In [ ]:

In [54]:
```
import pandas as pd
data = pd.read_csv("insurance2.csv")
```

In [57]:
```
data.head()
```

Out[57]:

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

In [59]:
```
data.isna().sum()
```

Out[59]:
```
0       1
1       1
2       0
3       0
4       1
      ..
1333    0
1334    1
1335    1
1336    0
1337    1
Name: insuranceclaim, Length: 1338, dtype: int64
```

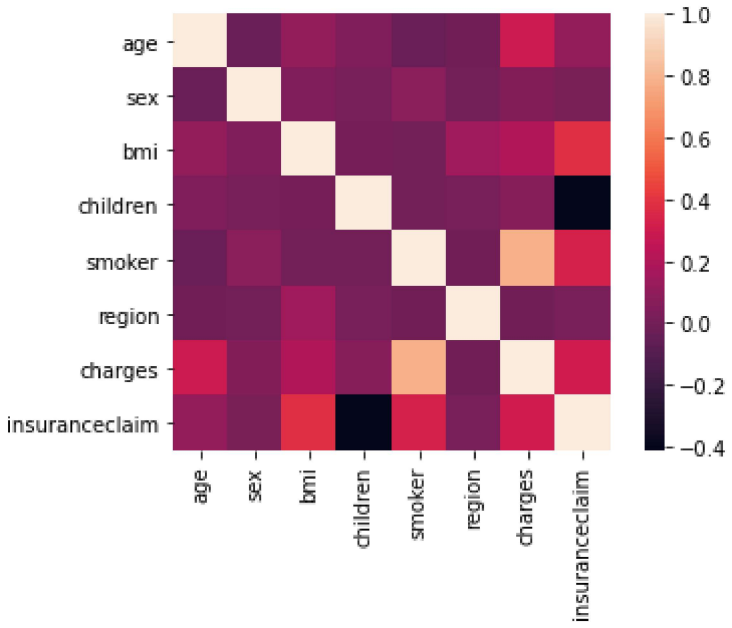
In [61]:
```
data.corr()
```

Out[61]:

	age	sex	bmi	children	smoker	region	charges	insuranceclaim
age	1.000000	-0.020856	0.109272	0.042469	-0.025019	0.002127	0.299008	0.113723
sex	-0.020856	1.000000	0.046371	0.017163	0.076185	0.004588	0.057292	0.031565
bmi	0.109272	0.046371	1.000000	0.012759	0.003750	0.157566	0.198341	0.384198
children	0.042469	0.017163	0.012759	1.000000	0.007673	0.016569	0.067998	-0.409526
smoker	-0.025019	0.076185	0.003750	0.007673	1.000000	-0.002181	0.787251	0.333261
region	0.002127	0.004588	0.157566	0.016569	-0.002181	1.000000	-0.006208	0.020891
charges	0.299008	0.057292	0.198341	0.067998	0.787251	-0.006208	1.000000	0.309418
insuranceclaim	0.113723	0.031565	0.384198	-0.409526	0.333261	0.020891	0.309418	1.000000

In [62]:
```
import seaborn as sns
sns.heatmap(data.corr(), square=True)
```

Out[62]:
```
<AxesSubplot:>
```



In [63]:
```
data1 = data.drop("insuranceclaim", axis=1)
data1 = (data1 - data1.mean())/data1.std()
```

In [64]:
```
data1
```

Out[64]:

	age	sex	bmi	children	smoker	region	charges
0	-1.438227	-1.010141	-0.453151	-0.908274	1.969850	1.343402	0.298472
1	-1.509401	0.989221	0.509431	-0.078738	-0.507273	0.438331	-0.953333
2	-0.797655	0.989221	0.383164	1.580335	-0.507273	0.438331	-0.728402
3	-0.441782	0.989221	-1.305043	-0.908274	-0.507273	-0.466741	0.719574
4	-0.512957	0.989221	-0.292447	-0.908274	-0.507273	-0.466741	-0.776512
...	...	...	...	...	...	...	...
1333	0.768185	0.989221	0.050278	1.580335	-0.507273	-0.466741	-0.220468
1334	-1.509401	-1.010141	0.206062	-0.908274	-0.507273	-1.371813	-0.913661
1335	-1.509401	-1.010141	1.014499	-0.908274	-0.507273	0.438331	-0.961237
1336	-1.295877	-1.010141	-0.797515	-0.908274	-0.507273	1.343402	-0.930014
1337	1.551106	-1.010141	-0.261290	-0.908274	1.969850	-0.466741	1.310563

1338 rows × 7 columns

In [66]:
```
from sklearn.model_selection import train_test_split
x = data1
y = data["insuranceclaim"]
```

In [75]:
```
x_train, x_test, y_train, y_test = train_test_split(x,y,random_state = 10, test_size =
```

In [76]:
```
from sklearn.linear_model import LogisticRegression
reg = LogisticRegression()
```

```
reg.fit(x_train,y_train)
y_pred = reg.predict(x_test)
```

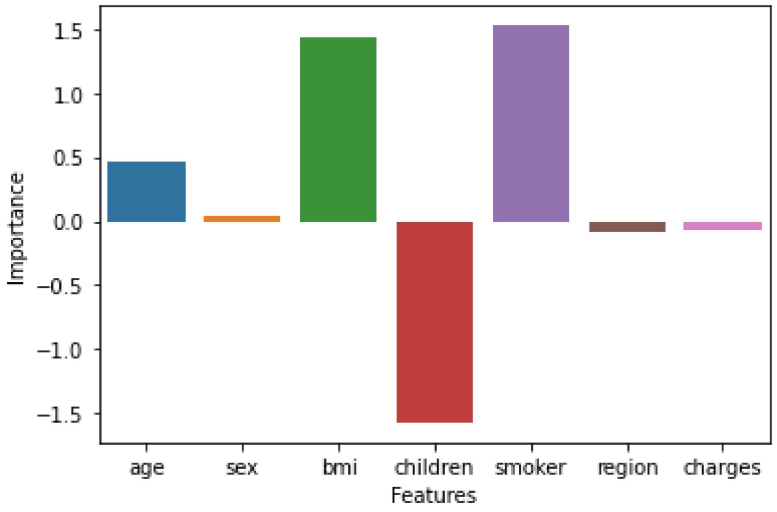
In [77]:
```
from sklearn.metrics import accuracy_score
accuracy = accuracy_score(y_pred, y_test)
accuracy
```

Out[77]:
```
0.8694029850746269
```

In [88]:
```
coef = list(reg.coef_[0])
label = list(data1.columns)
features = pd.DataFrame()
features['Features'] = label
features['Importance'] = coef
sns.barplot(x = features['Features'], y = features['Importance'])
```

```
<zip object at 0x0000023FD0137A00>
```

Out[88]:
```
<AxesSubplot:xlabel='Features', ylabel='Importance'>
```



In [97]:
```
print(set(zip(features['Features'], features['Importance'])))
```

```
{('sex', 0.037468365592143324), ('children', -1.5740102487063619), ('region', -0.0920397
9782575374), ('bmi', 1.4290694918696698), ('charges', -0.07455476697629447), ('smoker',
1.5257882747063194), ('age', 0.4644925867190356)}
```

In [ ]:
```

```