Let’s import the dataset and the necessary Python libraries that we need for this task:

In [1]:

```
import numpy as np
import pandas as pd
data = pd.read_csv(r"C:\Users\SHREE\Downloads\Python CODES\Health Insurance Premium Prediction with Machine Learning\Heal
data.head()
```

Out[1]:

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520

Before moving forward, let’s have a look at whether this dataset contains any null values or not:

In [2]:

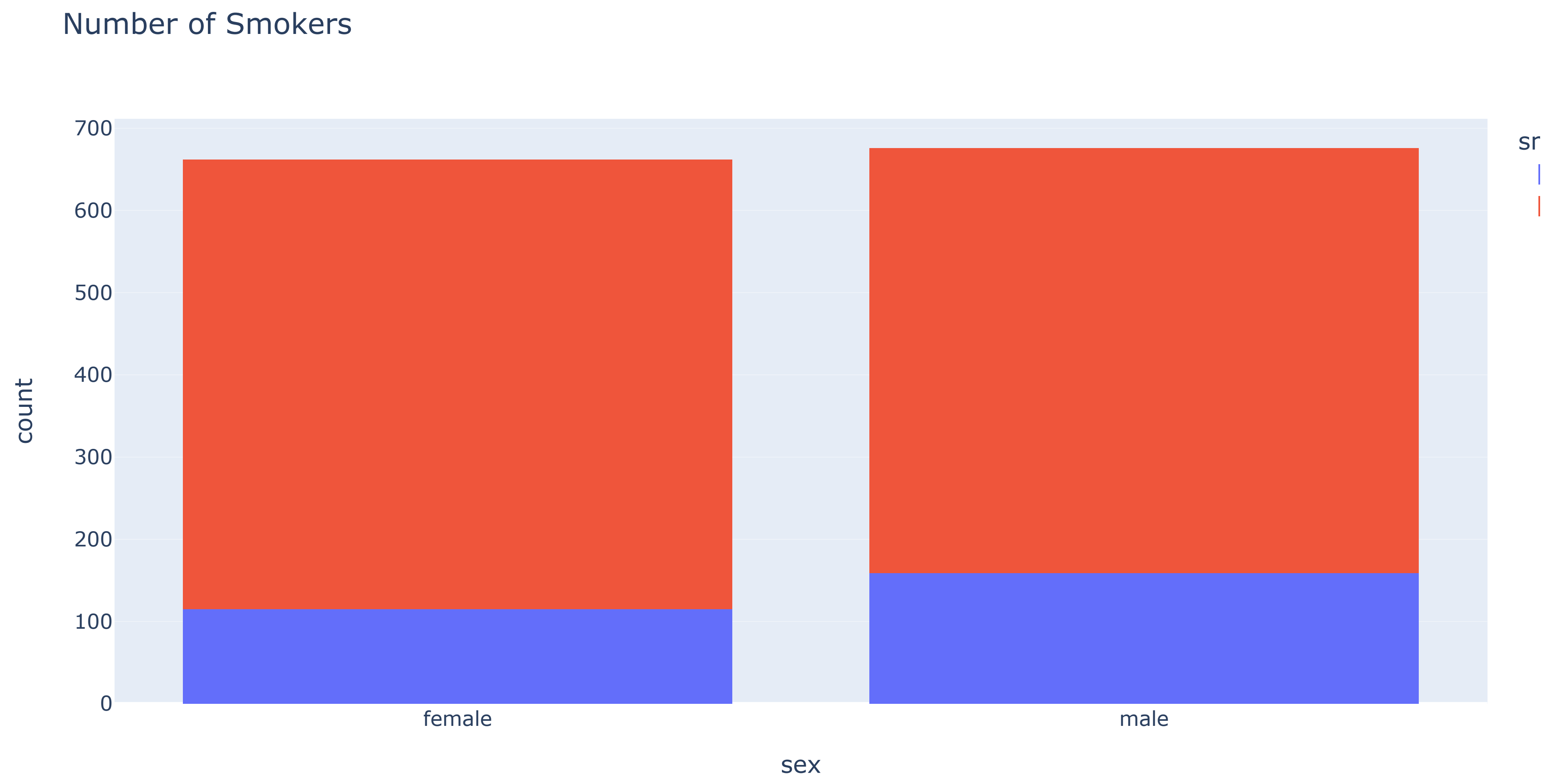
```
data.isnull().sum()
```

Out[2]:

```
age         0
sex         0
bmi         0
children    0
smoker      0
region      0
charges     0
dtype: int64
```

The dataset is therefore ready to be used. After getting the first impressions of this data, I noticed the “smoker” column, which indicates whether the person smokes or not. This is an important feature of this dataset because a person who smokes is more likely to have major health problems compared to a person who does not smoke. So let’s look at the distribution of people who smoke and who do not:

In [3]:
```
import plotly.express as px
data = data
figure = px.histogram(data, x = "sex", color = "smoker", title= "Number of Smokers")
figure.show()
```



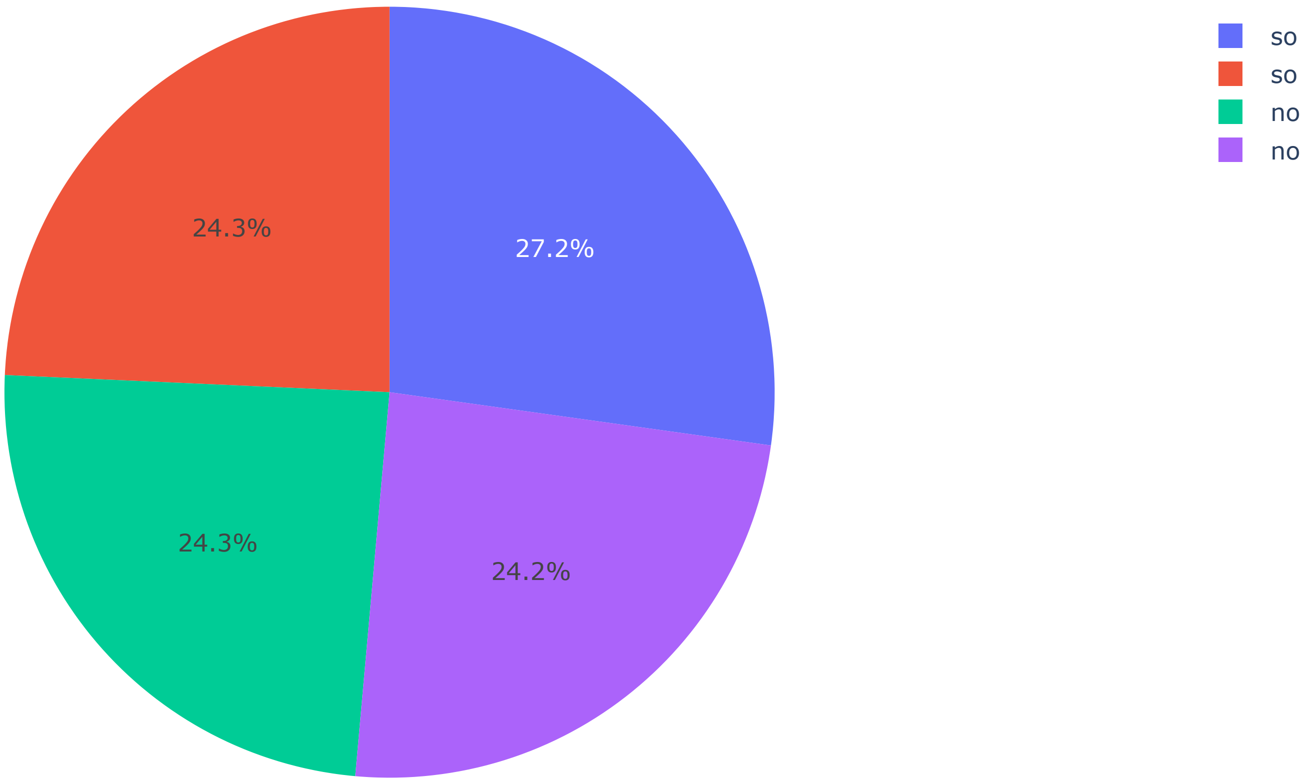
According to the above visualisation, 547 females, 517 males don’t smoke, and 115 females, 159 males do smoke. It is important to use this feature while training a machine learning model, so now I will replace the values of the “sex” and “smoker” columns with 0 and 1 as both these columns contain string values:

```
In [4]: data["sex"] = data["sex"].map({"female": 0, "male": 1})
        data["smoker"] = data["smoker"].map({"no": 0, "yes": 1})
        print(data.head())
```

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

Now let’s have a look at the distribution of the regions where people are living according to the dataset:

```
In [5]: import plotly.express as px
        pie = data["region"].value_counts()
        regions = pie.index
        population = pie.values
        fig = px.pie(data, values=population, names=regions)
        fig.show()
```



Now let’s have a look at the correlation between the features of this dataset:

```
In [6]: print(data.corr())
```

```
               age       sex       bmi  children    smoker   charges
age       1.000000 -0.020856  0.109272  0.042469 -0.025019  0.299008
sex      -0.020856  1.000000  0.046371  0.017163  0.076185  0.057292
bmi       0.109272  0.046371  1.000000  0.012759  0.003750  0.198341
children  0.042469  0.017163  0.012759  1.000000  0.007673  0.067998
smoker   -0.025019  0.076185  0.003750  0.007673  1.000000  0.787251
charges   0.299008  0.057292  0.198341  0.067998  0.787251  1.000000
```

Health Insurance Premium Prediction Model

Now let’s move on to training a machine learning model for the task of predicting health insurance premiums. First, I’ll split the data into training and test sets:

```
In [7]: x = np.array(data[["age", "sex", "bmi", "smoker"]])
        y = np.array(data["charges"])

        from sklearn.model_selection import train_test_split
        xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size=0.2, random_state=42)
```

After using different machine learning algorithms, I found the random forest algorithm as the best performing algorithm for this task. So here I will train the model by using the random forest regression algorithm:

```
In [8]: from sklearn.ensemble import RandomForestRegressor
        forest = RandomForestRegressor()
        forest.fit(xtrain, ytrain)
```

```
Out[8]: ▾ RandomForestRegressor

        RandomForestRegressor()
```

Now let’s have a look at the predicted values of the model:

In [9]:

```
ypred = forest.predict(xtest)
data = pd.DataFrame(data={"Predicted Premium Amount": ypred})
print(data.head())
```

```
   Predicted Premium Amount
0              10195.399276
1               5593.074187
2              28390.352115
3               9643.372430
4              34670.883579
```

myr