- Import libraries such as Pandas, matplotlib, NumPy, and seaborn and load the insurance dataset

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
data=pd.read_csv('insurance.csv')
print(data)
```

```
      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 x 7 columns]
```

- Check the shape of the data along with the data types of the column

```
data.shape
```

```
(1338, 7)
```

```
data.dtypes
```

```
age           int64
sex          object
bmi         float64
children      int64
smoker       object
region       object
charges     float64
dtype: object
```

- Check missing values in the dataset and find the appropriate measures to fill in the missing values

```
data.isna().sum()
```

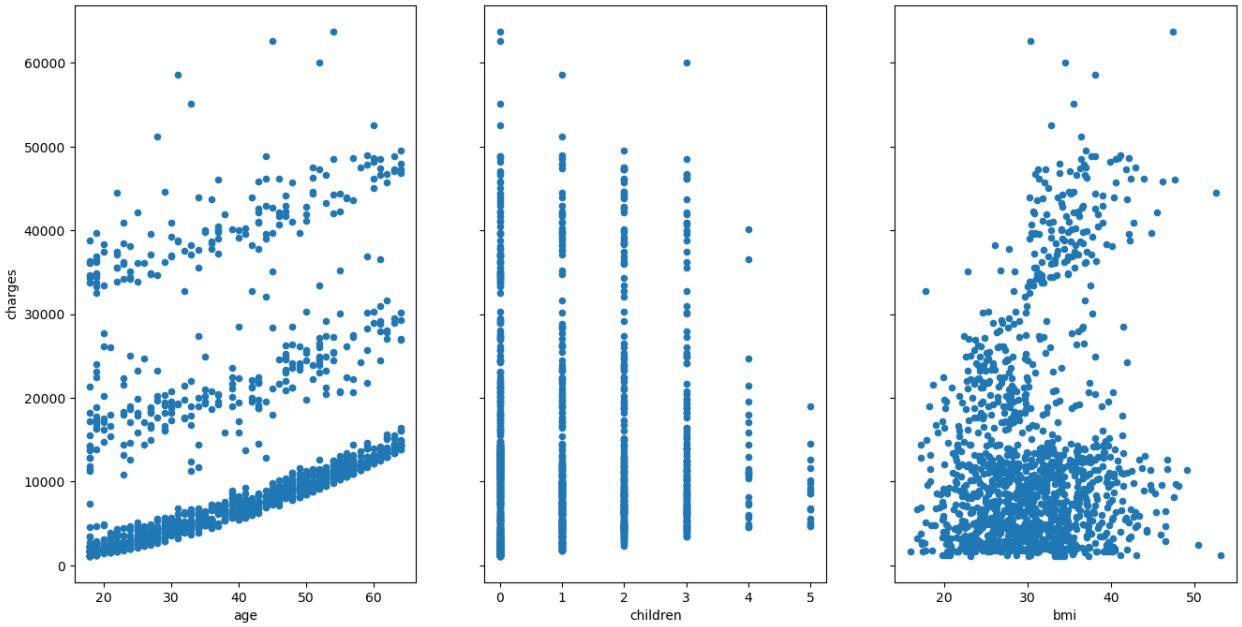
```
age         0
sex         0
bmi         0
children    0
smoker      0
```

```
region      0
charges     0
dtype: int64
```

- Explore the relationship between the feature and target column using a count plot of categorical columns and a scatter plot of numerical columns

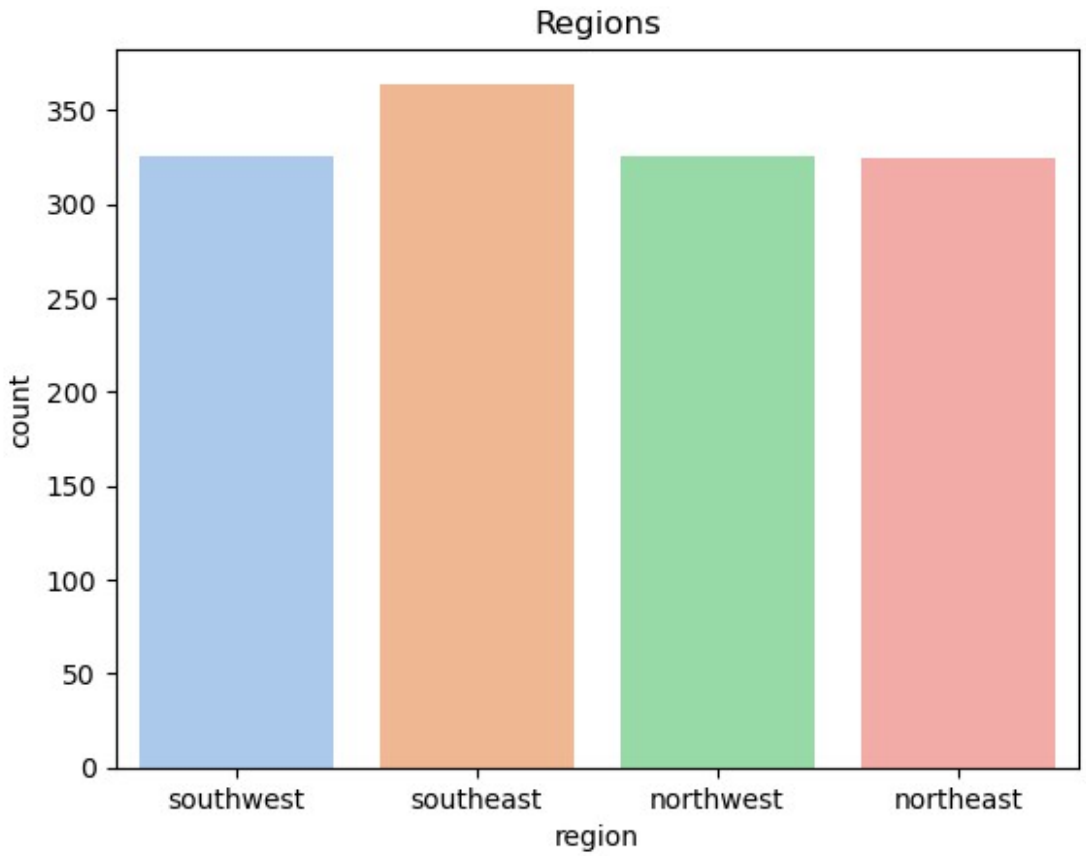
```
fig,axs=plt.subplots(1,3,sharey=True)
data.plot(kind='scatter',x='age',y='charges',ax=axs[0],figsize=(16,8))
data.plot(kind='scatter',x='children',y='charges',ax=axs[1])
data.plot(kind='scatter',x='bmi',y='charges',ax=axs[2])
```

```
<Axes: xlabel='bmi', ylabel='charges'>
```




```
sns.countplot(data=data,x='region',palette='pastel')
plt.title('Regions')
plt.show()
```

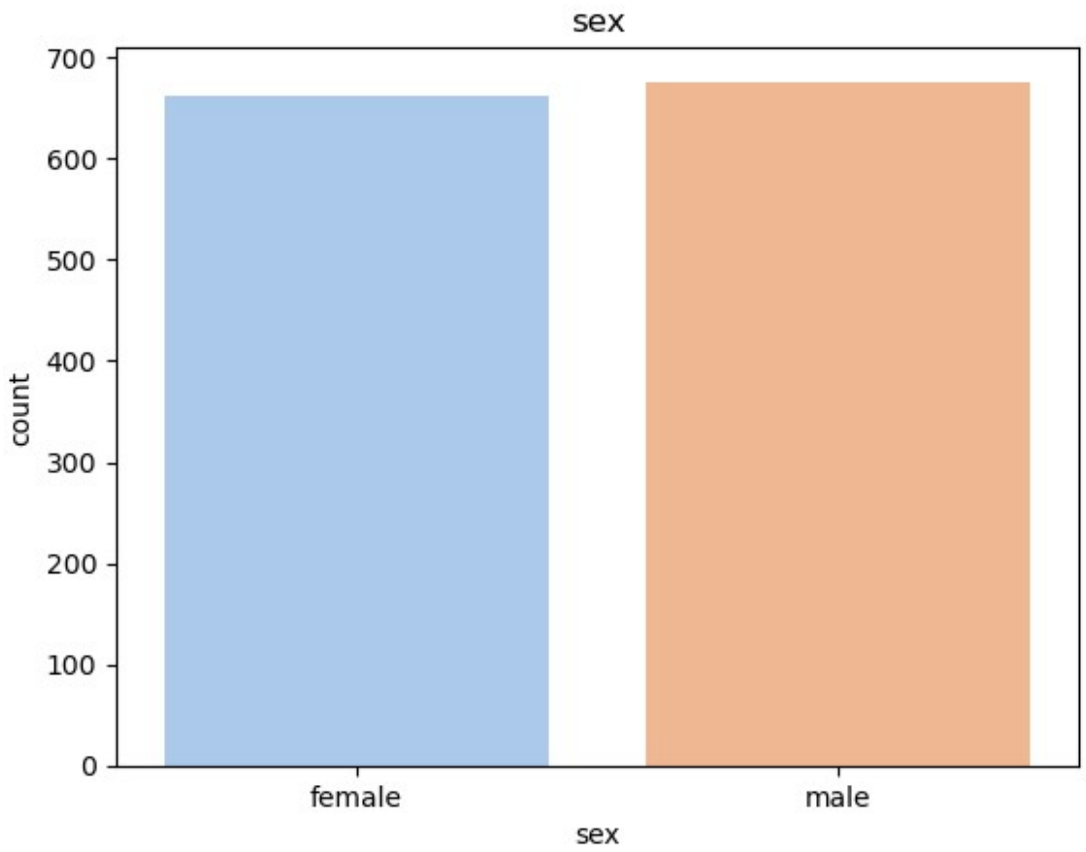
```
C:\Users\91798\AppData\Local\Temp\ipykernel_18568\4135800785.py:1:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.countplot(data=data,x='region',palette='pastel')
```




```
sns.countplot(data=data,x='sex',palette='pastel')
plt.title('sex')
```

```
C:\Users\91798\AppData\Local\Temp\ipykernel_18568\333834983.py:1:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.countplot(data=data,x='sex',palette='pastel')
```

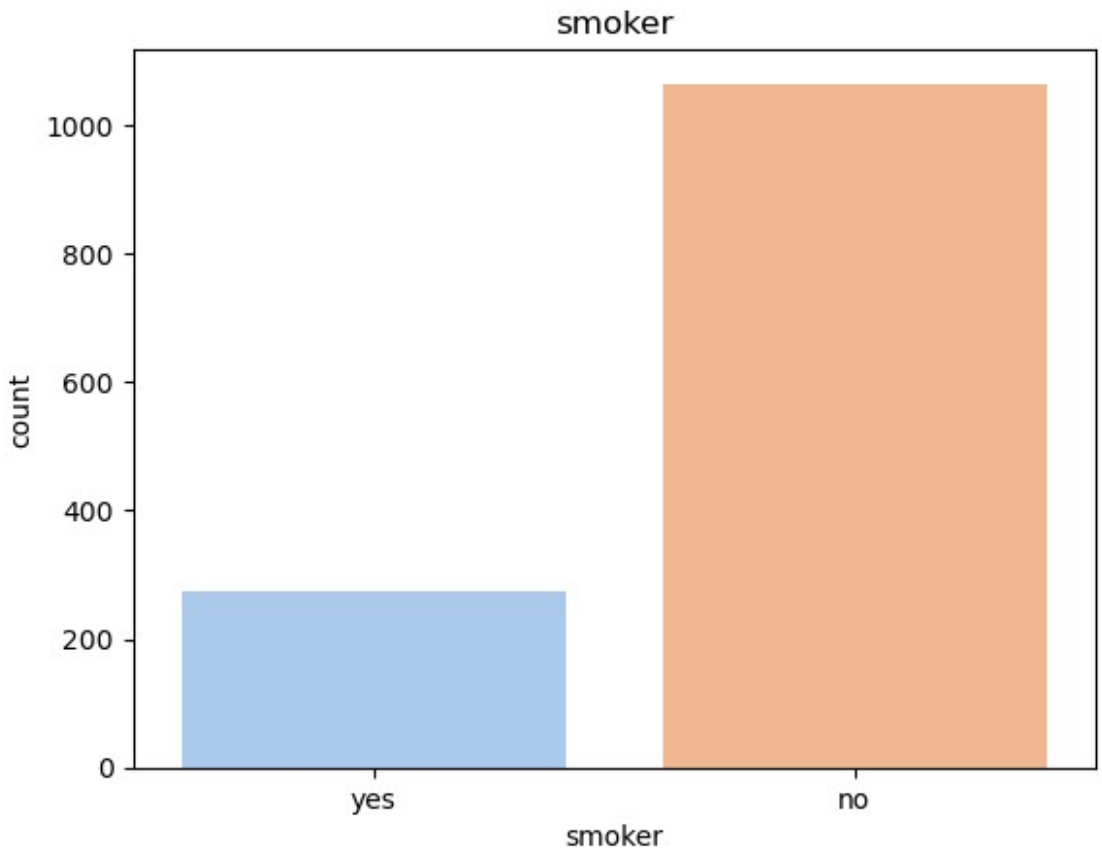
```
Text(0.5, 1.0, 'sex')
```




```
sns.countplot(data=data,x='smoker',palette='pastel')

plt.title('smoker')
```

```
C:\Users\91798\AppData\Local\Temp\ipykernel_21016\3487440372.py:1:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.countplot(data=data,x='smoker',palette='pastel')
```

```
Text(0.5, 1.0, 'smoker')
```



- Perform data visualization using plots of feature vs feature

plt.figure(figsize=(10,6)) sns.boxplot(x='children',y='age',data=data,palette='pastel') plt.show()

#Draw a bar plot for age vs charge

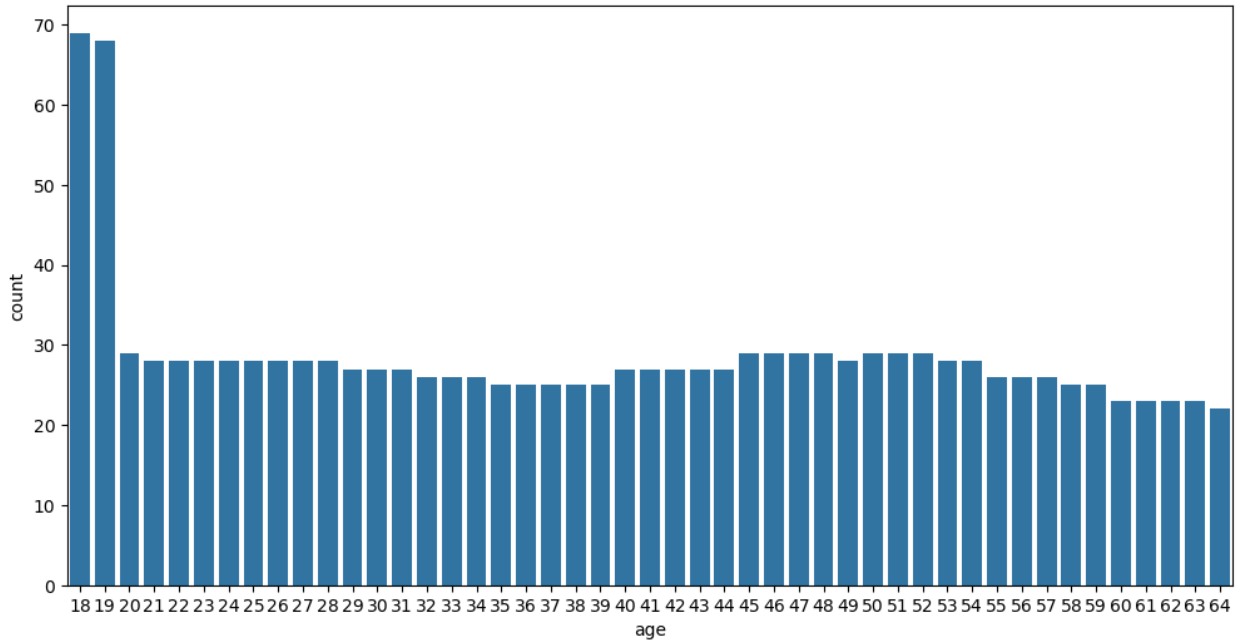
#Draw a bar plot for region vs charge

#Draw a bar plot for sex vs charge

#Draw a bar plot for bmi vs charge

#Draw a bar plot for smoker vs charge

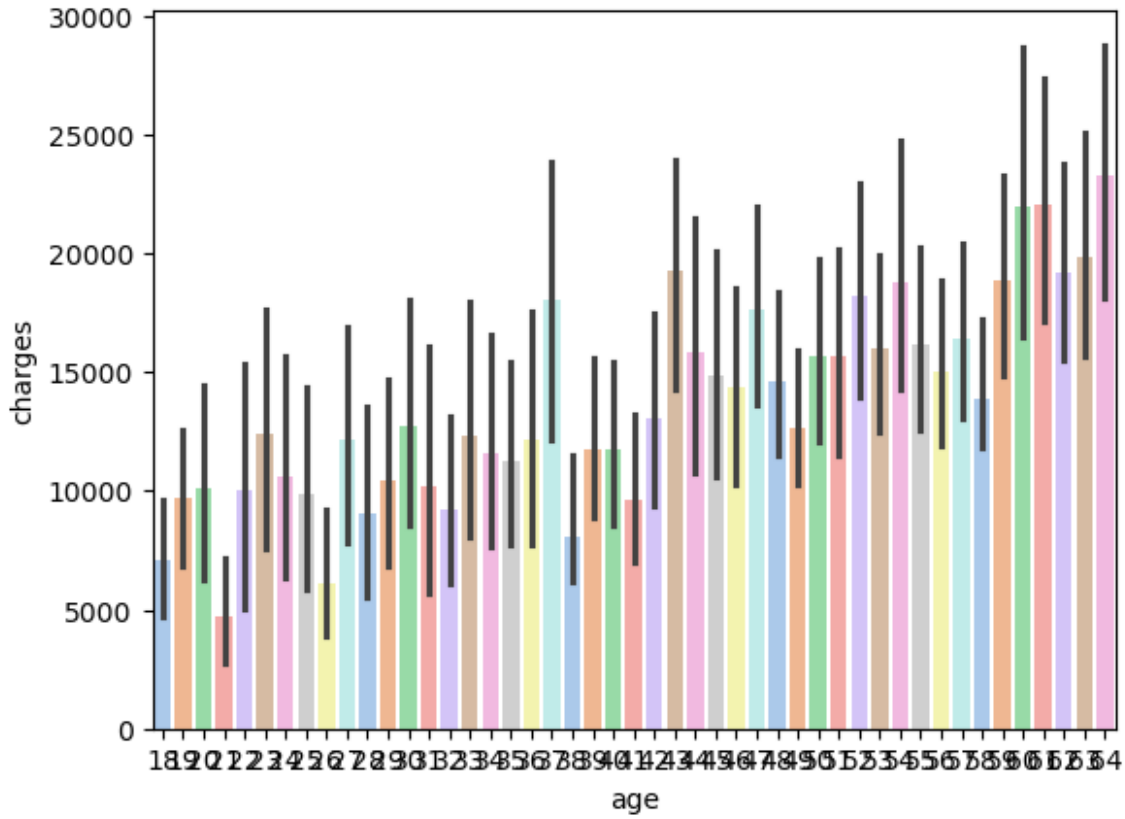
```
import seaborn as sns
plt.figure(figsize=(12,6))
sns.countplot(x='age',data=data)
plt.show()
```

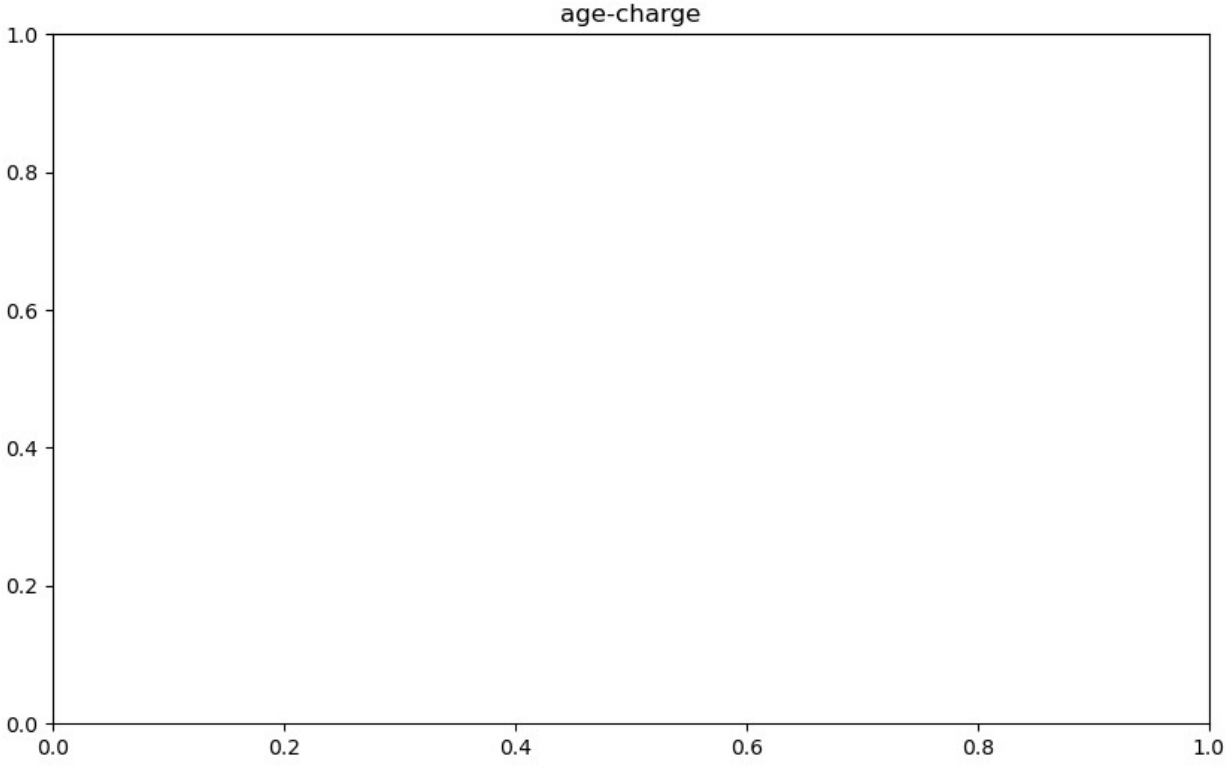


- Check if the number of premium charges for smokers or non-smokers is increasing as they are aging

```
sns.barplot(data=data,x='age',y='charges',palette='pastel')
plt.figure(figsize=(10,6))
plt.title('age-charge')

plt.show()
```

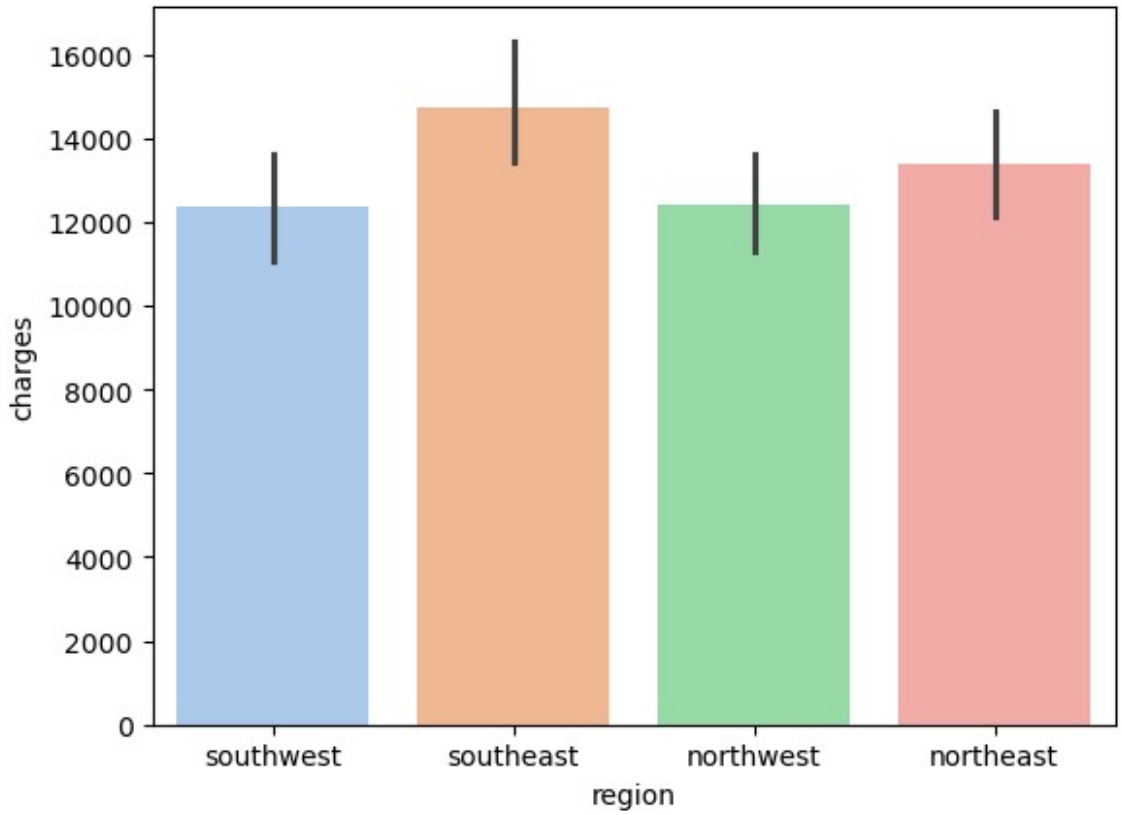
```
C:\Users\91798\AppData\Local\Temp\ipykernel_21016\568792742.py:1:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.barplot(data=data,x='age',y='charges',palette='pastel')
```

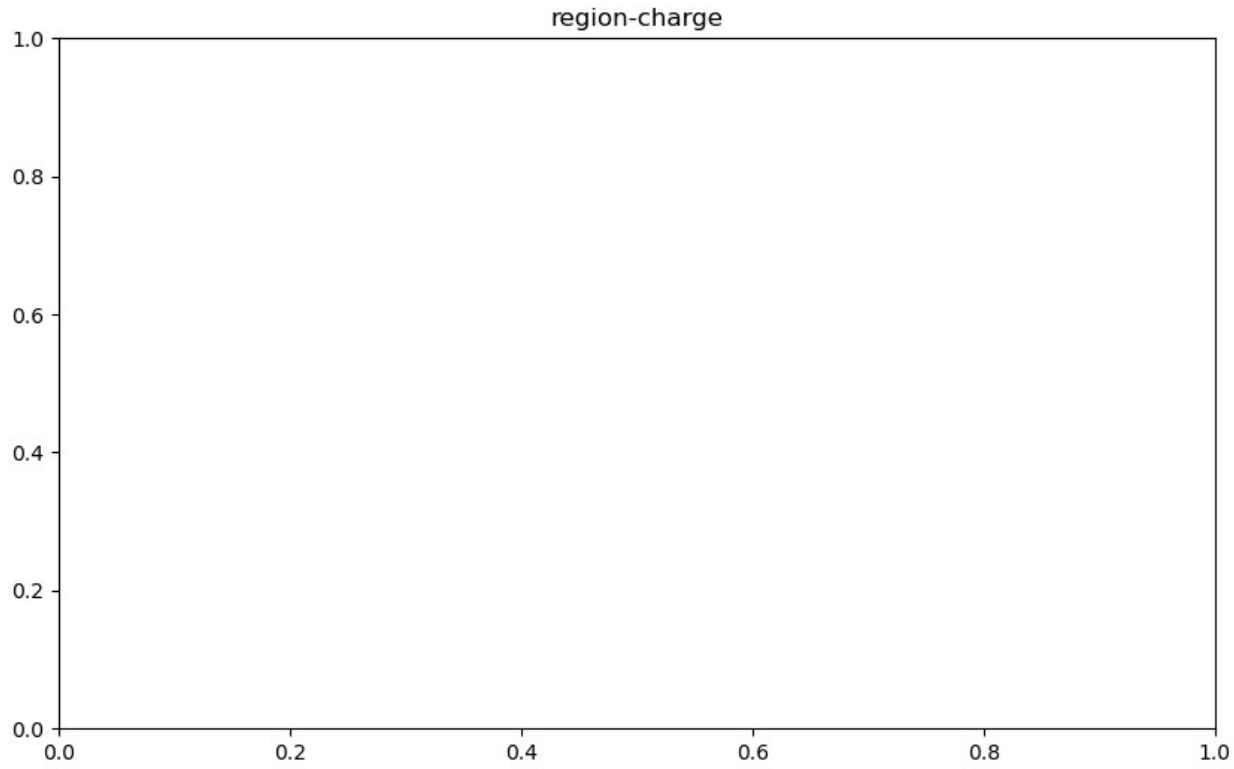

30000
25000
20000
15000
10000
5000
0
charges
age


age-charge
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

```
sns.barplot(data=data,x='region',y='charges',palette='pastel')
plt.figure(figsize=(10,6))
plt.title('region-charge')

plt.show()
```

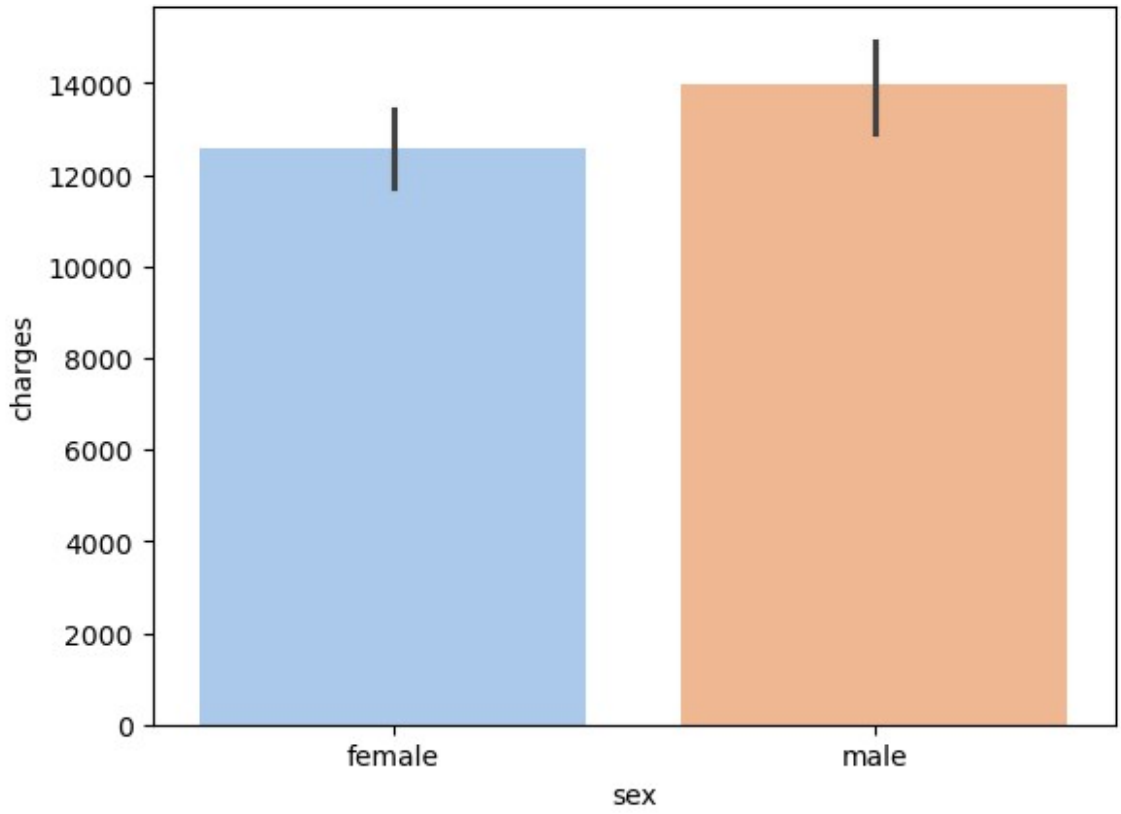
```
C:\Users\91798\AppData\Local\Temp\ipykernel_21016\2041983473.py:1:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.barplot(data=data,x='region',y='charges',palette='pastel')
```

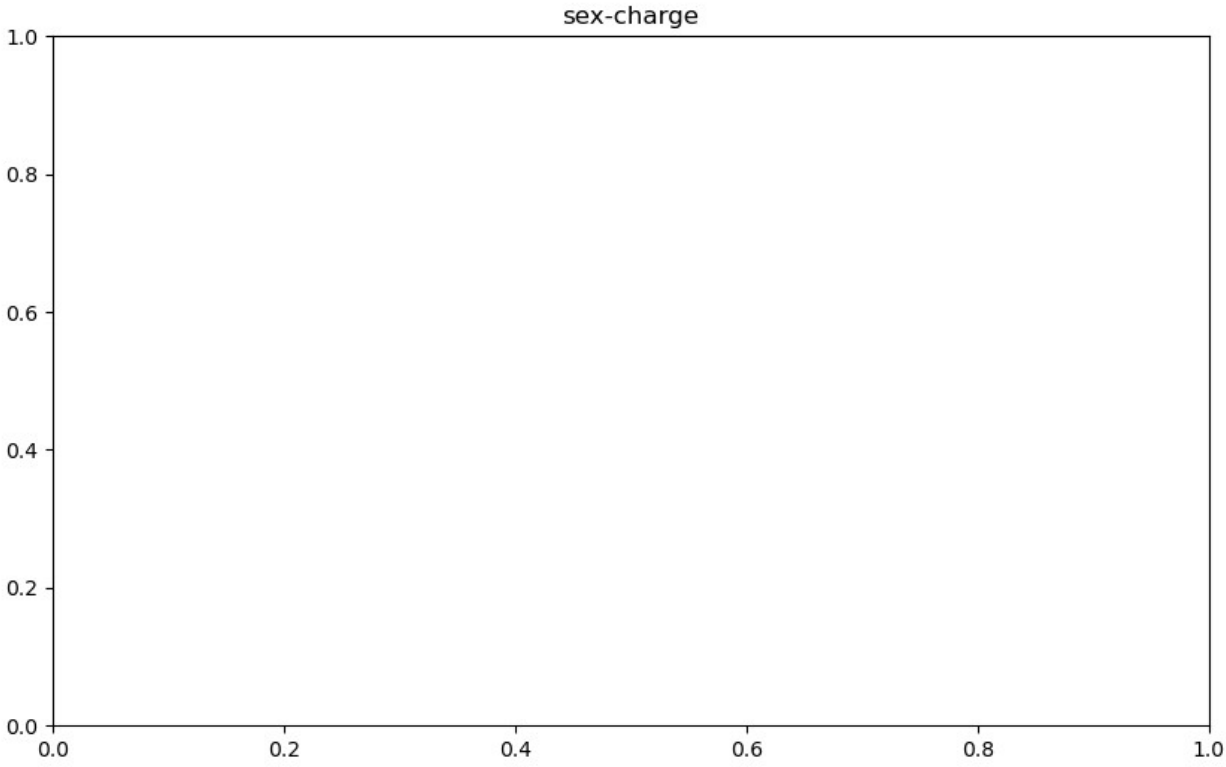





```
sns.barplot(data=data,x='sex',y='charges',palette='pastel')
plt.figure(figsize=(10,6))
plt.title('sex-charge')

plt.show()
```

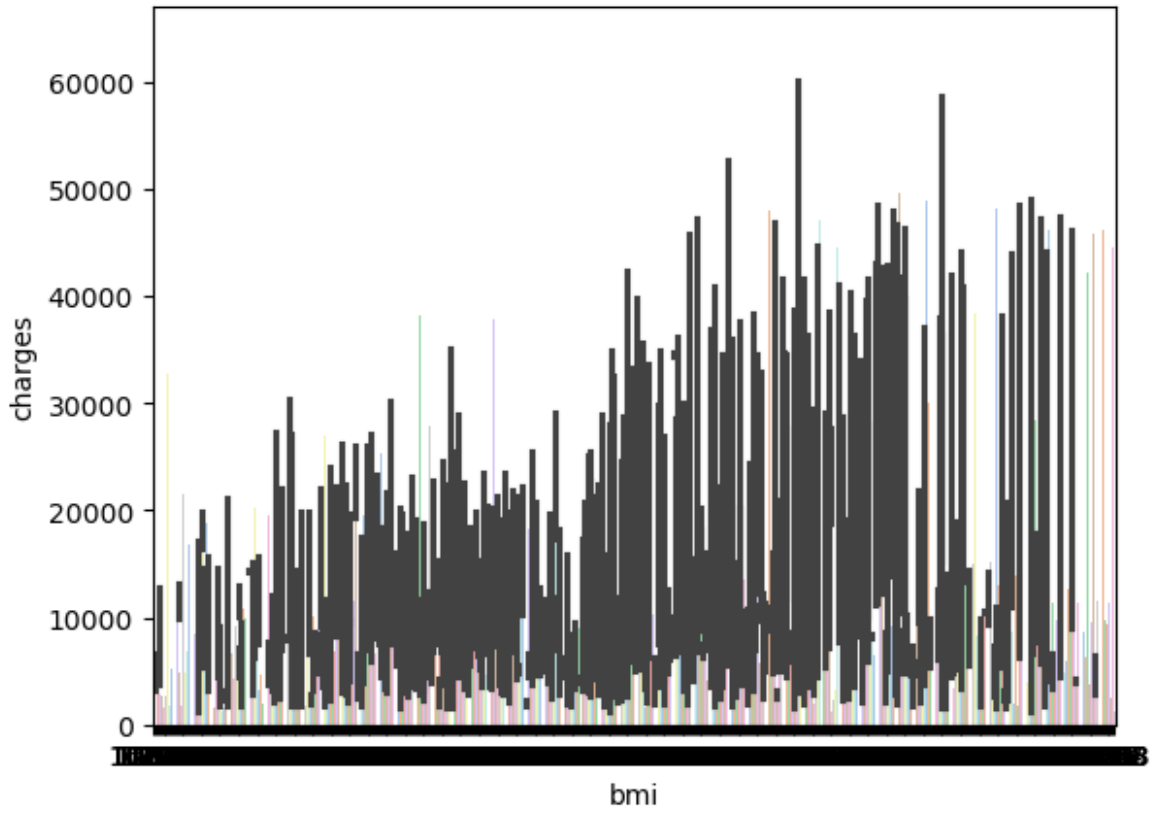
```
C:\Users\91798\AppData\Local\Temp\ipykernel_21016\188371249.py:1:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.barplot(data=data,x='sex',y='charges',palette='pastel')
```

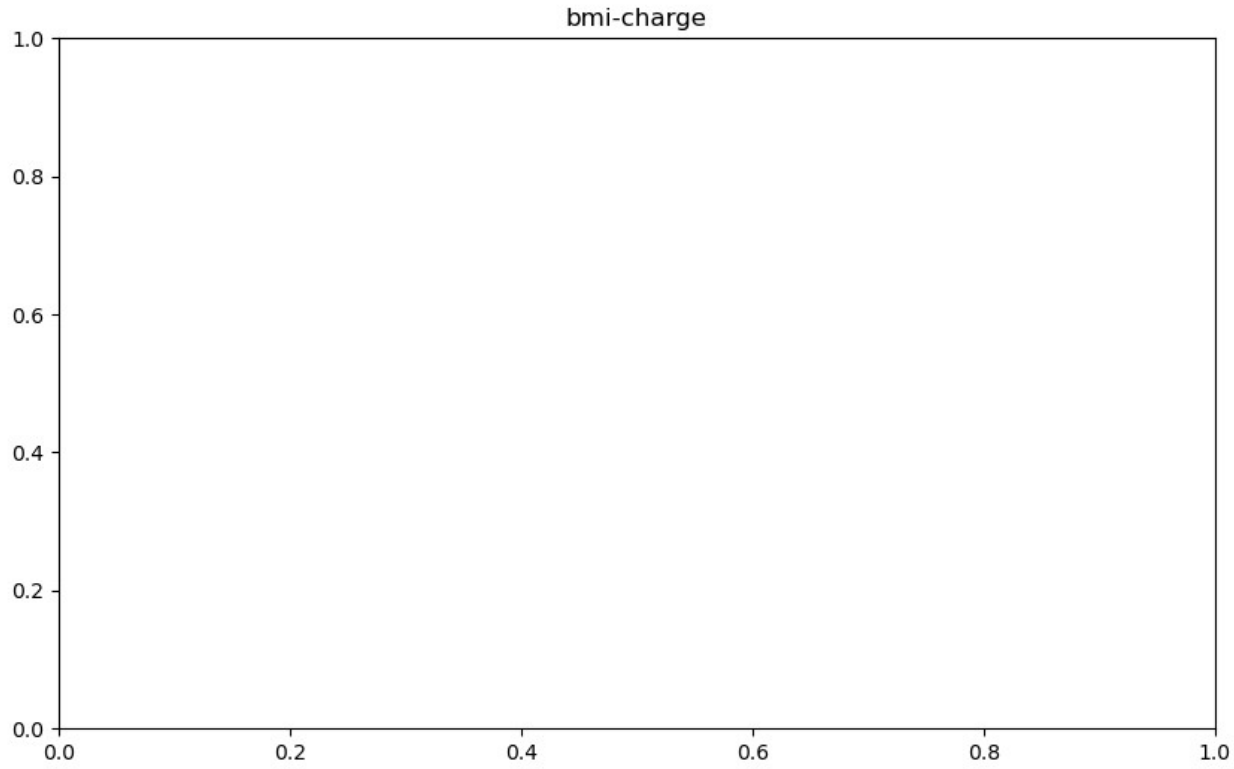

14000
12000
10000
8000
6000
4000
2000
0
charges
female
male
sex


sex-charge
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

```
sns.barplot(data=data,x='bmi',y='charges',palette='pastel',)
plt.figure(figsize=(10,6))
plt.title('bmi-charge')

plt.show()
```

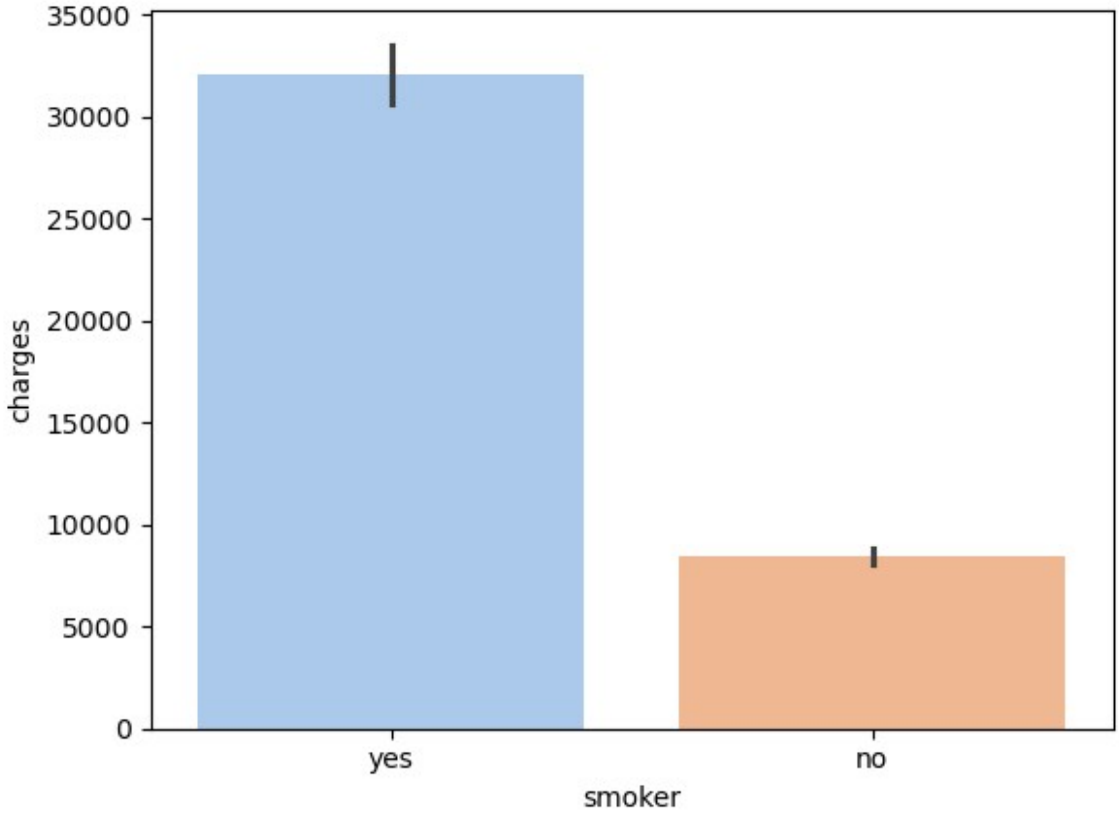
```
C:\Users\91798\AppData\Local\Temp\ipykernel_21016\475725946.py:1:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.barplot(data=data,x='bmi',y='charges',palette='pastel',)
```

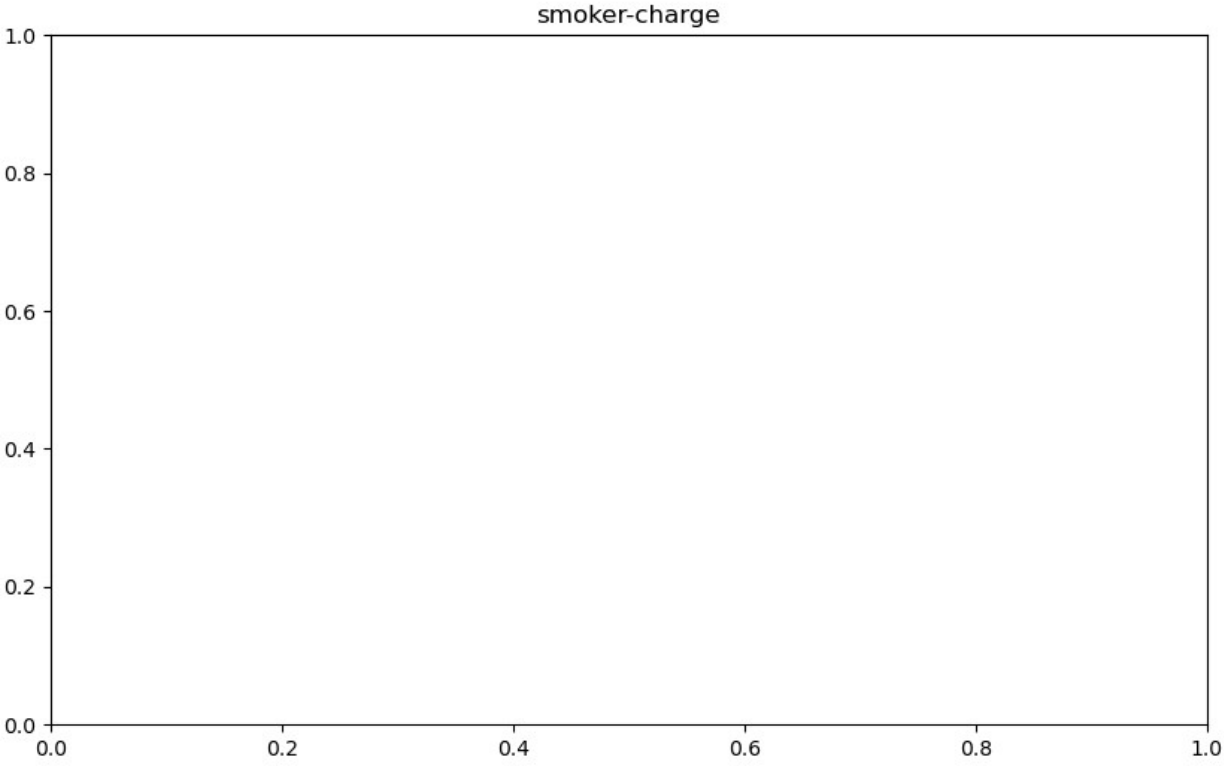





```
sns.barplot(data=data,x='smoker',y='charges',palette='pastel',)
plt.figure(figsize=(10,6))
plt.title('smoker-charge')

plt.show()
```

```
C:\Users\91798\AppData\Local\Temp\ipykernel_21016\3911709832.py:1:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.barplot(data=data,x='smoker',y='charges',palette='pastel',)
```


35000
30000
25000
20000
15000
10000
5000
0
charges
yes
no
smoker


smoker-charge
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0