```
import warnings
warnings.filterwarnings('ignore')
```

```
# Import the numpy and pandas package

import numpy as np
import pandas as pd
```

```
# Read the given CSV file, and view some sample records

medical = pd.read_csv('../content/Medical Price Dataset.csv')
medical.head()
```

	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

```
#Determining the number of rows and columns
medical.shape
```

```
(1338, 7)
```

```
medical.describe()  #summary of all the numeric columns in the dataset
```

	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

```
medical.info()  #Datatypes of each column
```

```
<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
```

```
#Checking missing values
medical.isnull().sum()
```

```
age         0
sex         0
bmi         0
children    0
smoker      0
region      0
charges     0
dtype: int64
```

```
#Mapping
medical['sex'] = medical['sex'].map({'male': 0, 'female': 1})
medical['smoker'] = medical['smoker'].map({'yes': 1, 'no': 0})
medical.head()
```

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

```
#Import necessary libraies
import matplotlib.pyplot as plt
import seaborn as sns
```

```
#Binning the age column.
bins = [17,35,55,1000]
slots = ['Young adult','Senior Adult','Elder']

medical['Age_range']=pd.cut(medical['age'],bins=bins,labels=slots)
```

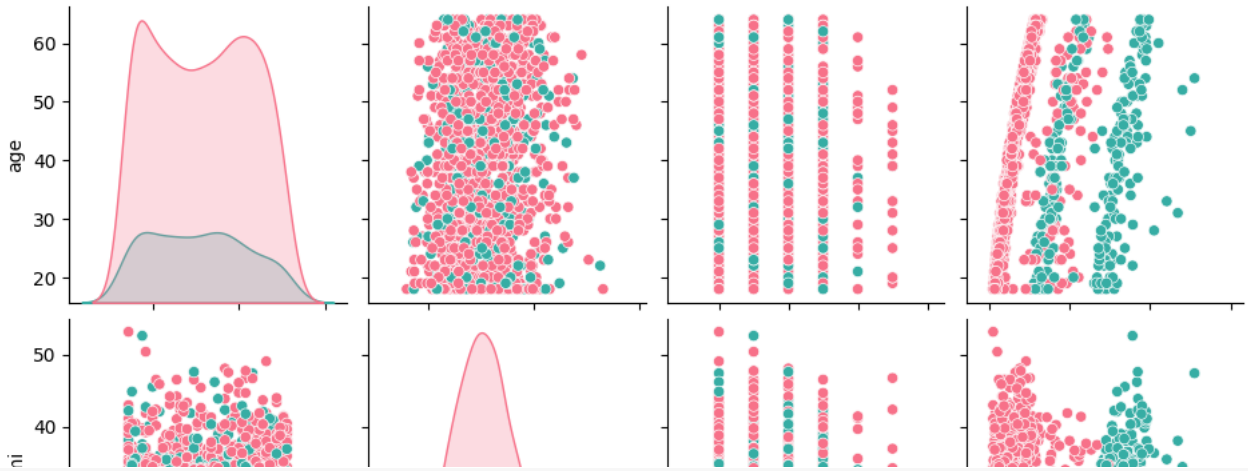
```
medical.head()
```

	age	sex	bmi	children	smoker	region	charges	Age_range
0	19	1	27.900	0	1	southwest	16884.92400	Young adult
1	18	0	33.770	1	0	southeast	1725.55230	Young adult
2	28	0	33.000	3	0	southeast	4449.46200	Young adult
3	33	0	22.705	0	0	northwest	21984.47061	Young adult
4	32	0	28.880	0	0	northwest	3866.85520	Young adult

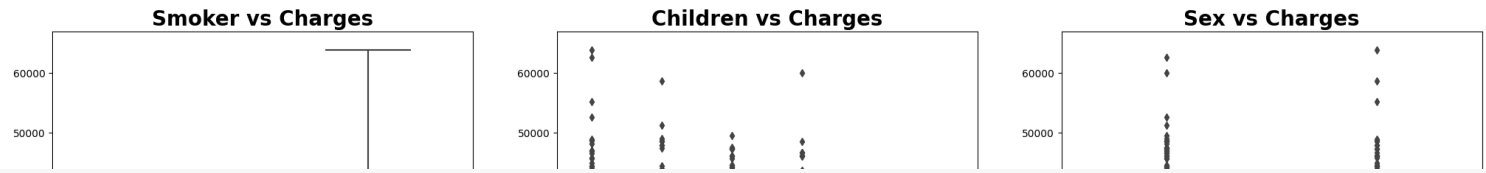
```
medical.nunique().sort_values()
```

```
sex             2
smoker          2
Age_range       3
region          4
children        6
age            47
bmi           548
charges      1337
dtype: int64
```

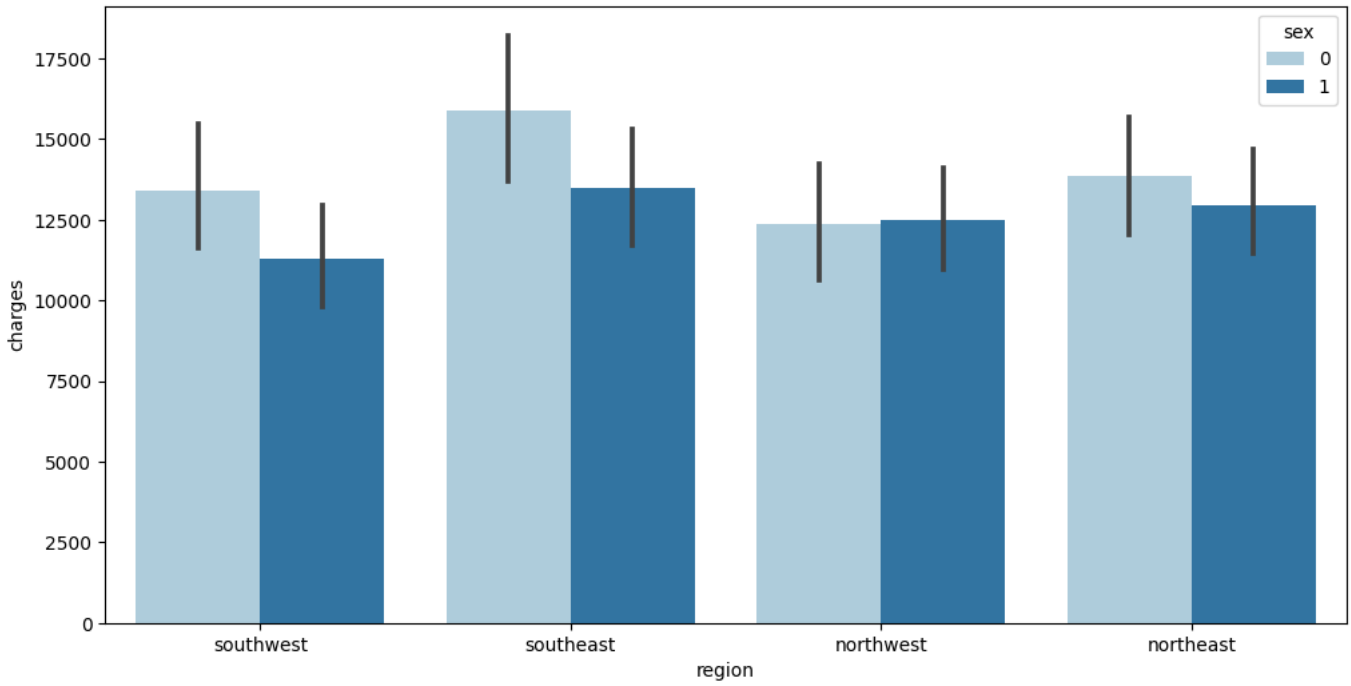
```
#Pairplot of all numerical variables
sns.pairplot(medical, vars=["age", 'bmi','children','charges'],hue='smoker',palette="husl")
plt.show()
```



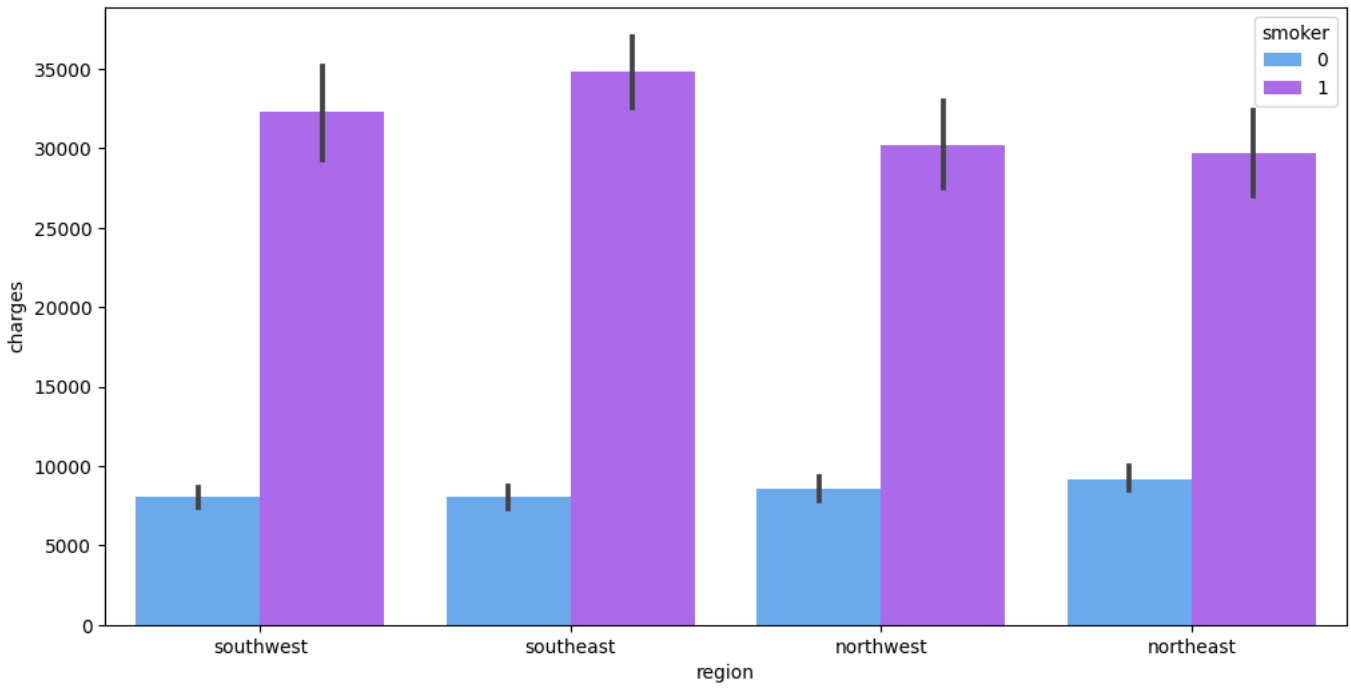

```
plt.figure(figsize=(25, 16))
plt.subplot(2,3,1)
sns.boxplot(x = 'smoker', y = 'charges', data = medical)
plt.title('Smoker vs Charges',fontweight="bold", size=20)
plt.subplot(2,3,2)
sns.boxplot(x = 'children', y = 'charges', data = medical,palette="husl")
plt.title('Children vs Charges',fontweight="bold", size=20)
plt.subplot(2,3,3)
sns.boxplot(x = 'sex', y = 'charges', data = medical, palette= 'husl')
plt.title('Sex vs Charges',fontweight="bold", size=20)
plt.subplot(2,3,4)
sns.boxplot(x = 'region', y = 'charges', data = medical,palette="bright")
plt.title('Region vs Charges',fontweight="bold", size=20)
plt.subplot(2,3,5)
sns.boxplot(x = 'Age_range', y = 'charges', data = medical, palette= 'husl')
plt.title('Age vs Charges',fontweight="bold", size=20)
plt.show()
```



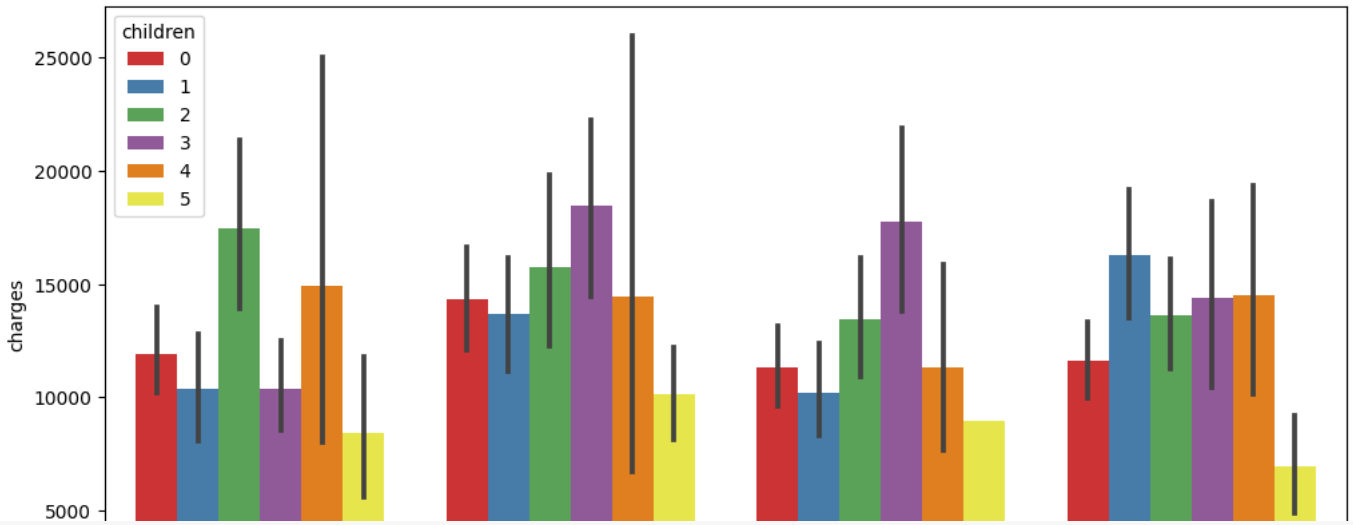

```
plt.figure(figsize=(12,6))
sns.barplot(x='region', y='charges', hue='sex', data=medical, palette='Paired')
plt.show()
```



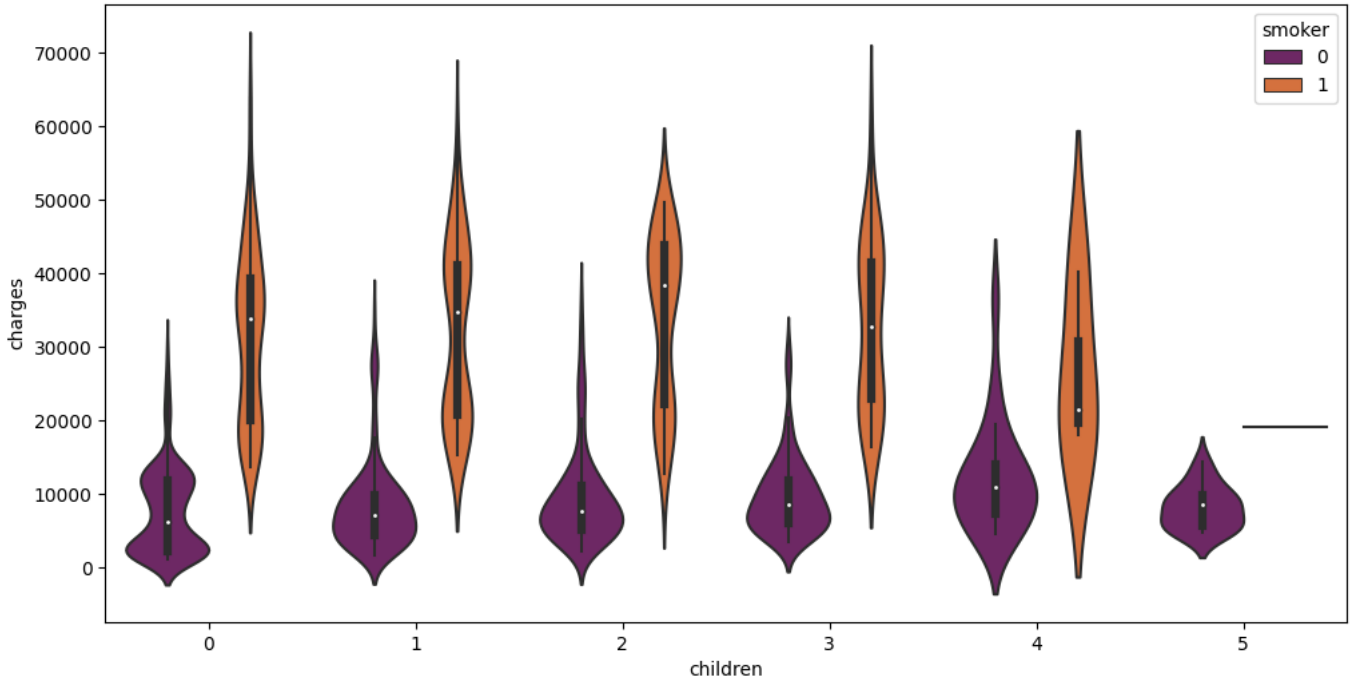

```
plt.figure(figsize=(12,6))
sns.barplot(x = 'region', y = 'charges',hue='smoker', data=medical, palette='cool')
plt.show()
```



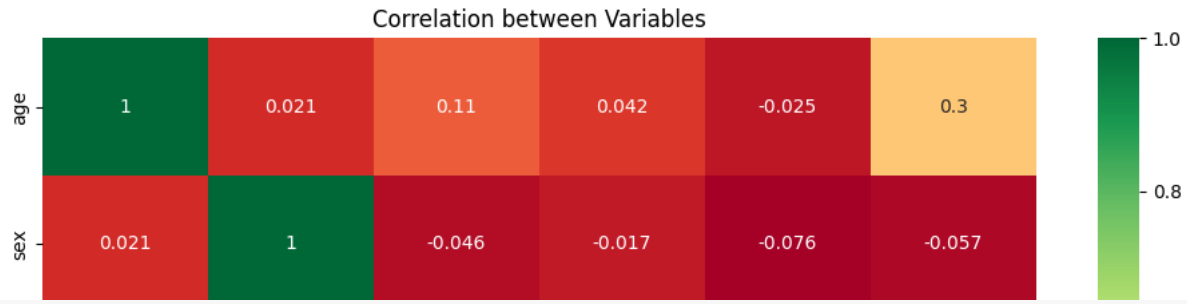

```
plt.figure(figsize=(12,6))
sns.barplot(x='region', y='charges', hue='children', data=medical, palette='Set1')
plt.show()
```




```
plt.figure(figsize=(12,6))
sns.violinplot(x = 'children', y = 'charges', data=medical, hue='smoker', palette='inferno')
plt.show()
```




```
#Heatmap to see correlation between variables
plt.figure(figsize=(12, 8))
sns.heatmap(medical.corr(), cmap='RdYlGn', annot = True)
plt.title("Correlation between Variables")
plt.show()
```




```
medical.head()
```

	age	sex	bmi	children	smoker	region	charges	Age_range
0	19	1	27.900	0	1	southwest	16884.92400	Young adult
1	18	0	33.770	1	0	southeast	1725.55230	Young adult
2	28	0	33.000	3	0	southeast	4449.46200	Young adult
3	33	0	22.705	0	0	northwest	21984.47061	Young adult
4	32	0	28.880	0	0	northwest	3866.85520	Young adult

```
region=pd.get_dummies(medical.region,drop_first=True)
Age_range=pd.get_dummies(medical.Age_range,drop_first=True)
children= pd.get_dummies(medical.children,drop_first=True,prefix='children')
```

```
medical=pd.concat([region,Age_range,children,medical],axis=1)
medical.head()
```

	northwest	southeast	southwest	Senior Adult	Elder	children_1	children_2	children_3	children_4	children_5	age	sex	bmi	children	smoker	
0	0	0	1	0	0	0	0	0	0	0	19	1	27.900	0	1	s
1	0	1	0	0	0	1	0	0	0	0	18	0	33.770	1	0	s
2	0	1	0	0	0	0	0	1	0	0	28	0	33.000	3	0	s

```
medical.drop(['region', 'Age_range', 'age','children'], axis = 1, inplace = True)
medical.head()
```

	northwest	southeast	southwest	Senior Adult	Elder	children_1	children_2	children_3	children_4	children_5	sex	bmi	smoker	charges
0	0	0	1	0	0	0	0	0	0	0	1	27.900	1	16884.92400
1	0	1	0	0	0	1	0	0	0	0	0	33.770	0	1725.55230
2	0	1	0	0	0	0	0	1	0	0	0	33.000	0	4449.46200
3	1	0	0	0	0	0	0	0	0	0	0	22.705	0	21984.47061
4	1	0	0	0	0	0	0	0	0	0	0	28.880	0	3866.85520

```
medical.shape
```

```
(1338, 14)
```

```
plt.figure(figsize=(15, 10))
sns.heatmap(medical.corr(), cmap='YlGnBu', annot = True)
plt.show()
```


```
from sklearn.model_selection import train_test_split

# We specify this so that the train and test data set always have the same rows, respectively
#np.random.seed(0)
medical_train, medical_test = train_test_split(medical, train_size = 0.7, random_state = 100)
```

```
print(medical_train.shape)
print(medical_test.shape)
```

```
(936, 14)
(402, 14)
```

```
from sklearn.preprocessing import MinMaxScaler
```

```
medical.head()
```

	northwest	southeast	southwest	Senior Adult	Elder	children_1	children_2	children_3	children
0	0	0	1	0	0	0	0	0	
1	0	1	0	0	0	1	0	0	
2	0	1	0	0	0	0	0	1	
3	1	0	0	0	0	0	0	0	

+ Code + Text

```
#Instantiate an object
scaler = MinMaxScaler()

#Create a list of numeric variables
num_vars=['bmi','charges']

#Fit on data
medical_train[num_vars] = scaler.fit_transform(medical_train[num_vars])
medical_train.head()
```

	northwest	southeast	southwest	Senior Adult	Elder	children_1	children_2	children_3	children_4	children_5	sex	bmi	smoker	charges
966	1	0	0	1	0	0	1	0	0	0	0	0.237692	1	0.364661
522	0	0	0	1	0	0	0	0	0	0	1	0.483051	0	0.139579
155	1	0	0	1	0	0	0	0	0	0	0	0.633844	0	0.093008
671	0	0	0	0	0	0	0	0	0	0	1	0.408932	0	0.045040
1173	1	0	0	1	0	0	1	0	0	0	0	0.357815	0	0.085173

0s completed at 5:31 PM