In [8]:

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

In [9]:

```
df1 = pd.read_csv('insurance.csv')
```

In [10]:

```
df1
```

Out[10]:

	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 [11]:

```
df1.head()
```

Out[11]:

	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 [12]:
```
df1.tail()
```

Out[12]:

	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 [13]:
```
df1.shape
```

Out[13]:
```
(1338, 7)
```

In [14]:
```
df1.describe()
```

Out[14]:

	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 [15]:
```
df1.describe(include = object)
```

Out[15]:

	sex	smoker	region
count	1338	1338	1338
unique	2	2	4
top	male	no	southeast
freq	676	1064	364

In [17]:
```
df1.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 [18]:
```
df1.sample()
```

Out[18]:

	age	sex	bmi	children	smoker	region	charges
695	26	female	40.185	0	no	northwest	3201.24515

In [22]:
```
df1.columns
```

Out[22]:
```
Index(['age', 'sex', 'bmi', 'children', 'smoker', 'region', 'charg
es'], dtype='object')
```

In [23]:
```
df1.isnull().sum()
```

Out[23]:
```
age         0
sex         0
bmi         0
children    0
smoker      0
region      0
charges     0
dtype: int64
```

In [24]:
```
df1.nunique()
```

Out[24]:
```
age           47
sex            2
bmi          548
children       6
smoker         2
region         4
charges     1337
dtype: int64
```

In [25]:
```
df1['age'].unique()
```

Out[25]:
```
array([19, 18, 28, 33, 32, 31, 46, 37, 60, 25, 62, 23, 56, 27, 52,
30, 34,
       59, 63, 55, 22, 26, 35, 24, 41, 38, 36, 21, 48, 40, 58, 53,
43, 64,
       20, 61, 44, 57, 29, 45, 54, 49, 47, 51, 42, 50, 39])
```

```
In [29]: df1['age'].value_counts()
```

```
Out[29]: 18    69
         19    68
         50    29
         51    29
         47    29
         46    29
         45    29
         20    29
         48    29
         52    29
         22    28
         49    28
         54    28
         53    28
         21    28
         26    28
         24    28
         25    28
         28    28
         27    28
         23    28
         43    27
         29    27
         30    27
         41    27
         42    27
         44    27
         31    27
         40    27
         32    26
         33    26
         56    26
         34    26
         55    26
         57    26
         37    25
         59    25
         58    25
         36    25
         38    25
         35    25
         39    25
         61    23
         60    23
         63    23
         62    23
         64    22
         Name: age, dtype: int64
```

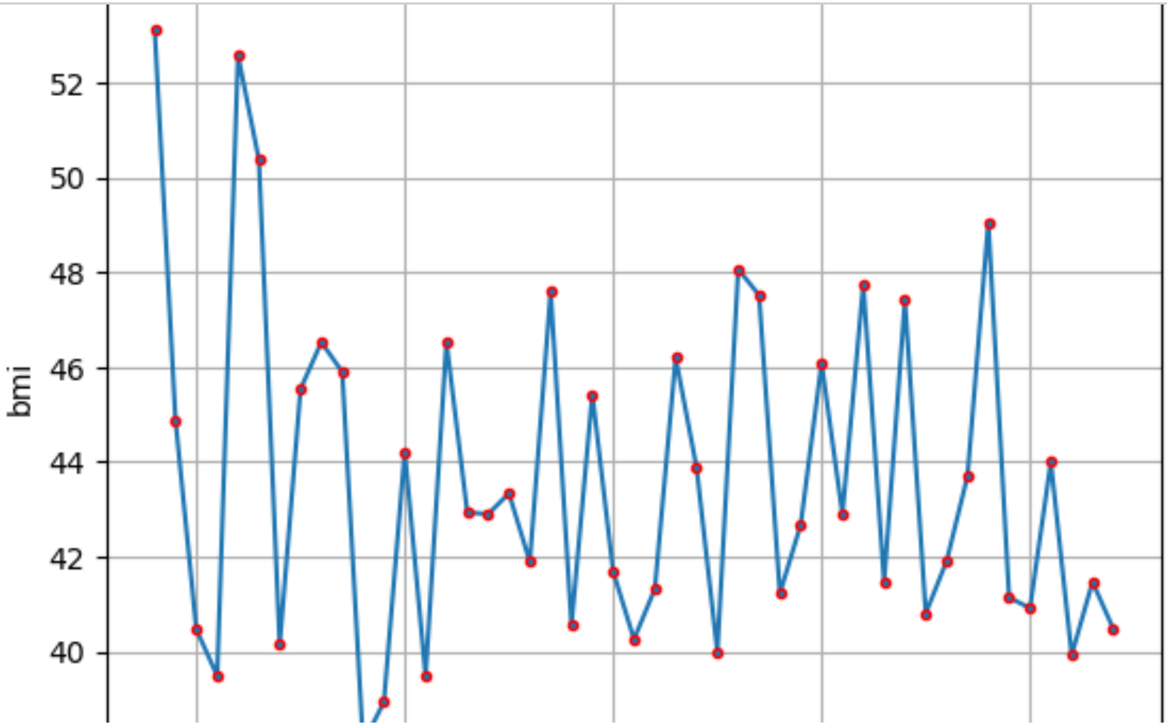
In [34]:

```
df1.groupby('age')['bmi'].max()
```

Out[34]:

```
age
18    53.130
19    44.880
20    40.470
21    39.490
22    52.580
23    50.380
24    40.150
25    45.540
26    46.530
27    45.900
28    38.060
29    38.940
30    44.220
31    39.490
32    46.530
33    42.940
34    42.900
35    43.340
36    41.895
37    47.600
38    40.565
39    45.430
40    41.690
41    40.260
42    41.325
43    46.200
44    43.890
45    39.995
46    48.070
47    47.520
48    41.230
49    42.680
50    46.090
51    42.900
52    47.740
53    41.470
54    47.410
55    40.810
56    41.910
57    43.700
58    49.060
59    41.140
60    40.920
61    44.000
62    39.930
63    41.470
64    40.480
Name: bmi, dtype: float64
```

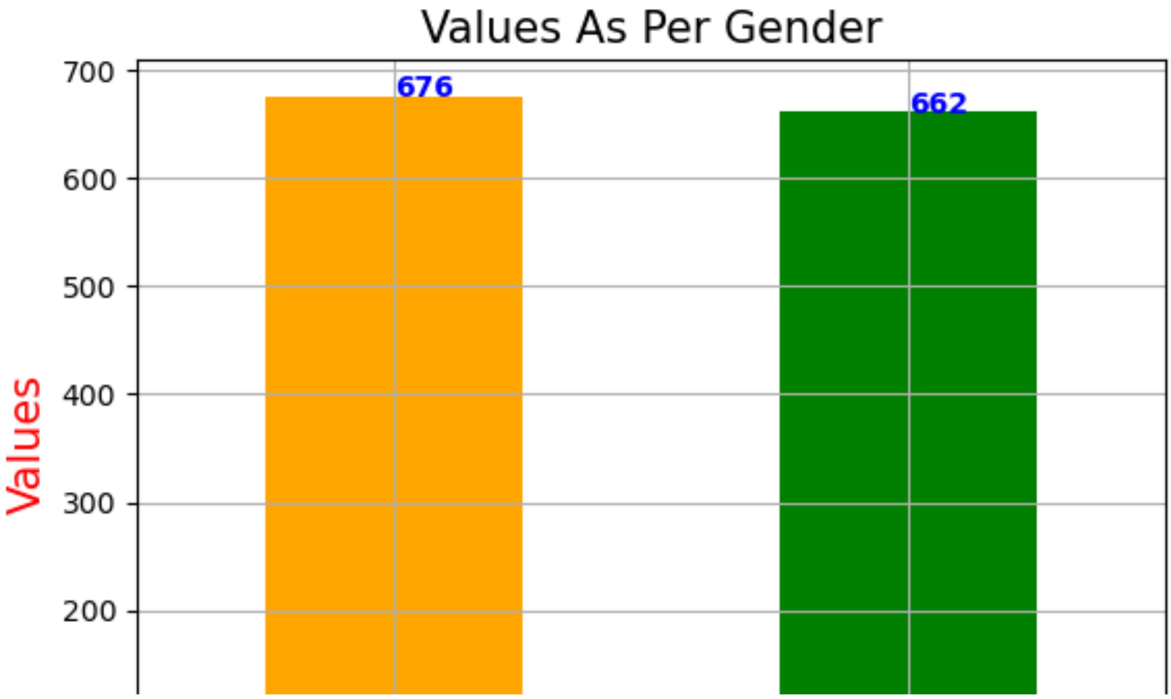
In [113]:
```
df1.groupby('age')['bmi'].max().plot(kind='line',marker='.',mec='r'
plt.title('Age vs bmi',size=15)
plt.grid()
plt.show()
```



In [51]:
```
df1['sex'].value_counts()
```

Out[51]:
```
male      676
female    662
Name: sex, dtype: int64
```

In [115]:
```
df1['sex'].value_counts().plot(kind='bar',color=['orange','green'])
plt.grid()
plt.text(0,675,'676',color='blue',fontweight='bold')
plt.text(1,661,'662',color='blue',fontweight='bold')
plt.title('Values As Per Gender',fontsize=15,c='black')
plt.xlabel('Gender',fontsize=15,c='red')
plt.ylabel('Values',fontsize=15,c='red')
plt.show()
```

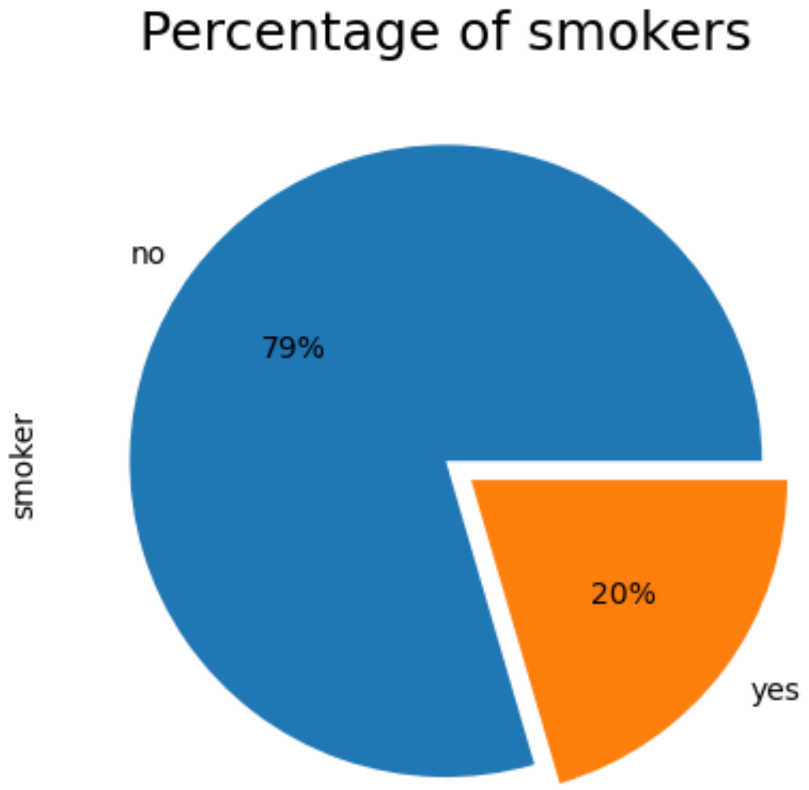


In [56]:
```
df1['smoker'].value_counts()
```

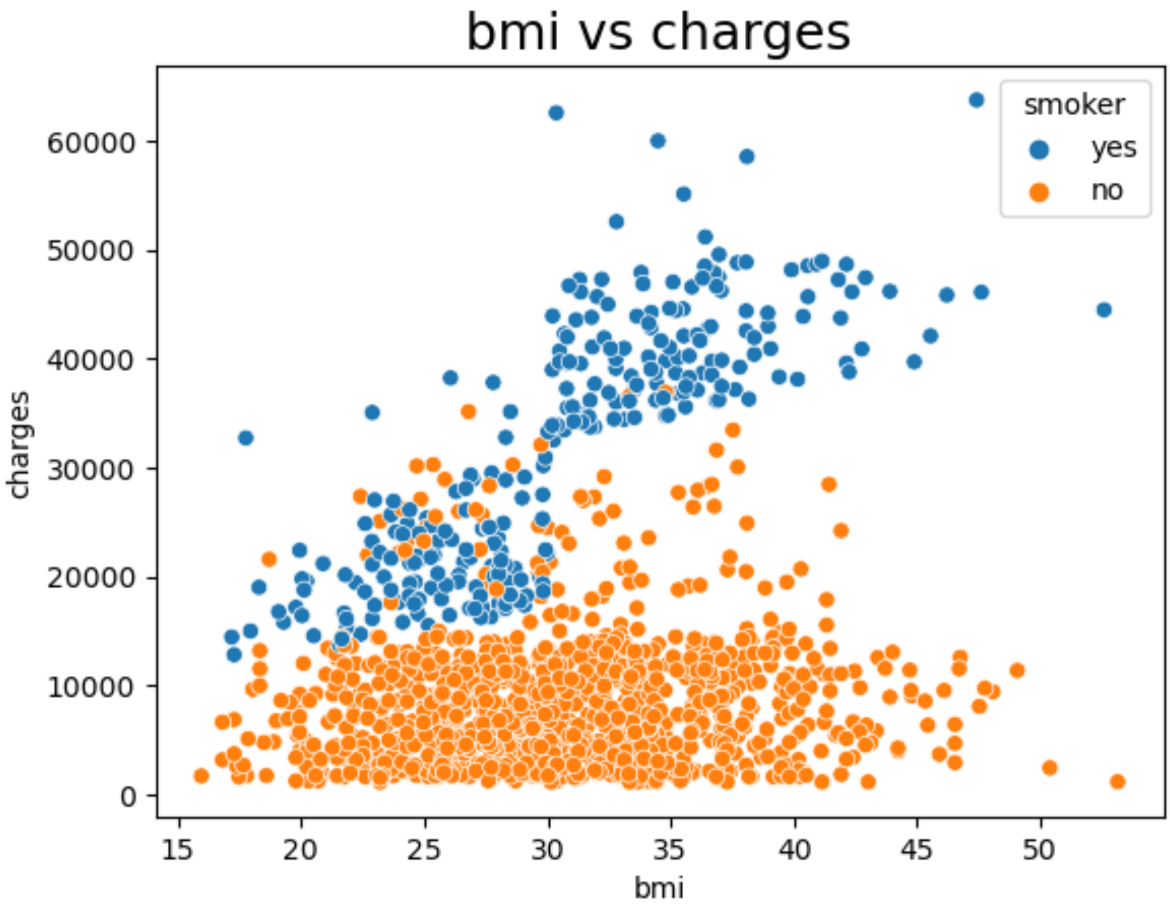
Out[56]:
```
no     1064
yes     274
Name: smoker, dtype: int64
```

In [119]:

```
df1['smoker'].value_counts().plot(kind='pie',autopct='%i%%',explode
plt.title('Percentage of smokers',fontsize=18,c='black')
plt.show()
```



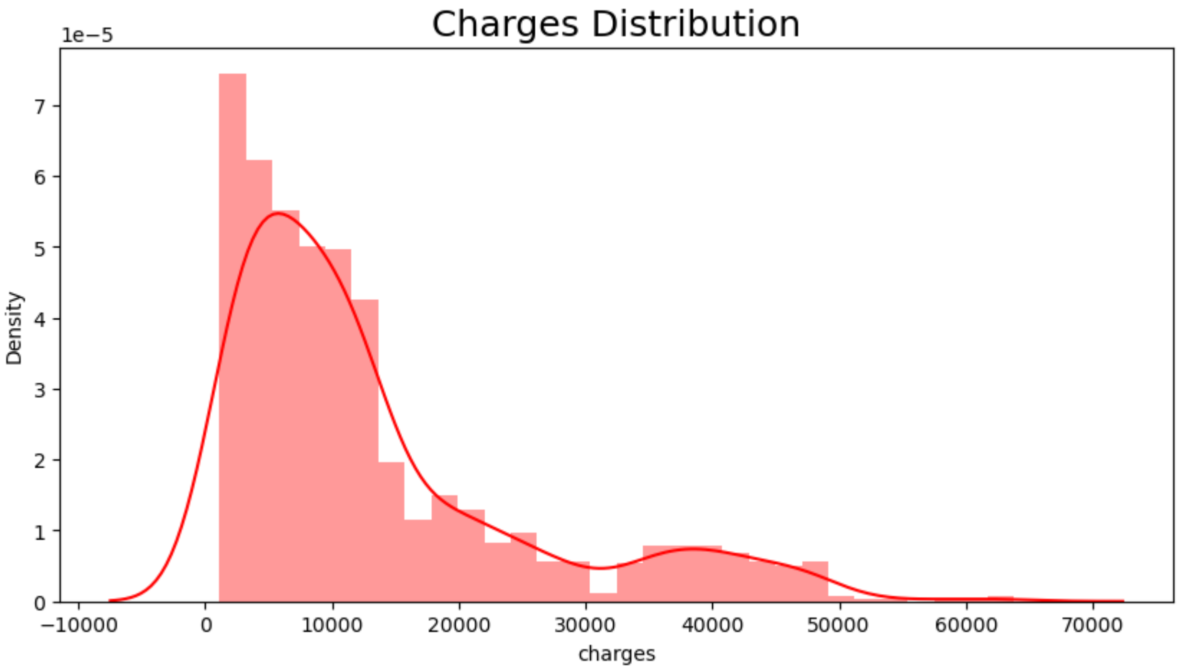
```
In [118]: sns.scatterplot(data=df1,x='bmi',y='charges',hue='smoker')
          plt.title('bmi vs charges',size=18)
          plt.show()
```



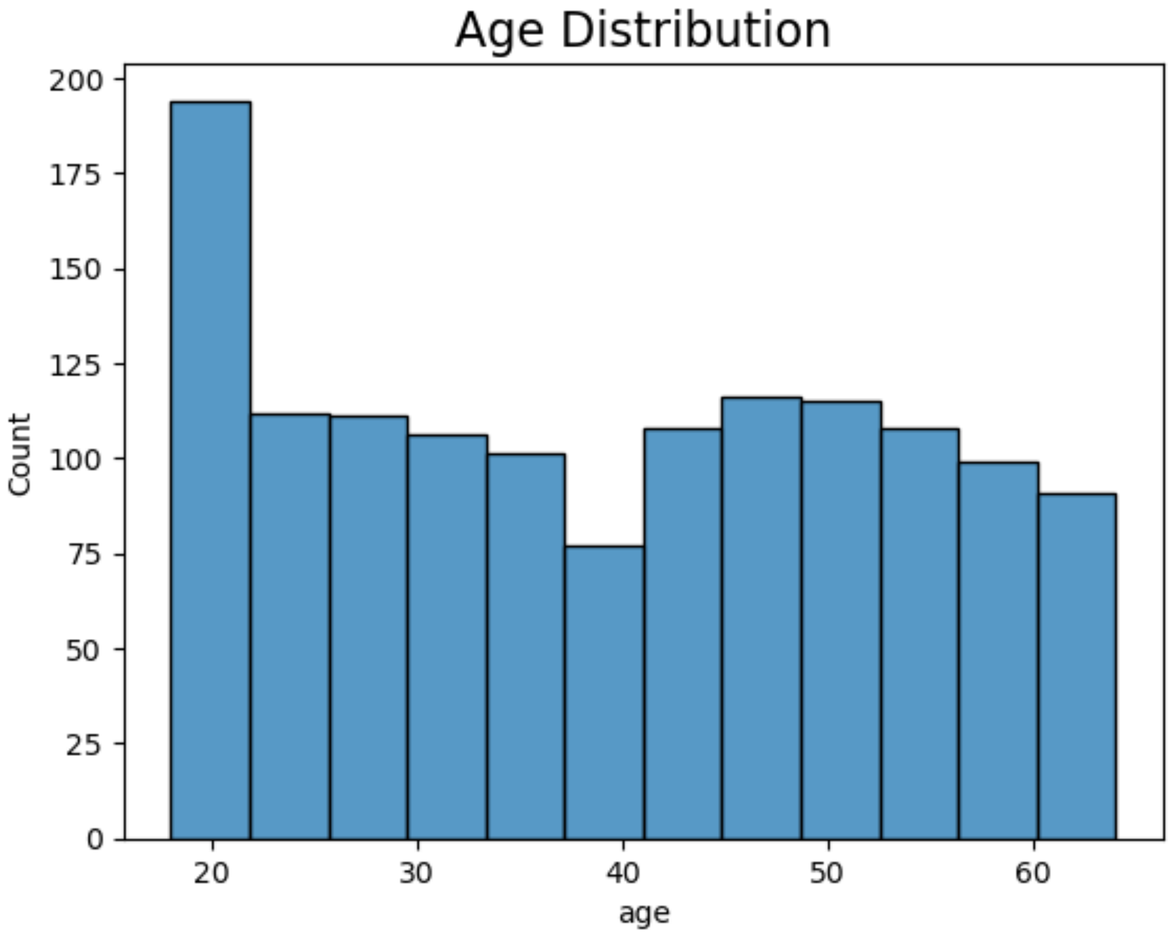
In [101]:

```
plt.figure(figsize=(10,5))
sns.distplot(df1.charges,color='red')
plt.title('Charges Distribution',size=18)
plt.show()
```

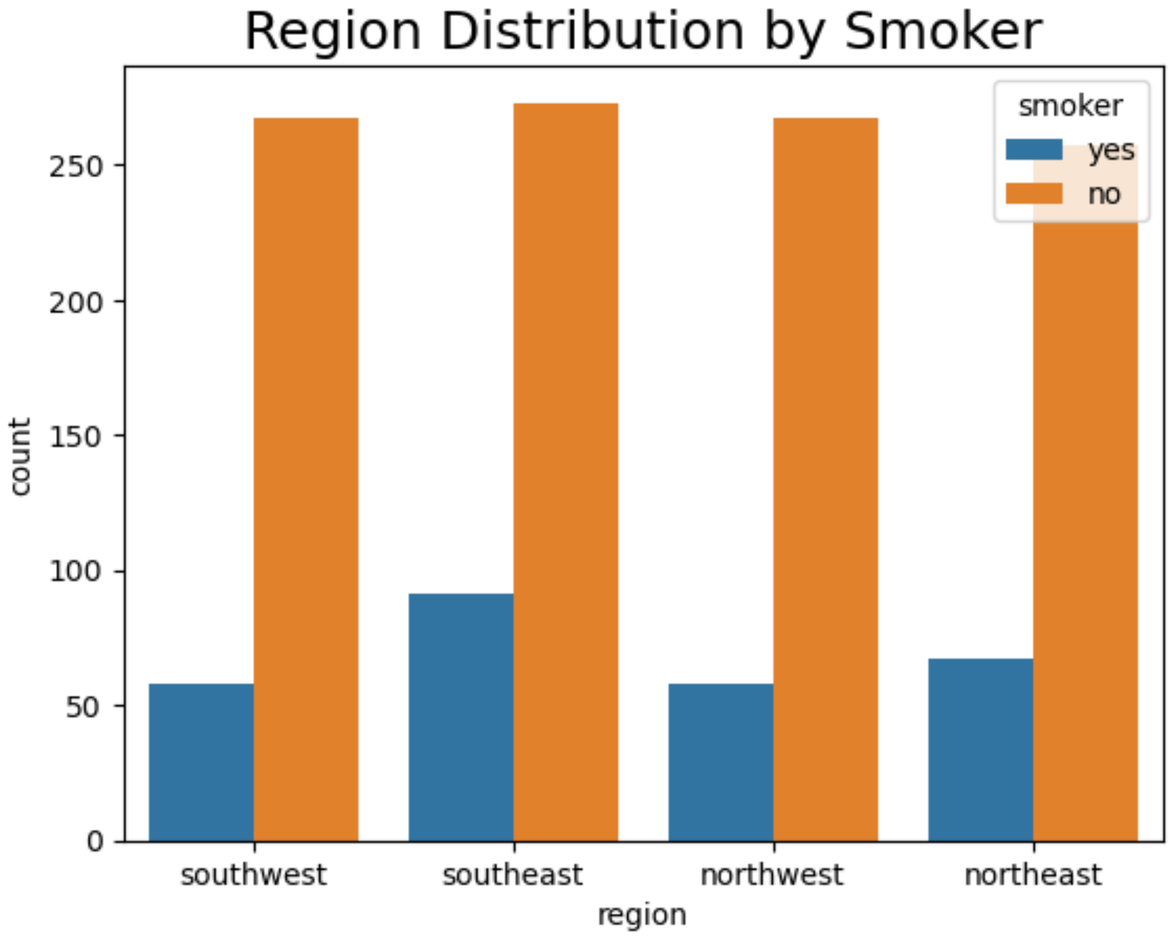
```
/var/folders/4r/_fbllh5n3539mmkqv4sj1h_c0000gn/T/ipykernel_81572/3
575971251.py:2: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level fun
ction with
similar flexibility) or `histplot` (an axes-level function for his
tograms).

For a guide to updating your code to use the new functions, please
see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751
(https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)

  sns.distplot(df1.charges,color='red')
```



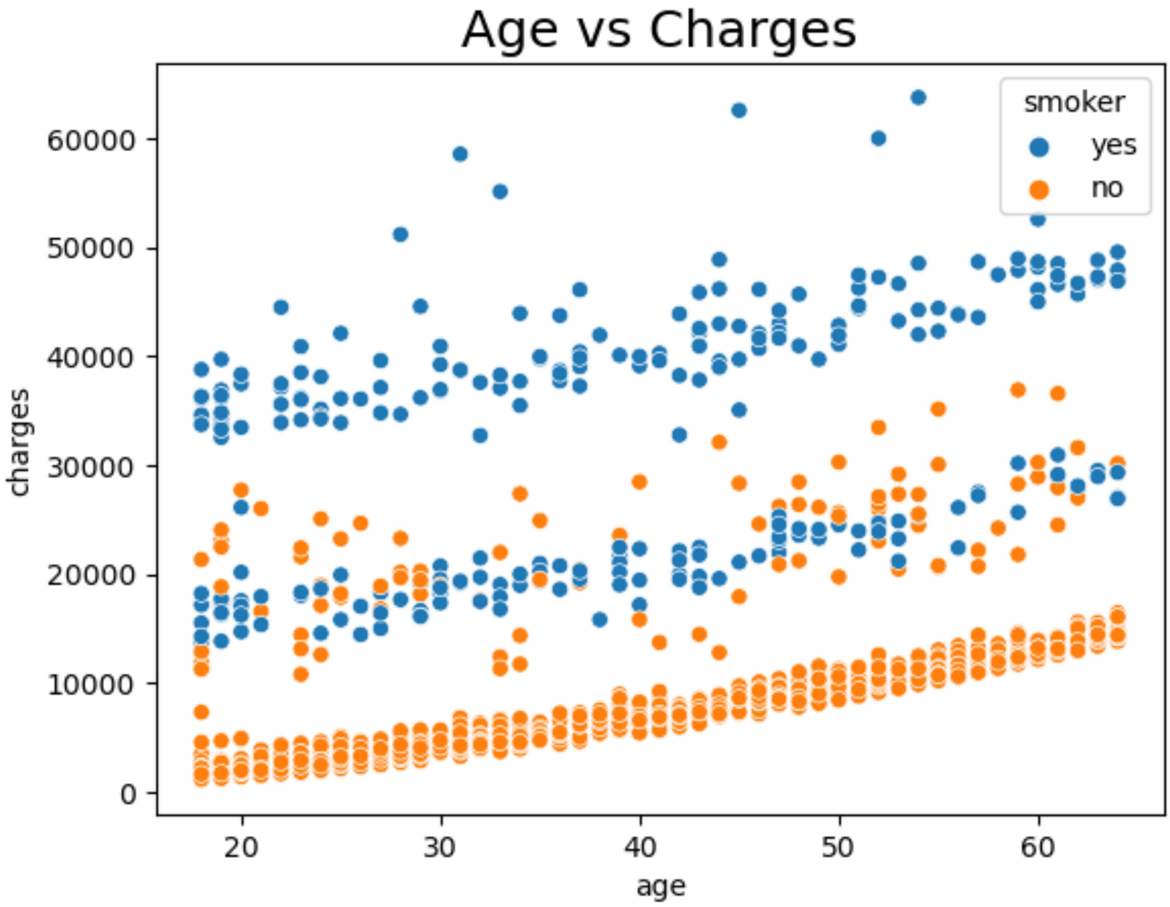
```
In [100]: sns.histplot(df1.age)
          plt.title('Age Distribution',size=16)
          plt.show()
```



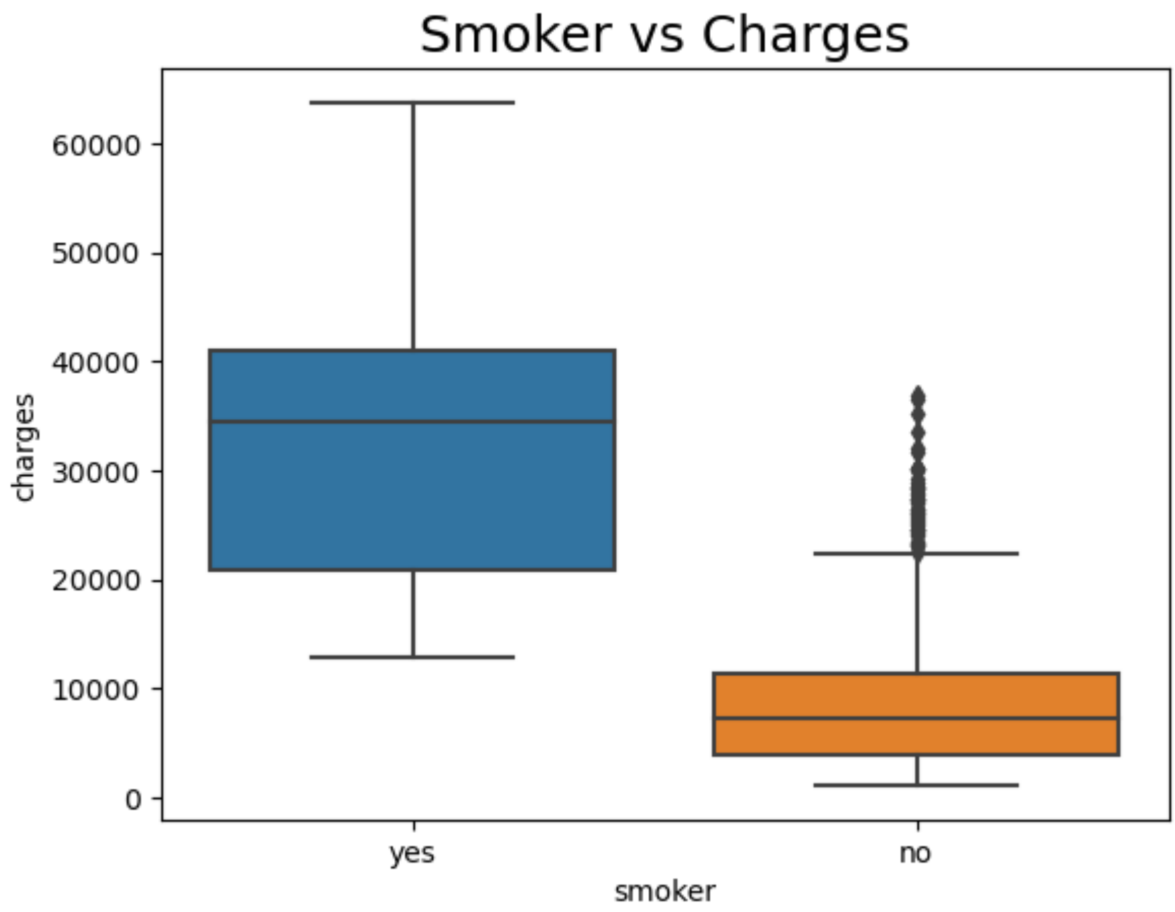
In [99]:
```
sns.countplot(data=df1,x='region',hue='smoker')
plt.title('Region Distribution by Smoker', size=18)
plt.show()
```



In [98]:
```
sns.scatterplot(data=df1,x='age',y='charges',hue='smoker')
plt.title('Age vs Charges',size=18)
plt.show()
```



In [103]:
```
sns.boxplot(data=df1,x='smoker',y='charges')
plt.title('Smoker vs Charges',size=18)
plt.show()
```



In [104]:
```
df1.corr()
```

```
/var/folders/4r/_fbllh5n3539mmkqv4sj1h_c0000gn/T/ipykernel_81572/4
73017434.py:1: FutureWarning: The default value of numeric_only in
DataFrame.corr is deprecated. In a future version, it will default
to False. Select only valid columns or specify the value of numeri
c_only to silence this warning.
  df1.corr()
```

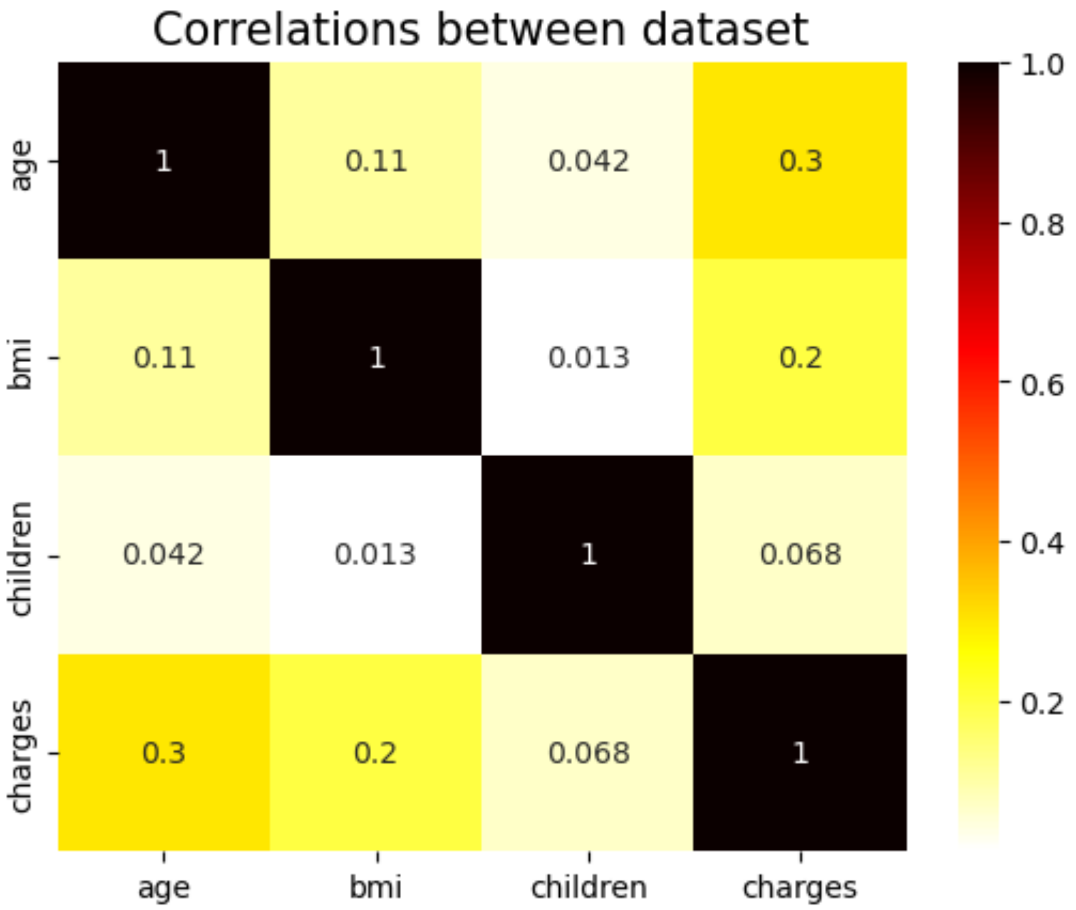
Out[104]:

	age	bmi	children	charges
age	1.000000	0.109272	0.042469	0.299008
bmi	0.109272	1.000000	0.012759	0.198341
children	0.042469	0.012759	1.000000	0.067998
charges	0.299008	0.198341	0.067998	1.000000

In [112]:

```
sns.heatmap(df1.corr(),annot=True,cmap='hot_r')
plt.title('Correlations between dataset',size=15)
plt.show()
```

```
/var/folders/4r/_fbllh5n3539mmkqv4sj1h_c0000gn/T/ipykernel_81572/2
512831035.py:1: FutureWarning: The default value of numeric_only i
n DataFrame.corr is deprecated. In a future version, it will defau
lt to False. Select only valid columns or specify the value of num
eric_only to silence this warning.
  sns.heatmap(df1.corr(),annot=True,cmap='hot_r')
```



In [ ]: