Statistics Assessment
Ishita Sarkar

1. Import the necessary libraries

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats as stats
from scipy.stats import skew
from scipy.stats import chisquare,chi2_contingency
from scipy.stats import ttest_ind
from scipy.stats import f_oneway
import copy
```

0

The necessary libraries needed for the analysis are numpy which is used for working with arrays, pandas is used for analyzing data, matplotlib to plot various graphs, seaborn for boxplot, next is scipy.stats all of the statistics functions are located in the sub-package scipy.stats, last but not the least copy which is used to copy dataframe into new variables.

---

2. Read the data as a data frame

```
insurance =
pd.read_csv(r"C:\Users\ISHITA\Desktop\GreatLearning\statistics\insurance.cs
v")
insurance
```

Out[47]:

	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

3

1338 rows × 7 columns

Statistics Assessment
Ishita Sarkar

3. Perform basic EDA which should include the following and print out your insights at every step.
a. Shape of the data b. Data type of each attribute c. Checking the presence of missing values d. 5-point summary of numerical attributes e. Distribution of ‘bmi’, ‘age’ and ‘charges’ columns. f. Measure of skewness of ‘bmi’, ‘age’ and ‘charges’ columns g. Checking the presence of outliers in ‘bmi’, ‘age’ and ‘charges columns h. Distribution of categorical columns (include children) i. Pair plot that includes all the columns of the data frame

a. Shape of the data

```
In [48]:  insurance.shape
Out[48]:  (1338, 7)
```

The shape function is used to obtain the shape of the dataframe such as the number of rows and columns.

b. Data type of each attribute

```
In [49]:  insurance.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
```

2
7

```
In [50]:  insurance.dtypes
Out[50]:  age           int64
          sex          object
          bmi         float64
          children      int64
          smoker       object
          region       object
          charges     float64
          dtype: object
```

There are 2 ways one can check for the data type of each attribute, one by using ‘.info()’ function and the other by using the ‘.dtypes’.

Statistics Assessment

Ishita Sarkar

c. Checking the presence of missing values

```
In [51]: insurance.isnull().sum()

Out[51]: age         0
         sex         0
         bmi         0
         children    0
         smoker      0
         region      0
         charges     0
         dtype: int64
```

```
In [52]: insurance.isna()
```

Out[52]:

	age	sex	bmi	children	smoker	region	charges
0	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...
1333	False	False	False	False	False	False	False
1334	False	False	False	False	False	False	False
1335	False	False	False	False	False	False	False
1336	False	False	False	False	False	False	False
1337	False	False	False	False	False	False	False

1338 rows × 7 columns

```
In [53]: insurance.isnull().values.any()

Out[53]: False
```

So, there are 3 ways to check for null values in dataframe first using '.isnull' function which returns an overall 'True' or 'False' value for each column, second '.isna()' which gives a 'true' or 'false' value for each value, lastly it's 'isnull().values.any()'.

d. 5-point summary of numerical attributes

```
In [54]: insurance.describe()
```

Out[54]:

	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

The describe function is used to return the overall description of the daraframe.

Statistics Assessment
Ishita Sarkar

e. Distribution of 'bmi', 'age' and 'charges' columns.

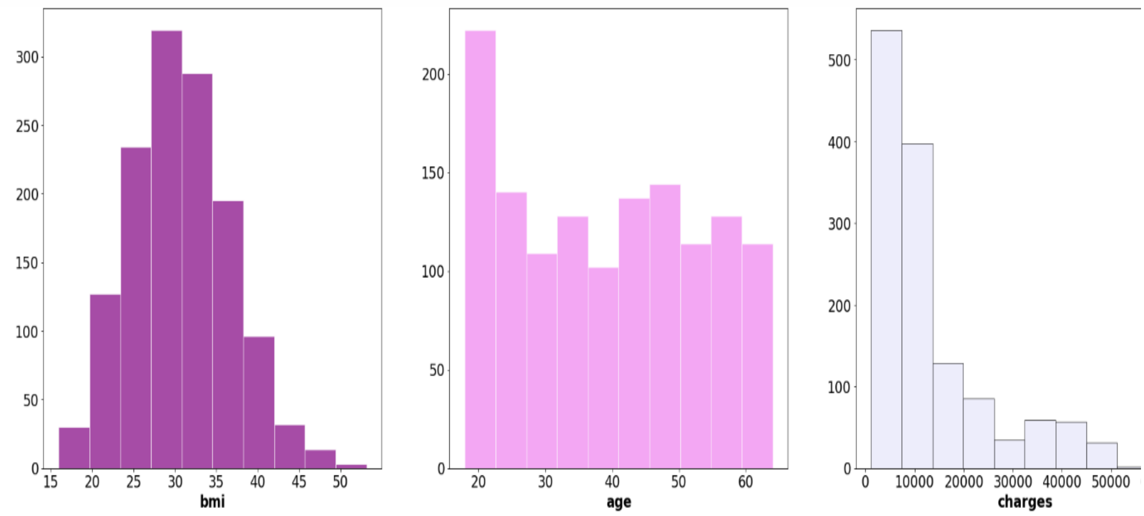
In [55]:
```
plt.figure(figsize= (45,45))

plt.subplot(4,4,1)
plt.hist(insurance.bmi, color='purple', edgecolor = 'white', alpha = 0.7)
plt.xlabel('bmi',fontsize=20,fontweight = 'bold')
plt.xticks(size = 20)
plt.yticks(size = 20)

plt.subplot(4,4,2)
plt.hist(insurance.age, color='violet', edgecolor = 'white', alpha = 0.7)
plt.xlabel('age',fontsize=20,fontweight = 'bold')
plt.xticks(size = 20)
plt.yticks(size = 20)

plt.subplot(4,4,3)
plt.hist(insurance.charges, color='lavender', edgecolor = 'black', alpha = 0.7)
plt.xlabel('charges',fontsize=20,fontweight = 'bold')
plt.xticks(size = 20)
plt.yticks(size = 20)

plt.show()
```



Little Right Skewed   Normal Distribution   Right Skewed

f. Measure of skewness of 'bmi', 'age' and 'charges' columns

In [69]:
```
Skewness = pd.DataFrame({'Skewness' : [skew(insurance.bmi),skew(insurance.age),skew(insurance.charges)]},index=['bmi','
Skewness
```

Out[69]:

	Skewness
bmi	0.283729
age	0.055610
charges	1.514180

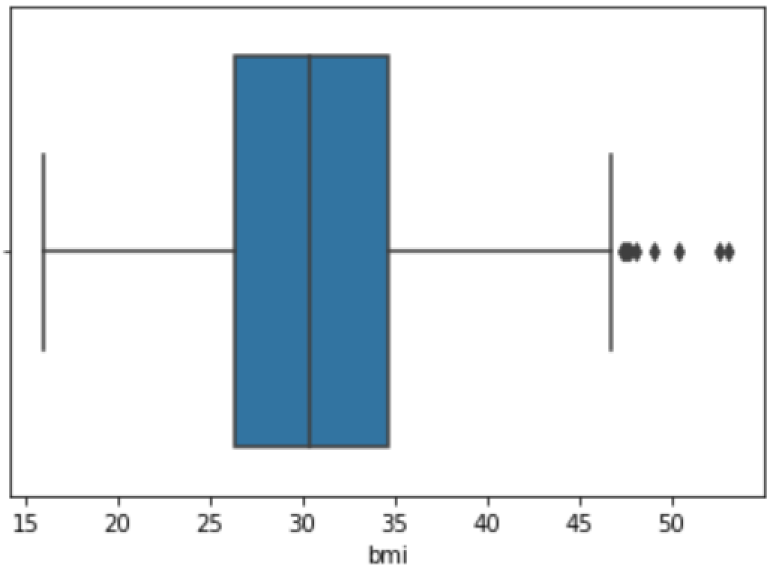
As we observed above after calculating the skewness we can state that 'bmi' is slightly skewed, 'age' is normally distributed but 'charges' are 'right skewed'.

Statistics Assessment
Ishita Sarkar

g. Checking the presence of outliers in ‘bmi’, ‘age’ and ‘charges columns

In [105]:
```
#bmi
q25=insurance['bmi'].quantile(0.25)
q75=insurance['bmi'].quantile(0.75)
IQR=q75-q25
cut_off = IQR * 1.5
low= q25 - cut_off
up=q75 + cut_off
outliers = [x for x in insurance['bmi'] if x < low or x > up]
len(outliers),outliers
```
Out[105]:
```
(9, [49.06, 48.07, 47.52, 47.41, 50.38, 47.6, 52.58, 47.74, 53.13])
```

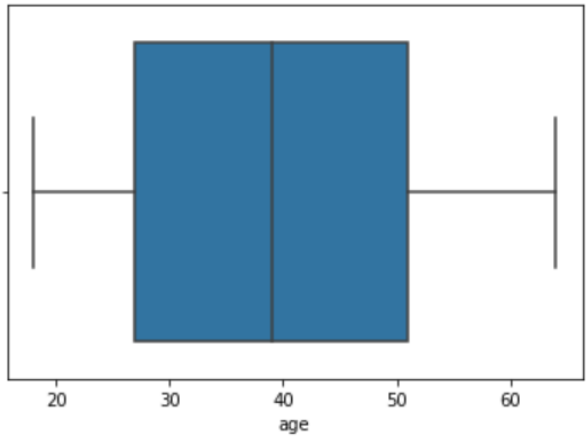
In [98]:
```
sns.boxplot(x='bmi',data =insurance)
```
Out[98]:
```
<AxesSubplot:xlabel='bmi'>
```



In [110]:
```
#age
q25=insurance['age'].quantile(0.25)
q75=insurance['age'].quantile(0.75)
IQR=q75-q25
cut_off = IQR * 1.5
low= q25 - cut_off
up=q75 + cut_off
outliers = [x for x in insurance['age'] if x < low or x > up]
len(outliers),outliers
```
Out[110]:
```
(0, [])
```

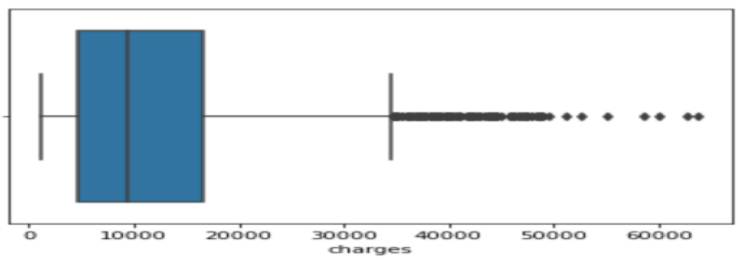
Statistics Assessment
Ishita Sarkar

In [107]:
```
sns.boxplot(x='age',data =insurance)
```
Out[107]: <AxesSubplot:xlabel='age'>



In [116]:
```
#charges
q25=insurance['charges'].quantile(0.25)
q75=insurance['charges'].quantile(0.75)
IQR=q75-q25
cut_off = IQR * 1.5
low= q25 - cut_off
up=q75 + cut_off
Df=insurance
outliers = [x for x in insurance['charges'] if x < low or x > up]
len(outliers),outliers
```
Out[116]:
```
(139,
 [39611.7577,
  36837.467,
  37701.8768,
  38711.0,
  35585.576,
  51194.55914,
  39774.2763,
  48173.361,
  38709.176,
  37742.5757,
  47496.49445,
```

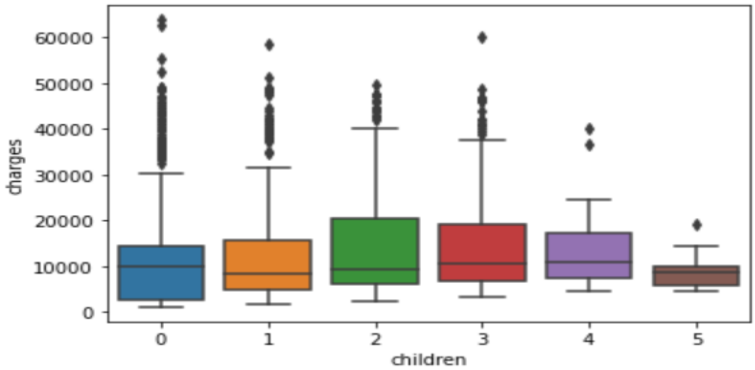
In [97]:
```
sns.boxplot(x='charges',data =insurance)
```
Out[97]: <AxesSubplot:xlabel='charges'>



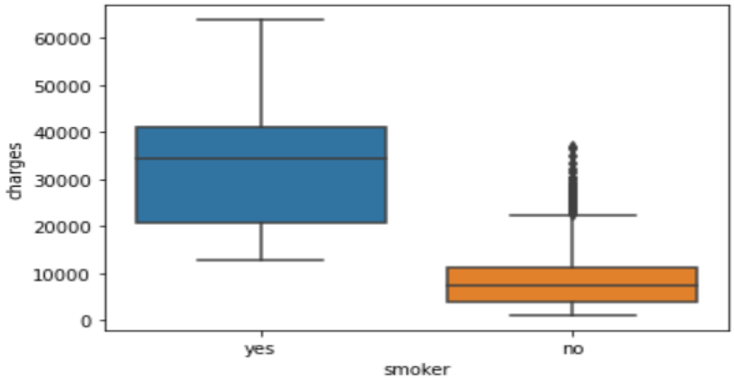
Statistics Assessment
Ishita Sarkar

h. Distribution of categorical columns (include children)

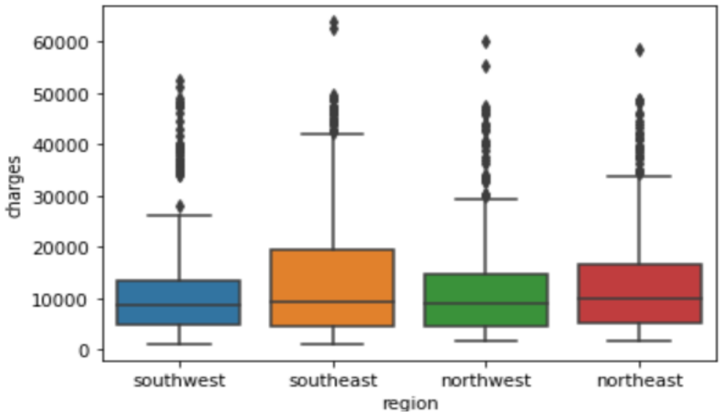
In [119]:
```
sns.boxplot(x='children', y='charges', data= insurance)
```
Out[119]: <AxesSubplot:xlabel='children', ylabel='charges'>



In [120]:
```
sns.boxplot(x='smoker', y='charges', data= insurance)
```
Out[120]: <AxesSubplot:xlabel='smoker', ylabel='charges'>

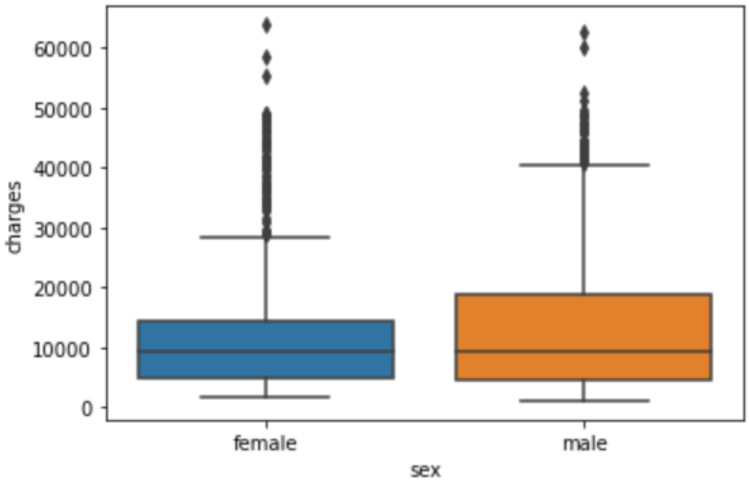


In [122]:
```
sns.boxplot(x='region', y='charges', data= insurance)
```
Out[122]: <AxesSubplot:xlabel='region', ylabel='charges'>



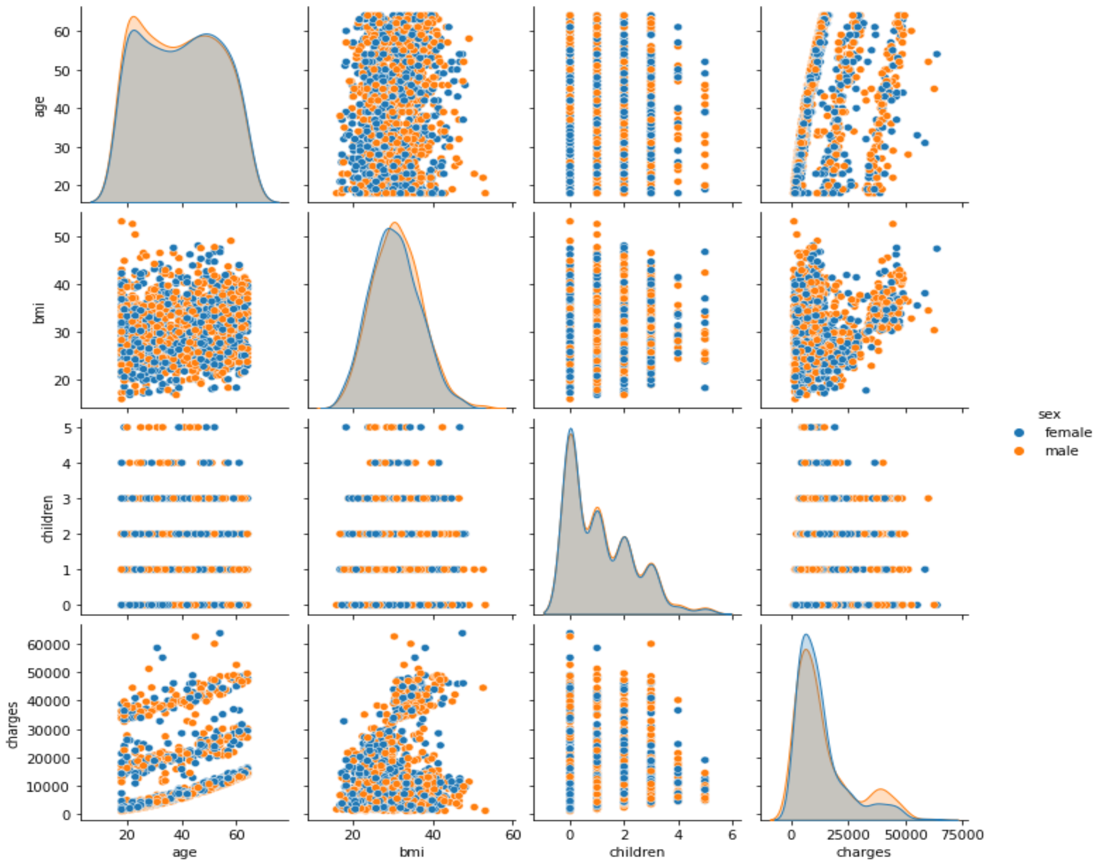
Statistics Assessment
Ishita Sarkar

In [123]:
```
sns.boxplot(x='sex', y='charges', data= insurance)
```
Out[123]: <AxesSubplot:xlabel='sex', ylabel='charges'>



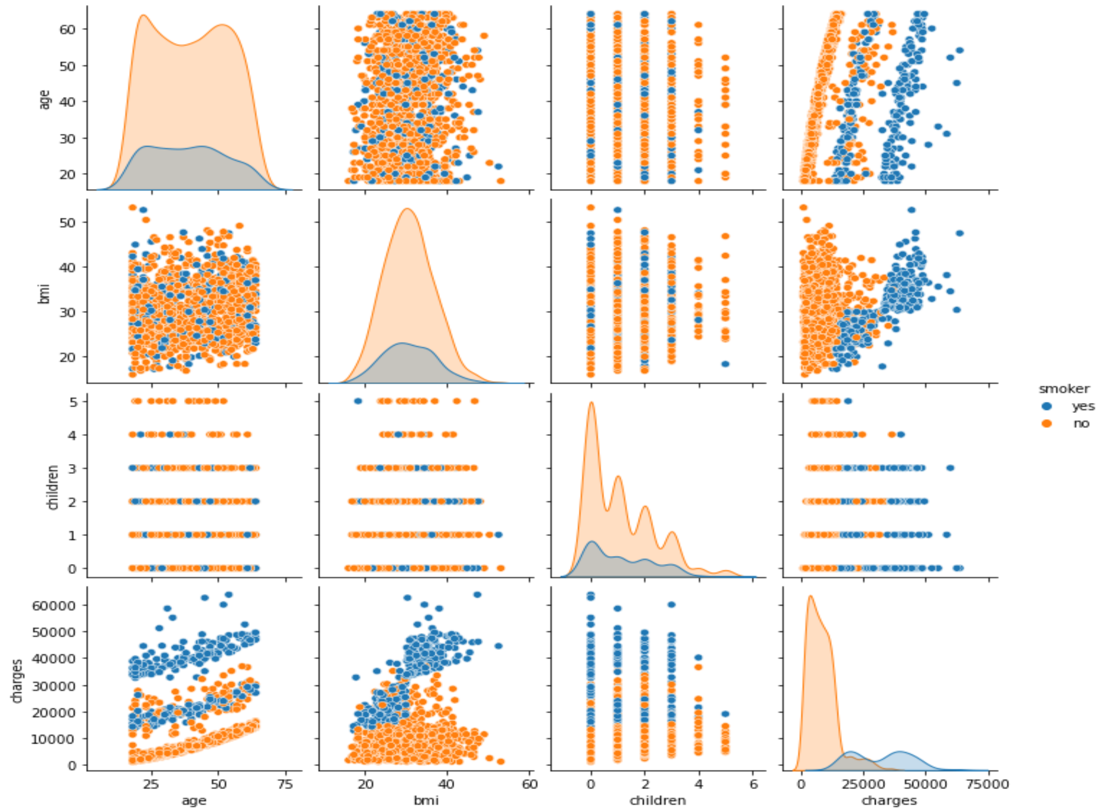
i. Pair plot that includes all the columns of the data frame

In [124]:
```
sns.pairplot(insurance,hue='sex')
```

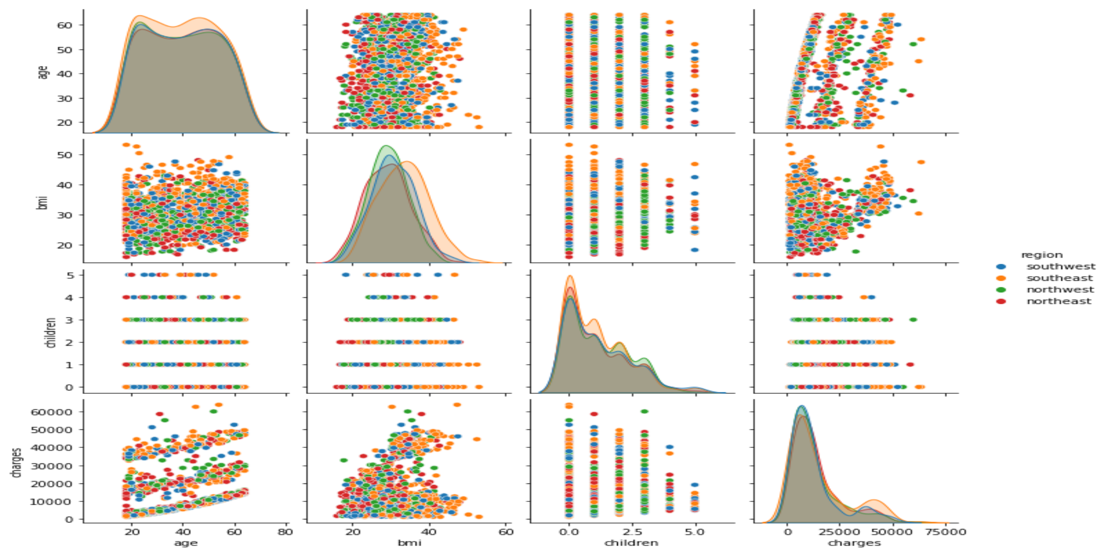


Statistics Assessment
Ishita Sarkar

```
In [125]:  sns.pairplot(insurance,hue='smoker')
```




```
In [126]:  sns.pairplot(insurance,hue='region')
```



Statistics Assessment
Ishita Sarkar

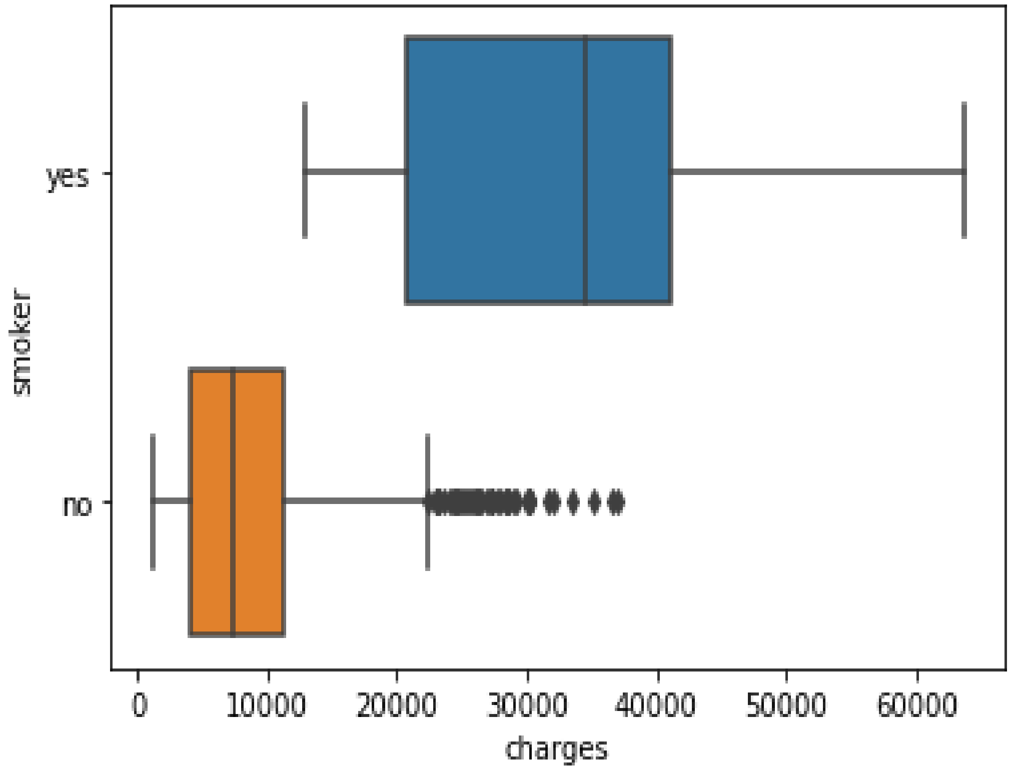
4. Answer the following questions with statistical evidence
a) Do charges of people who smoke differ significantly from the people who don't? b) Does bmi of males differ significantly from that of females? c) Is the proportion of smokers significantly different in different genders? d) Is the distribution of bmi across women with no children, one child and two children, the same?

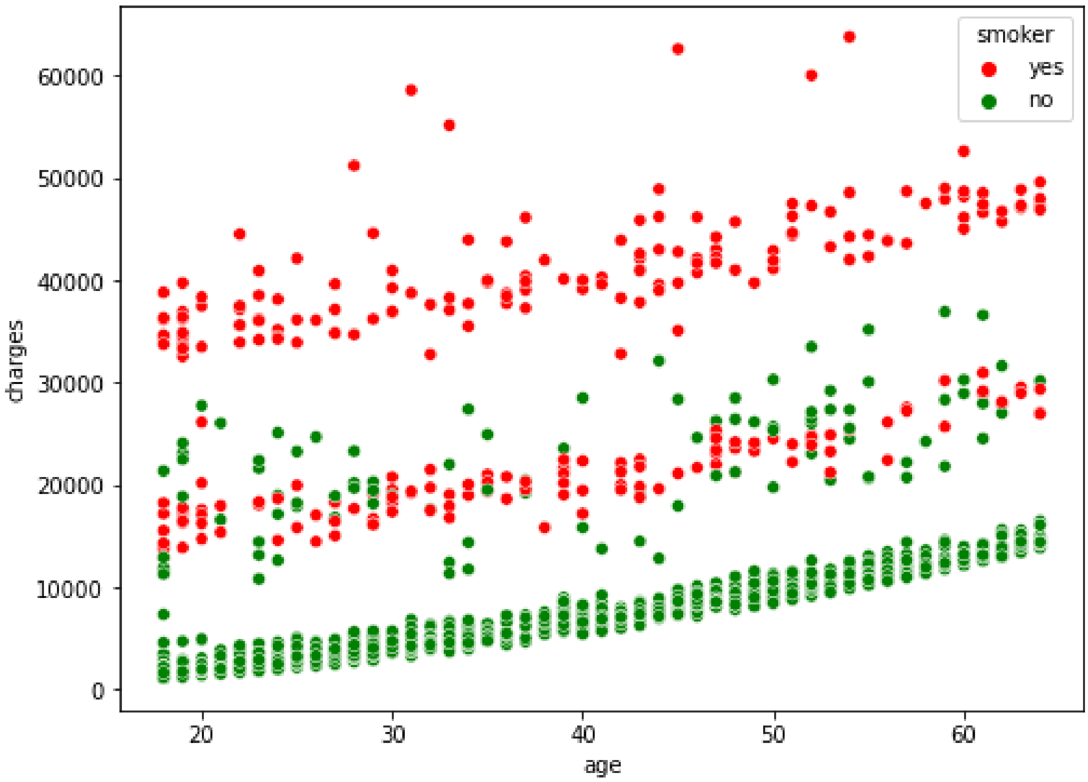
a) Do charges of people who smoke differ significantly from the people who don't?

In [128]:
```
smoker = insurance[insurance['smoker'] == 'yes']
print('smokers = ', len(smoker))
nonsmoker = insurance[insurance['smoker'] == 'no']
print('Non-smokers =', len(nonsmoker))
print("mean value for smokers = ", smoker['charges'].mean())
print("mean value for non-smokers = ",nonsmoker['charges'].mean())
sns.boxplot(x="charges", y="smoker", data=insurance)
```

```
smokers =  274
Non-smokers = 1064
mean value for smokers =  32050.23183153285
mean value for non-smokers =  8434.268297856199
```

Statistics Assessment
Ishita Sarkar





Statistics Assessment
Ishita Sarkar

From the above analysis it is clearly observed that the people who are smokers or smoke are charged comparatively higher than compared to a non-smoker.

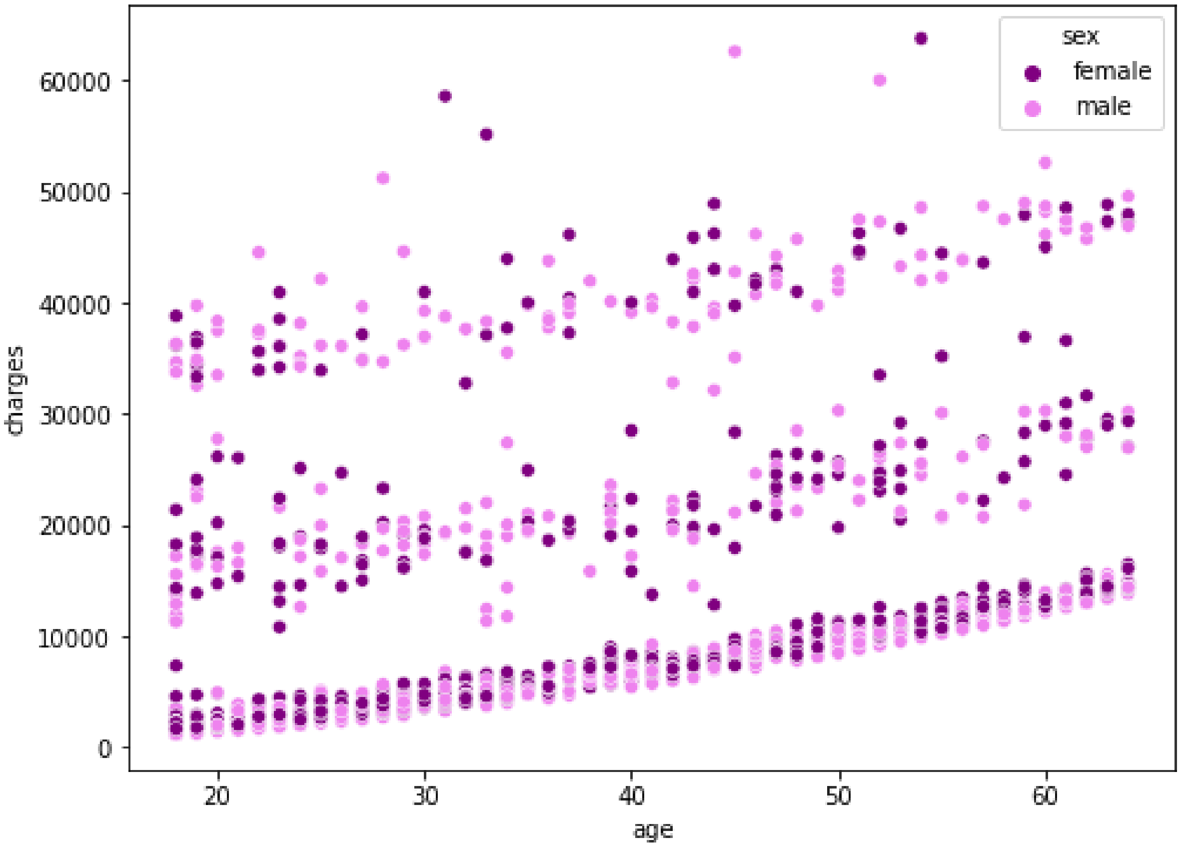
b) Does bmi of males differ significantly from that of females?

In [146]:

```
female = insurance[insurance['sex'] == 'female']
male = insurance[insurance['sex'] == 'male']
print('no. of male =', len(male))
print('no. of female =', len(female))
print("average bmi for male =", male['bmi'].mean())
print("average bmi for female =", female['bmi'].mean())
stats, p_value = ttest_ind(male['bmi'], female['bmi'],axis =0)
print("Tstatistic and Pvalue", stats, p_value)
```

```
no. of male = 676
no. of female = 662
average bmi for male = 30.943128698224832
average bmi for female = 30.377749244713023
Tstatistic and Pvalue 1.696752635752224 0.08997637178984932
```

Statistics Assessment
Ishita Sarkar



It's observed that Gender/sex has no impact on the 'bmi' value as the pvalue is greater than 0.05.

c) Is the proportion of smokers significantly different in different genders?

In [135]:
```
crosstab = pd.crosstab(insurance['sex'], insurance['smoker'])
crosstab
chi2_contingency(crosstab)
```

Out[135]:
```
(7.39291081459996,
 0.006548143503580696,
 1,
 array([[526.43348281, 135.56651719],
        [537.56651719, 138.43348281]]))
```

It is seen that the the proportion of smokers is different with respect to gender, as the pvalue is less than 0.05.

d) Is the distribution of bmi across women with no children, one child and two children, the same?

In [145]:
```
female_df = copy.deepcopy(insurance[insurance['sex'] == 'female'])
z=female_df[female_df.children == 0]['bmi']
o=female_df[female_df.children == 1]['bmi']
t=female_df[female_df.children == 2]['bmi']
fstat, pvalue = stats.f_oneway(z,o,t)
print(pvalue)
```

Statistics Assessment
Ishita Sarkar

Since all the mean values are same, the pvalue will be greater than 0.05, which refers to null hypothesis. Therefore the distribution of 'bmi' values across women with no children, one child and 2 children is same.