In [4]:
```
import pandas as pd
pd.plotting.register_matplotlib_converters()
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
print("Setup Complete")
import numpy as np
```

```
Setup Complete
```

In [5]:
```
dt = pd.read_csv('insurance.csv')
dt
```

Out[5]:

	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 [6]:
```
dt.head(5)
```

Out[6]:

	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 [7]:
```
dt.tail(5)
```

Out[7]:

	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 [8]:
```
#getting a random sample of my data set
dt.sample(n = 5)
```

Out[8]:

	age	sex	bmi	children	smoker	region	charges
522	51	female	33.915	0	no	northeast	9866.30485
776	40	male	32.300	2	no	northwest	6986.69700
614	20	female	33.000	0	no	southeast	1880.07000
64	20	female	22.420	0	yes	northwest	14711.74380
219	24	female	23.210	0	no	southeast	25081.76784

In [9]:
```
dt.charges.unique()
```

Out[9]:
```
array([16884.924 ,  1725.5523,  4449.462 , ...,  1629.8335,  2007.945 ,
       29141.3603])
```

In [10]:
```
dt['smoker'].dtype
```

Out[10]:
```
dtype('O')
```

In [11]:
```
#checking for missing values

dt.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 [12]:
```
dt.isna().sum()
```

Out[12]:
```
age         0
sex         0
bmi         0
children    0
smoker      0
region      0
charges     0
dtype: int64
```

In [13]:
```
dt.columns.to_list()
```

Out[13]:
```
['age', 'sex', 'bmi', 'children', 'smoker', 'region', 'charges']
```

In [14]:
```
dt.charges.mean()
```

Out[14]:
```
13270.422265141257
```

In [15]:
```
dt.charges.min()
```

Out[15]:
```
1121.8739
```

In [16]:
```
dt.charges.max()
```

Out[16]:
```
63770.42801
```

In [17]:
```
dt.charges.median()
```

Out[17]:
```
9382.033
```

In [18]:
```
dt.charges.mode()
```

Out[18]:
```
0    1639.5631
Name: charges, dtype: float64
```

In [19]:
```
dt.bmi.max()
```

Out[19]:
```
53.13
```

In [20]:
```
dt.bmi.min()
```

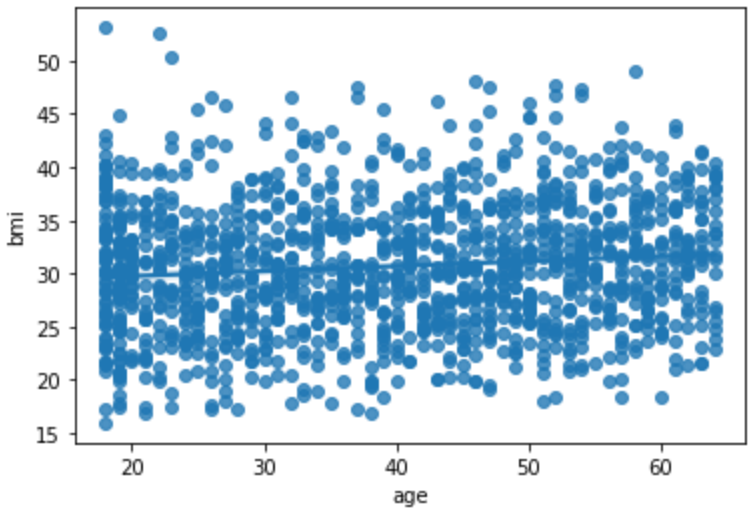
Out[20]:
```
15.96
```

In [21]:
```
dt.describe()
```

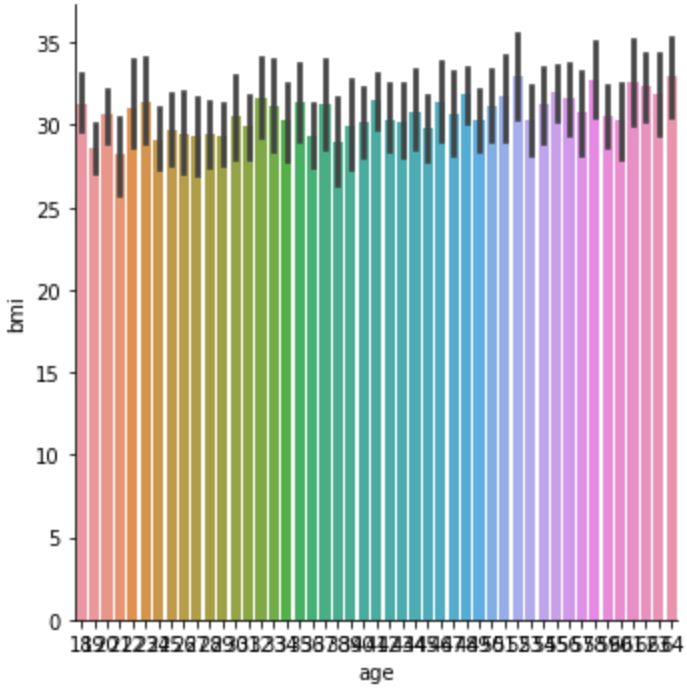
Out[21]:

	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 [22]:
```
ans = sns.regplot(x = 'age', y = 'bmi', data = dt)
```



In [23]:
```
ans = sns.catplot(x = 'age', y = 'bmi', data =dt, kind = 'bar');
```



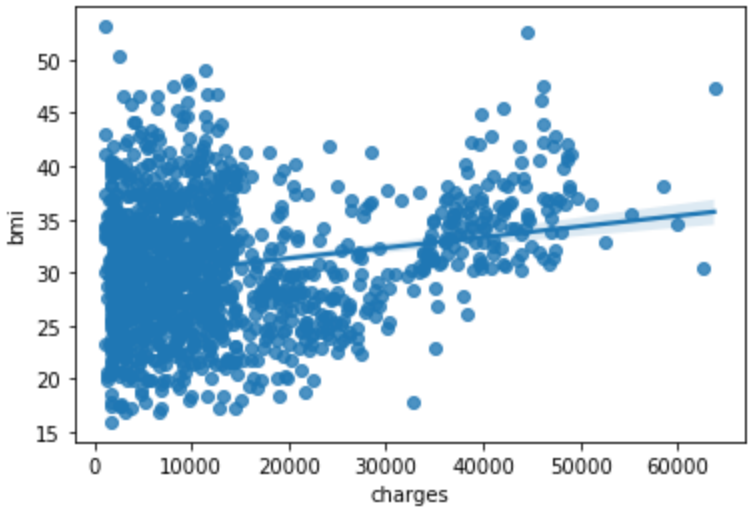
In [24]:
```
cols = ['charges', 'bmi', 'sex']
new_dt = dt[cols]
new_dt
```

Out[24]:

	charges	bmi	sex
0	16884.92400	27.900	female
1	1725.55230	33.770	male
2	4449.46200	33.000	male
3	21984.47061	22.705	male
4	3866.85520	28.880	male
...	...	...	...
1333	10600.54830	30.970	male
1334	2205.98080	31.920	female
1335	1629.83350	36.850	female
1336	2007.94500	25.800	female
1337	29141.36030	29.070	female

1338 rows × 3 columns

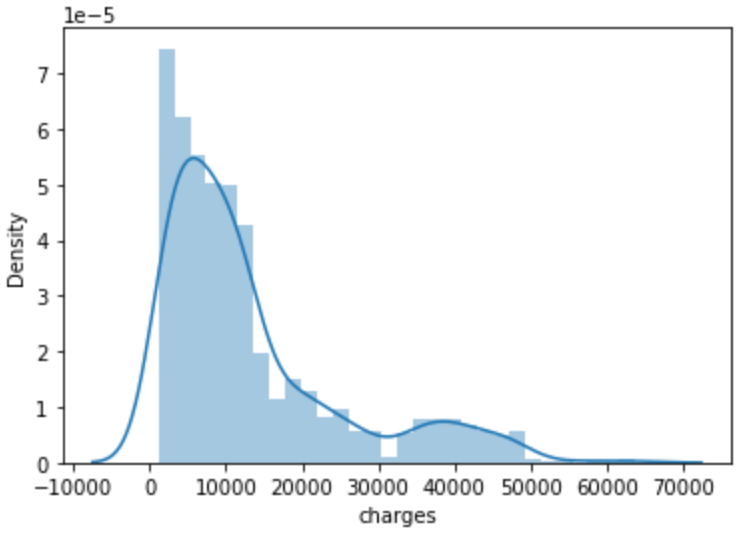
In [25]:
```
ans = sns.regplot(x = 'charges', y = 'bmi', data = dt)
```



In [26]:
```
sns.distplot(dt.charges.dropna(axis=0))
```

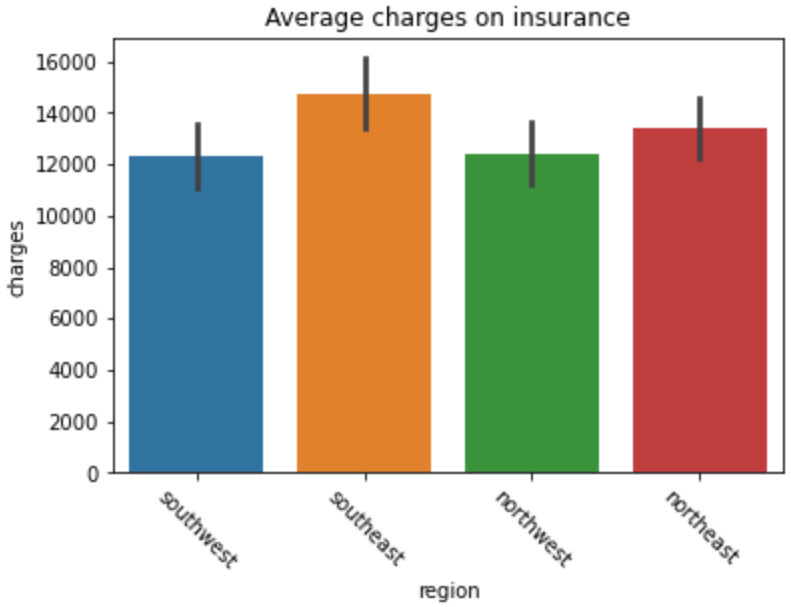
```
C:\Users\DELL\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning:
`distplot` is a deprecated function and will be removed in a future version. Please adap
t your code to use either `displot` (a figure-level function with similar flexibility) o
r `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[26]:
```
<AxesSubplot:xlabel='charges', ylabel='Density'>
```



In [27]:
```
p = sns.barplot(x = 'region', y = 'charges',
                data =dt, estimator = np.mean);
p.set(title = 'Average charges on insurance')
p.set_xticklabels(p.get_xticklabels(), rotation = -45)
```

Out[27]:
```
[Text(0, 0, 'southwest'),
 Text(1, 0, 'southeast'),
 Text(2, 0, 'northwest'),
 Text(3, 0, 'northeast')]
```



In [28]:
```
#Lets look at bmi greater than 30
#we'll have to create a new data set
large_bmi = dt[dt['bmi']>30]
large_bmi
```

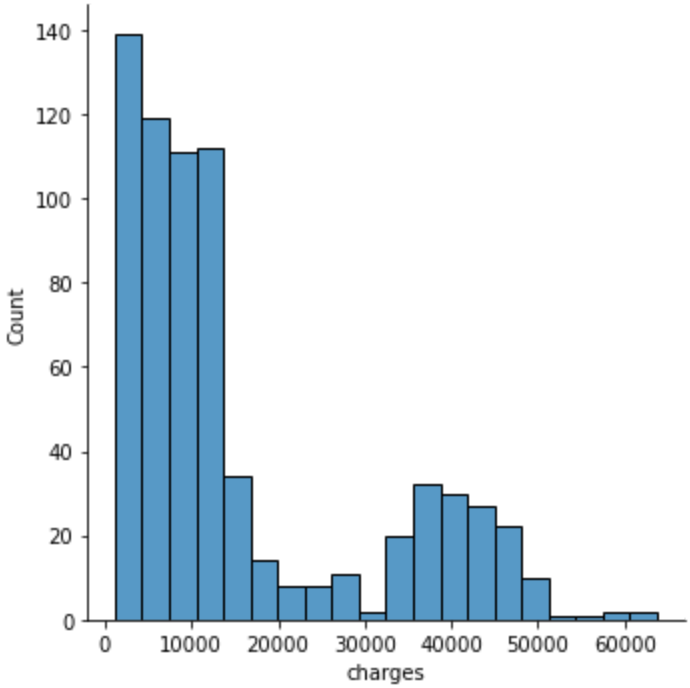
Out[28]:

	age	sex	bmi	children	smoker	region	charges
1	18	male	33.77	1	no	southeast	1725.55230
2	28	male	33.00	3	no	southeast	4449.46200
6	46	female	33.44	1	no	southeast	8240.58960
12	23	male	34.40	0	no	southwest	1826.84300
13	56	female	39.82	0	no	southeast	11090.71780
...	...	...	...	...	...	...	...
1331	23	female	33.40	0	no	southwest	10795.93733
1332	52	female	44.70	3	no	southwest	11411.68500
1333	50	male	30.97	3	no	northwest	10600.54830
1334	18	female	31.92	0	no	northeast	2205.98080
1335	18	female	36.85	0	no	southeast	1629.83350

705 rows × 7 columns

In [29]:
```
sns.displot(large_bmi.charges)
```

Out[29]:
```
<seaborn.axisgrid.FacetGrid at 0x101487f0>
```


140
120
100
80
60
40
20
0
Count
0
10000
20000
30000
40000
50000
60000
charges

In [ ]: