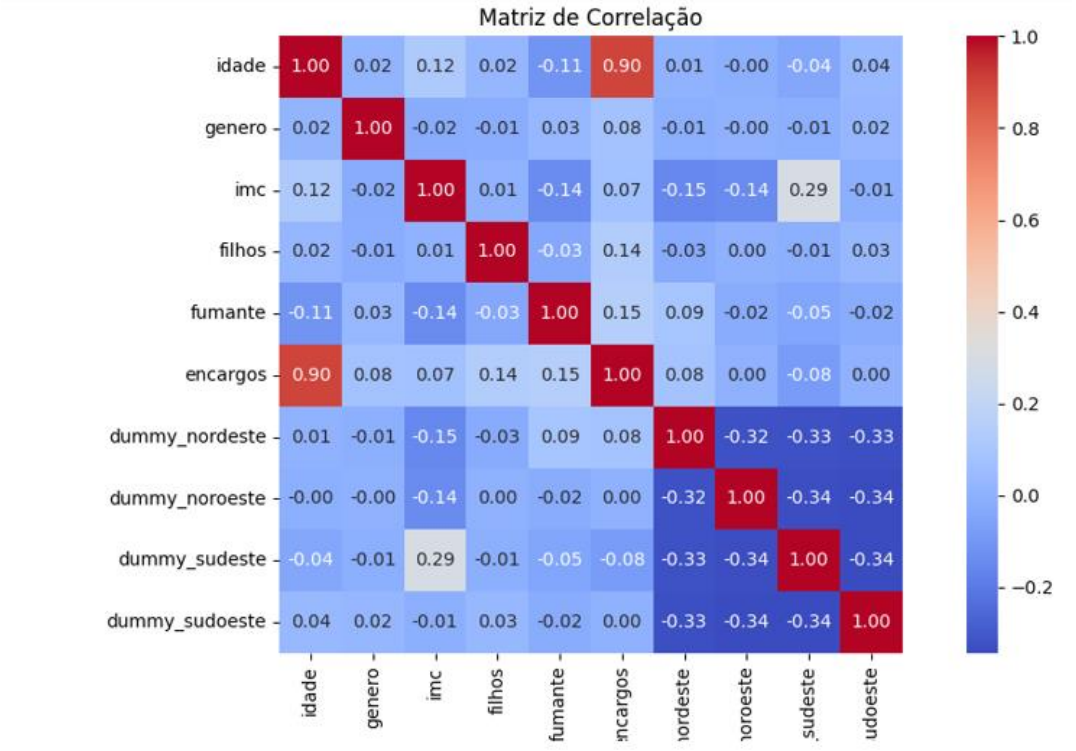
Pode-se concluir que muitos fumantes eram *outliers*, mas não há variáveis suficientes para explicar por que essa categoria apresentava encargos excessivamente altos.

```
176  print("Estatísticas Descritivas dos Dados sem outliers:\n")
177  print(dados.describe())
178
179  print("\nPercentual de Outliers Removidos da Base")
180  linhas_sem_outliers = dados.genero.count()
181  percentual_outliers_removidos = (linhas_com_outliers - linhas_sem_outliers) / linhas_com_outliers
182  print(f"\nTotal de outliers removidos: {percentual_outliers_removidos:.2%}")
```

```
Estatísticas Descritivas dos Dados sem outliers:

              idade         imc      filhos      encargos
count   980.000000  980.000000  980.000000    980.000000
mean     38.845918   30.492469    1.056122   7112.209423
std      13.984076    6.150983    1.211923   3869.930786
min      18.000000   15.960000    0.000000   1121.873900
25%      26.000000   26.101250    0.000000   3701.622875
50%      39.000000   30.200000    1.000000   6789.108725
75%      51.000000   34.320000    2.000000  10411.043550
max      64.000000   53.130000    5.000000  14988.432000

Percentual de Outliers Removidos da Base

Total de outliers removidos: 26.76%
```

Padronização dos Dados

Após a remoção dos *outliers*, os dados foram padronizados utilizando o **LabelEncoder.** O LabelEncoder é uma classe da biblioteca Scikit-learn utilizada para converter variáveis categóricas em valores numéricos. Na sequência foi construída a Matriz de Correlação.



A matriz de correlação acima fornece informações sobre a relação entre as diferentes variáveis do conjunto de dados. Cada célula da matriz representa o coeficiente de correlação entre duas variáveis. Algumas observações importantes sobre a matriz:

- A diagonal principal possui o valor 1.0, pois a correlação de uma variável consigo mesma é sempre 1.
- Os valores abaixo da diagonal principal são simétricos em relação à diagonal, pois a correlação entre as variáveis x e y é a mesma que a correlação entre y e x.
- Os valores positivos indicam uma relação direta entre as variáveis, enquanto os valores negativos indicam uma relação inversa.
- Algumas correlações se destacam, como a forte correlação positiva entre "imc" e "idade" (0.90), a moderada correlação positiva entre "encargos" e "idade" (0.90), e a forte correlação negativa entre "dummy_nordeste" e "dummy_sudeste" (-0.34).

Após analisar a matriz de correlação construída realizei o teste de normalidade que é um procedimento estatístico utilizado para avaliar se um conjunto de dados segue uma distribuição normal. O primeiro teste de normalidade que realizei foi **o teste de Shapiro-Wilk.**

```
210   # Teste de normalidade
211   _, p_value_idade = stats.shapiro(dados['idade'])
212   _, p_value_encargos = stats.shapiro(dados['encargos'])
213   print(f"\nTeste de Shapiro-Wilk para normalidade:")
214   print(f"Idade: p-value = {p_value_idade:.5f}")
215   print(f"Encargos: p-value = {p_value_encargos:.5f}")
```

```
Teste de Shapiro-Wilk para normalidade:
Idade: p-value = 0.00000
Encargos: p-value = 0.00000
```

Com base nos resultados do teste de Shapiro-Wilk apresentados podem-se fazer as seguintes interpretações:

Idade:

- O p-value é menor que o nível de significância usual de 0,05.
- Isso significa que podemos rejeitar a hipótese nula de que a variável "idade" segue uma distribuição normal.
- Ou seja, a distribuição de idade nos dados não pode ser considerada normal.

Encargos:

- O p-value também é menor que 0,05.
- Portanto, rejeita-se a hipótese nula e conclui-se que a variável "encargos" não segue uma distribuição normal.
- A distribuição de encargos apresenta desvios significativos em relação à normalidade.

Na sequência foi realizado o teste-t para comparação de médias entre fumantes e não-fumantes.

```
217   # Teste T para comparação de média de fumantes e não- fumantes
218
219   fumantes = dados[dados['fumante'] == 1]['encargos']
220   nao_fumantes = dados[dados['fumante'] == 0]['encargos']
221   t_stat, p_value_t = stats.ttest_ind(fumantes, nao_fumantes)
222   print(f"\nTeste t para comparação de médias entre fumantes e não fumantes:")
223   print(f"t-statistic = {t_stat:.5f}, p-value = {p_value_t:.5f}")
```

```
Teste t para comparação de médias entre fumantes e não fumantes:
t-statistic = 4.82338, p-value = 0.00000
```

Dado que o p-value é muito menor que 0,05, rejeita-se a hipótese nula. Portanto, pode-se concluir que existe uma diferença estatisticamente significativa nas médias de encargos entre fumantes e não-fumantes. **Isso sugere que os fumantes, em média, têm encargos significativamente maiores do que os não-fumantes.**

Validação Cruzada

```
# Aplicando a Validação Cruzada K-Fold

forest_model = RandomForestRegressor(n_estimators=10, random_state=42)
kfold = KFold(n_splits=5, shuffle=True)
scores = cross_val_score(forest_model, x, y, cv=5)
mean_mse = scores.mean()
print(f"\nK-Fold (R^2) Scores: {scores}")
print(f"Média do Erro Médio Quadrático (MSE) utilizando Cross-Validation: {mean_mse:.5f}")
print(f"Raiz do Erro Médio Quadrático (RMSE) utilizando Cross-Validation: {np.sqrt(mean_mse):.5f}")
```

```
K-Fold (R^2) Scores: [0.96334621 0.86315521 0.93473627 0.92907752 0.69219854]
Média do Erro Médio Quadrático (MSE) utilizando Cross-Validation: 0.87650
Raiz do Erro Médio Quadrático (RMSE) utilizando Cross-Validation: 0.93622
```

O código apresentado acima realiza a validação cruzada em um modelo de regressão utilizando a classe RandomForestRegressor do Scikit-learn. O primeiro, segundo terceiro e quarto scores (0.963, 0.863, 0.934, 0.929) indicam que o modelo é capaz de explicar uma grande proporção da variabilidade dos dados nesses *folds*. **Esses valores são considerados altos, sugerindo um bom ajuste do modelo.** O quinto score (0.692) é significativamente mais baixo em comparação com os outros. Isso indica que, para esse *fold* específico, o modelo não está performando tão bem, explicando apenas cerca de 69% da variabilidade dos dados. Isso pode ser devido a características específicas dos dados nesse *fold* ou a uma divisão que não representa bem o conjunto completo.

O MSE médio de 0.87650 indica que, em média, o erro quadrático das previsões do modelo em relação aos valores reais é relativamente baixo.

O RMSE de 0.93622 fornece uma medida do erro médio das previsões em unidades da variável de saída. Neste caso, o RMSE é razoável, especialmente considerando que o MSE é relativamente baixo.

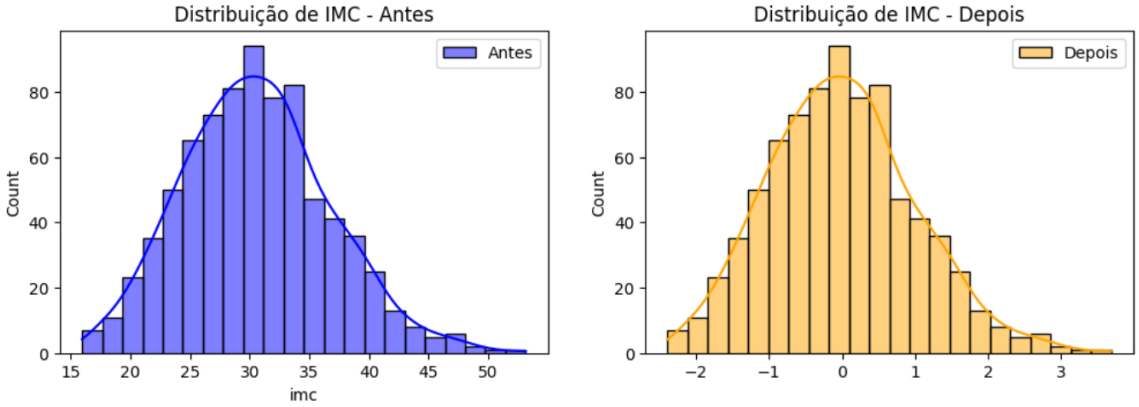
Na sequência do projeto, houve a divisão entre treino e teste onde **20% dos dados totais foram reservados para teste, enquanto 80% foram usados para treinamento.**

```
247  # Divisão entre Treino e Teste
248
249  x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2, random_state = 42)
```

Dando continuidade, a classe StandardScaler da biblioteca Scikit-learn foi utilizada para padronizar os dados de entrada.

```
251  # Padronizando Dados de Treino e Teste
252
253  scaler_std = StandardScaler()
254  scaler_std.fit(x_train)
255  x_dados_std_train = scaler_std.fit_transform(x_train)
256  x_dados_std_test = scaler_std.transform(x_test)
257  print("\n",x_dados_std_train,"\n")
258  print(x_dados_std_test,"\n")
```

Após a padronização dos dados, criei dois gráficos de distribuição do IMC para verificar a frequência após a padronização.



O Objetivo dos gráficos acima foi identificar se houveram diferenças nos dados não padronizados em relação aos padronizados. Porém, conforme é visto acima, nota-se que a frequência permanece a mesma.

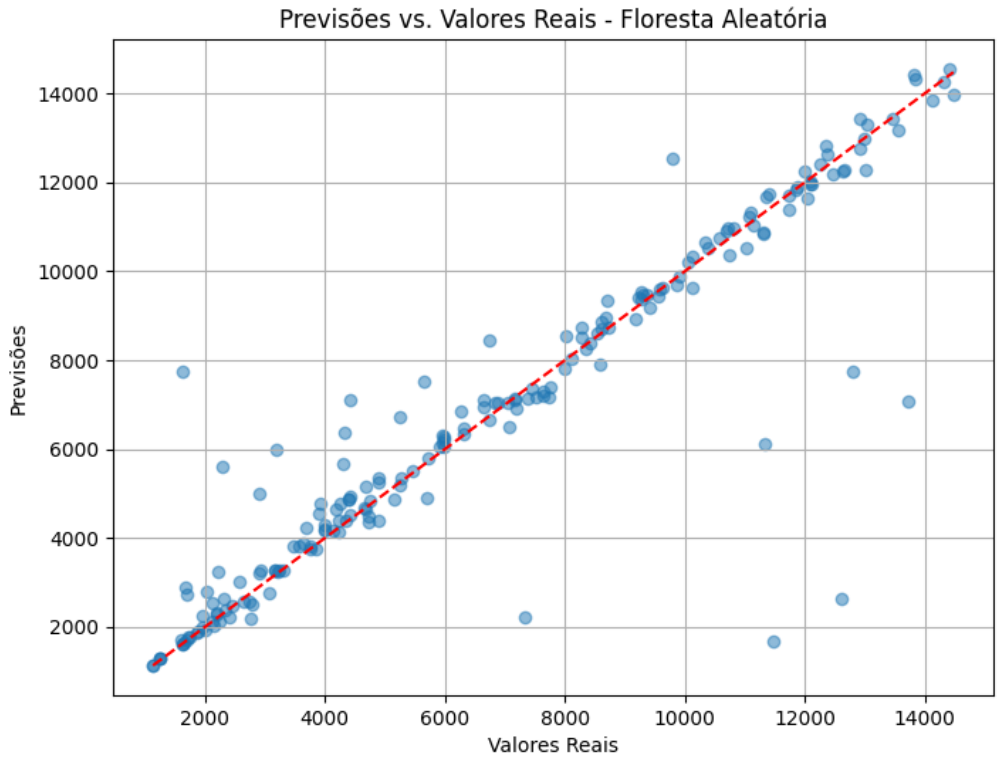
Trabalhando com o Modelo de Random Forest

O modelo foi treinado pelo algoritmo do Random Forest e ele foi utilizado para fazer previsões nos dados de teste.

```
# Iniciando o Modelo de Random Forest

print("\nIniciando o modelo de Random Forest")
forest_model.fit(x_dados_std_train, y_train) # Treinando o modelo
y_pred_forest = forest_model.predict(x_dados_std_test) #Fazendo previsões nos dados de teste

# Avaliando o desempenho do modelo
mse_forest = mean_squared_error(y_test, y_pred_forest)
rmse_forest = np.sqrt(mse_forest)
r2_forest = r2_score(y_test, y_pred_forest)
print(f"\nErro médio quadrático (MSE) - Floresta Aleatória: {mse_forest:.5f}")
print(f"Raiz do erro médio quadrático (RMSE) - Floresta Aleatória: {rmse_forest:.5f}")
print(f"Coeficiente de determinação (R^2) - Floresta Aleatória: {r2_forest:.5f}\n")
```

```
Iniciando o modelo de Random Forest

Erro médio quadrático (MSE) - Floresta Aleatória: 2183632.98808
Raiz do erro médio quadrático (RMSE) - Floresta Aleatória: 1477.71208
Coeficiente de determinação (R^2) - Floresta Aleatória: 0.85241

MAPE:  11.34%
```

Os resultados sugerem que o modelo de Floresta Aleatória **está se saindo bem em termos de precisão e capacidade de explicar a variabilidade nos dados.** O MSE e RMSE são relativamente baixos, indicando que as previsões estão próximas dos valores reais, e um R^2 de aproximadamente 0.85 sugere um bom ajuste. O MAPE (Erro Percentual Médio Absoluto) de 11.34% é também um indicador positivo, mostrando que as previsões são razoavelmente precisas em termos percentuais.



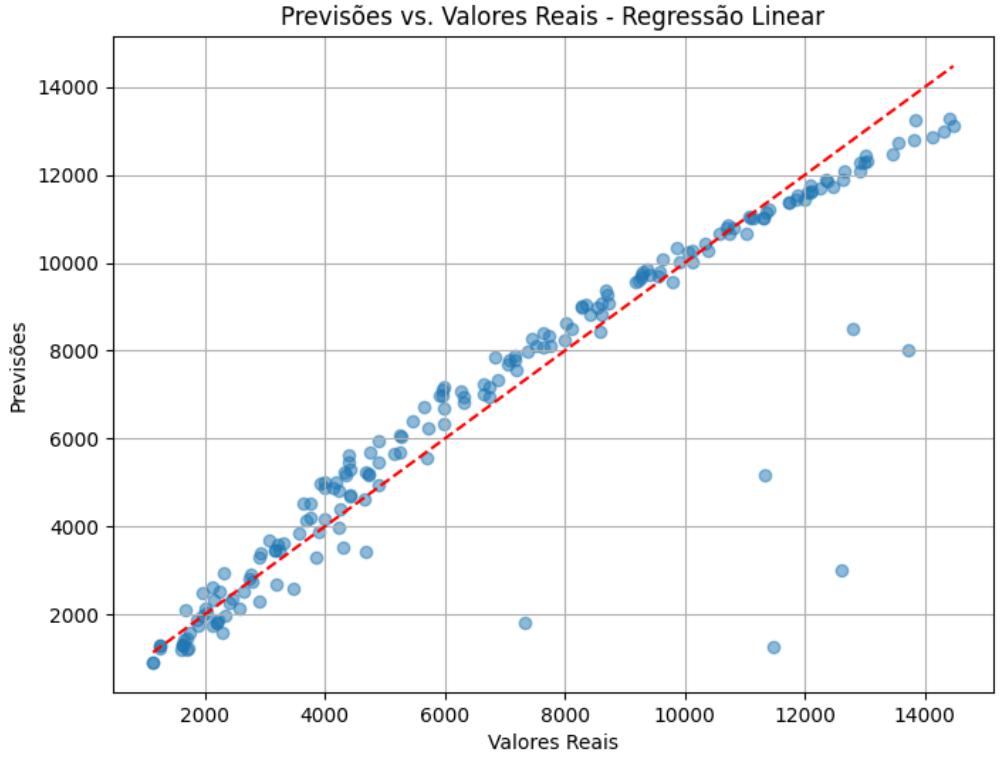
Ao analisar os gráficos verifica-se que grande parte dos pontos estão muito próximo a reta, apesar de existirem, em algumas regiões, pontos fora dela. Pode-se assim afirmar que **o modelo de Random Forest conseguiu um alinhamento favorável dos dados** seguindo uma certa tendência, mantendo assim o coeficiente de determinação em 0,85.

Trabalhando com o Modelo de Regressão Linear

Para que eu pudesse ter uma base para comparação, inclusive com relação a questão de performance, fiz o mesmo procedimento com a regressão linear.

```
Iniciando Modelo de Regressão Linear

Erro médio quadrático (MSE): 1933190.55178
Erro médio quadrático (RMSE): 1390.39223
Coeficiente de determinação (R^2): 0.86933

MAPE:  11.18%
```

Os resultados mostram que o modelo de Regressão Linear **apresenta um bom ajuste aos dados**, com um MSE e RMSE relativamente baixos, um R^2 alto e um MAPE razoável. **Isso sugere que o modelo é capaz de fazer previsões precisas para essa tarefa.**



Ao analisar o gráfico de Regressão Linear verifica-se que apesar dos pontos não estarem todos sobre a reta, o modelo conseguiu um melhor alinhamento dos dados. **O**

modelo obteve o coeficiente de determinação de 0,86. Em questão de desempenho, a regressão linear apresentou uma performance superior ao Random Forest.