Statistical Learning Project on Insurance Dataset

Domain:- Healthcare

Context:- Leveraging customer information is paramount for most businesses. In the case of an insurance company, attributes of customers like the ones mentioned below can be crucial in making business decisions. Hence, knowing to explore and generate value out of such data can be an invaluable skill to have.

Data Description:- Insurance.csv => The data at hand contains medical costs of people characterized by certain attributes.

Objective:- We want to see if we can dive deep into this data to find some valuable insights.

Task 1: **Import the necessary libraries (2 marks)**

In [1]:
```
#Importing Necessary Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats as stats
from sklearn.preprocessing import LabelEncoder
```

Task 2: **Read the data as a data frame (2 marks)**

In [2]:
```
#Read the data from file with name "insurance.csv" using the pandas library to print data in Dataframe.
ins = pd.read_csv("insurance.csv")
```

In [3]:
```
#Print First 5 Rows of DataFrame
ins.head()
```

Out[3]:

	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

Task 3: Perform Basic EDA

Task 3(a): **EDA - Print Shape of the data (2 marks)**

In [4]:
```
#Check the Number of Rows and Columns present in DataFrame.
ins.shape
```

Out[4]:
```
(1338, 7)
```

Task 3(b): **EDA - Print Data type of each attribute (2 marks)**

In [5]:
```
#Print a concise summary of a DataFrame
ins.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
```

Task 3(c): **EDA - Checking the presence of missing values (3 marks)**

In [45]:
```
#Check for Presence of any Null Values in DataFrame
ins.isnull().sum()
```

Out[45]:
```
age         0
sex         0
bmi         0
children    0
smoker      0
region      0
charges     0
dtype: int64
```

Task 3(d): **EDA - Print 5 point summary of numerical attributes (3 marks)**

In [7]:
```
ins.describe() #Prints 5 point summary
```

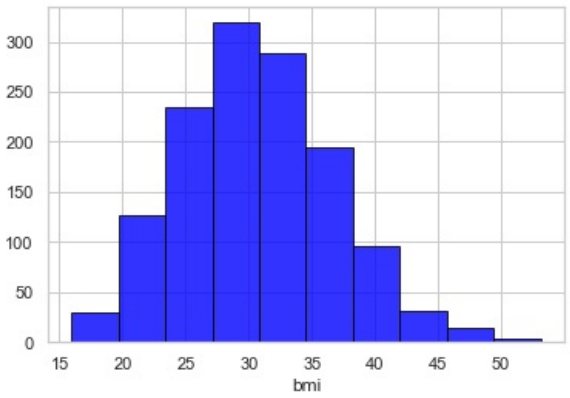
Out[7]:

	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

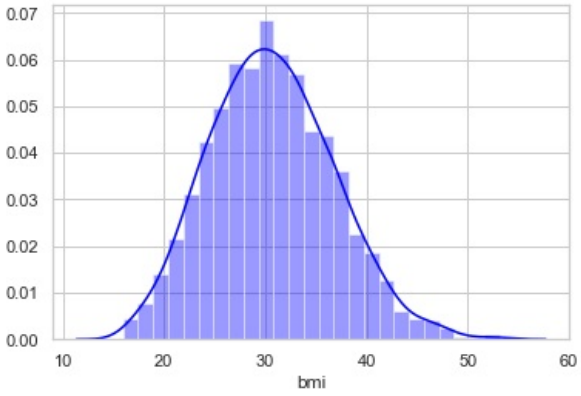
Task 3(e): **EDA - Show Distribution of 'bmi', 'age' and 'charges' columns. (4 marks)**

In [8]:
```
#Displaying Grid
sns.set(style="whitegrid")
```

In [9]:
```
#Plots to see the distribution of the continuous features of Column "bmi".
plt.hist(ins.bmi, color='blue', edgecolor = 'black', alpha=0.8)
plt.xlabel('bmi')
plt.show()
sns.distplot(ins['bmi'], color='blue')
```



Out[9]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1940ea03488>
```

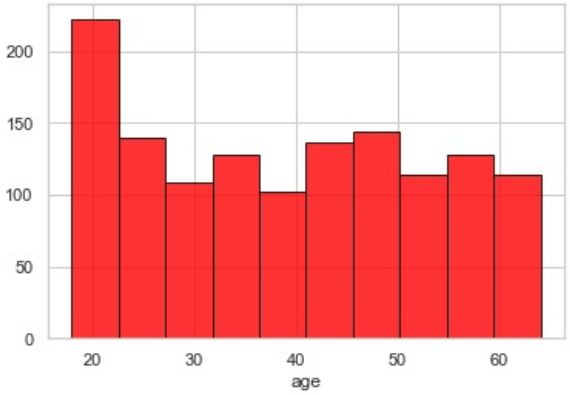


The Histogram Plot for BMI shows that it is in a considerable good shape and not much left skewness is present. Very less people with lower bmi exists in the dataset.

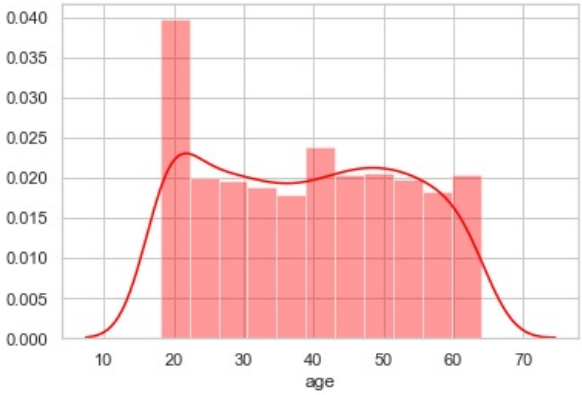
The Distribution Plot for BMI shows that it is approx normally distributed.

In [10]:
```
#Plots to see the distribution of the continuous features of Column "age".
plt.hist(ins.age, color='red', edgecolor = 'black', alpha=0.8)
```

```
plt.xlabel('age')
plt.show()
sns.distplot(ins['age'], color='red')
```



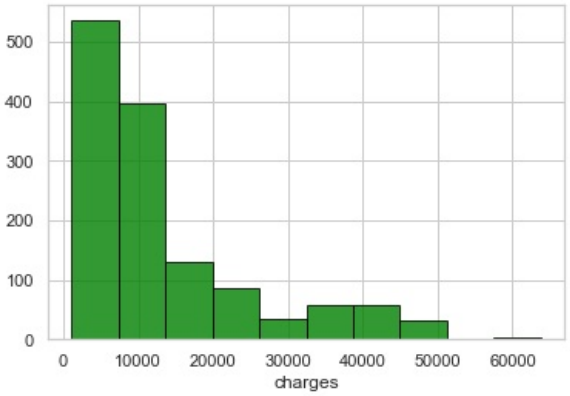
Out[10]: <matplotlib.axes._subplots.AxesSubplot at 0x1940e27a048>



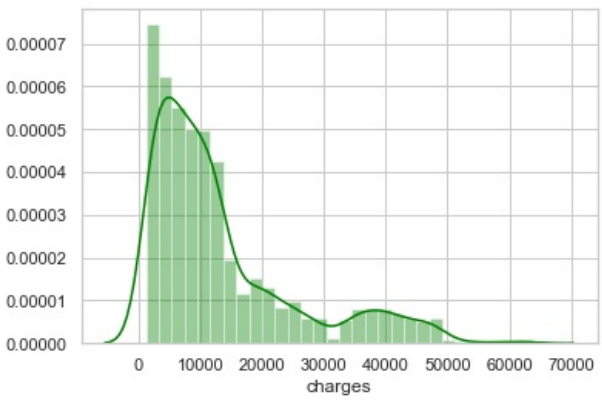
The Histogram Plot for Age describes that highest participation is done by the age around 20yrs old customers. Though the data is very very slightly more for higher age people is present.

The Distribution Plot for Age shows that it is distributed quiet uniformly.

In [11]:
```
#Plots to see the distribution of the continuous features of Column "charges".
plt.hist(ins.charges, color='green', edgecolor = 'black', alpha=0.8)
plt.xlabel('charges')
plt.show()
sns.distplot(ins['charges'], color='green')
```



Out[11]: <matplotlib.axes._subplots.AxesSubplot at 0x1940ebaf688>



The Histogram Plot for Charges shows that there is a High left skewness in the dataset and it tells that mostly low individual medical costs is only billed by health insurance.

The Distribution Plot for Charges describes that Charges are highly skewed.

Task 3(f): **EDA - Measure skewness of 'bmi', 'age' and 'charges' columns (2 marks)**

In [12]:
```
# Measure the skeweness of the required columns.
Skew = pd.DataFrame({'Skewness' : [stats.skew(ins.bmi),
                                   stats.skew(ins.age),stats.skew(ins.charges)]},
                    index=['bmi','age','charges'])
Skew
```

Out[12]:

	Skewness
bmi	0.283729
age	0.055610
charges	1.514180

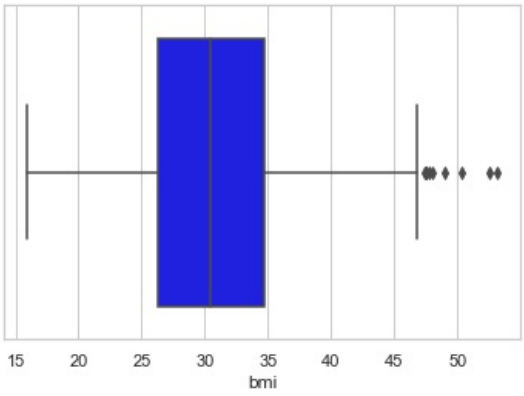
Task 3(g): **EDA - Checking the presence of outliers in 'bmi', 'age' and 'charges' columns (4 marks)**

In [13]:
```
#Plotting Box Plot for Column "bmi".
sns.boxplot(ins['bmi'], color='blue')

#Calculating Quantile1, Quantile3, Interquantile Range (IQR) for column "bmi".
Q1 = np.percentile(ins['bmi'], 25)
Q3 = np.percentile(ins['bmi'], 75)
IQR = Q3 - Q1

#Identifying the presence of outliers in "bmi" column.
outliers_bmi = [x for x in ins['bmi'] if x < (Q1-1.5*IQR) or x > (Q3+1.5*IQR)]
print('Identified outliers for bmi out of total no. of records:', len(outliers_bmi))
```

```
Identified outliers for bmi out of total no. of records: 9
```

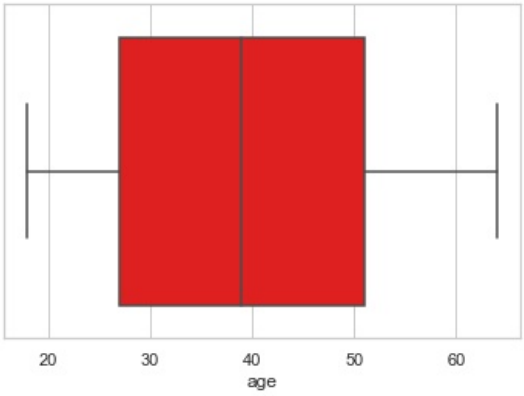


In [14]:
```
#Plotting Box Plot for Column "age".
sns.boxplot(ins['age'], color='red')

#Calculating Quantile1, Quantile3, Interquantile Range (IQR) for column "age".
Q1 = np.percentile(ins['age'], 25)
Q3 = np.percentile(ins['age'], 75)
IQR = Q3 - Q1

#Identifying the presence of outliers in "age" column.
outliers_age = [x for x in ins['age'] if x < (Q1-1.5*IQR) or x > (Q3+1.5*IQR)]
print('Identified outliers for age out of total no. of records:', len(outliers_age))
```

```
Identified outliers for age out of total no. of records: 0
```

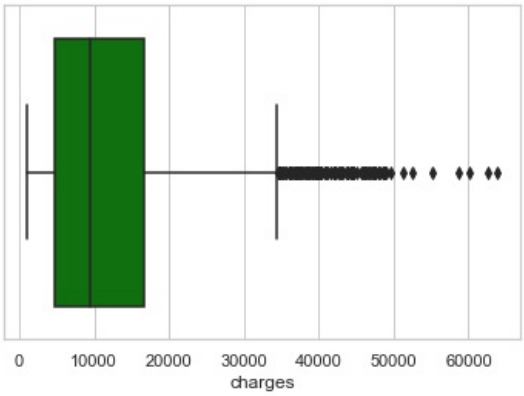


In [15]:
```
#Plotting Box Plot for Column "charges".
sns.boxplot(ins['charges'], color='green')

#Calculating Quantile1, Quantile3, Interquantile Range (IQR) for column "charges".
Q1 = np.percentile(ins['charges'], 25)
Q3 = np.percentile(ins['charges'], 75)
IQR = Q3 - Q1

#Identifying the presence of outliers in "charges" column.
outliers_charges = [x for x in ins['charges'] if x < (Q1-1.5*IQR) or x > (Q3+1.5*IQR)]
print('Identified outliers for charges out of total no. of records:', len(outliers_charges))
```

```
Identified outliers for charges out of total no. of records: 139
```

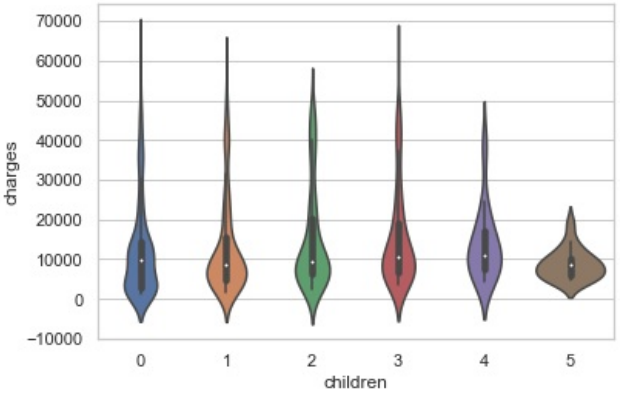


Task 3(h): **EDA - Show Distribution of categorical columns (include children) (4 marks)**

=> Plotting a Bivariate Distribution Plot for categorical column "children" against column "charges".

In [16]:
```
#Plotting a Violin Plot for categorical column "children" against column "charges"
sns.violinplot(x='children', y='charges', data=ins, split=True)
```

Out[16]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1940edd2fc8>
```

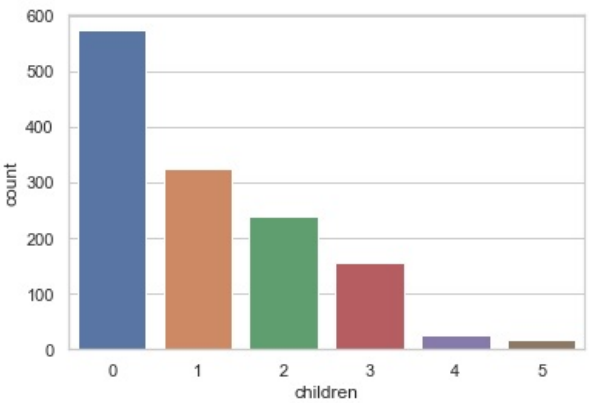


From this Bivariate Plot we can observe that in some cases we see the extremly higher charges are paid by people having no child while least charges are paid when having 5 children.

=> Plotting a Univariate Estimation Plot for categorical column "children".

In [17]:
```
#Plotting a Count Plot for categorical column "children".
sns.countplot(ins['children'])
```

Out[17]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1940ee7a648>
```

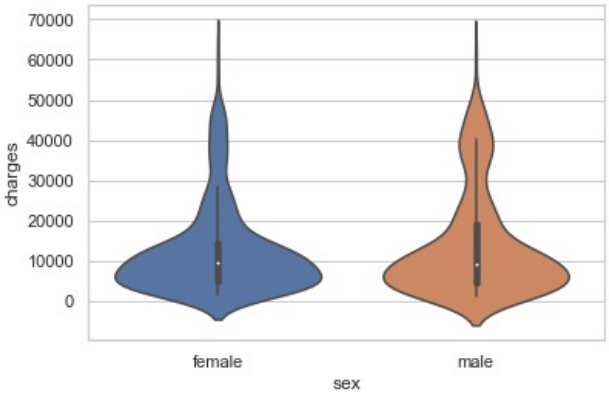


From this Univariate Plot we can observe that maximum people doesn't have any child and very less people have 5 children.

=> Plotting a Bivariate Distribution Plot for categorical column "sex" against column "charges".

In [18]:
```
#Plotting a Violin Plot for categorical column "sex" against column "charges".
sns.violinplot(x='sex', y='charges', data=ins, split=True)
```

Out[18]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1940eee3e08>
```

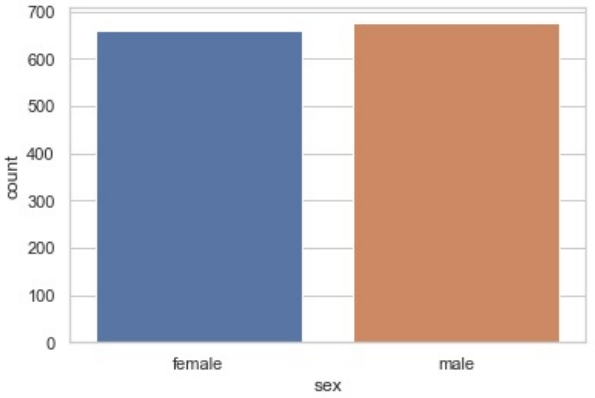


From this Bivariate Plot we can observe that in both the male and female, we see many among them had paid the extreme charges.

=> Plotting a Univariate Estimation Plot for categorical column "sex".

In [19]:
```
#Plotting a Count Plot for categorical column "sex".
sns.countplot(ins['sex'])
```

Out[19]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1940ef52588>
```

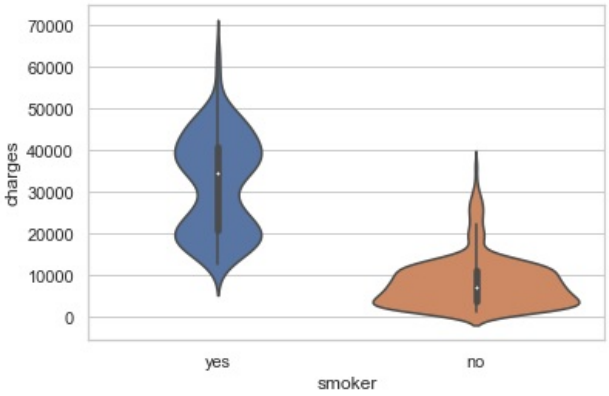


From this Univariate Plot we can observe that the gender ratio of insurance contractor is not significantly different.

=> Plotting a Bivariate Distribution Plot for categorical column "smoker" against column "charges".

In [20]:
```
#Plotting a Violin Plot for categorical column "smoker" against column "charges".
sns.violinplot(x='smoker', y='charges', data=ins, split=True)
```

Out[20]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1940efba688>
```

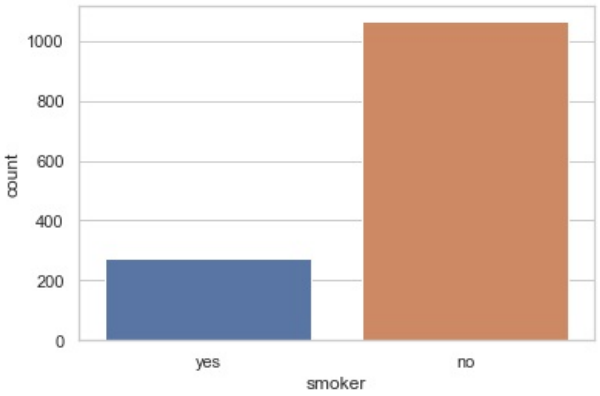


From this Bivariate Plot we can observe that, Smokers pay higher medical costs billed by health insurance than the non-smokers. However, there are some outliers exists in the non-smokers who pay higher charges.

=> Plotting a Univariate Estimation Plot for categorical column "smoker".

In [21]:
```
#Plotting a Count Plot for categorical column "smoker".
sns.countplot(ins['smoker'])
```

Out[21]: <matplotlib.axes._subplots.AxesSubplot at 0x1940ef59bc8>

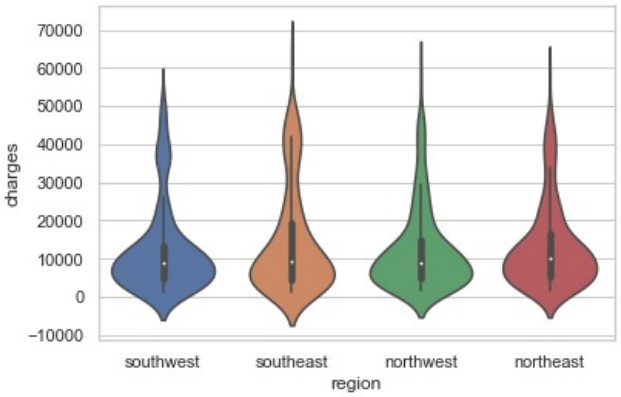


From this Univariate Plot we can observe that the count of non-smokers is quite high than the smokers in the dataset.

Plotting a Bivariate Distribution Plot for categorical column "region" against column "charges".

In [22]:
```
#Plotting a Violin Plot for categorical column "region" against column "charges".
sns.violinplot(x='region', y='charges', data=ins, split=True)
```

Out[22]: <matplotlib.axes._subplots.AxesSubplot at 0x1940ed478c8>

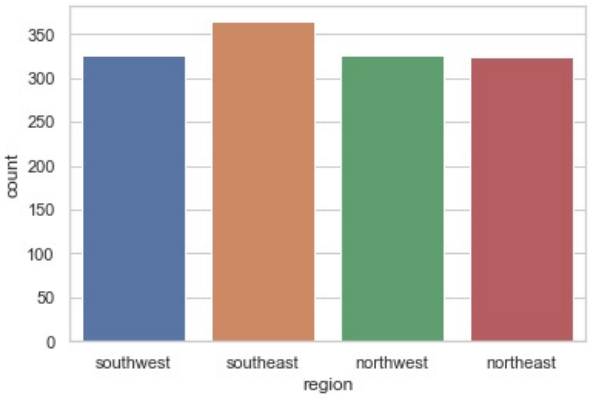


From this Bivariate Plot we can say that Each location is having some extreme cases of higher charges.Though Southeast customers pay higher charges more.

=> Plotting a Univariate Estimation Plot for categorical column "region".

In [23]:
```
#Plotting a Count Plot for categorical column "region".
sns.countplot(ins['region'])
```

Out[23]: <matplotlib.axes._subplots.AxesSubplot at 0x1940ecff788>



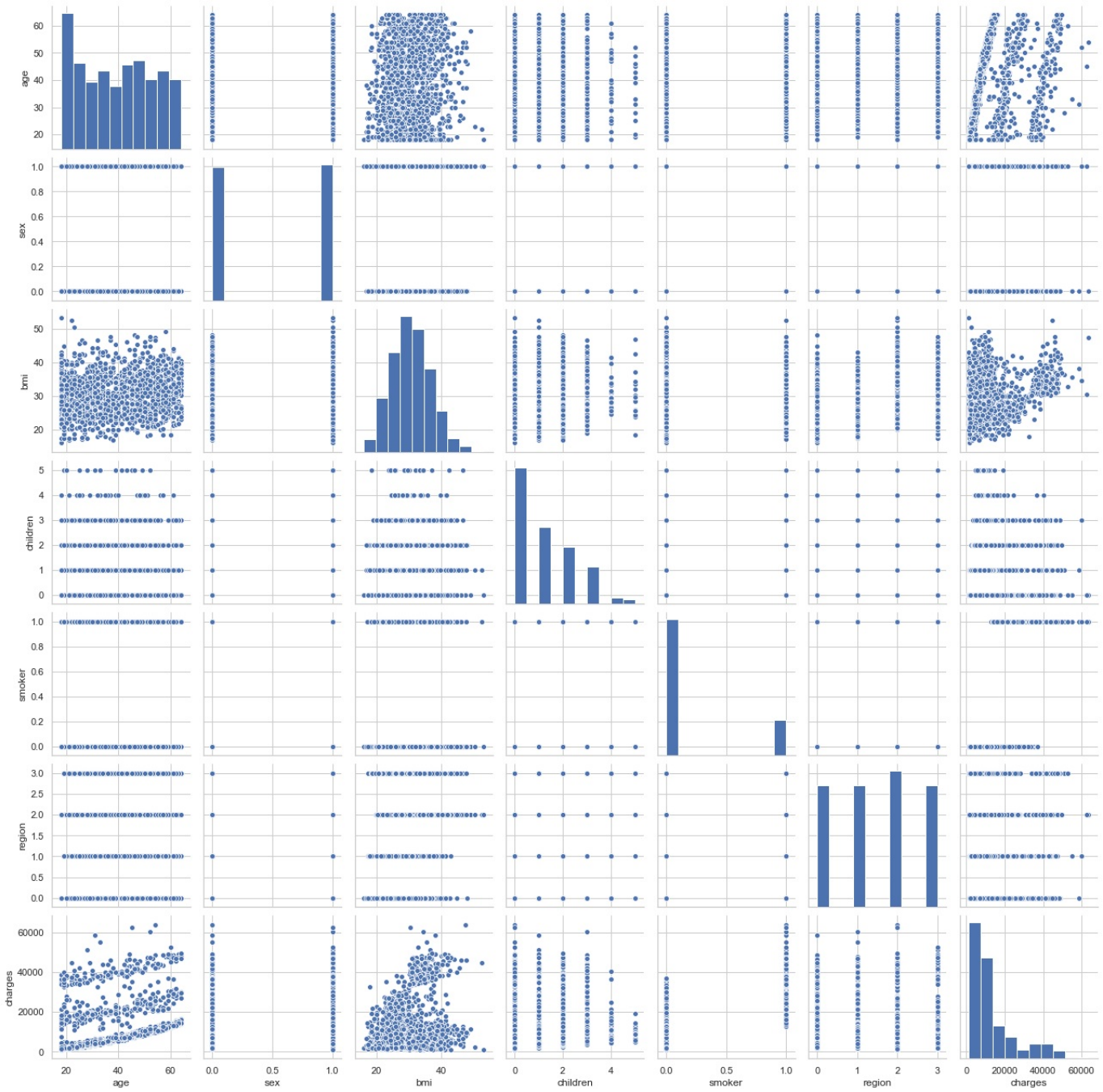
From this Univariate Plot we can observe that people are distributed evenly accross all regions. However, people live more in Southeast region in compare to other 3 regions.

Task 3(i): **EDA - Print a Pair plot that includes all the columns of the data frame (4 marks)**

In [24]:
```
#Creating a copy of the original dataset.
ins_encoded = ins.copy()

#Label encoding the variables before doing a pairplot because pairplot ignores strings
ins_encoded.loc[:,['sex', 'smoker', 'region']] = ins.loc[:,['sex', 'smoker', 'region']].apply(LabelEncoder().fi
```

```
#Plotting a Pair Plot for the Label Encoded Data
sns.pairplot(ins_encoded)
plt.show()
```



The only obvious correlation of 'charges' is with 'smoker'.It seems like smokers spent more money than non-smokers.

There's an pattern which should be noted i.e. between 'age' and 'charges. It concludes that As Age increases, Medical bills also increases. This could be because for the same ailment, older people are charged more than the younger ones. Since, when age increases complexity to cure the ailment becomes difficilt and expensive also.

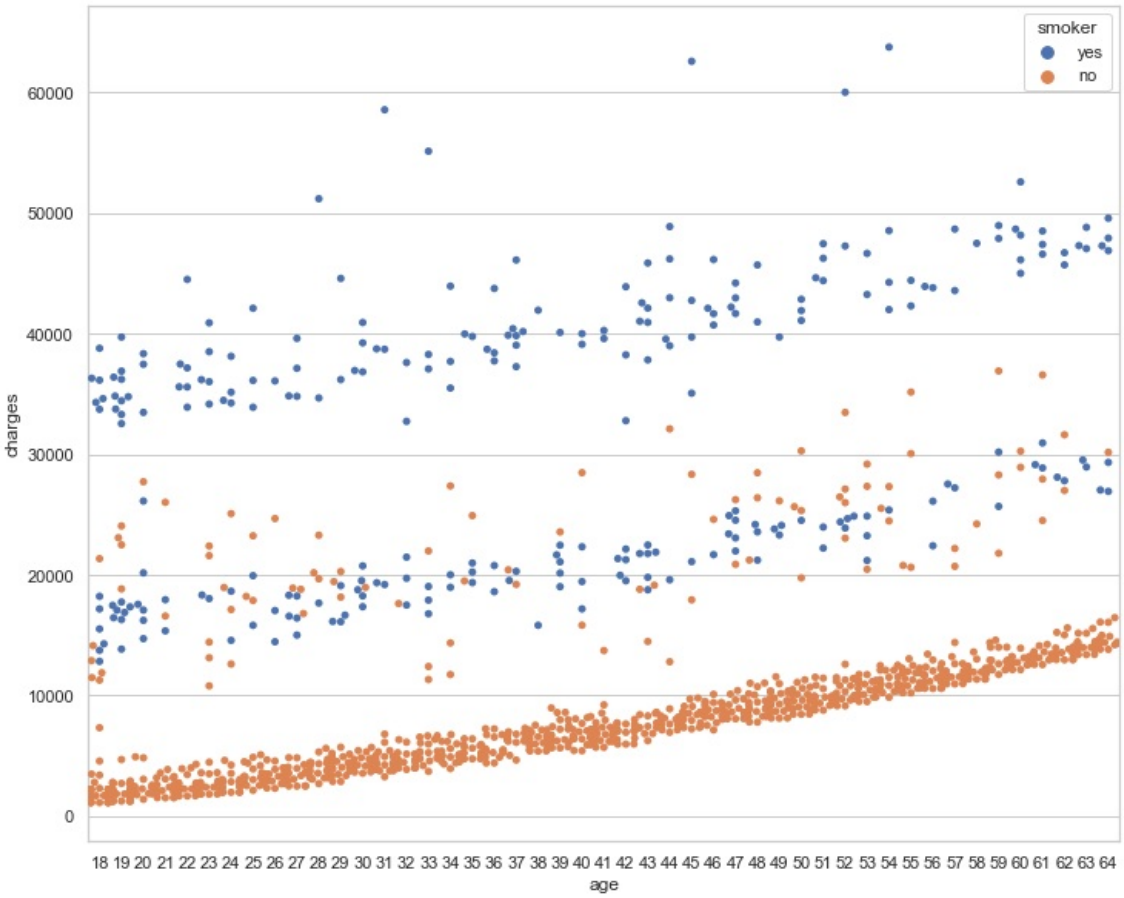
Task4: Answer the following questions with statistical evidence

Task 4(a): **Do charges of people who smoke differ significantly from the people who don't? (7 marks)**

In [25]:
```
#Display the count of smokers and non-smokers.
ins.smoker.value_counts()
```

Out[25]:
```
no     1064
yes     274
Name: smoker, dtype: int64
```

In [28]:
```
#Plotting a Swarm Plot through which comparison can be made that whether
#charges of smokers differ significantly from the non-smokers or not.
plt.figure(figsize=(12,10))
sns.swarmplot(ins.age, ins.charges,hue=ins.smoker)
plt.show()
```



Through visualization we can clearly see that smokers differ significantly from the no-smokers.

In [40]:
```
#Applying T-test to determine the impact of smoking on the charges.
Ho = "Charges of smoker and non-smoker are same"
Ha = "Charges of smoker and non-smoker are not the same"

# Selecting charges corresponding to smokers as an array
x = np.array(ins[ins.smoker == 'yes'].charges)
# Selecting charges corresponding to non-smokers as an array
y = np.array(ins[ins.smoker == 'no'].charges)

#Performing an Independent t-test
t, p_value  = stats.ttest_ind(x,y, axis = 0)  #Performing an Independent t-test

print(p_value)
```

```
8.271435842177219e-283
```

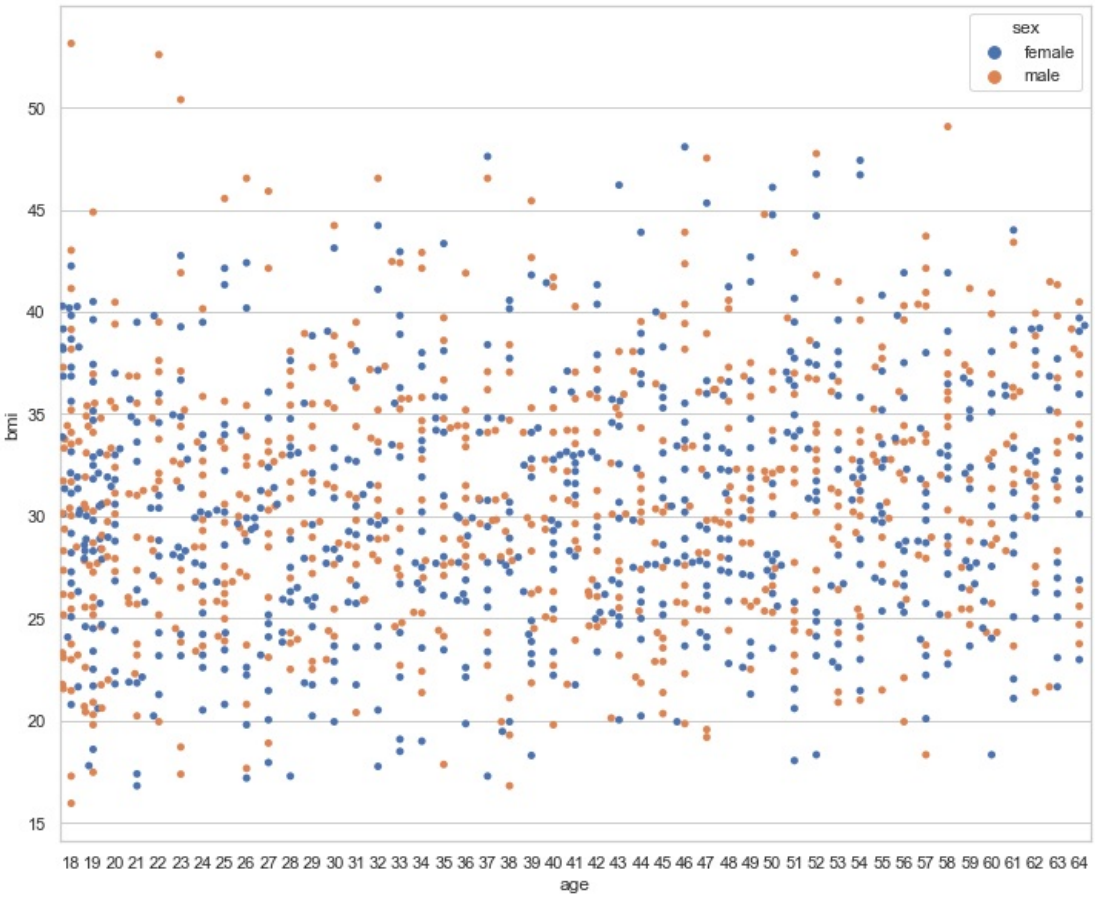
Conclusion: Rejecting the Null hypothesis as the p-value is lesser than 0.05. It tells us that the paid charges by the smokers and non-smokers is significantly different.Smokers pay higher charges in comparison to the non-smokers

Task 4(b): **Does bmi of males differ significantly from that of females? (7 marks)**

In [41]:
```
#Display the count of males and females.
ins.sex.value_counts()
```

Out[41]:
```
male      676
female    662
Name: sex, dtype: int64
```

In [42]:
```
#Plotting a Swarm Plot through which comparison can be made that whether
#bmi of males differ significantly from that of females or not.
plt.figure(figsize=(12,10))
sns.swarmplot(ins.age, ins.bmi,hue=ins.sex)
plt.show()
```



Through vizualisation here, It is very difficult to come to an conclusion regarding whether bmi differs significantly on basis of gender or not. Hence, we will check dependency of bmi on gender.

In [43]:
```
###### Check dependency of bmi on gender
Ho = "Bmi does not change significantly on basis of Gender"
Ha = "Bmi change significantly on basis of Gender"

# Selecting bmi corresponding to males as an array
x = np.array(ins[ins.sex == 'male'].bmi)
# Selecting bmi corresponding to females as an array
y = np.array(ins[ins.sex == 'female'].bmi)

#Performing an Independent t-test
t, p_value  = stats.ttest_ind(x,y, axis = 0)

print(p_value)
```

```
0.08997637178984932
```

Conclusion: Accepting Null hypothesis as the p-value is greater than 0.05. Hence, BMI does not change significantly on basis of Gender.

Task 4(c): **Is the proportion of smokers significantly different in different genders? (7 marks)**

In [44]:
```
# We will perform Chi_square test to check the proportion of smokers differs as per gender or not.
#Lets take Hypothesis Ho & Ha.
Ho = "Smoking Habits doesn't differs with the Gender."
Ha = "Smoking Habits differs with the Gender."

#Create cross-tabulation table for showing the frequency of columns "sex" and "smoker".
crosstab = pd.crosstab(ins['sex'],ins['smoker'])

#Applying Chi_square test and calculating p-value.
chi, p_value, dof, expected =  stats.chi2_contingency(crosstab)
print(p_value)
```

```
0.006548143503580696
```

Conclusion: Rejecting Null hypothesis as the p-value is less than 0.05. Hence, Smoking Habits differs with the Gender.

Task 4(d): **Is the distribution of bmi across women with no children, one child and two children, the same ? (7 marks)**

In [38]:
```
# Applying anova test to check the distribution of bmi across women with no children, one child and two childre
#Lets take Hypothesis Ho & Ha.
Ho = "No. of children has no effect on bmi"
Ha = "No. of children has an effect on bmi"

#Create a separate dataframe for the data of all the females present in the original dataframe.
```

```
female_df = (ins[ins['sex'] == 'female']).copy()

zero = female_df[female_df.children == 0]['bmi']
one = female_df[female_df.children == 1]['bmi']
two = female_df[female_df.children == 2]['bmi']

#Applying Anova Test and calculating p-value.
f_stat, p_value = stats.f_oneway(zero,one,two)
print(p_value)
```

```
0.7158579926754841
```

Conclusion: Accepting Null hypothesis as the p-value is greater than 0.05. Hence, it tells the number of children does not bring any difference in women's bmi.

Mini Project 2 Finished !!!

Submitted By: Raunak Choudhary

Contact: raunakchoudhary17@gmail.com