In [1]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:

```
data = pd.read_csv("insurance.csv")
```

In [3]:

```
data
```

Out[3]:

	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
...	...	...	...	...	...	...	...
1333	50	male	30.970	3	no	northwest	10600.54830
1334	18	female	31.920	0	no	northeast	2205.98080
1335	18	female	36.850	0	no	southeast	1629.83350
1336	21	female	25.800	0	no	southwest	2007.94500
1337	61	female	29.070	0	yes	northwest	29141.36030

1338 rows × 7 columns

In [4]:

```
data.head()
```

Out[4]:

	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

In [5]:

```
data.tail()
```

Out[5]:

	age	sex	bmi	children	smoker	region	charges
1333	50	male	30.97	3	no	northwest	10600.5483
1334	18	female	31.92	0	no	northeast	2205.9808
1335	18	female	36.85	0	no	southeast	1629.8335
1336	21	female	25.80	0	no	southwest	2007.9450
1337	61	female	29.07	0	yes	northwest	29141.3603

In [6]:

```
data.shape
```

Out[6]:

```
(1338, 7)
```

In [7]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1338 entries, 0 to 1337
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   age       1338 non-null   int64  
 1   sex       1338 non-null   object 
 2   bmi       1338 non-null   float64
 3   children  1338 non-null   int64  
 4   smoker    1338 non-null   object 
 5   region    1338 non-null   object 
 6   charges   1338 non-null   float64
dtypes: float64(2), int64(2), object(3)
memory usage: 73.3+ KB
```

In [8]:

```
data.describe()
```

Out[8]:

	age	bmi	children	charges
count	1338.000000	1338.000000	1338.000000	1338.000000
mean	39.207025	30.663397	1.094918	13270.422265
std	14.049960	6.098187	1.205493	12110.011237
min	18.000000	15.960000	0.000000	1121.873900
25%	27.000000	26.296250	0.000000	4740.287150
50%	39.000000	30.400000	1.000000	9382.033000
75%	51.000000	34.693750	2.000000	16639.912515
max	64.000000	53.130000	5.000000	63770.428010

In [9]:

```
data.describe().columns
```

Out[9]:

```
Index(['age', 'bmi', 'children', 'charges'], dtype='object')
```

In [10]:

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

Out[10]:

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

In [11]:

```
import warnings
warnings.filterwarnings("ignore")
```

In [12]:

```
data['sex'].value_counts()
```

Out[12]:

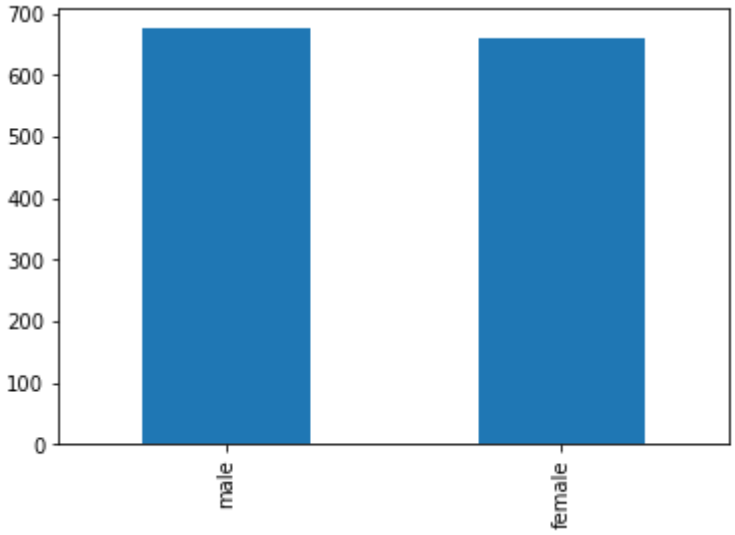
```
male      676
female    662
Name: sex, dtype: int64
```

In [13]:

```
data['sex'].value_counts().sort_index(ascending=False).plot(kind='bar')
```

Out[13]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2756037e20>
```



In [14]:

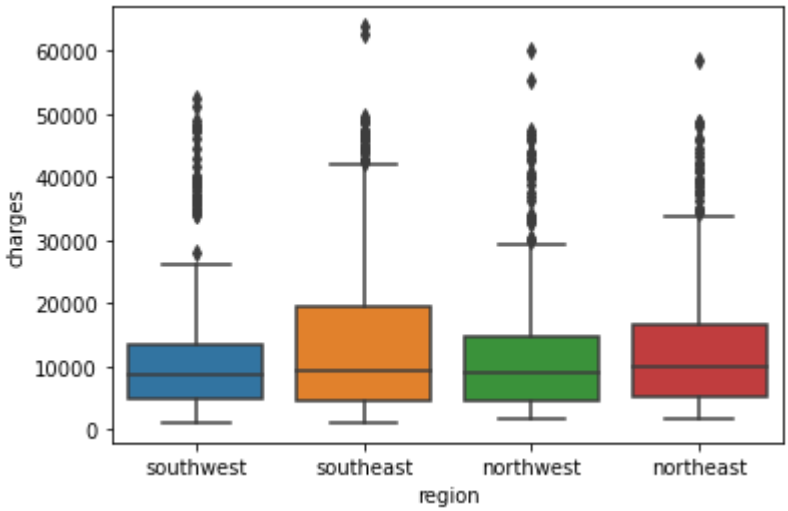
```
factors=['age', 'sex', 'bmi', 'children', 'charges']
data.groupby('sex')[factors].mean()
```

Out[14]:

sex	age	bmi	children	charges
female	39.503021	30.377749	1.074018	12569.578844
male	38.917160	30.943129	1.115385	13956.751178

In [15]:

```
sns.boxplot(x='region', y='charges', data=data);
```



In [19]:

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
```

In [21]:

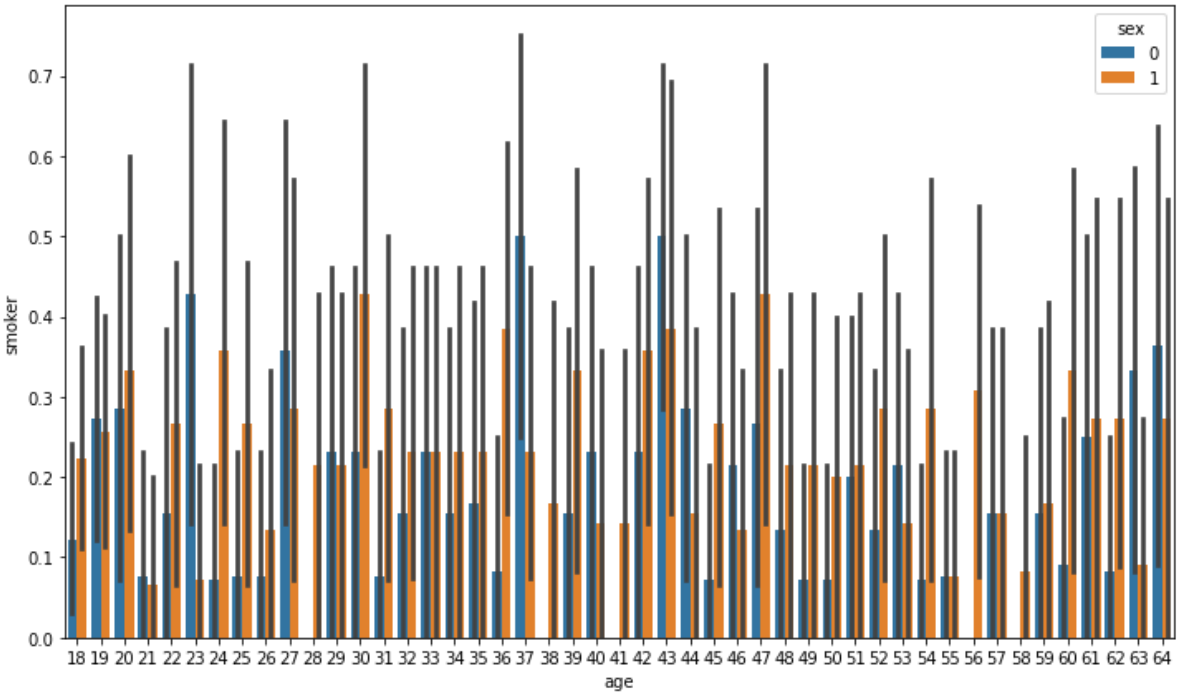
```
data.sex = le.fit_transform(data.sex)
data.smoker = le.fit_transform(data.smoker)
data.region = le.fit_transform(data.region)
```

In [24]:

```
plt.figure(figsize = (12,7))
sns.barplot(x = data.age, y = data.smoker, hue = data.sex)
```

Out[24]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2756c14100>
```

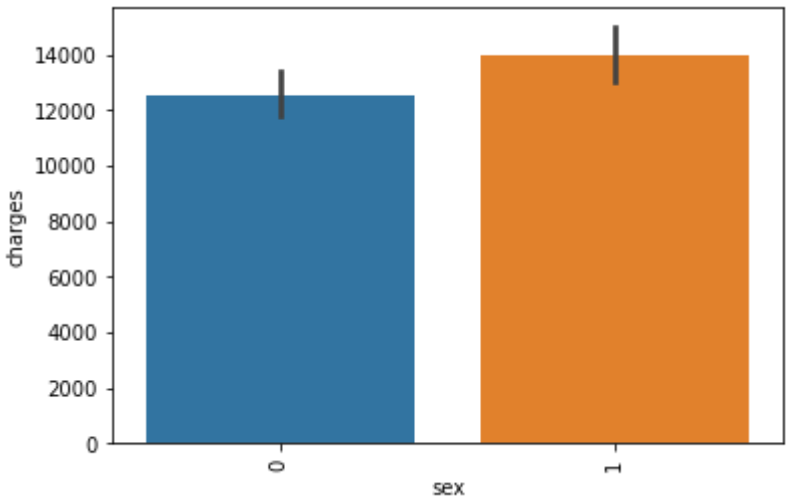


In [25]:

```
plt.xticks(rotation = 90)
sns.barplot(x = data.sex, y = data.charges)
```

Out[25]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x276bcf98b0>
```

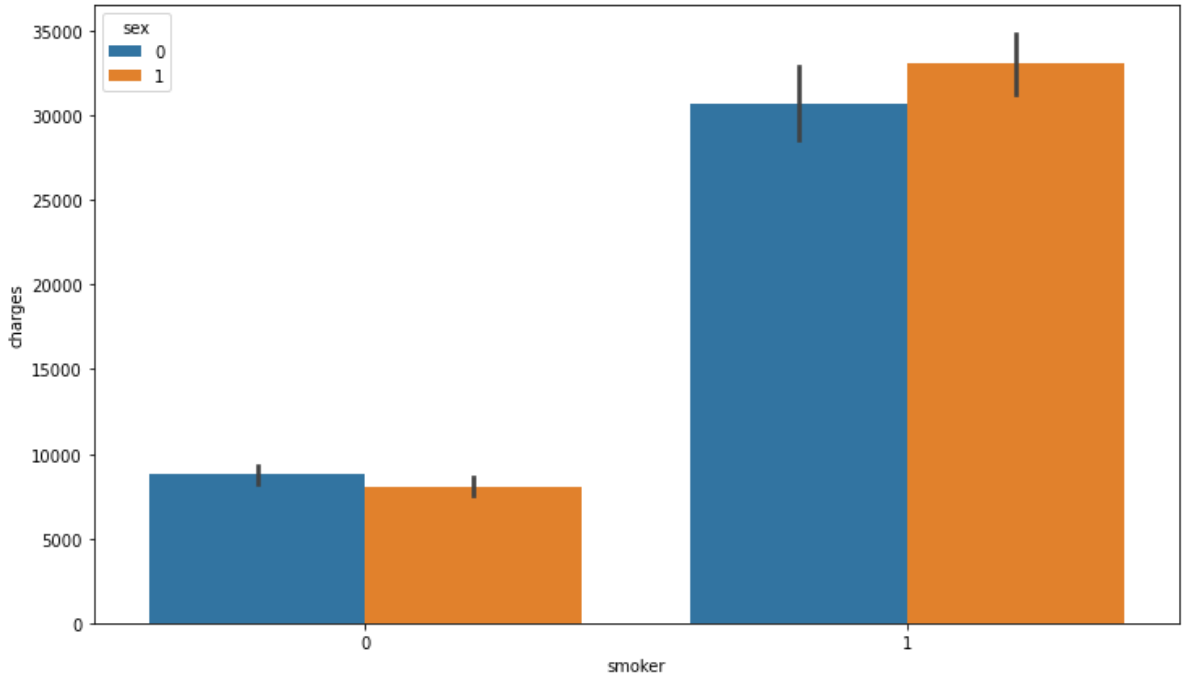


In [27]:

```
plt.figure(figsize = (12,7))
sns.barplot(x = data.smoker, y = data.charges, hue = data.sex)
```

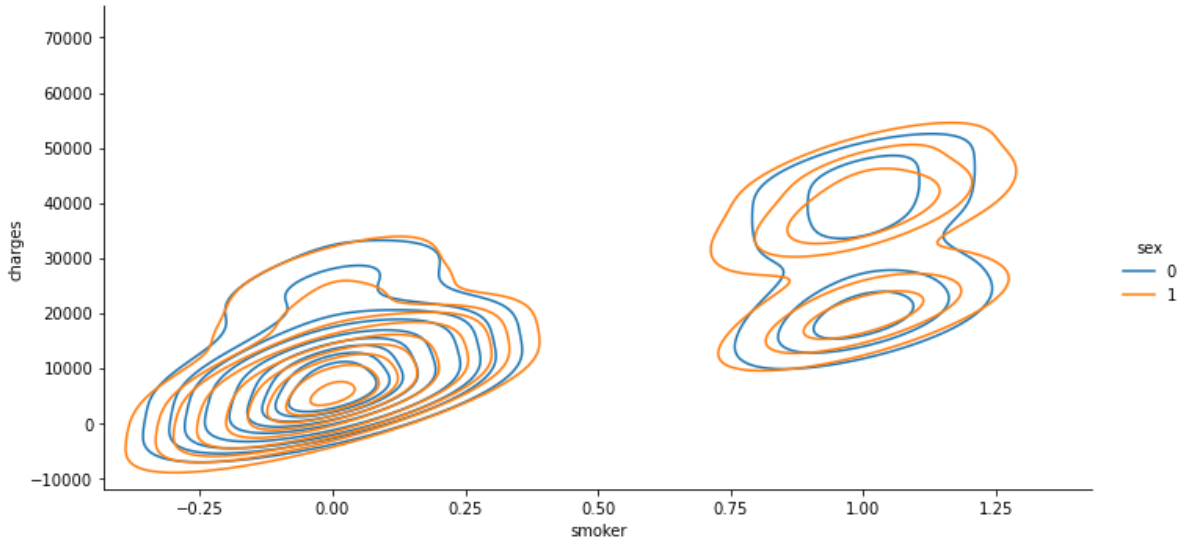
Out[27]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x276b72acd0>
```



In [28]:

```
sns.displot(kind='kde', x='smoker', y = 'charges', hue='sex', data=data, aspect=2);
```

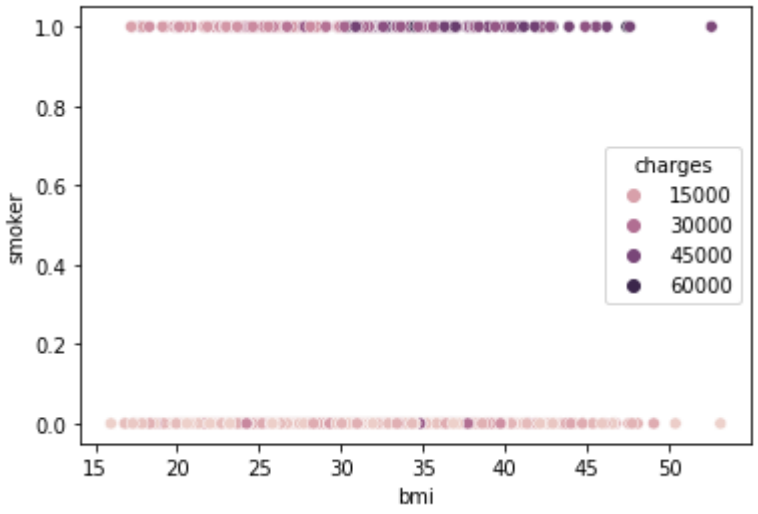


In [32]:

```
sns.scatterplot(x="bmi", y="smoker", hue = 'charges', data=data)
```

Out[32]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x276c6a2400>
```



In [33]:

```
data_factors = data[['age', 'sex', 'bmi', 'children', 'smoker', 'region', 'charges']]
```

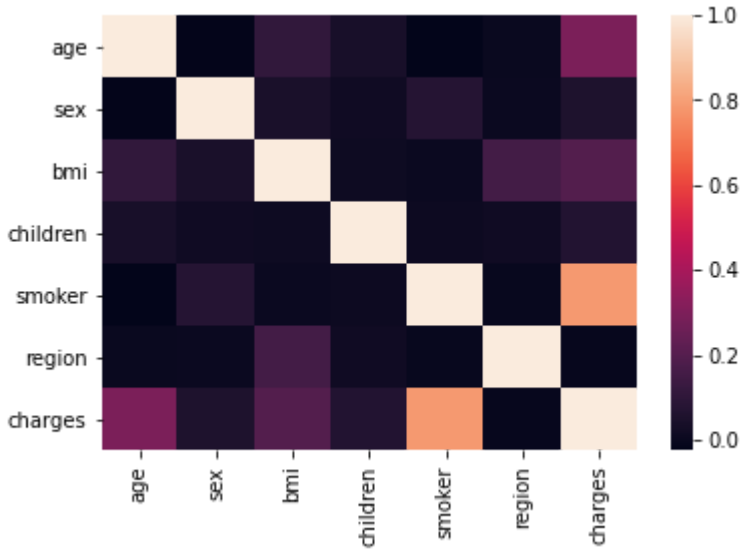
In [34]:

```
print(data_factors.corr())
sns.heatmap(data_factors.corr())
```

```
                age       sex       bmi  children    smoker    region    cha
rges
age        1.000000 -0.020856  0.109272  0.042469 -0.025019  0.002127  0.29
9008
sex       -0.020856  1.000000  0.046371  0.017163  0.076185  0.004588  0.05
7292
bmi        0.109272  0.046371  1.000000  0.012759  0.003750  0.157566  0.19
8341
children   0.042469  0.017163  0.012759  1.000000  0.007673  0.016569  0.06
7998
smoker    -0.025019  0.076185  0.003750  0.007673  1.000000 -0.002181  0.78
7251
region     0.002127  0.004588  0.157566  0.016569 -0.002181  1.000000 -0.00
6208
charges    0.299008  0.057292  0.198341  0.067998  0.787251 -0.006208  1.00
0000
```

Out[34]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x275613e6d0>
```



In [36]:

```
x = data.drop(['charges'], axis =1)
```

In [37]:

```
y = data.charges
```

In [38]:

```
x.shape
```

Out[38]:

```
(1338, 6)
```

In [39]:

```
y.shape
```

Out[39]:

```
(1338,)
```

In [40]:

```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size = 0.2)
```

In [42]:

```
model= LinearRegression()
model.fit(X_train, y_train)
```

Out[42]:

```
LinearRegression()
```

In [43]:

```
y_pred = model.predict(X_test)
```

In [44]:

```
print("Training Accuracy :", model.score(X_train, y_train))
print("Testing Accuracy :", model.score(X_test, y_test))
```

```
Training Accuracy : 0.7412934137769827
Testing Accuracy : 0.7866891777516701
```

In [50]:

```
from sklearn.tree import DecisionTreeRegressor
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size = 0.2)
```

In [53]:

```
model1 = DecisionTreeRegressor()
model1.fit(X_train, y_train)
```

Out[53]:

```
DecisionTreeRegressor()
```

In [54]:

```
y_pred = model1.predict(X_test)
```

In [55]:

```
print("Training Accuracy :", model1.score(X_train, y_train))
print("Testing Accuracy :", model1.score(X_test, y_test))
```

```
Training Accuracy : 0.999510039812348
Testing Accuracy : 0.6727314220123752
```

In [56]:

```
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size = 0.2)
```

In [57]:

```
model2 = DecisionTreeRegressor()
model2.fit(X_train, y_train)
```

Out[57]:

```
DecisionTreeRegressor()
```

In [58]:

```
y_pred = model2.predict(X_test)
```

In [59]:

```
print("Training Accuracy :", model2.score(X_train, y_train))
print("Testing Accuracy :", model2.score(X_test, y_test))
```

```
Training Accuracy : 0.9983206672851662
Testing Accuracy : 0.7331681300379966
```