PROJECT TITLE: PERSONAL MEDICAL COST PREDICTION MODEL USING LINEAR REGRESSION

DOMAIN: DATA ANALYTICS

NAME: PRATHIBA.D

DEPARTMENT: COMPUTER SCIENCE

PROJECT OVERVIEW

CONTENTS

DIYA PROJECT **PRATHIBA.D**

MACHINE LEARNING MODEL USING LINEAR REGRESSION

Linear Regression Model:

Linear regression algorithm shows a linear relationship between a dependent (y) and one or more independent (y) variables. It is one of the easiest and most popular machine learning algorithms and a statistical method used for predictive analysis.

The main aim of this model is to find the best fit linear line between the dependent and independent variable.

Linear Regression is of two types: Simple and Multiple.

Simple Linear Regression is where only one independent variable is present and the model has to find the linear relationship of it with the dependent variable

Equation of Simple Linear Regression, where b_o is the intercept, b_1 is coefficient or slope, x is the independent variable and y is the dependent variable.

$$y=b_0+b_1x$$

In Multiple Linear Regression there are more than one independent variables for the model to find the relationship.

Equation of Multiple Linear Regression, where bo is the intercept, $b_1,b_2,b_3,b_4\ldots,b_n$ are coefficients or slopes of the independent variables $x_1,x_2,x_3,x_4\ldots,x_n$ and y is the dependent variable.

$$y=b_0+b_1x_1+b_2x_2+b_3x_3\ldots.+b_nx_n$$

PERSONAL MEDICAL COST PREDICTION MODEL

ABSTRACT

Everyone's life revolves around their health. Good health is an asset. Health refers to a person's ability to cope up with the environment on a physical, emotional, mental, and social level. Changes in food habits, high levels of stress, sleeplessness, alcohol consumption , smoking and numerous other factors tend to exploit human health. As a consequence, when people become ill a lot of medical expenses are incurred.

So, a machine learning model can be can be built which can make people understand the factors that make them unfit thereby creating a lot of medical expenses. It could identify and estimate medical expense if someone has such factors.

OBJECTIVE

· Predict the future medical expenses of subjects based on certain features by building a robust machine learning model.

· Identifying the factors affecting the medical expenses of the subjects based on the model's output.

DATASET DESCRIPTION

The data has been imported from the Kaggle data science platform. This dataset contains 1338 rows and 7 columns. The columns present in the dataset are 'age',' sex','bmi', 'children', smoker', 'region', and 'expenses'.

Charges

The Expenses column is the target column and the rest others are independent columns. Independent columns are those which will predict the outcome.

Age

The first column is Age. Age is an important factor for predicting medical expenses because young people are generally healthier than old ones and the medical expenses for young people will be lesser compared to old people.

Sex

The Next column is sex, which has two Categories in this column: Male and Female. The sex of the person can also play a vital role in predicting the medical expenses of a subject.

BMI

The third column is the BMI is Body Mass Index.An ideal BMI is in the 18.5 to 24.9 range. People with very low or very high ‘bmi’ are more likely to require medical assistance, resulting in higher costs.

Children

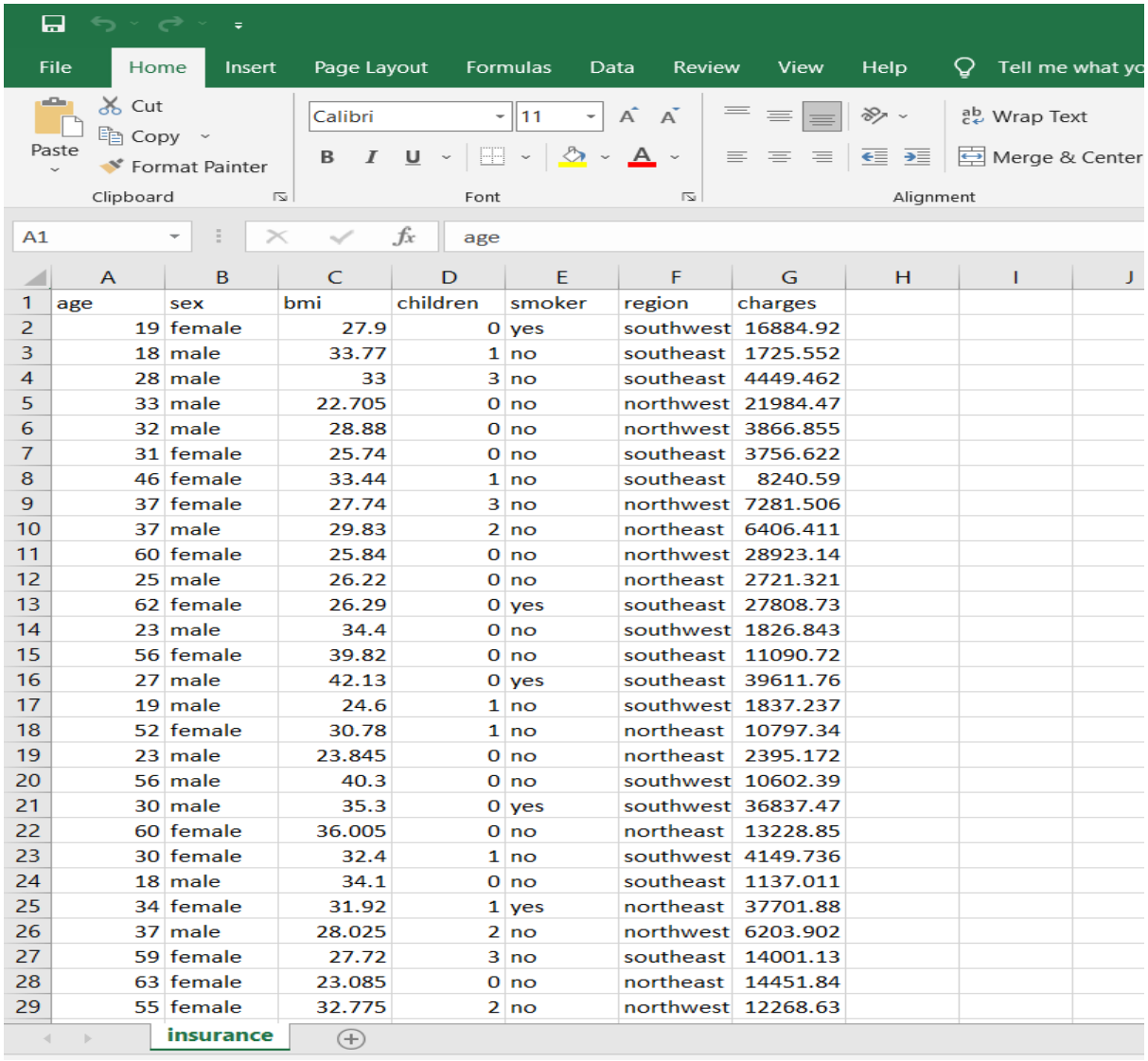
The fourth column is the ‘children’ column, which contains information on how many children the patients have. Persons who have children are under more pressure because of their children’s education, and other needs than people who do not have children.

Smoker

The fifth is the ‘smoker’ column. The Smoking factor is also considered to be one of the most important factors as the people who smoke are always at risk when their age reaches 50 to 60.

Region

Some regions are hygienic, clean, tidy and prosperous but some regions are not, and this information affects health which is related to medical expenses.



	A	B	C	D	E	F	G
1	age	sex	bmi	children	smoker	region	charges
2	19	female	27.9	0	yes	southwest	16884.92
3	18	male	33.77	1	no	southeast	1725.552
4	28	male	33	3	no	southeast	4449.462
5	33	male	22.705	0	no	northwest	21984.47
6	32	male	28.88	0	no	northwest	3866.855
7	31	female	25.74	0	no	southeast	3756.622
8	46	female	33.44	1	no	southeast	8240.59
9	37	female	27.74	3	no	northwest	7281.506
10	37	male	29.83	2	no	northeast	6406.411
11	60	female	25.84	0	no	northwest	28923.14
12	25	male	26.22	0	no	northeast	2721.321
13	62	female	26.29	0	yes	southeast	27808.73
14	23	male	34.4	0	no	southwest	1826.843
15	56	female	39.82	0	no	southeast	11090.72
16	27	male	42.13	0	yes	southeast	39611.76
17	19	male	24.6	1	no	southwest	1837.237
18	52	female	30.78	1	no	northeast	10797.34
19	23	male	23.845	0	no	northeast	2395.172
20	56	male	40.3	0	no	southwest	10602.39
21	30	male	35.3	0	yes	southwest	36837.47
22	60	female	36.005	0	no	northeast	13228.85
23	30	female	32.4	1	no	southwest	4149.736
24	18	male	34.1	0	no	southeast	1137.011
25	34	female	31.92	1	yes	northeast	37701.88
26	37	male	28.025	2	no	northwest	6203.902
27	59	female	27.72	3	no	southeast	14001.13
28	63	female	23.085	0	no	northeast	14451.84
29	55	female	32.775	2	no	northwest	12268.63

Personal medical cost prediction dataset

RESULTS IMPLEMENTATION

In [43]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly.graph_objects as go
import warnings
warnings.filterwarnings('ignore')

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from  sklearn import metrics
```

Importing required python libraries

In [14]:
```
#loading the medical insurance data from a csv file to a pandas dataframe
insurance_data_path='../Raw data/insurance.csv'
```

Storing the csv file path

In [15]:
```
insurance_data= pd.read_csv(insurance_data_path)
```

In [16]:
```
#first 5 rows of the dataframe
insurance_data.head()
```

Out[16]:

	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 [17]:
```
#last 5 rows of the dataframe
insurance_data.tail()
```

Out[17]:

	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

Reading the csv file into a dataframe using pandas

```
In [18]:
1 insurance_data.size
```

```
Out[18]: 9366
```

```
In [19]:
1 #number of rows and columns in the dataset
2 insurance_data.shape
```

```
Out[19]: (1338, 7)
```

```
In [20]:
1 insurance_data.columns
```

```
Out[20]: Index(['age', 'sex', 'bmi', 'children', 'smoker', 'region', 'charges'], dtype='object')
```

Dimensions of the crop dataframe

```
In [22]:
1 #information about the dataset
2 insurance_data.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
```

Concise summary of the crop dataframe

```
In [23]:
1 #statistical measures of the dataset
2 insurance_data.describe()
```

Out[23]:

	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

Statistics computed for the crop dataframe

```
In [44]:
1 insurance_data.isnull().sum()
```

Out[44]:

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

Checking for missing values

DATA VISUALIZATION

<u>Visualization Libraries Used:</u>

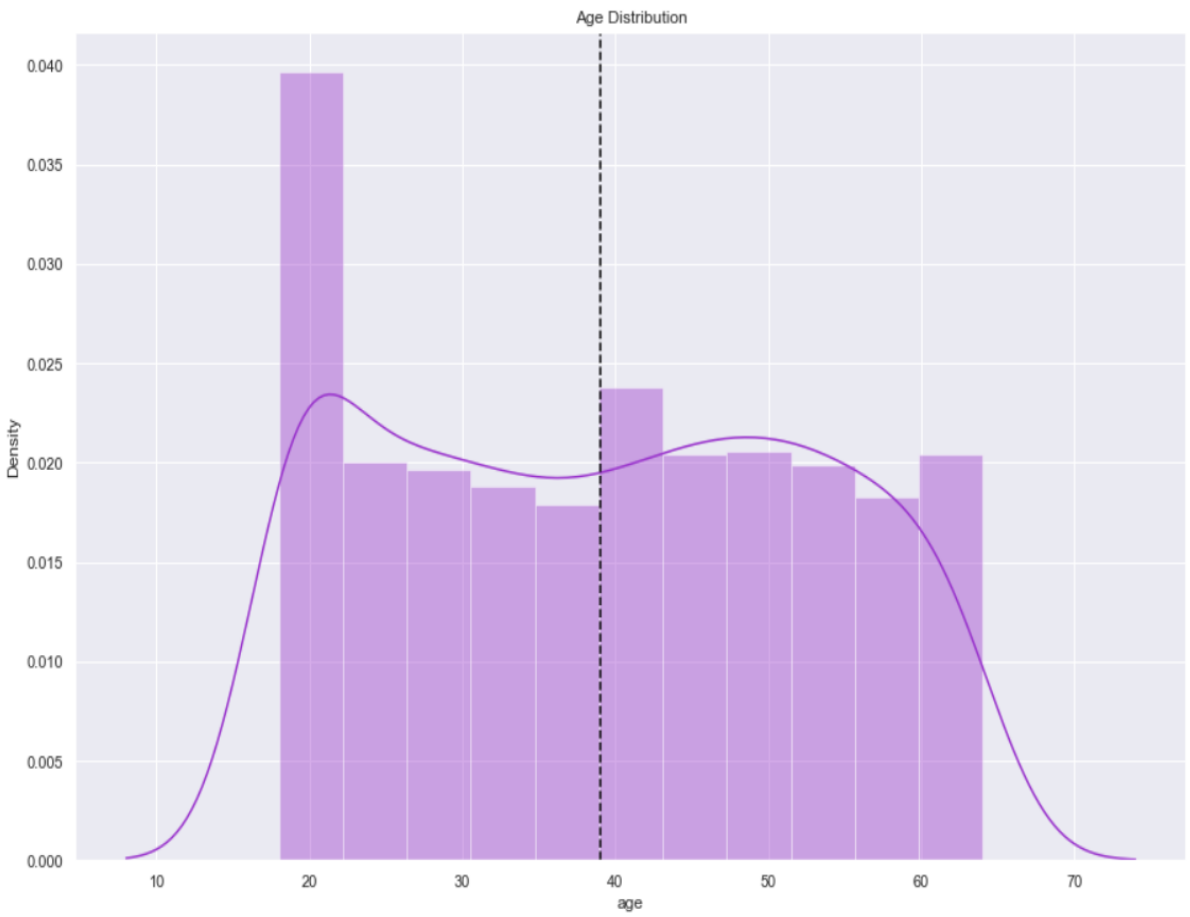
- **Matplotlib**
- **Seaborn**
- **Plotly**

UNIVARIATE ANALYSIS

Uni means one and variate means variable. In univariate analysis, there is only one dependable variable. The objective of univariate analysis is to derive the data, define and summarize it, and analyze the pattern present in it. In a dataset, it explores each variable separately.

Distribution plot using Seaborn

In [45]:
```
#distribution of age values
sns.set()
colour='darkorchid'
plt.figure(figsize=(15,10))
sns.distplot(insurance_data['age'],color=colour)
plt.axvline(39, linestyle = '--', color = 'black', label = 'mean Age')
plt.title('Age Distribution')
plt.show()
```



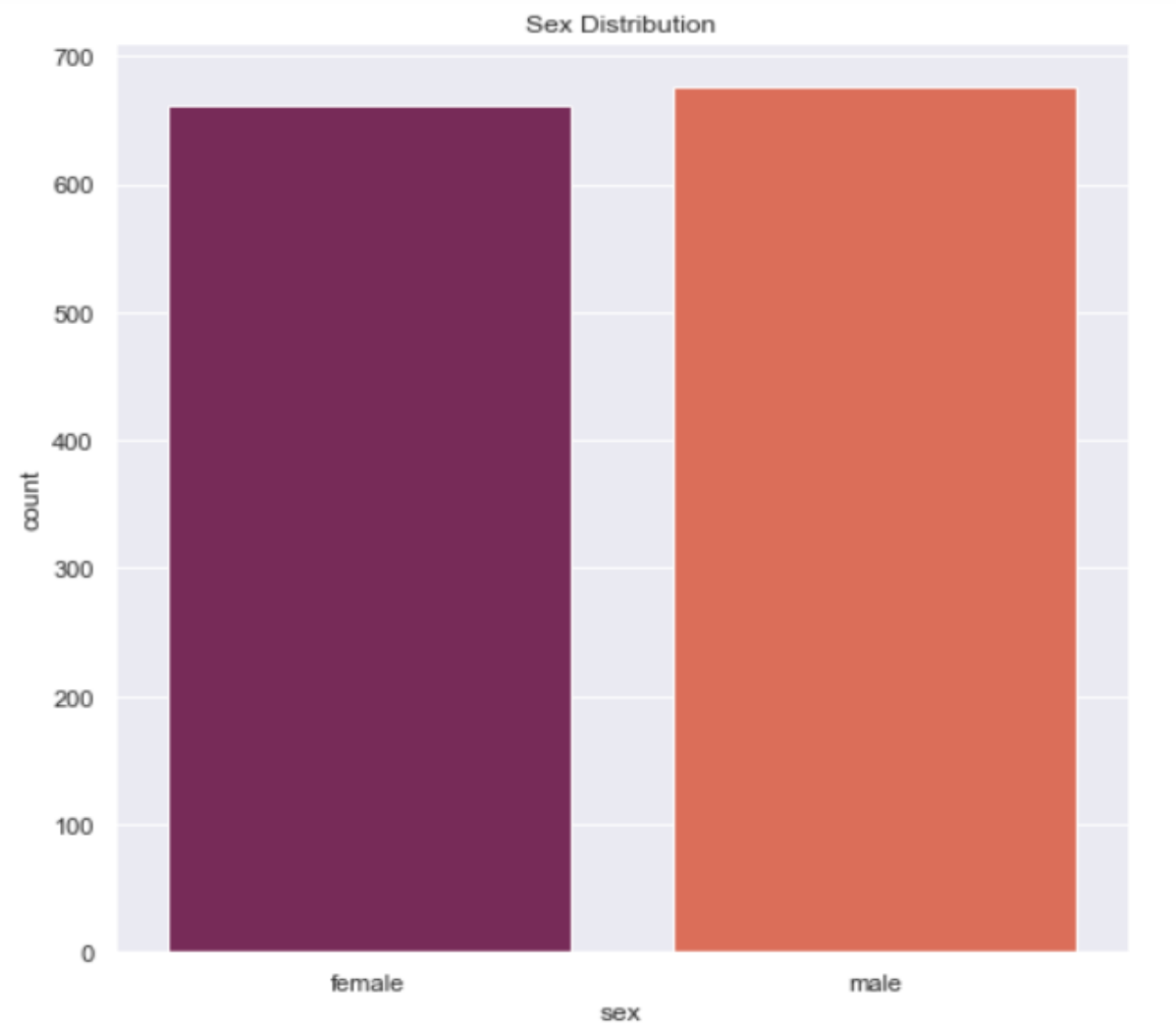
Distribution of Age values

Countplot using Seaborn

In [26]:
```
#Countplot for gender column
plt.figure(figsize=(8,8))
sns.countplot(x='sex',data=insurance_data,palette='rocket')
plt.title('Sex Distribution')
plt.show()
```

In [27]:
```
insurance_data['sex'].value_counts()
```

Out[27]:
```
male      676
female    662
Name: sex, dtype: int64
```



Countplot of gender column

Distribution plot using seaborn

In [28]:

```
#bmi distribution
sns.set()
plt.figure(figsize=(15,10))
sns.distplot(insurance_data['bmi'],color="navy")
plt.title('BMI Distribution')
plt.show()
```

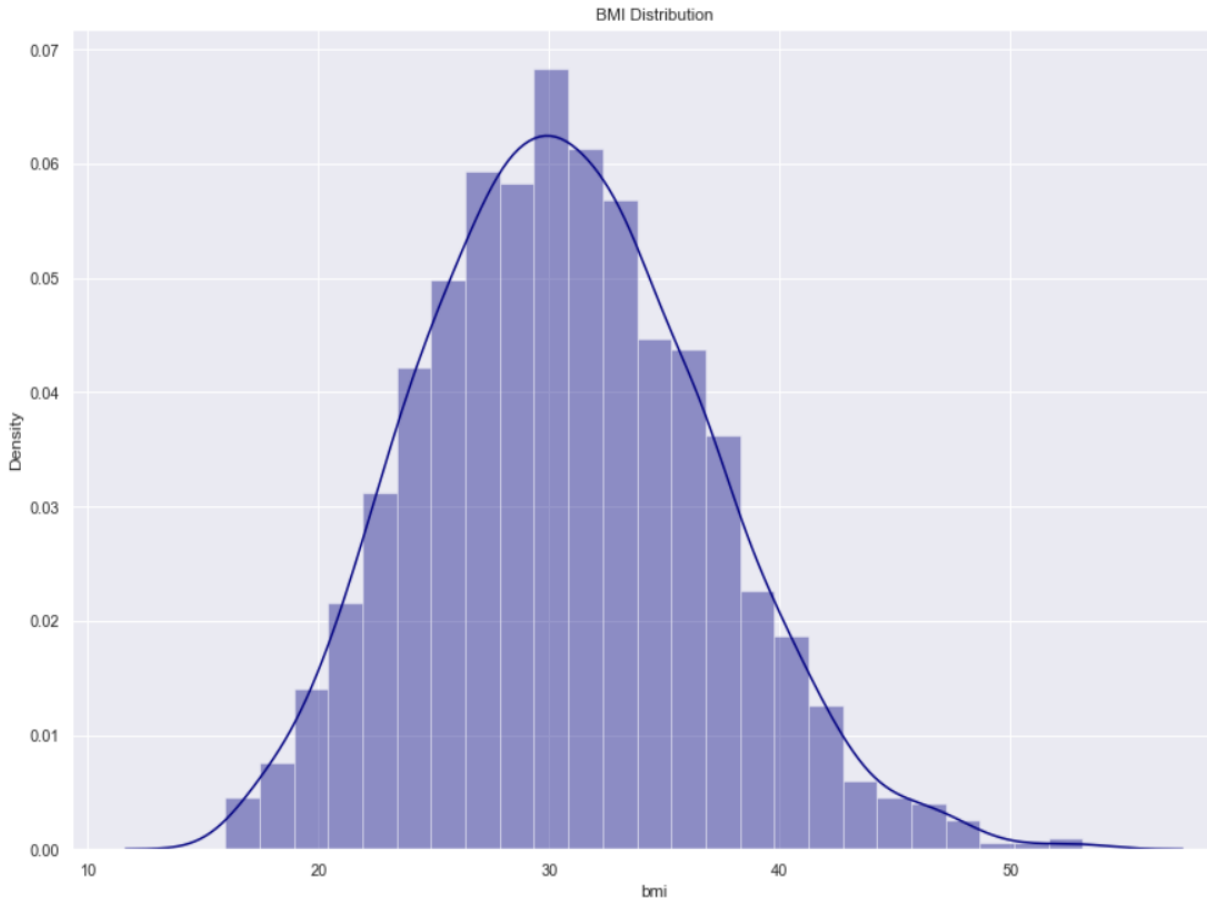
BMI Weight classification

Below 18.---------->Underweight

18.5-24.9---------->Normal

25.0-29.9---------->Overweight

30.0 or higher------>Obese



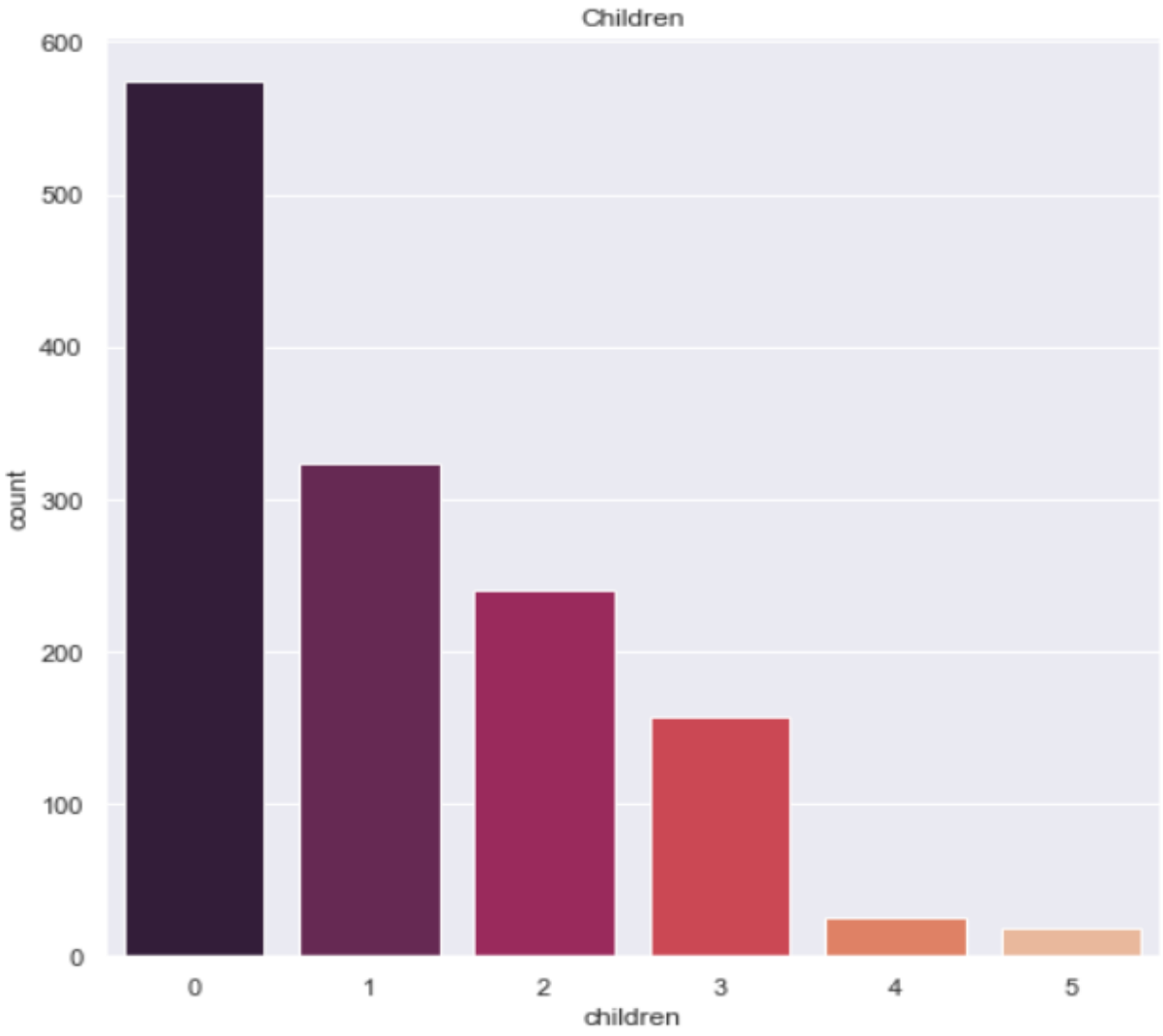
BMI Distribution

Countplot using seaborn and matplotlib

In [29]:
```
plt.figure(figsize=(8,8))
sns.countplot(x='children',data=insurance_data,palette="rocket")
plt.title('Children')
plt.show()
```

In [30]:
```
insurance_data['children'].value_counts()
```

Out[30]:
```
0    574
1    324
2    240
3    157
4     25
5     18
Name: children, dtype: int64
```



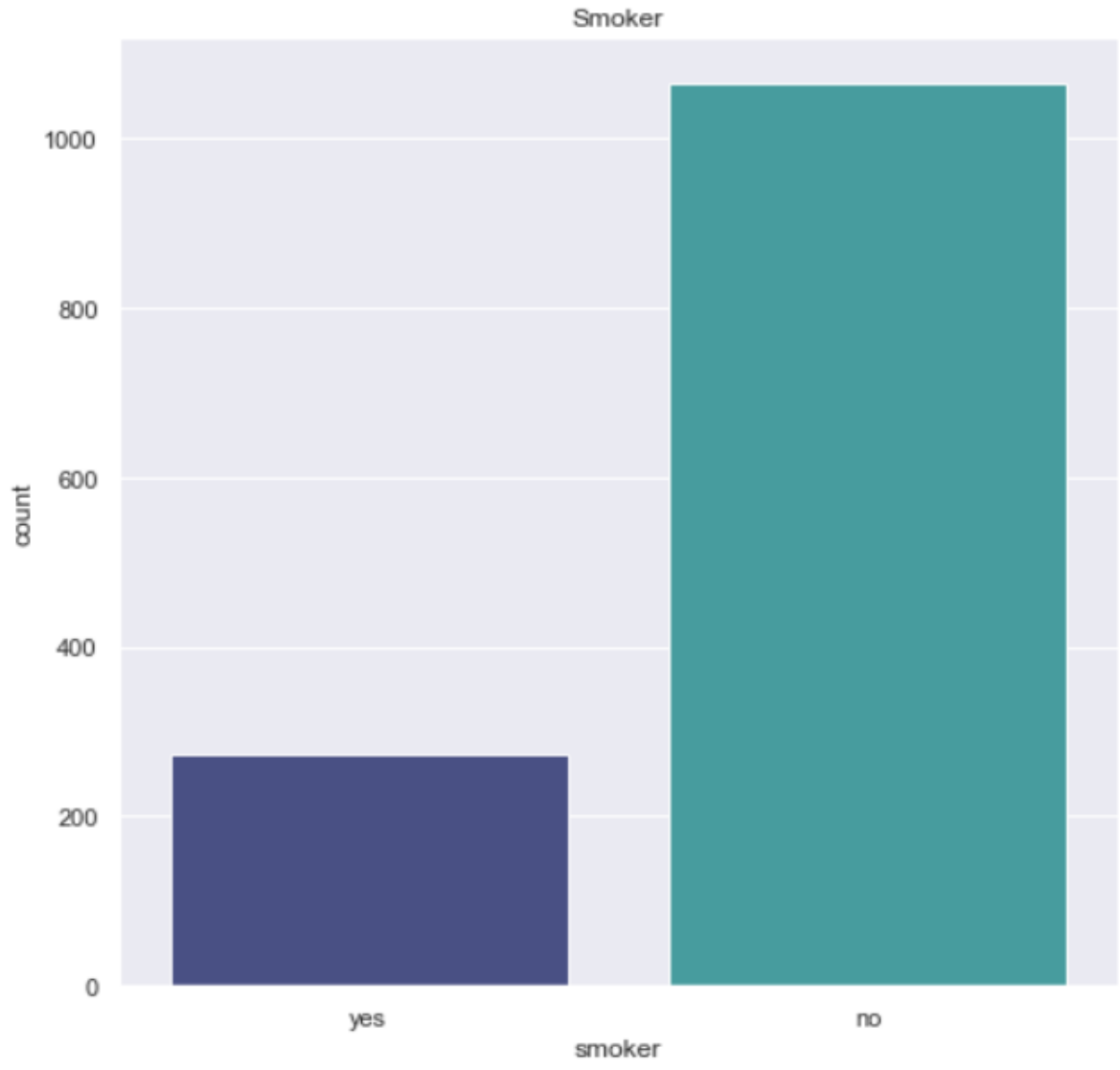
Countplot of children column

Countplot using seaborn and matplotlib

```
In [31]:
1 plt.figure(figsize=(8,8))
2 sns.countplot(x='smoker',data=insurance_data,palette="mako")
3 plt.title('Smoker')
4 plt.show()
```

```
In [32]:
1 insurance_data['smoker'].value_counts()
```

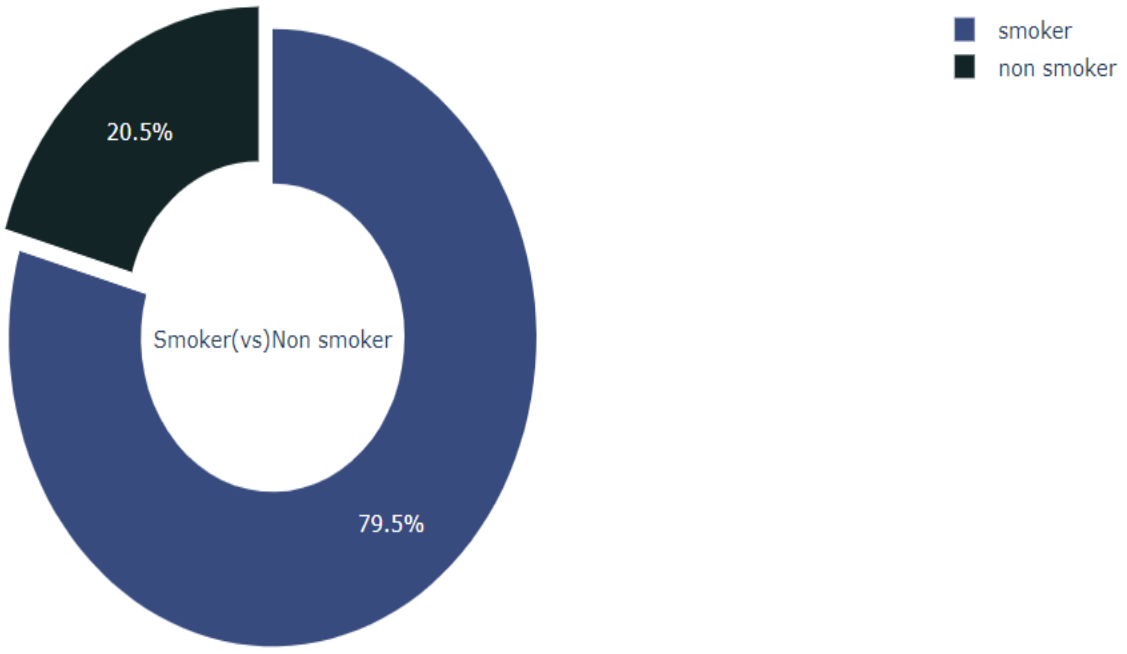
```
Out[32]: no     1064
         yes     274
         Name: smoker, dtype: int64
```



Countplot of smoker column

Pie chart using plotly

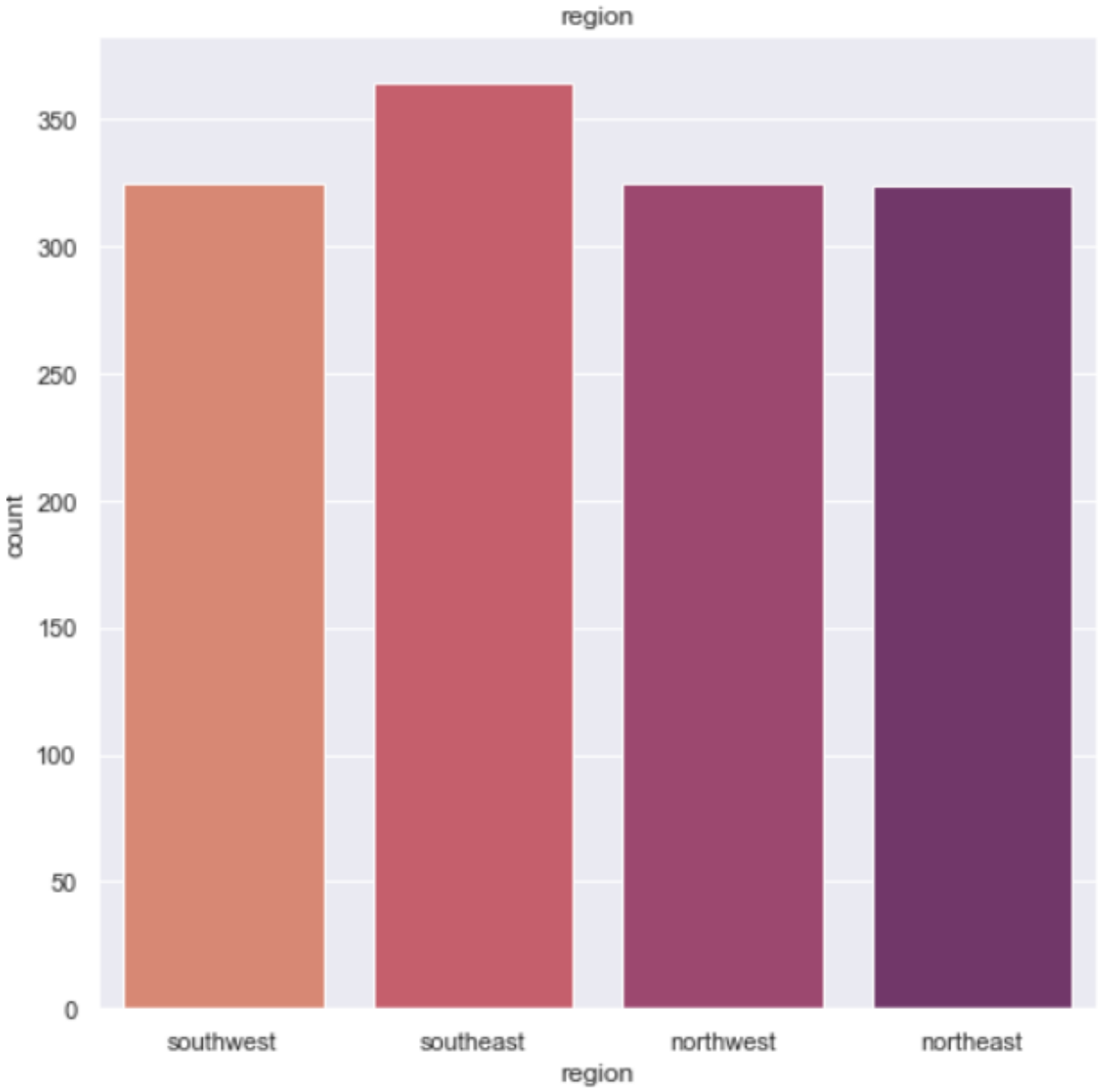
In [33]:
```
night_colors = ['rgb(56, 75, 126)', 'rgb(18, 36, 37)', 'rgb(34, 53, 101)',
                'rgb(36, 55, 57)', 'rgb(6, 4, 4)']
values = insurance_data['smoker'].value_counts().to_list()
labels=['smoker','non smoker']
fig = go.Figure(go.Pie(labels=labels,values= values,name='smoker',hole = 0.5,title='Smoker(vs)Non smoker',pull=[0,0.09],mark
fig.show()
```



Pie chart depicting percentage of smokers and non-smokers

Countplot using seaborn and matplotlib

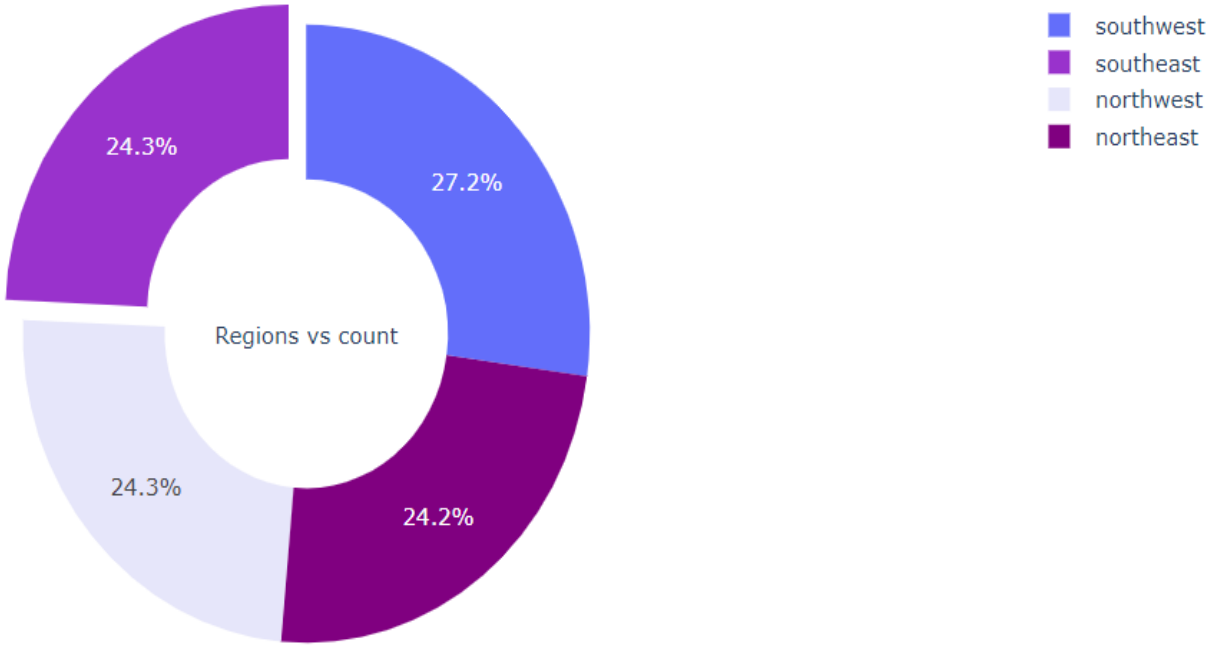
In [34]:
```
plt.figure(figsize=(8,8))
sns.countplot(x='region',data=insurance_data,palette='flare')
plt.title('region')
plt.show()
```



Countplot for regions

Pie chart using plotly

In [47]:
```
colors=['darkcoral','darkorchid','lavender','purple']
values = insurance_data['region'].value_counts().to_list()
labels=['southwest','southeast','northwest','northeast']
fig = go.Figure(go.Pie(labels=labels,values= values,hole = 0.5,title='Regions vs count',pull=[0,0.09,0,0],marker_colors=colo
fig.show()
```



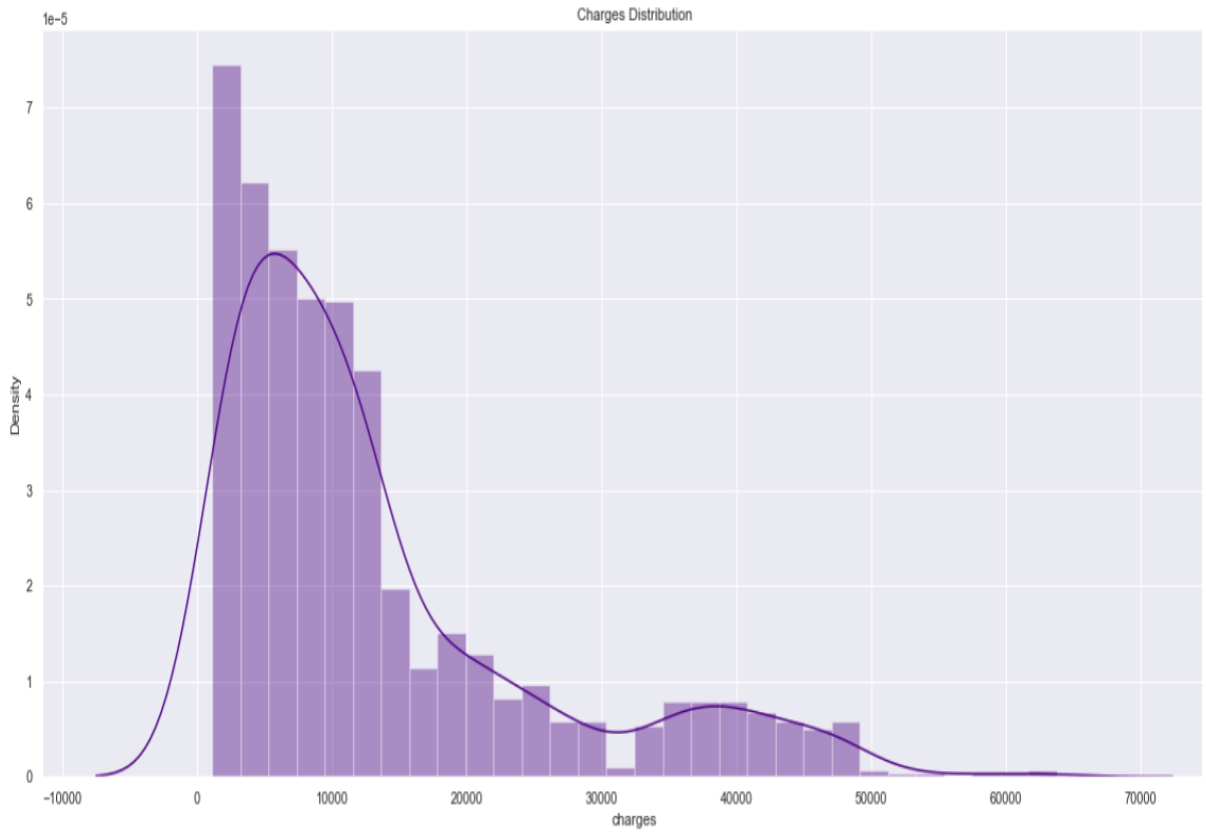
Pie chart depicting various regions and their count

```
In [36]: 1 insurance_data['region'].value_counts()

Out[36]: southeast    364
         southwest    325
         northwest    325
         northeast    324
         Name: region, dtype: int64
```

Distribution plot using seaborn

```
In [37]: 1 #charges distribution
         2 sns.set()
         3 plt.figure(figsize=(20,9))
         4 sns.distplot(insurance_data['charges'],color='indigo')
         5 plt.title('Charges Distribution')
         6 plt.show()
```



Distribution of charges

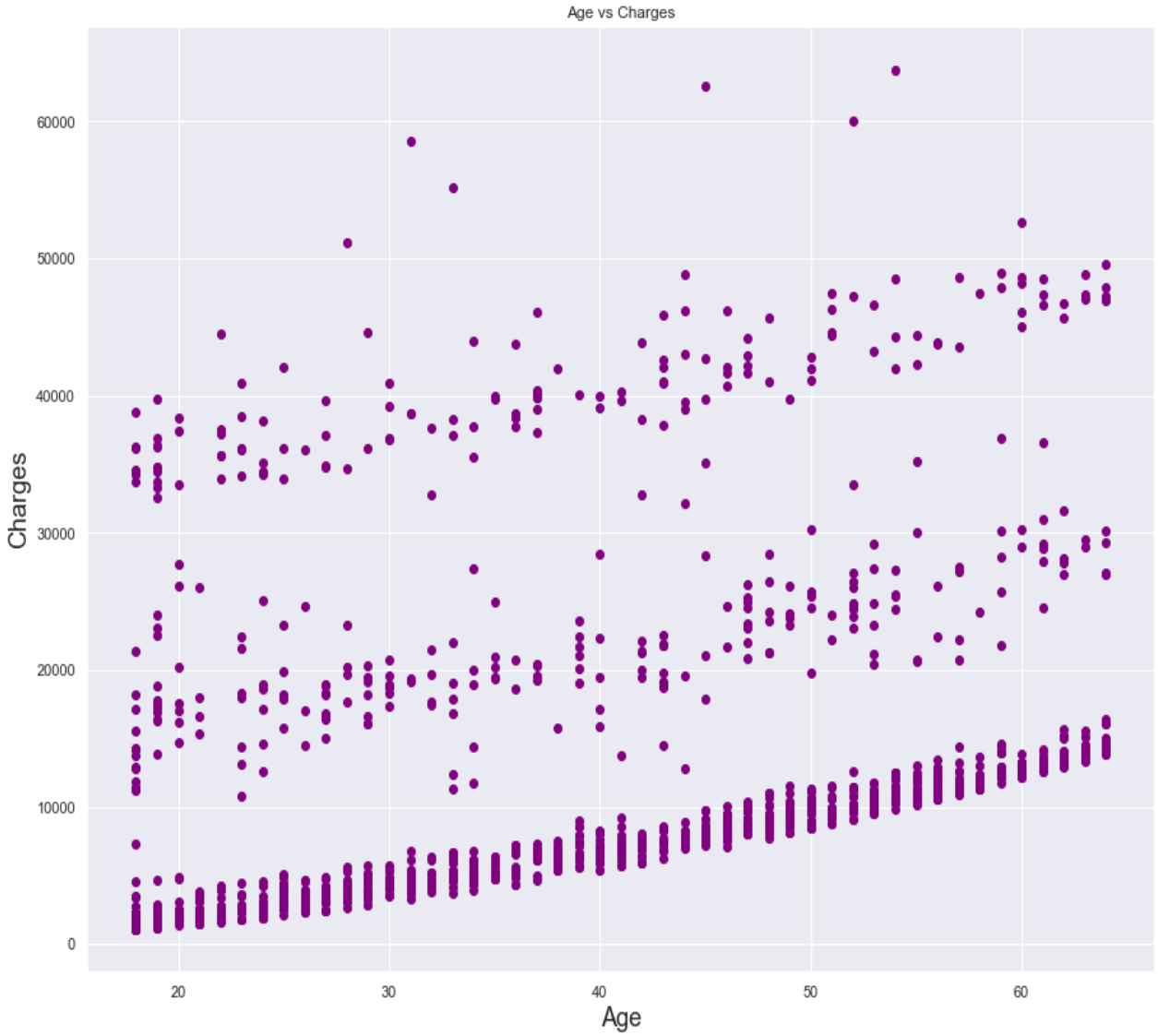
BIVARIATE ANALYSIS

This type of data involves two different variables. The analysis of this type of data deals with causes and relationships and the analysis is done to find out the relationship among the two variables.

Scatterplot using plotly

In [38]:

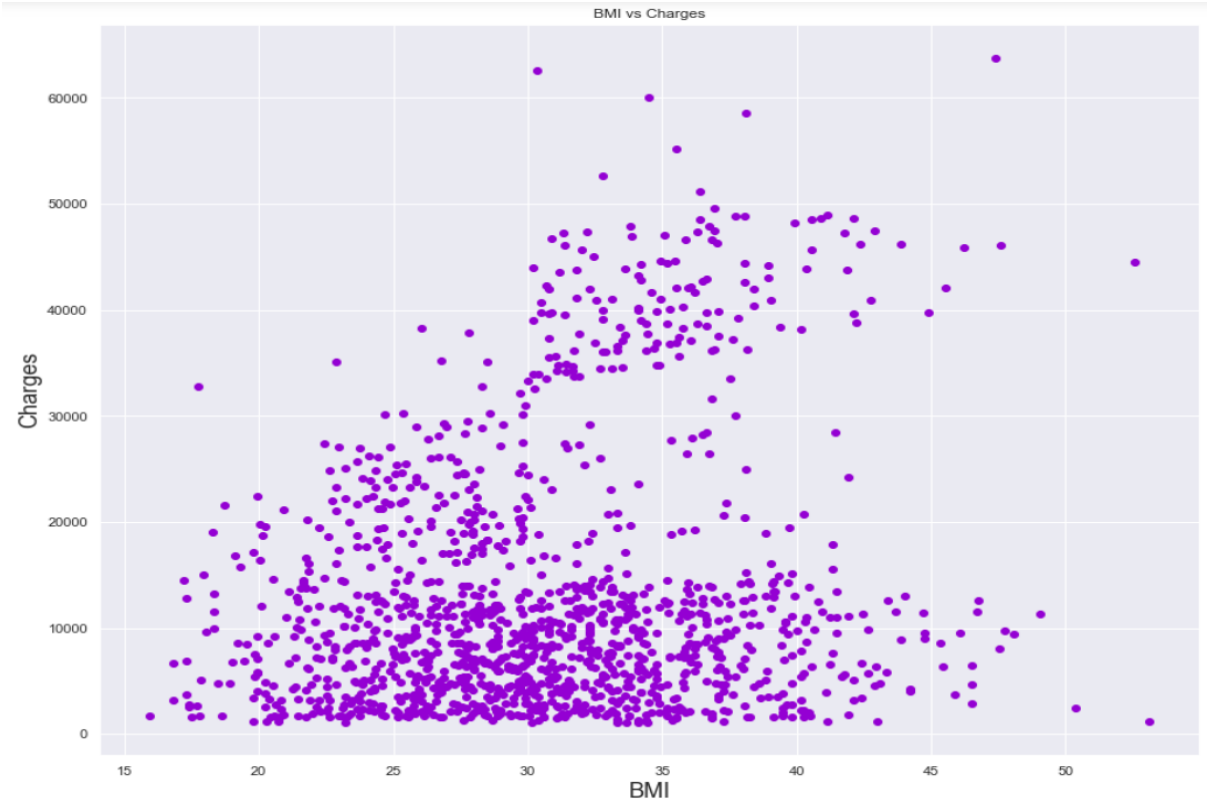
```
plt.figure(figsize=(15,12))
plt.scatter(insurance_data['age'],insurance_data['charges'],color="purple")
plt.xlabel('Age',size=18)
plt.ylabel('Charges',size=18)
plt.title('Age vs Charges')
plt.show()
```



Scatterplot for Age vs Charges

Scatterplots using plotly

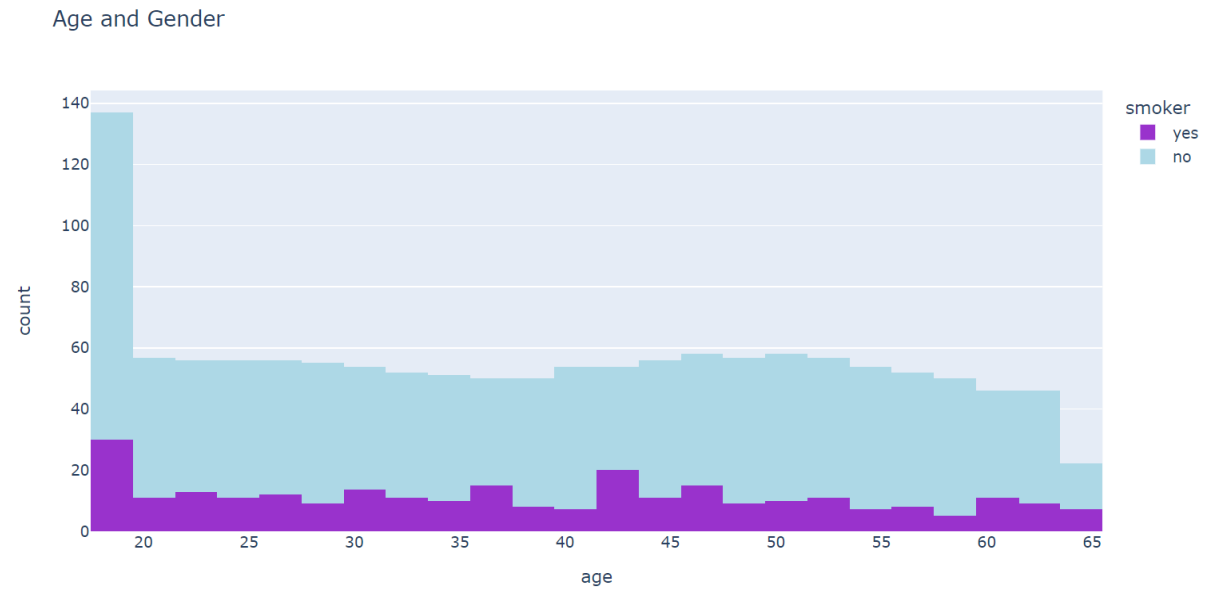
In [39]:
```
plt.figure(figsize=(15,12))
plt.scatter(insurance_data['bmi'],insurance_data['charges'],color="darkviolet")
plt.xlabel('BMI',size=18)
plt.ylabel('Charges',size=18)
plt.title('BMI vs Charges')
plt.show()
```



Scatterplot for BMI vs Charges

Histogram using plotly

```
In [40]:  1 fig = px.histogram(data_frame=insurance_data, x = 'age',color = 'smoker', title = "Age and Gender
          2 fig.show()
```

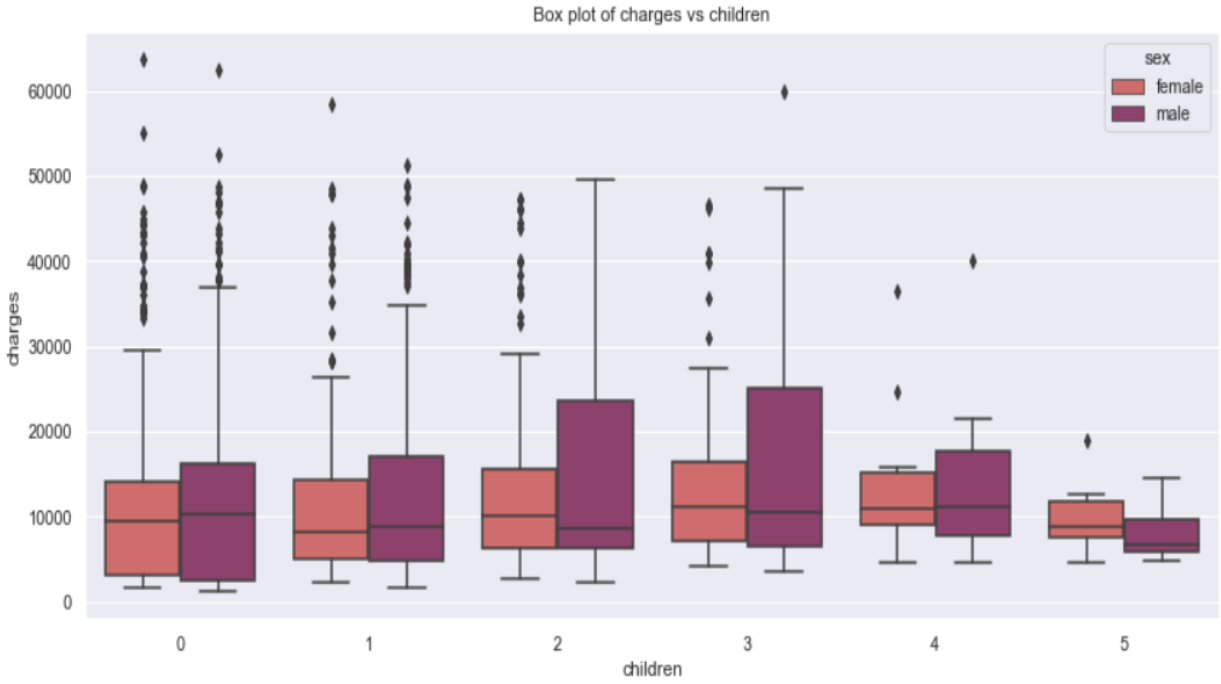


Histogram for age vs gender

Boxplot using seaborn and matplotlib

In [41]:
```
plt.figure(figsize=(14,6))
sns.boxplot(x='children',y='charges',hue='sex',data=insurance_data,palette='flare')
plt.title('Box plot of charges vs children')
```

Out[41]: Text(0.5, 1.0, 'Box plot of charges vs children')



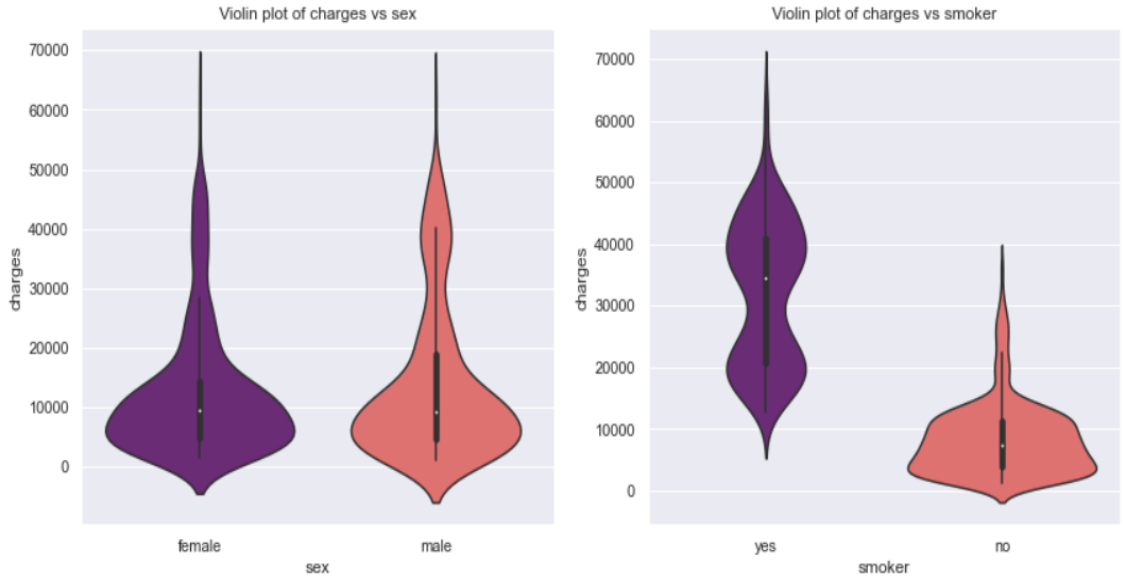
Boxplot of children vs charges

Violin plot using seaborn and matplotlib

In [51]:
```
fig=plt.figure(figsize=(14,6))
position=fig.add_subplot(121)
sns.violinplot(x='sex',y='charges',data=insurance_data,palette="magma",ax=position)
position.set_title('Violin plot of charges vs sex')

position=fig.add_subplot(122)
sns.violinplot(x='smoker',y='charges',data=insurance_data,palette="magma",ax=position)
position.set_title('Violin plot of charges vs smoker')
```

Out[51]: Text(0.5, 1.0, 'Violin plot of charges vs smoker')



(i) Violin plot of charges vs sex

(ii) Violin plot of charges vs smoker

DATA PREPROCESSING

Categorical Data Encoding

The categorical data in the dataset is encoded to fit and evaluate the model. The categorical features are: Sex, Smoker, Region

In [52]:
```
#Encoding the categorical features
```

In [56]:
```
#Encoding 'sex' column
insurance_data.replace({'sex':{'male':0,'female':1}},inplace=True)

#Encoding 'smoker' column
insurance_data.replace({'smoker':{'yes':0,'no':1}},inplace=True)

#Encoding 'region' column
insurance_data.replace({'region':{'southeast':0,'southwest':1,'northeast':2,'northwest':3}},
inplace=True)
```

In [57]:
```
1 insurance_data
```

Out[57]:

	age	sex	bmi	children	smoker	region	charges
0	19	1	27.900	0	0	1	16884.92400
1	18	0	33.770	1	1	0	1725.55230
2	28	0	33.000	3	1	0	4449.46200
3	33	0	22.705	0	1	3	21984.47061
4	32	0	28.880	0	1	3	3866.85520
...	...	...	...	...	...	...	...
1333	50	0	30.970	3	1	3	10600.54830
1334	18	1	31.920	0	1	2	2205.98080
1335	18	1	36.850	0	1	0	1629.83350
1336	21	1	25.800	0	1	1	2007.94500
1337	61	1	29.070	0	0	3	29141.36030

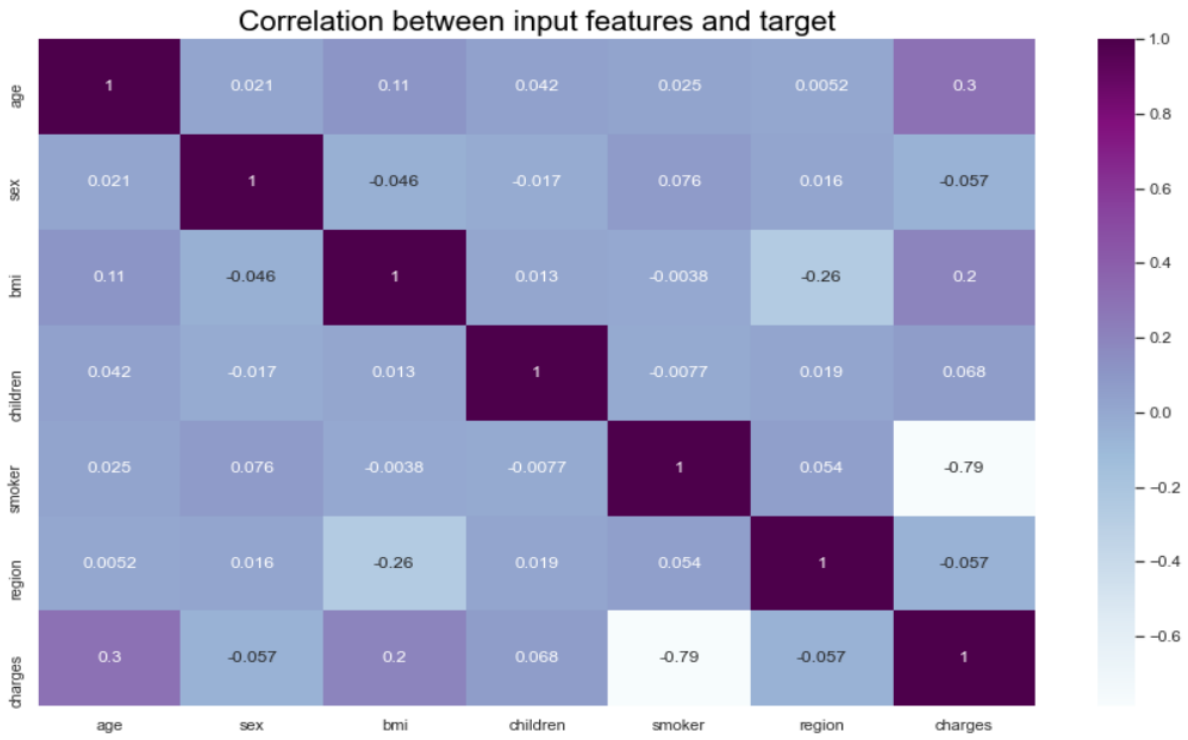
1338 rows × 7 columns

Correlation Analysis

Correlation ranges from -1 to +1. Values closer to zero means there is no linear trend between the two variables. The close to 1 the correlation is the more positively correlated they are; that is as one increases so does the other and the closer to 1 the stronger this relationship is.

Correlation Heatmap using Seaborn

In [55]:
```
1 fig, ax = plt.subplots(1, 1, figsize=(15, 9))
2 sns.heatmap(insurance_data.corr(), annot=True,cmap = 'BuPu')
3 plt.title('Correlation between input features and target', fontsize = 20, c='black')
4 plt.show()
```



Heatmap depicting correlation between data fields

Splitting the input features and target

In [58]:
```
1 inputs = insurance_data.drop(columns='charges',axis=1)
2 output= insurance_data['charges']
```

In [59]:
```
1 #contains input features
2 print(inputs)
```

```
      age  sex     bmi  children  smoker  region
0      19    1  27.900         0       0       1
1      18    0  33.770         1       1       0
2      28    0  33.000         3       1       0
3      33    0  22.705         0       1       3
4      32    0  28.880         0       1       3
...   ...  ...     ...       ...     ...     ...
1333   50    0  30.970         3       1       3
1334   18    1  31.920         0       1       2
1335   18    1  36.850         0       1       0
1336   21    1  25.800         0       1       1
1337   61    1  29.070         0       0       3

[1338 rows x 6 columns]
```

In [60]:
```
#contains target column
print(output)
```

```
0       16884.92400
1        1725.55230
2        4449.46200
3       21984.47061
4        3866.85520
            ...
1333    10600.54830
1334     2205.98080
1335     1629.83350
1336     2007.94500
1337    29141.36030
Name: charges, Length: 1338, dtype: float64
```

Splitting the data into train data and test data

In [65]:
```
x_train,x_test,y_train,y_test=train_test_split(inputs,output,test_size = 0.2,random_state =2)
```

In [66]:
```
print(x_train.shape,x_test.shape)
```

```
(1070, 6) (268, 6)
```

MODEL TRAINING

Linear Regression

In [67]:
```
#Loading the linear regression model
regressor = LinearRegression()
```

In [68]:
```
regressor.fit(x_train.values,y_train.values)
```

Out[68]:
```
LinearRegression()
```

Building a predictive system

In [73]:
```
input_data = (31,1,25.74,0,1,0)

#changing  input_data to a numpy array
input_data_as_numpy_array = np.asarray(input_data)

#reshape the array
input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)

prediction = regressor.predict(input_data_reshaped)
print(prediction)
```

```
[3760.0805765]
```

In [74]:
```
print('The insurance cost is USD',prediction[0])
```

```
The insurance cost is USD 3760.0805764960587
```

PERFORMANCE ANALYSIS

R-Squared score

The R2 score is a very important metric that is used to evaluate the performance of a regression-based machine learning model. It works by measuring the amount of variance in the predictions explained by the dataset. It is the difference between the samples in the dataset and the predictions made by the model.

Model Evaluation

In [69]:
```
#prediction on training data
train_data_pred=regressor.predict(x_train.values)
```

In [70]:
```
# R squared value
rsquare_train = metrics.r2_score(y_train,train_data_pred)
print('R squared value : ',rsquare_train)
```

```
R squared value :  0.751505643411174
```

In [71]:
```
#prediction on test data
test_data_pred=regressor.predict(x_test.values)
```

In [72]:
```
# R squared value
rsquare_test = metrics.r2_score(y_test,test_data_pred)
print('R squared value : ',rsquare_test)
```

```
R squared value :  0.7447273869684077
```

A good value of R square is obtained over the training and test data. So there is no overfitting.

RESULTS

- From the regression analysis, it is found that region and gender do not bring significant difference on charges.
- Age, BMI, no. of children and smoking are the ones that drive the charges.
- Smoking seems to have the most influence on the medical charges.

REFERENCES

https://ieeexplore.ieee.org/document/9374348

https://www.researchgate.net/publication/348559741_Predict_Health_Insurance_Cost_by_using_Machine_Learning_and_DNN_Regression_Models

https://www.kaggle.com/mirichoi0218/insurance/code

https://github.com/SahilChachra/Medical-Cost-Prediction