```
#1 BIBLIOTECAS
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score
from scipy.stats import pearsonr
from sklearn import metrics


#2 IMPORTANDO ARQUIVO
dados=pd.read_csv('insurance.csv')


#3 ANALISANDO OS DADOS I (ANÁLISE EXPLORATÓRIA DOS DADOS - AED)
print(dados.head())
print(dados.shape)
```

Mostrar saída oculta

```
#4 ANALISANDO OS DADOS II - AED
print(dados.dtypes)
```

Mostrar saída oculta

```
#5 ANALISANDO OS DADOS III - AED
dados.describe().round(2)
```

Mostrar saída oculta

```
#6 PRÉ PROCESSSANDO OS DADOS I
#Convertendo as variáveis SEX, SMOKER e REGION em numéricas (ENCODING)
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()

#sex
le.fit(dados.sex)
dados.sex = le.transform(dados.sex)

# smoker
le.fit(dados.smoker)
dados.smoker = le.transform(dados.smoker)

#region
le.fit(dados.region)
dados.region = le.transform(dados.region)

print(dados.head())
print(dados.shape)
```

```
   age  sex     bmi  children  smoker  region      charges
0   19    0  27.900         0       1       3  16884.92400
1   18    1  33.770         1       0       2   1725.55230
2   28    1  33.000         3       0       2   4449.46200
3   33    1  22.705         0       0       1  21984.47061
4   32    1  28.880         0       0       1   3866.85520
```

```
(1338, 7)
```

```
#7 ANALISANDO OS DADOS IV - AED
#CORRELAÇÕES
dados.corr().round(2)
```

	age	sex	bmi	children	smoker	region	charges
age	1.00	-0.02	0.11	0.04	-0.03	0.00	0.30
sex	-0.02	1.00	0.05	0.02	0.08	0.00	0.06
bmi	0.11	0.05	1.00	0.01	0.00	0.16	0.20
children	0.04	0.02	0.01	1.00	0.01	0.02	0.07
smoker	-0.03	0.08	0.00	0.01	1.00	-0.00	0.79
region	0.00	0.00	0.16	0.02	-0.00	1.00	-0.01
charges	0.30	0.06	0.20	0.07	0.79	-0.01	1.00

```
#8 FILTRANDO DADOS I
#FILTRO PARA SEPARAR SOMENTE OS FUMANTES
dados = dados[dados['smoker'] == 1]
print(dados.head())
print(dados.shape)
```

```
    age  sex    bmi  children  smoker  region      charges
0    19    0  27.90         0       1       3  16884.9240
11   62    0  26.29         0       1       2  27808.7251
14   27    1  42.13         0       1       2  39611.7577
19   30    1  35.30         0       1       3  36837.4670
23   34    0  31.92         1       1       0  37701.8768
(274, 7)
```

```
#9 FILTRANDO DADOS II
#FILTRO PARA SEPARAR SOMENTE AS II - MULHERES
dados = dados[dados['sex'] == 1]
print(dados.head())
print(dados.shape)
```

```
    age  sex    bmi  children  smoker  region      charges
14   27    1  42.13         0       1       2  39611.75770
19   30    1  35.30         0       1       3  36837.46700
29   31    1  36.30         2       1       3  38711.00000
30   22    1  35.60         0       1       3  35585.57600
34   28    1  36.40         1       1       3  51194.55914
(159, 7)
```

```
#10 ESCOLHA DAS VARIÁVEIS : IMC X GASTO COM SEGURO
X = dados['bmi'].values
Y = dados['charges'].values
```

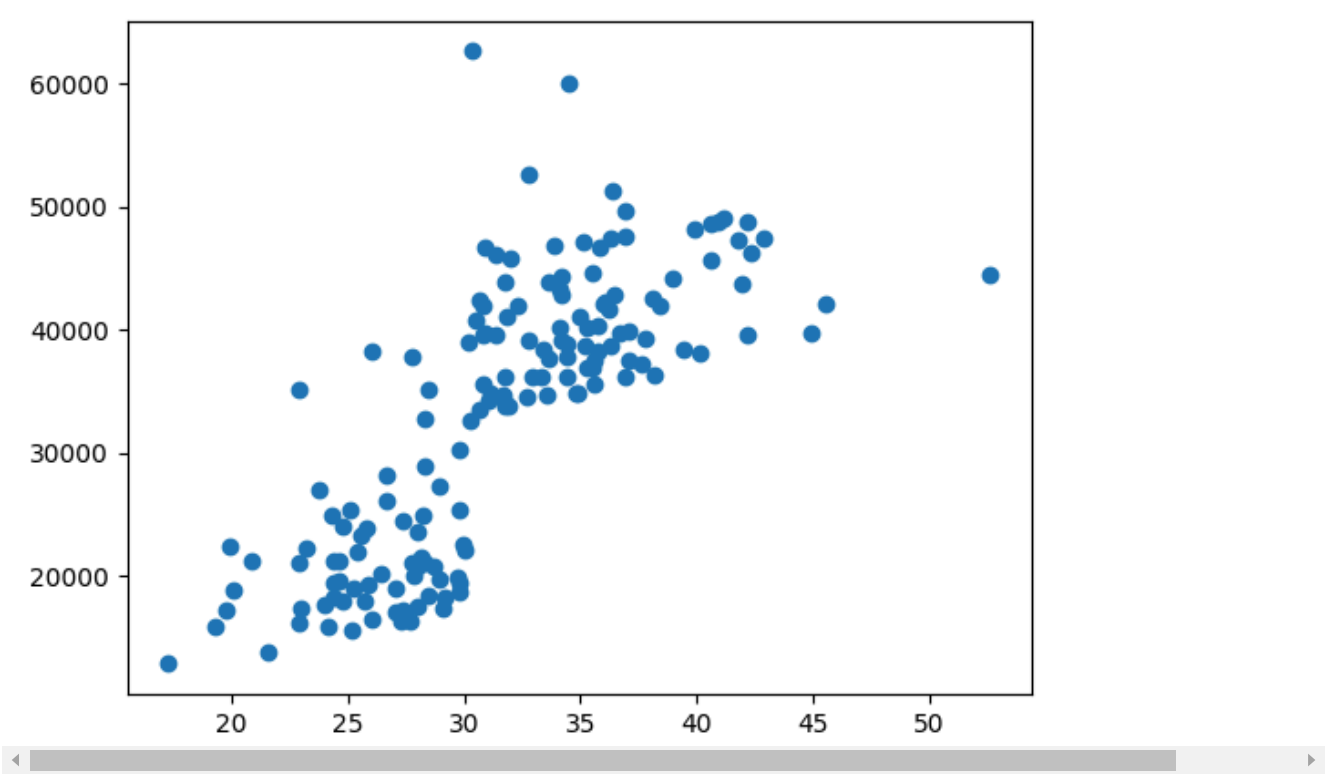
```
print(X)
```

Mostrar saída oculta

```
print(Y)
```

Mostrar saída oculta

```
#11 ANÁLISE GRÁFICA - AED
#Gráfico da relação entre IMC x Custo
plt.scatter(X, Y)
plt.show()
```




```
#12 PEARSON
#Calculo do r (Pearson)
r = pearsonr(X, Y)
print(f'Coeficiente de correlação: {r}')
```

```
Coeficiente de correlação: PearsonRResult(statistic=0.7693553500239402, pvalue=2.29005
```

```
#13 MLS I
#Separar os conjuntos TREINAMENTO e TESTE (70% / 30%)
x_train, x_test, y_train, y_test = train_test_split(X,Y,test_size=0.3)
```

Clique duas vezes (ou pressione "Enter") para editar

```
#Dados de x (Features)
print(x_train)
```

Mostrar saída oculta

```
print(x_test)
```

Mostrar saída oculta

```
#Dados de y (Target)
print(y_train)
```

Mostrar saída oculta

```
print(y_test)
```

Mostrar saída oculta

```
#14 PRÉ PROCESSSANDO OS DADOS II
# Carregar os dados no modelo de ML
# Transformar os dados de treino e teste em arrays coluna
x_train=x_train.reshape(-1,1)
y_train=y_train.reshape(-1,1)
x_test=x_test.reshape(-1,1)
y_test=y_test.reshape(-1,1)
```

```
print(x_train)
```

Mostrar saída oculta

```
print(y_train)
```

Mostrar saída oculta

```
print(x_test)
```

Mostrar saída oculta

```
print(y_test)
```

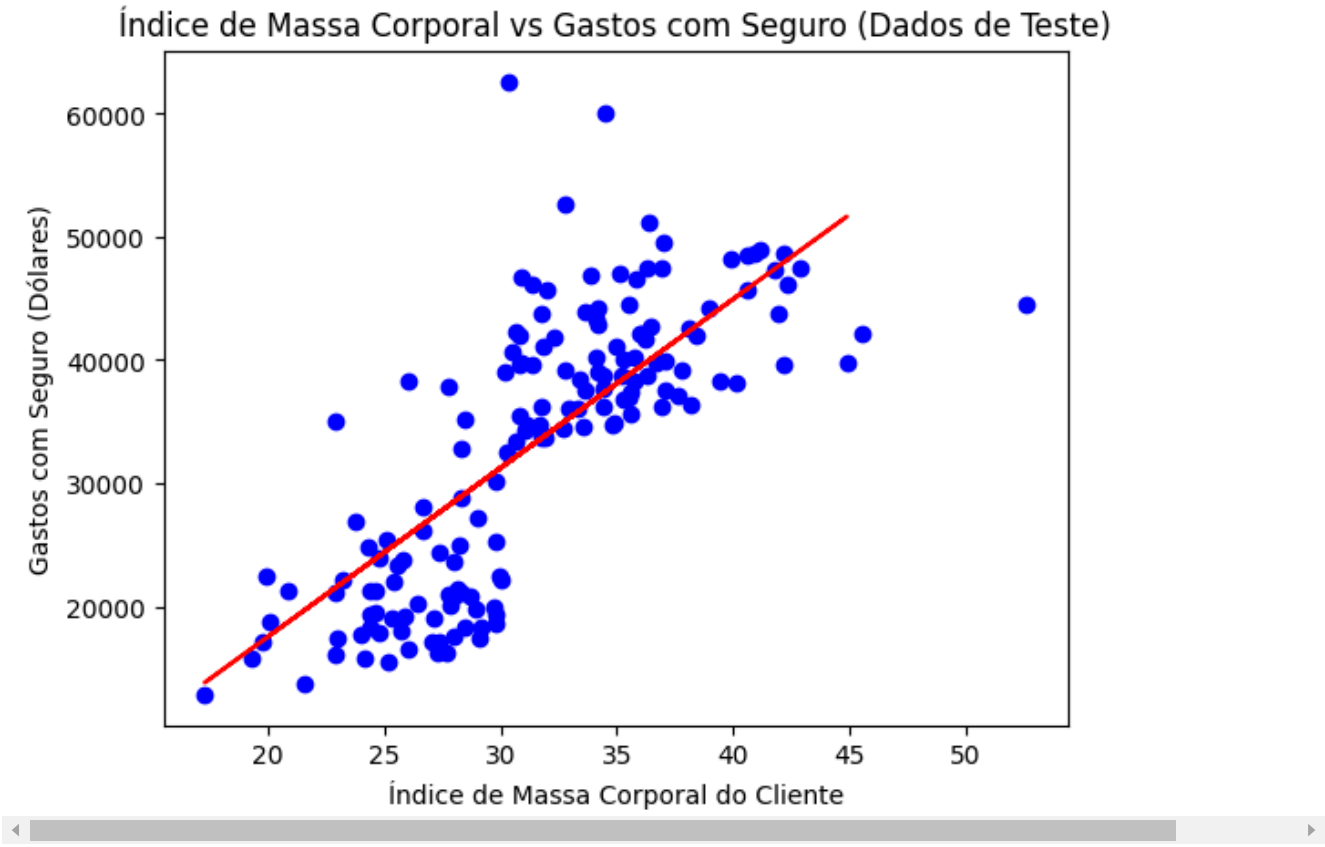
Mostrar saída oculta

```
# 15 MLS
# Aplicação do Método de MLS (Regressão Linear)
# 15.1 Ajuste do MODELO
reg = LinearRegression()
reg.fit(x_train,y_train)
# 15.2 Predição com o MODELO (TESTE COM x_teste -> pred)
pred = reg.predict(x_test)
print(pred)
```

Mostrar saída oculta

```
#16 ANÁLISE GRÁFICA - Dados Experimentais x Modelo
plt.scatter(X, Y, color="blue")
plt.plot(x_test, pred, color="red")
plt.title("Índice de Massa Corporal vs Gastos com Seguro (Dados de Teste)")
plt.xlabel("Índice de Massa Corporal do Cliente")
plt.ylabel("Gastos com Seguro (Dólares)")
```

```
Text(0, 0.5, 'Gastos com Seguro (Dólares)')
```




```
#17 CÁLCULO DO R2 (AJUSTE LINEAR)
r_squared = r2_score(y_test, pred)
print(f'Coeficiente r2: {r_squared}')
```

```
Coeficiente r2: 0.6871893267247464
```

```
#18 DETERMINAÇÃO DO AJUSTE (ERRO MÉDIO)
print('MAE (Erro):', metrics.mean_absolute_error(y_test, pred))
```

```
MAE (Erro): 5118.9430089859425
```