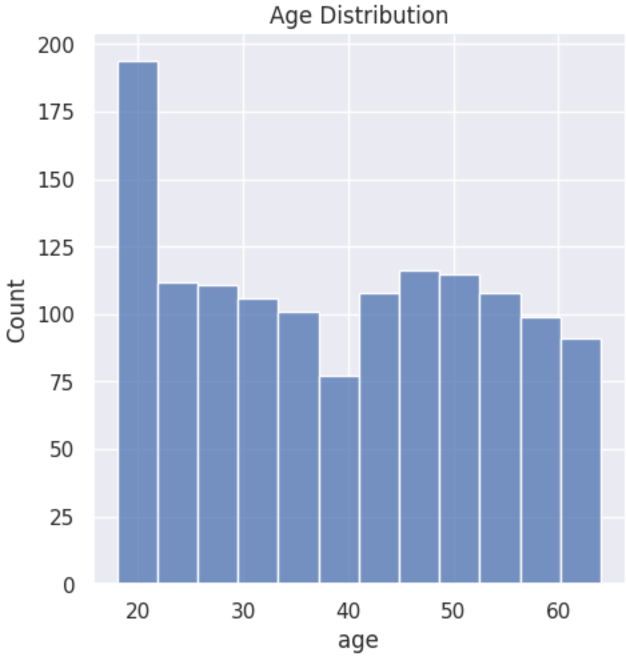
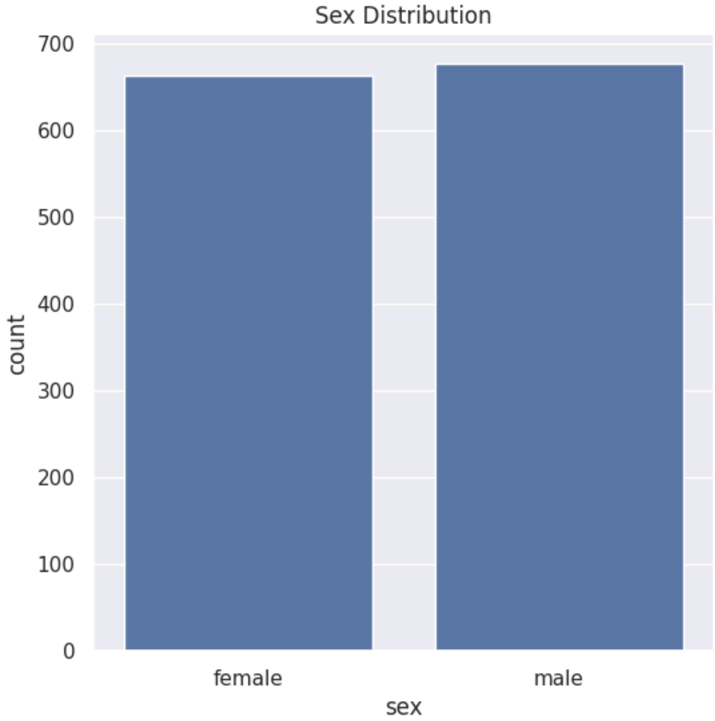
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

df = pd.read_csv("/content/insurance.csv")
#statistical Measures of the dataset
print("statistical Measures of the dataset")
df.describe()

#distribution of age value
print("Distribution of age value:")
sns.set()
plt.figure(figsize=(6,6))
sns.displot(df['age'])
plt.title('Age Distribution')
plt.show()

# Gender column
print("Distribution of Gender column:")
plt.figure(figsize=(6,6))
sns.countplot(x='sex', data=df)
plt.title('Sex Distribution')
plt.show()

# bmi distribution
print("Distribution of BMI:")
plt.figure(figsize=(6,6))
sns.displot(df['bmi'])
plt.title('BMI Distribution')
plt.show()

# children column
print("Distribution of Children column:")
plt.figure(figsize=(6,6))
sns.countplot(x='children', data=df)
plt.title('Children')
plt.show()

#smoker column
print("Distribution of Smoker column:")
plt.figure(figsize=(6,6))
sns.countplot(x='smoker', data=df)
plt.title('smoker')
plt.show()

# region column
print("Distribution of Region column:")
plt.figure(figsize=(6,6))
sns.countplot(x='region', data=df)
plt.title('region')
plt.show()

# distribution of charges value
print("Distribution of Charges:")
plt.figure(figsize=(6,6))
sns.displot(df['charges'])
plt.title('Charges Distribution')
plt.show()

# Convert categorical variables to numerical using one-hot encoding
df = pd.get_dummies(df, columns=['sex', 'smoker'], drop_first=True)

# Perform simple linear regression for each feature
features = ['age', 'bmi', 'children', 'sex_male', 'smoker_yes']

for feature in features:
    X = df[feature].values.reshape(-1, 1)
    y = df['charges'].values

    # Split the data into training and testing sets
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

    # Create and fit the model
    model = LinearRegression()
    model.fit(X_train, y_train)

    # Make predictions on the test set
    y_pred = model.predict(X_test)
    # Plot the regression line
    plt.figure(figsize=(8, 6))
    sns.scatterplot(x=X_test.flatten(), y=y_test, label='Actual Charges')
    sns.lineplot(x=X_test.flatten(), y=y_pred, color='red', label='Regression Line')
    plt.title(f'Simple Linear Regression for {feature}')
    plt.xlabel(feature)
    plt.ylabel('Charges')
    plt.legend()
    plt.show()

    # Print the mean squared error
    mse = mean_squared_error(y_test, y_pred)
    print(f'Mean Squared Error for {feature}: {mse:.2f}')
```

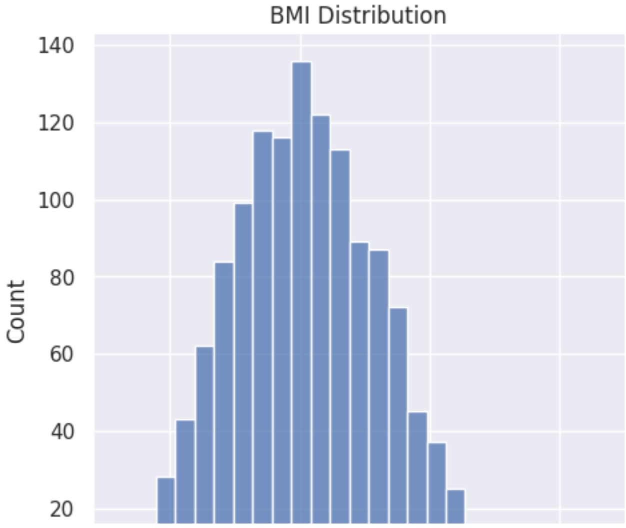
```
statistical Measures of the dataset
Distribution of age value:
<Figure size 600x600 with 0 Axes>
```

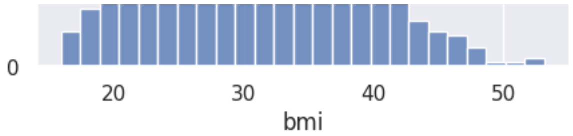


```
Distribution of Gender column:
```

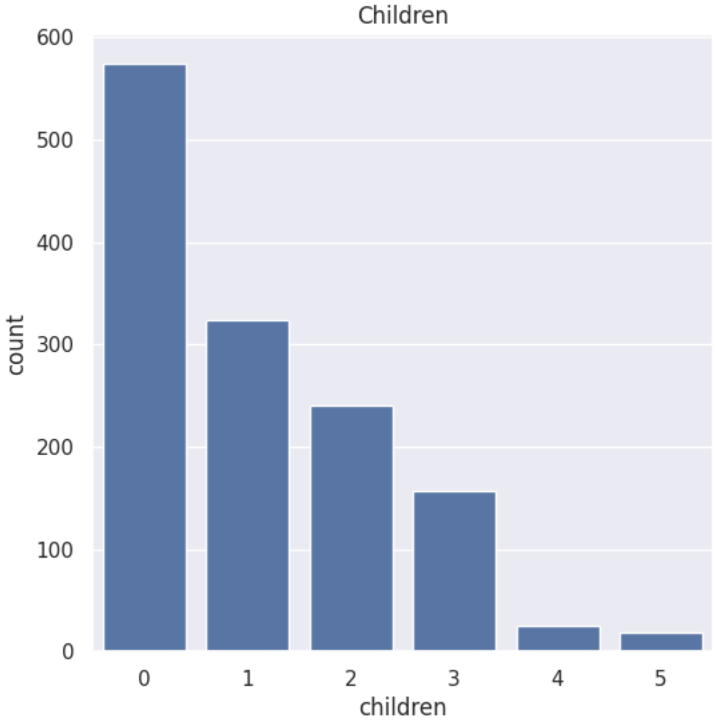


```
Distribution of BMI:
<Figure size 600x600 with 0 Axes>
```

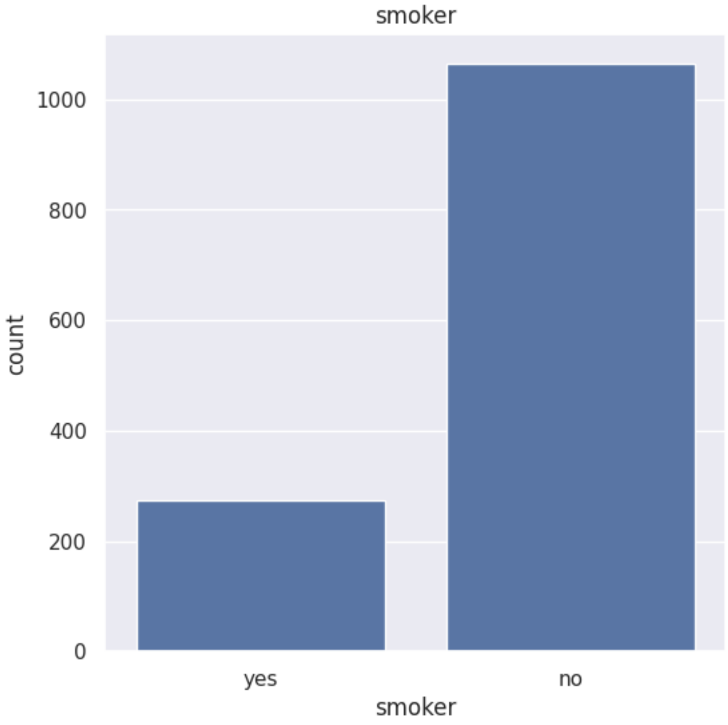




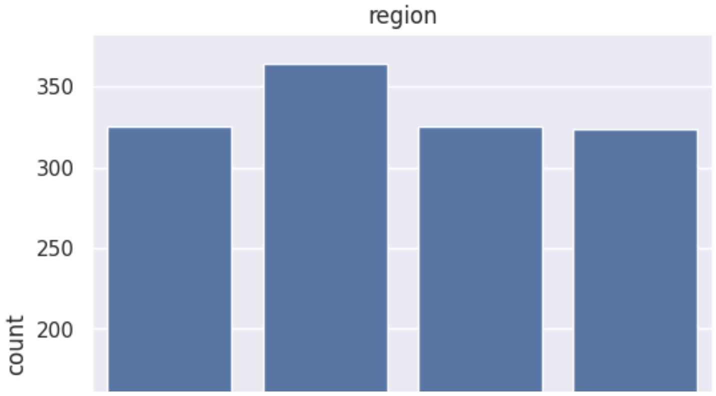
```
Distribution of Children column:
```

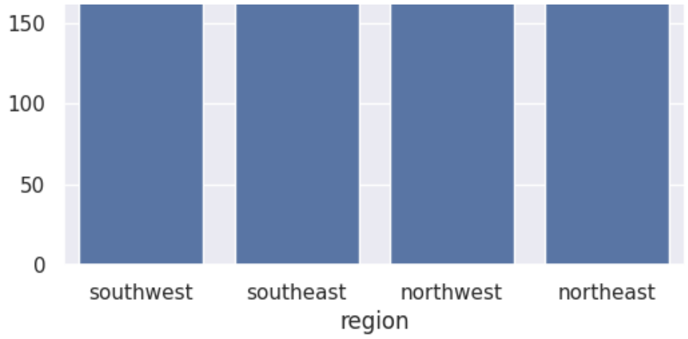


```
Distribution of Smoker column:
```

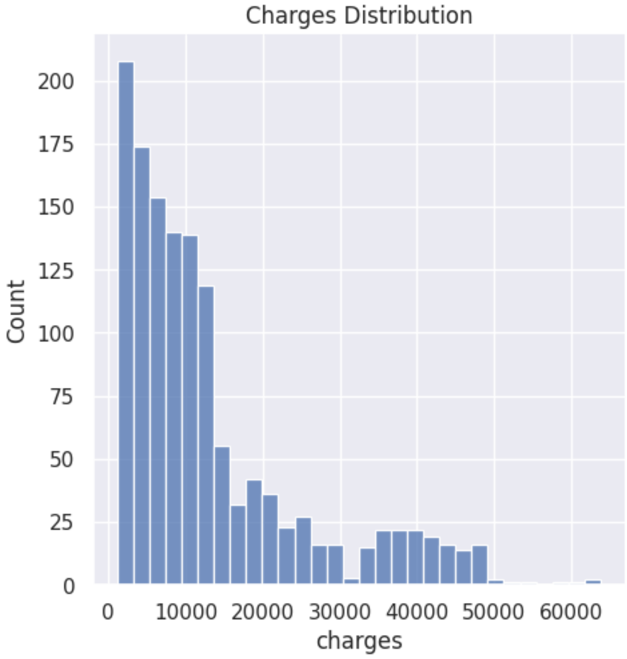


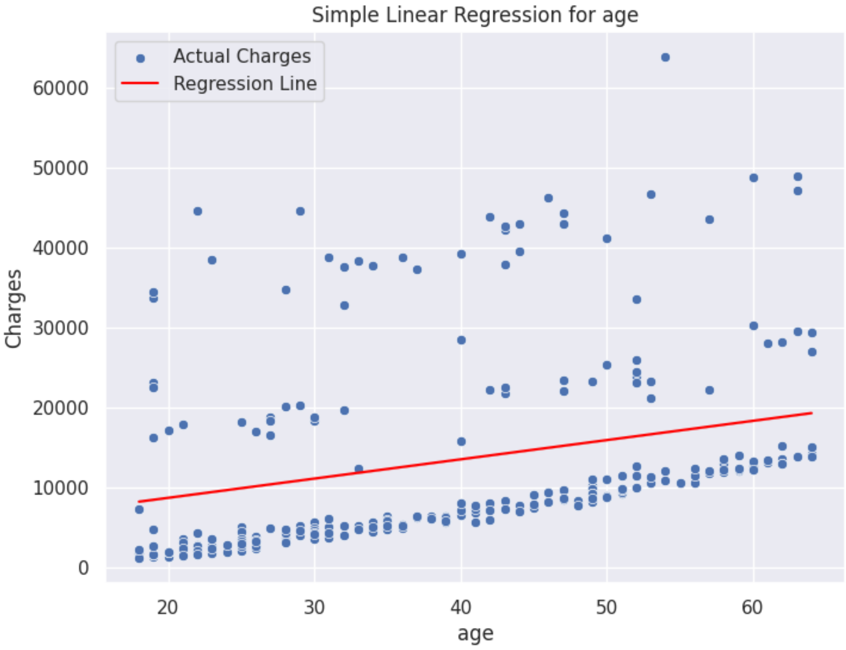
```
Distribution of Region column:
```



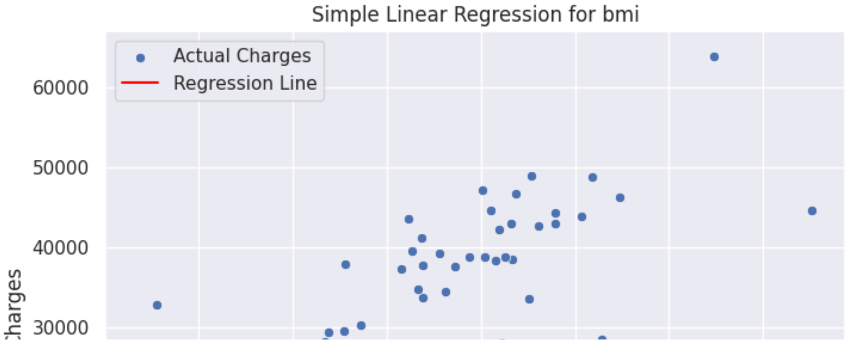


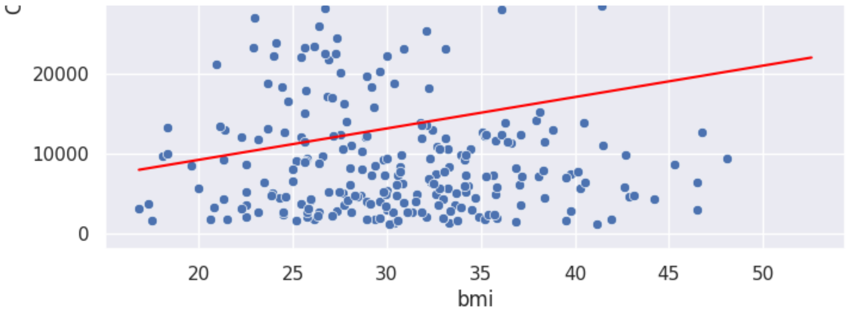
```
Distribution of Charges:
<Figure size 600x600 with 0 Axes>
```





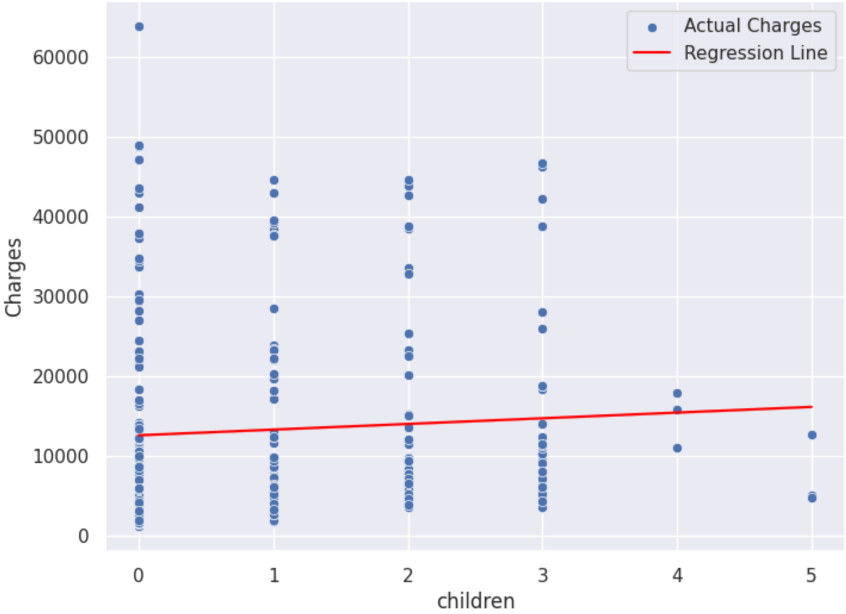
```
Mean Squared Error for age: 135983957.48
```





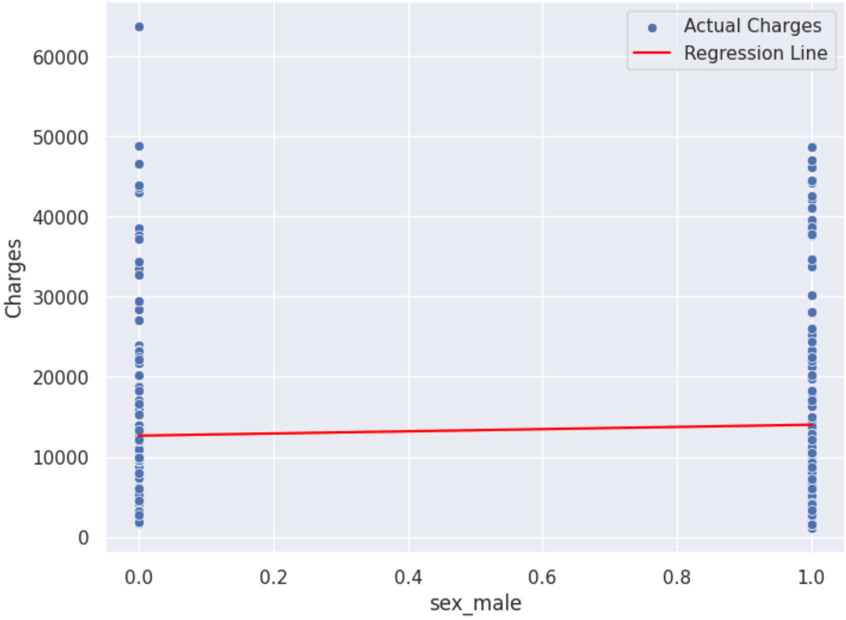
```
Mean Squared Error for bmi: 149085057.04
```

Simple Linear Regression for children



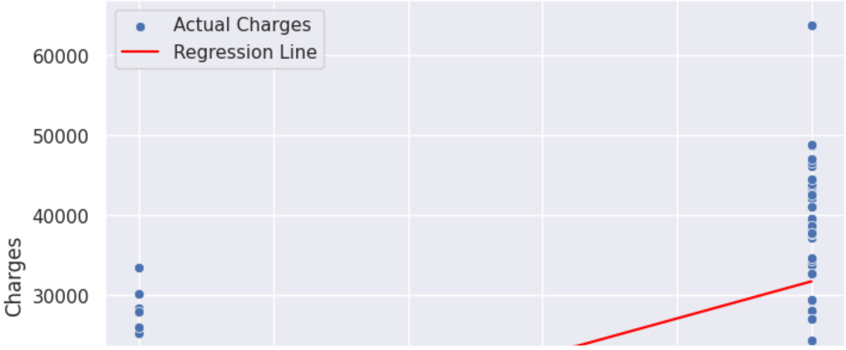
```
Mean Squared Error for children: 154985513.03
```

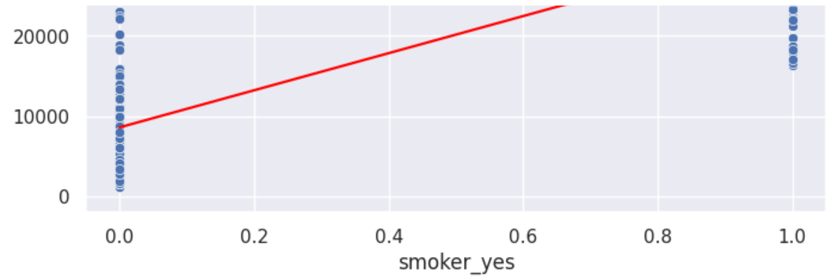
Simple Linear Regression for sex_male



```
Mean Squared Error for sex_male: 154843202.23
```

Simple Linear Regression for smoker_yes





```
Mean Squared Error for smoker_yes: 52745964.73
```

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
from sklearn import metrics

# loading the data from csv file to a Pandas DataFrame
insurance_dataset = pd.read_csv("/content/insurance.csv")
print("Information about the dataset:")
insurance_dataset.info()

#linear regression relation between
# encoding sex column
insurance_dataset.replace({'sex':{'male':0,'female':1}}, inplace=True)

3 # encoding 'smoker' column
insurance_dataset.replace({'smoker':{'yes':0,'no':1}}, inplace=True)

# encoding 'region' column
insurance_dataset.replace({'region':{'southeast':0,'southwest':1,'northeast':2,'northwest':3}}, inplace=True)

#insurance_dataset['bmi_squared'] = insurance_dataset['bmi']**2

X = insurance_dataset.drop(columns='charges', axis=1)
Y = insurance_dataset['charges']

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.3, random_state=2)

#loading the Linear Regression model
regressor = LinearRegression()
regressor.fit(X_train, Y_train)

#Prediction on training data
training_data_prediction =regressor.predict(X_train)
```