# ***Python Major Project***

**Submitted by:** Alavala Sai Goutham Reddy.

**Email Id:** gouthamsai.alavala@gmail.com

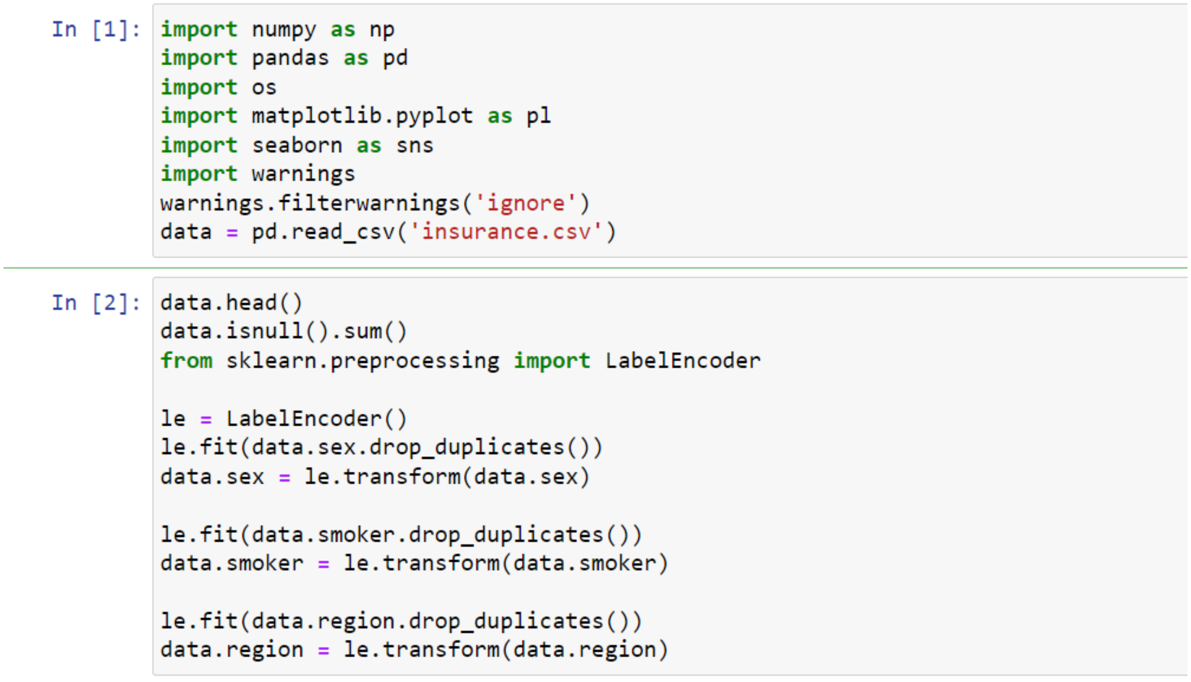
**Project Statement:**

Take any Dataset of your choice, perform EDA (Exploratory Data Analysis) and apply a suitable Classifier, Regressor or Clusterer and calculate the accuracy of the model.

**Project Solution:**

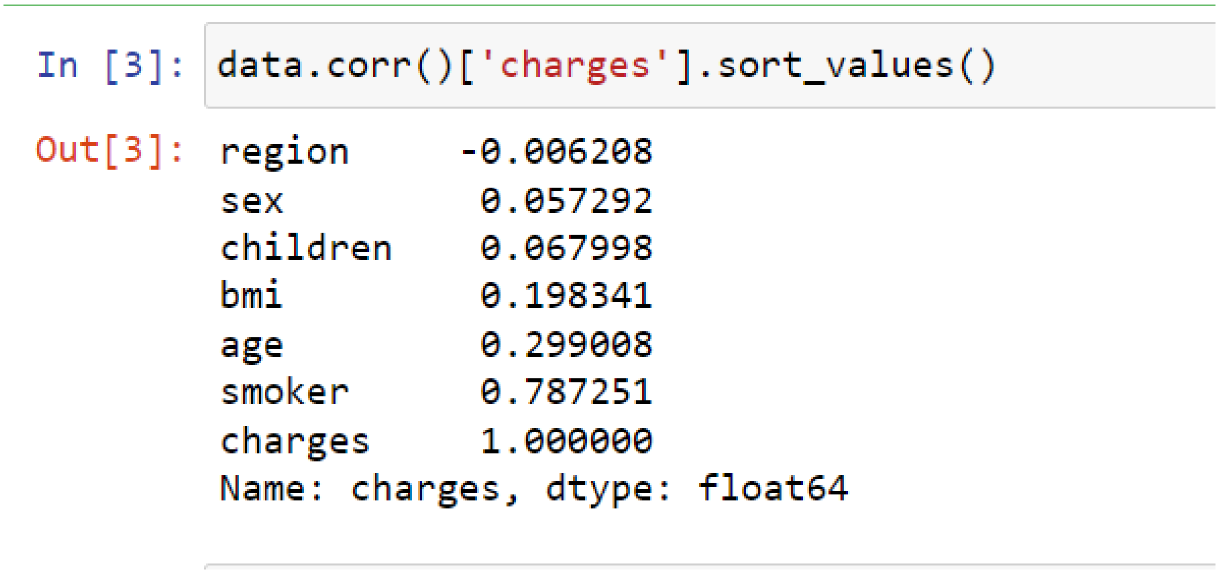
For this project we are using data set from kaggle known as insurance.csv which contains age, sex, Bmi, children, smoker, region, charges and we are using regression.

In the field of healthcare, it is important to understand the various factors that influence the cost of treatment for patients. As mentioned earlier, the cost of treatment can vary depending on a multitude of factors. Diagnosis, type of clinic, city of residence and age are just a few examples of factors that can impact treatment costs. While the absence of patient diagnosis data may seem like a hurdle, there are other crucial pieces of information that can help healthcare professionals make informed decisions about patient care. By leveraging regression analysis techniques and exploring the available data set, healthcare professionals can gain valuable insights that can ultimately lead to better patient outcomes.

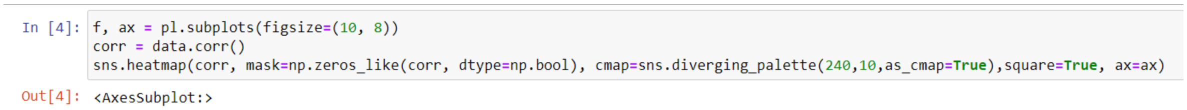


```
In [1]: import numpy as np
        import pandas as pd
        import os
        import matplotlib.pyplot as pl
        import seaborn as sns
        import warnings
        warnings.filterwarnings('ignore')
        data = pd.read_csv('insurance.csv')
```

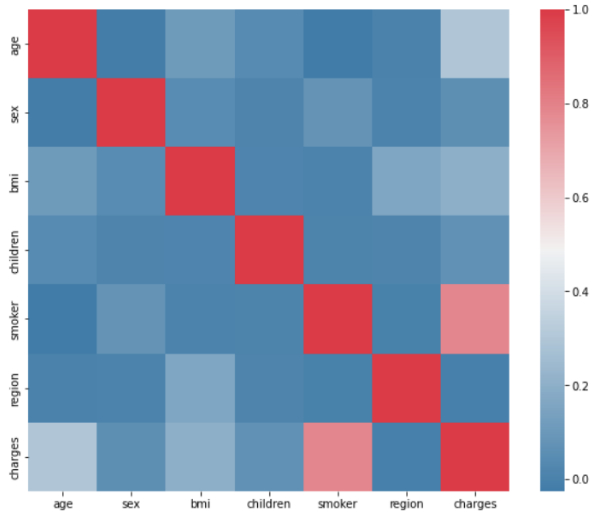
```
In [2]: data.head()
        data.isnull().sum()
        from sklearn.preprocessing import LabelEncoder

        le = LabelEncoder()
        le.fit(data.sex.drop_duplicates())
        data.sex = le.transform(data.sex)

        le.fit(data.smoker.drop_duplicates())
        data.smoker = le.transform(data.smoker)

        le.fit(data.region.drop_duplicates())
        data.region = le.transform(data.region)
```



```
In [3]: data.corr()['charges'].sort_values()

Out[3]: region     -0.006208
        sex         0.057292
        children    0.067998
        bmi         0.198341
        age         0.299008
        smoker      0.787251
        charges     1.000000
        Name: charges, dtype: float64
```

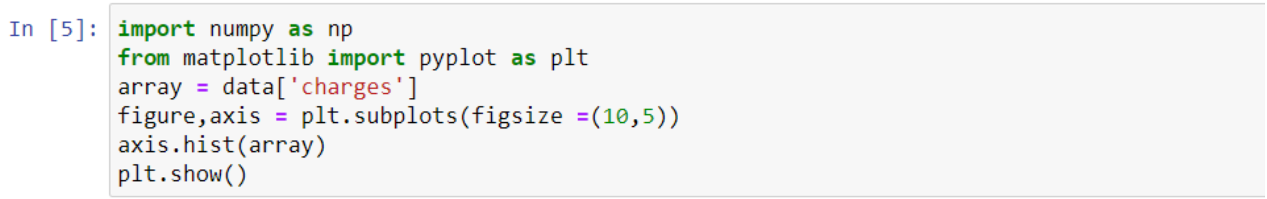


```
In [4]: f, ax = pl.subplots(figsize=(10, 8))
        corr = data.corr()
        sns.heatmap(corr, mask=np.zeros_like(corr, dtype=np.bool), cmap=sns.diverging_palette(240,10,as_cmap=True),square=True, ax=ax)

Out[4]: <AxesSubplot:>
```

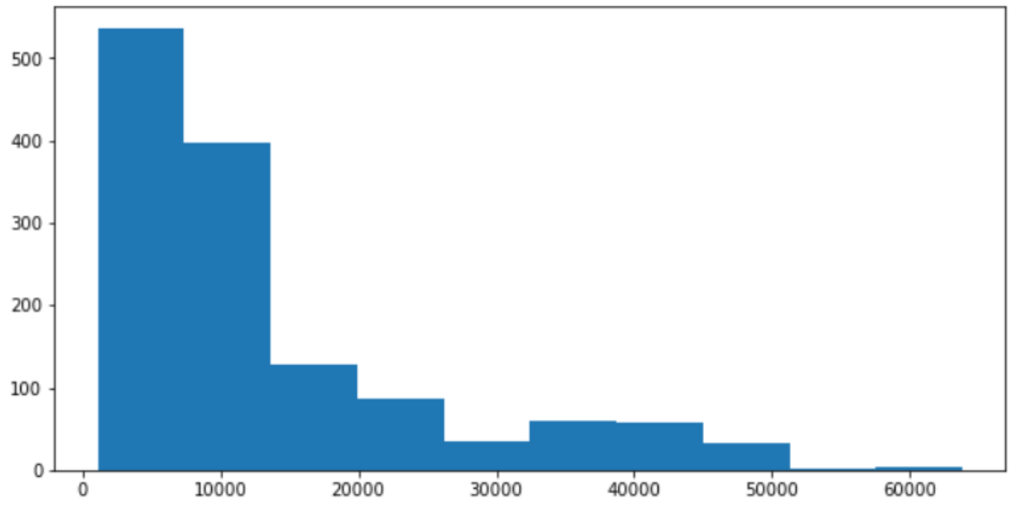


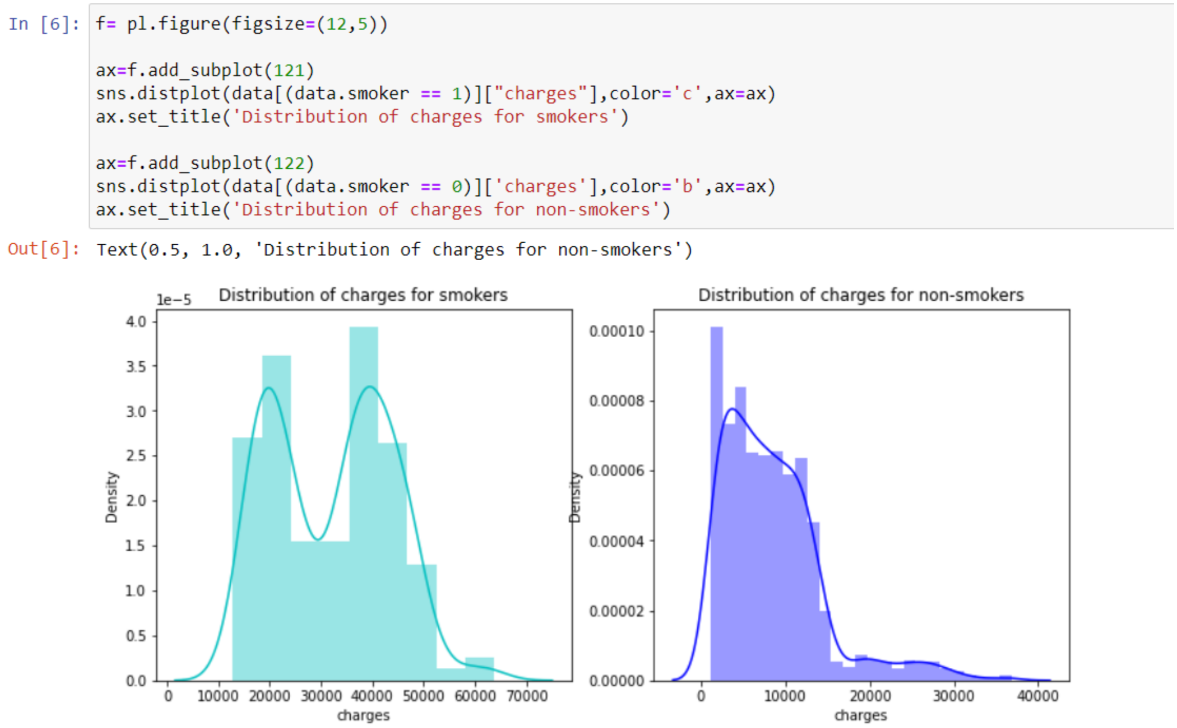
A strong correlation is observed only with the fact of smoking the patient.

Let's look at the distribution of charges. This will help us to know how much patients spend on treatment on average.



```
In [5]: import numpy as np
        from matplotlib import pyplot as plt
        array = data['charges']
        figure,axis = plt.subplots(figsize =(10,5))
        axis.hist(array)
        plt.show()
```

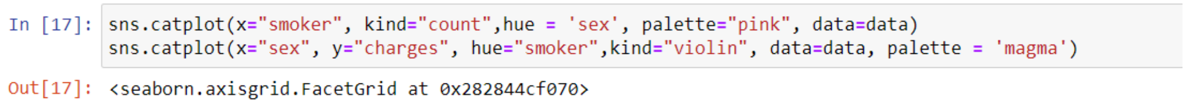




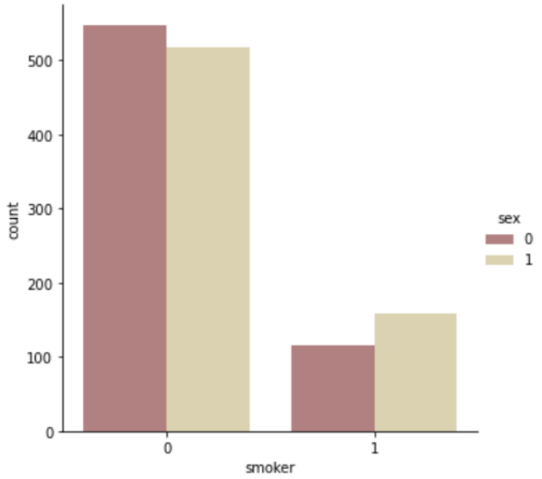
```
In [6]: f= pl.figure(figsize=(12,5))

        ax=f.add_subplot(121)
        sns.distplot(data[(data.smoker == 1)]["charges"],color='c',ax=ax)
        ax.set_title('Distribution of charges for smokers')

        ax=f.add_subplot(122)
        sns.distplot(data[(data.smoker == 0)]['charges'],color='b',ax=ax)
        ax.set_title('Distribution of charges for non-smokers')
```

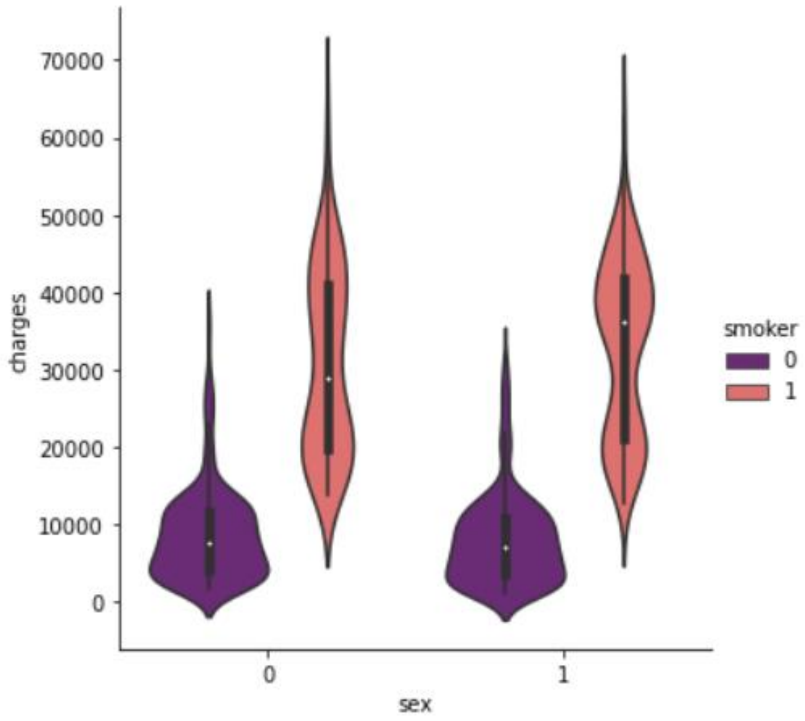
```
Out[6]: Text(0.5, 1.0, 'Distribution of charges for non-smokers')
```

Smoking patients spend more on treatment. But the number of non-smoking patients may be greater. So, we are going to check through it too.

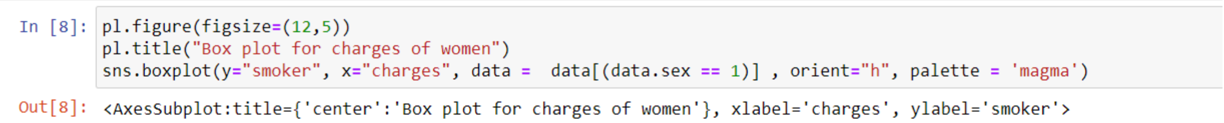


```
In [17]: sns.catplot(x="smoker", kind="count",hue = 'sex', palette="pink", data=data)
         sns.catplot(x="sex", y="charges", hue="smoker",kind="violin", data=data, palette = 'magma')

Out[17]: <seaborn.axisgrid.FacetGrid at 0x282844cf070>
```

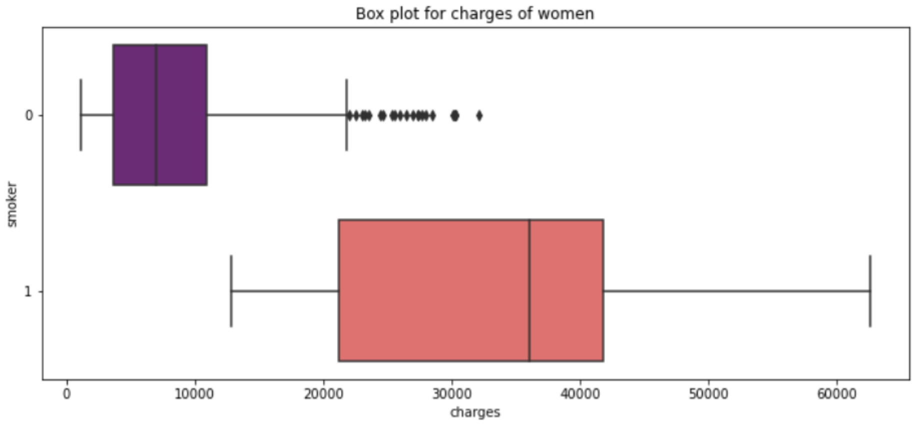


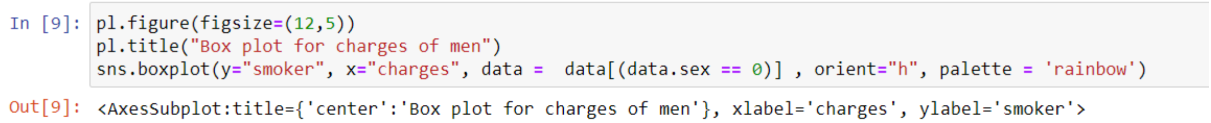


Here, women are coded with the symbol " 1 "and men with "0". Thus, non-smoking people and the truth more. Also, we can notice that more male smokers than women smokers. It can be assumed that the total cost of treatment in men will be more than in women, given the impact of smoking.

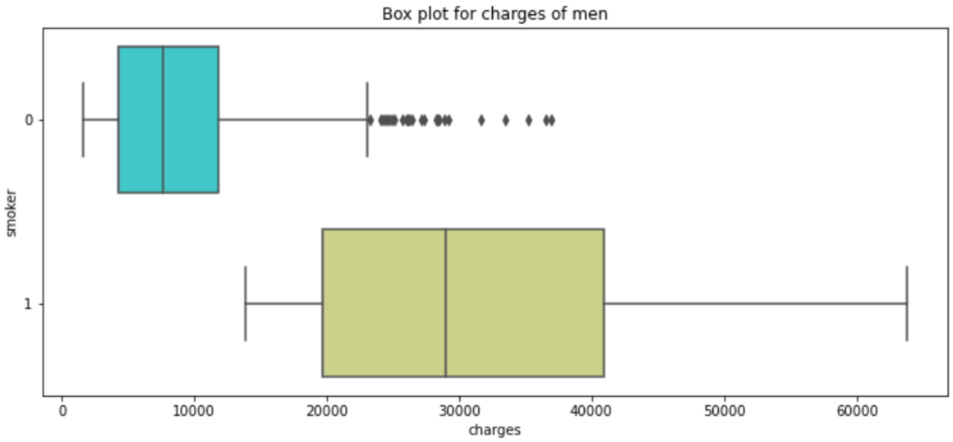


```
In [8]: pl.figure(figsize=(12,5))
        pl.title("Box plot for charges of women")
        sns.boxplot(y="smoker", x="charges", data =  data[(data.sex == 1)] , orient="h", palette = 'magma')

Out[8]: <AxesSubplot:title={'center':'Box plot for charges of women'}, xlabel='charges', ylabel='smoker'>
```

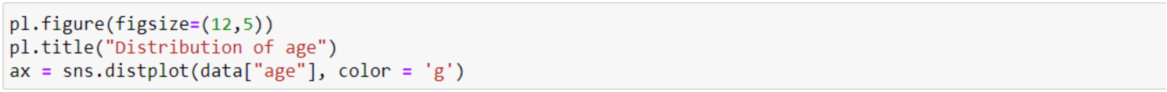




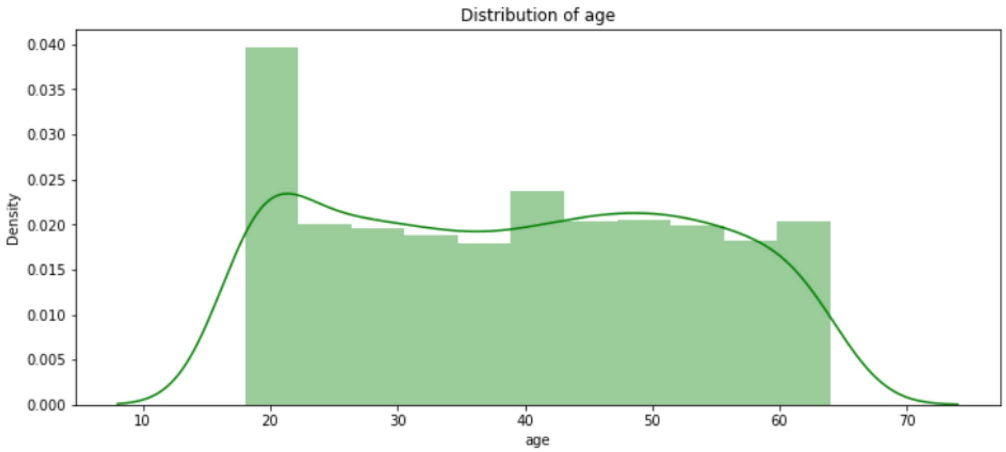
```
In [9]: pl.figure(figsize=(12,5))
        pl.title("Box plot for charges of men")
        sns.boxplot(y="smoker", x="charges", data =  data[(data.sex == 0)] , orient="h", palette = 'rainbow')

Out[9]: <AxesSubplot:title={'center':'Box plot for charges of men'}, xlabel='charges', ylabel='smoker'>
```



Let's pay attention to the age of the patients as it can also be an important factor for treatment and insurance. First, let's look at how age affects the cost of treatment, and look at patients of what age more in our data set.

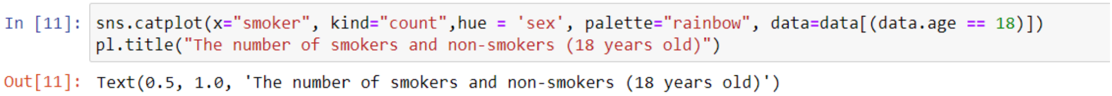


```
pl.figure(figsize=(12,5))
pl.title("Distribution of age")
ax = sns.distplot(data["age"], color = 'g')
```

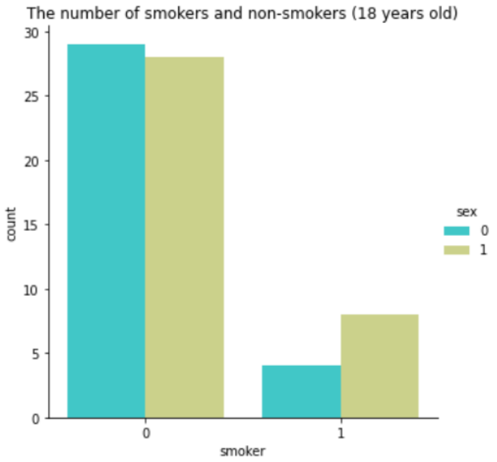


We have patients under 20 in our data set. 18 years is the minimum age of patients in our set. The maximum age is 64 years.

Let’s check whether there are smokers among patients 18 years.

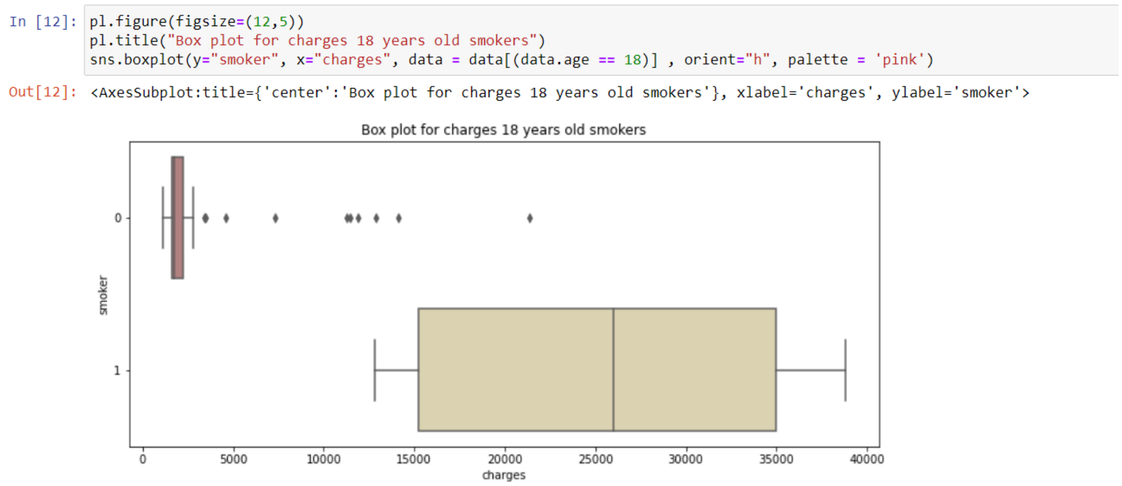


```
In [11]: sns.catplot(x="smoker", kind="count",hue = 'sex', palette="rainbow", data=data[(data.age == 18)])
         pl.title("The number of smokers and non-smokers (18 years old)")

Out[11]: Text(0.5, 1.0, 'The number of smokers and non-smokers (18 years old)')
```



Yes, there are 18 years old smokers.

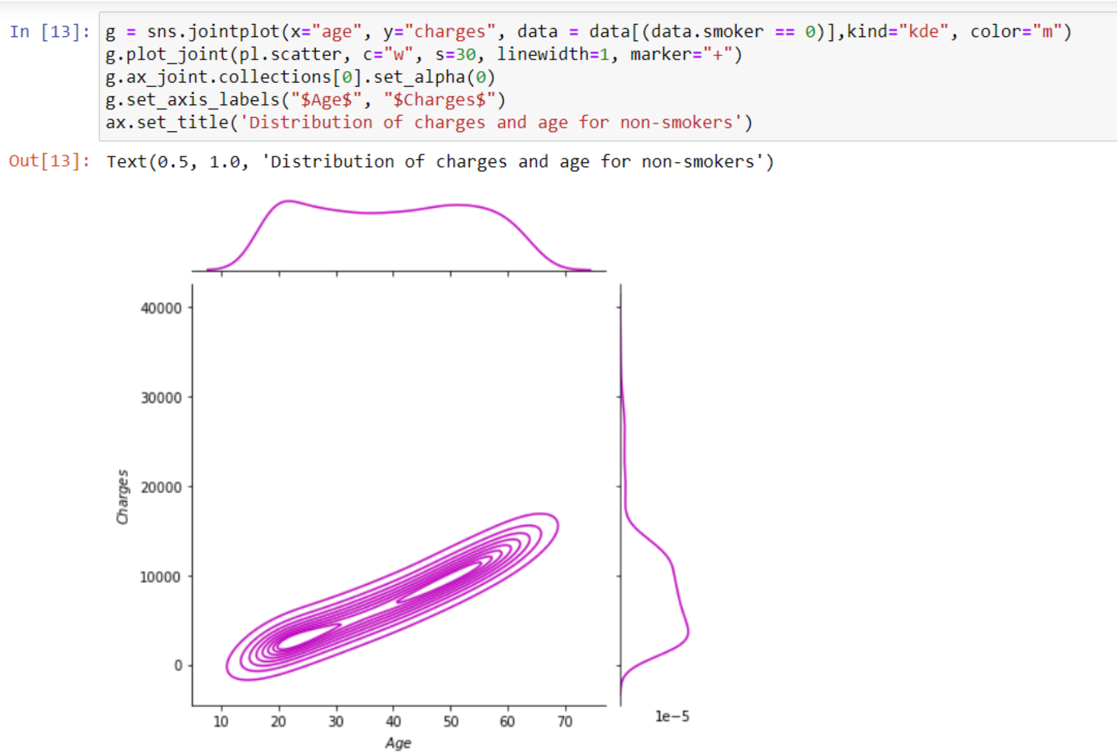
Let’s see whether smoking affects the cost of treatment at this age too.



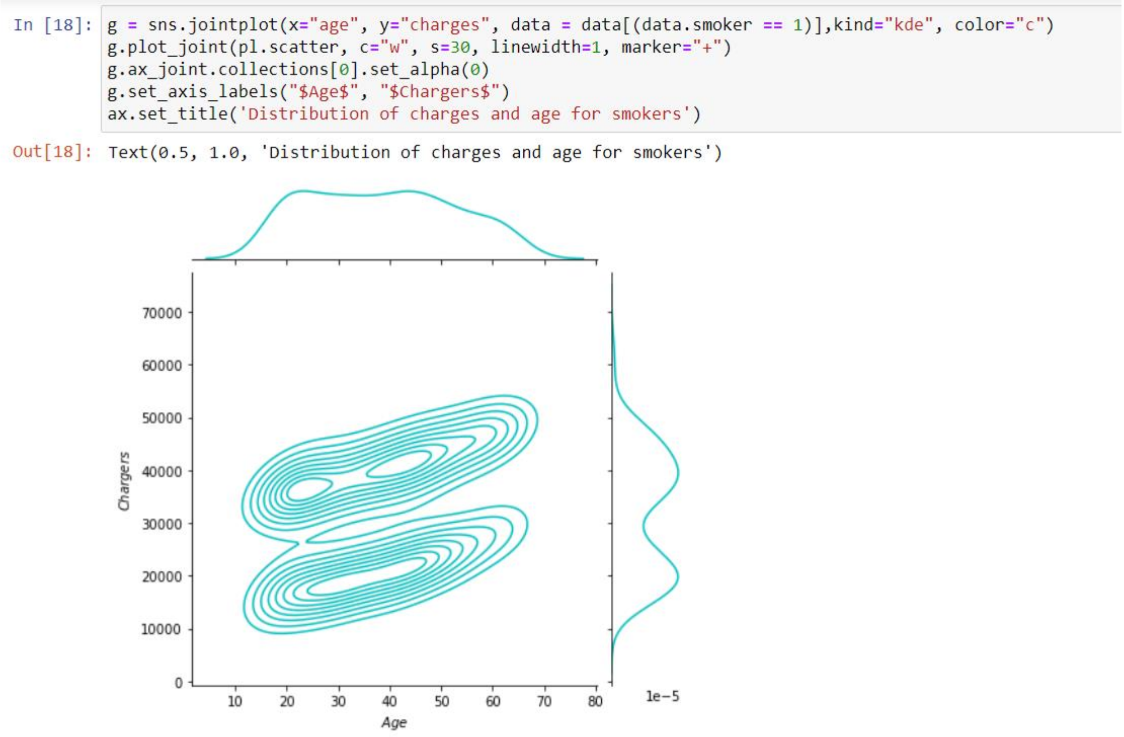
```
In [12]: pl.figure(figsize=(12,5))
         pl.title("Box plot for charges 18 years old smokers")
         sns.boxplot(y="smoker", x="charges", data = data[(data.age == 18)] , orient="h", palette = 'pink')

Out[12]: <AxesSubplot:title={'center':'Box plot for charges 18 years old smokers'}, xlabel='charges', ylabel='smoker'>
```

Even at the age of 18 smokers spend much more on treatment than non-smokers. Among non-smokers we are seeing some outliers.

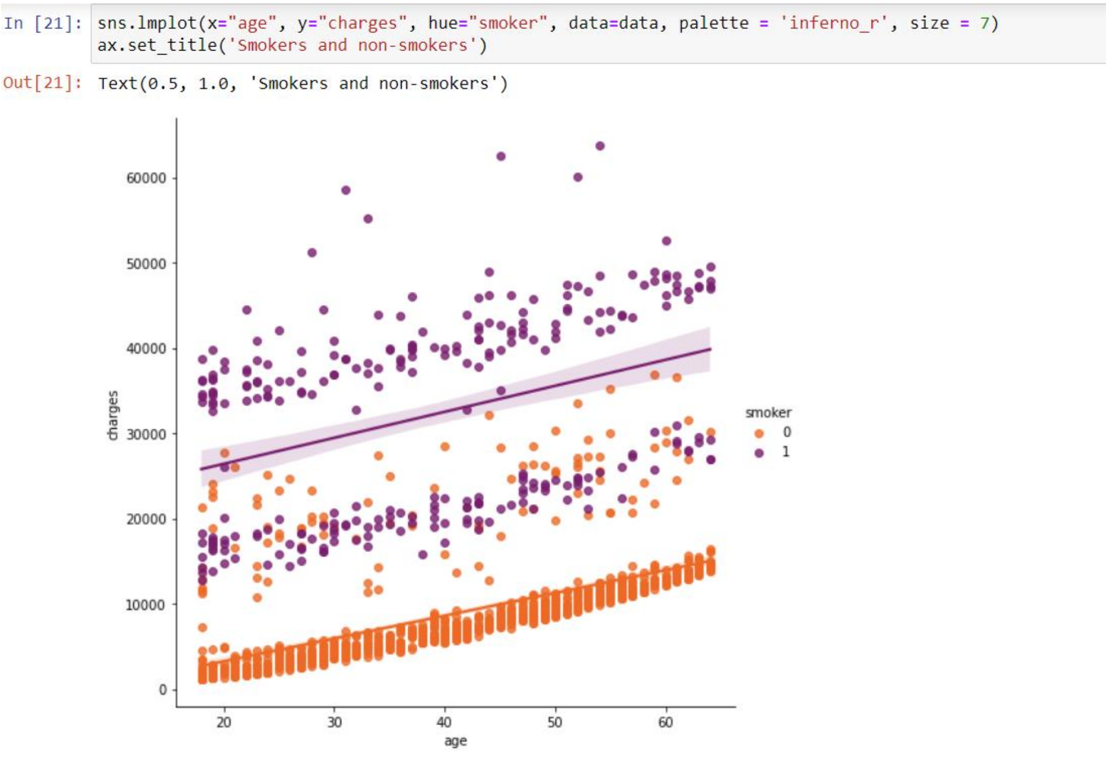
Let's check how the cost of treatment depends on the age of smokers and non-smokers patients.



```
In [13]: g = sns.jointplot(x="age", y="charges", data = data[(data.smoker == 0)],kind="kde", color="m")
         g.plot_joint(pl.scatter, c="w", s=30, linewidth=1, marker="+")
         g.ax_joint.collections[0].set_alpha(0)
         g.set_axis_labels("$Age$", "$Charges$")
         ax.set_title('Distribution of charges and age for non-smokers')

Out[13]: Text(0.5, 1.0, 'Distribution of charges and age for non-smokers')
```



```
In [18]: g = sns.jointplot(x="age", y="charges", data = data[(data.smoker == 1)],kind="kde", color="c")
         g.plot_joint(pl.scatter, c="w", s=30, linewidth=1, marker="+")
         g.ax_joint.collections[0].set_alpha(0)
         g.set_axis_labels("$Age$", "$Chargers$")
         ax.set_title('Distribution of charges and age for smokers')

Out[18]: Text(0.5, 1.0, 'Distribution of charges and age for smokers')
```

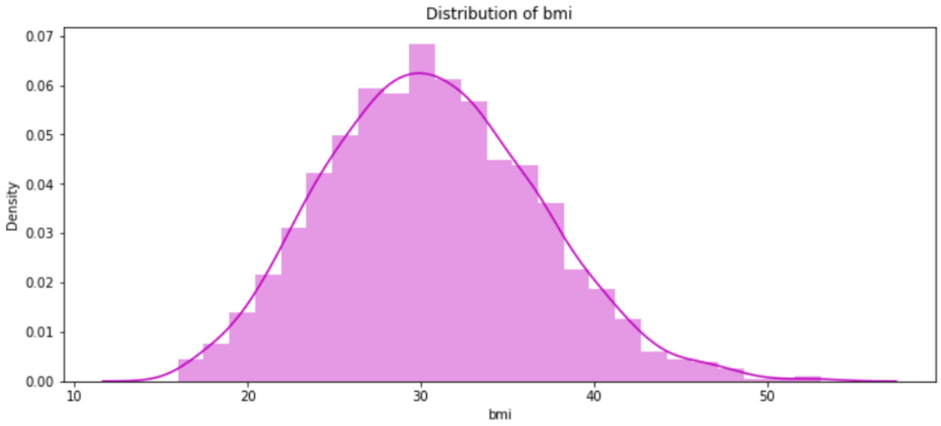


```
In [21]: sns.lmplot(x="age", y="charges", hue="smoker", data=data, palette = 'inferno_r', size = 7)
         ax.set_title('Smokers and non-smokers')

Out[21]: Text(0.5, 1.0, 'Smokers and non-smokers')
```

In non-smokers, the cost of treatment increases with age.

Now let’s compare in terms of “bmi” factor.

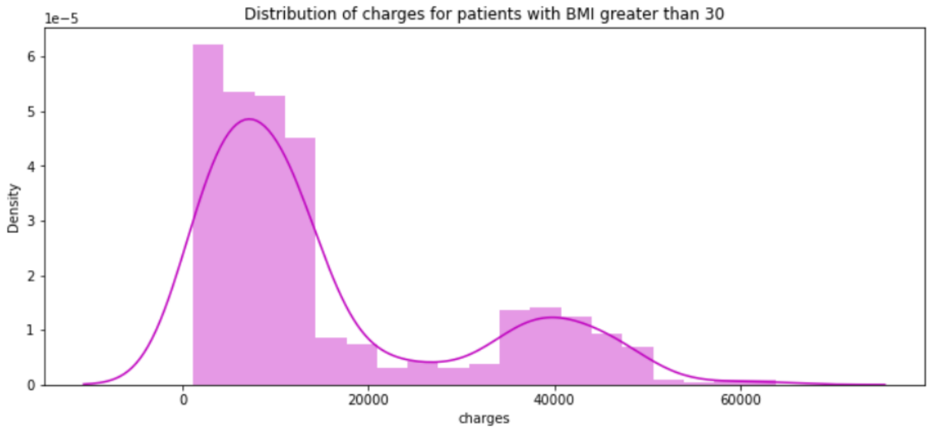
```
In [22]: pl.figure(figsize=(12,5))
         pl.title("Distribution of bmi")
         ax = sns.distplot(data["bmi"], color = 'm')
```



Distribution of bmi

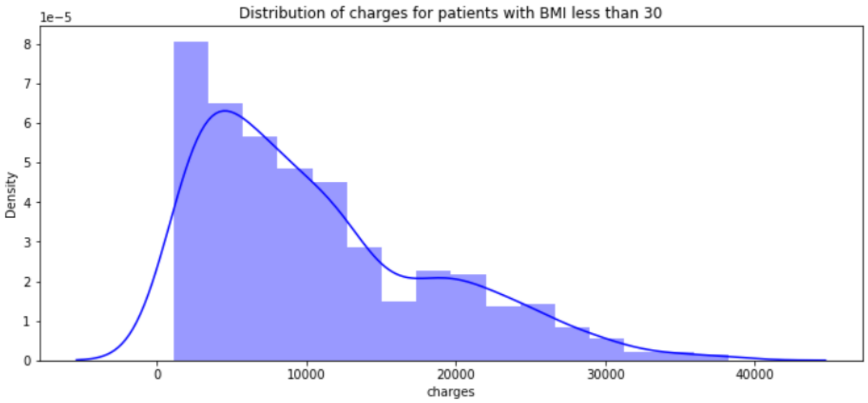
The average BMI in patients is 30.

```
In [23]: pl.figure(figsize=(12,5))
         pl.title("Distribution of charges for patients with BMI greater than 30")
         ax = sns.distplot(data[(data.bmi >= 30)]['charges'], color = 'm')
```



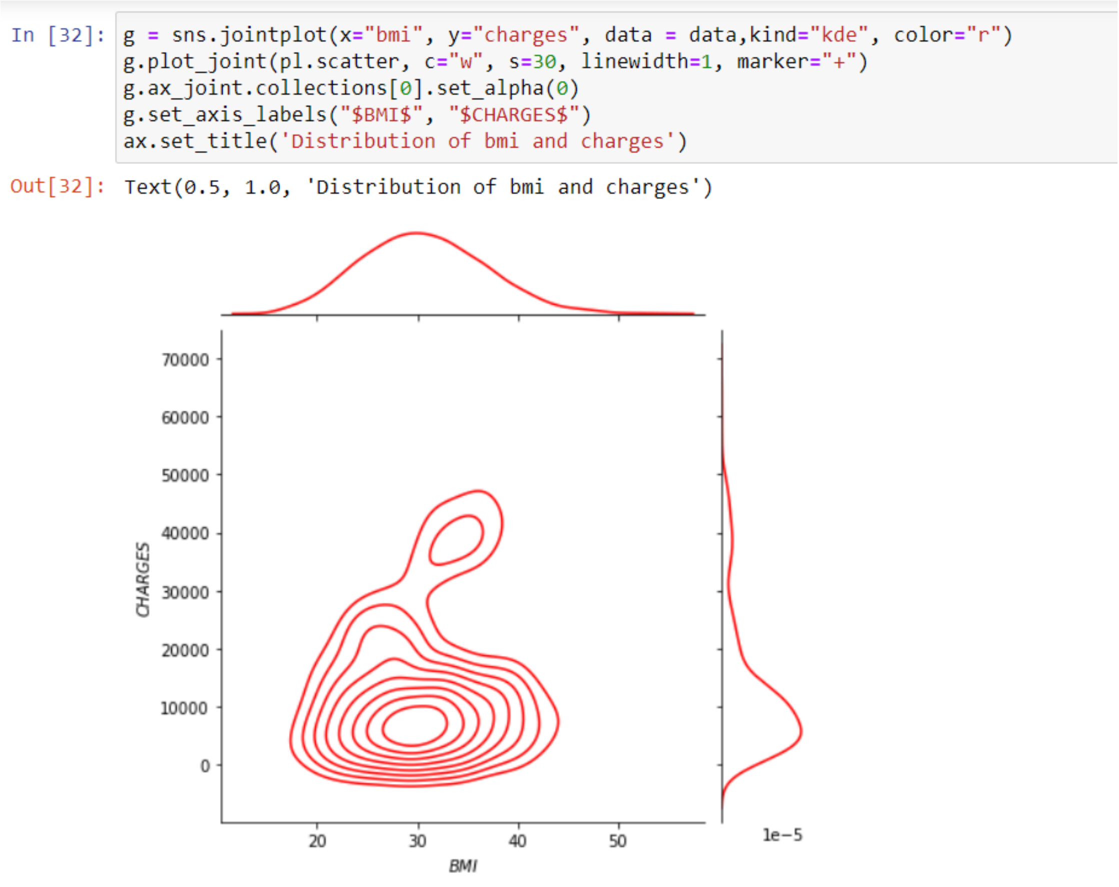
Distribution of charges for patients with BMI greater than 30

```
In [24]: pl.figure(figsize=(12,5))
         pl.title("Distribution of charges for patients with BMI less than 30")
         ax = sns.distplot(data[(data.bmi < 30)]['charges'], color = 'b')
```

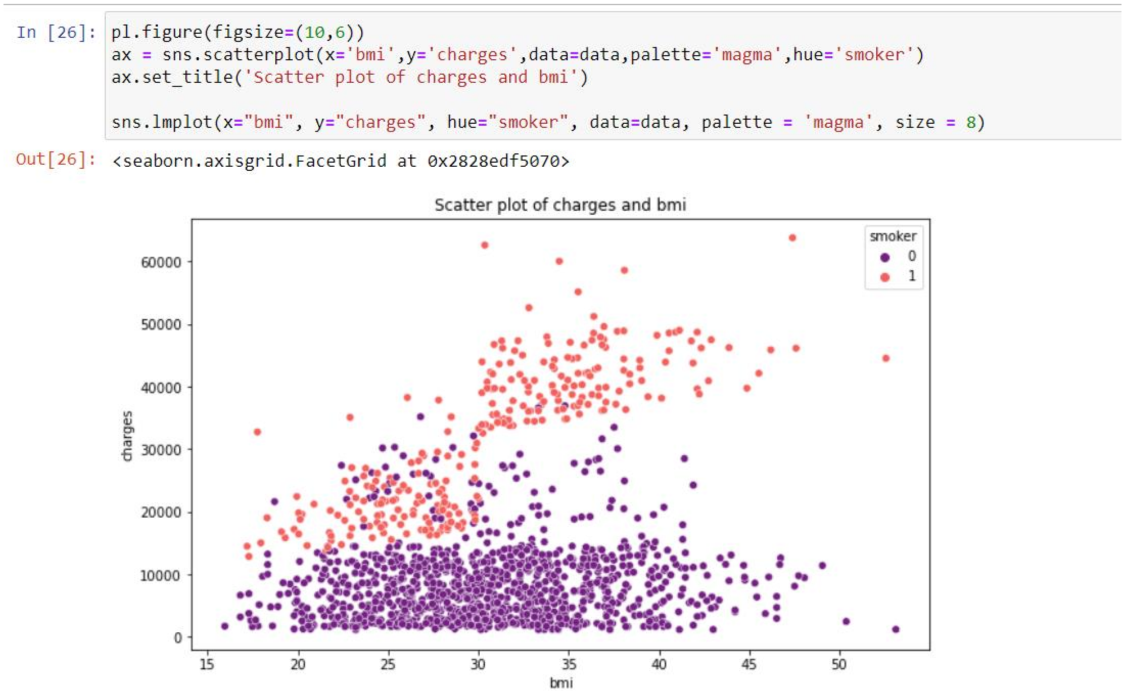


Distribution of charges for patients with BMI less than 30

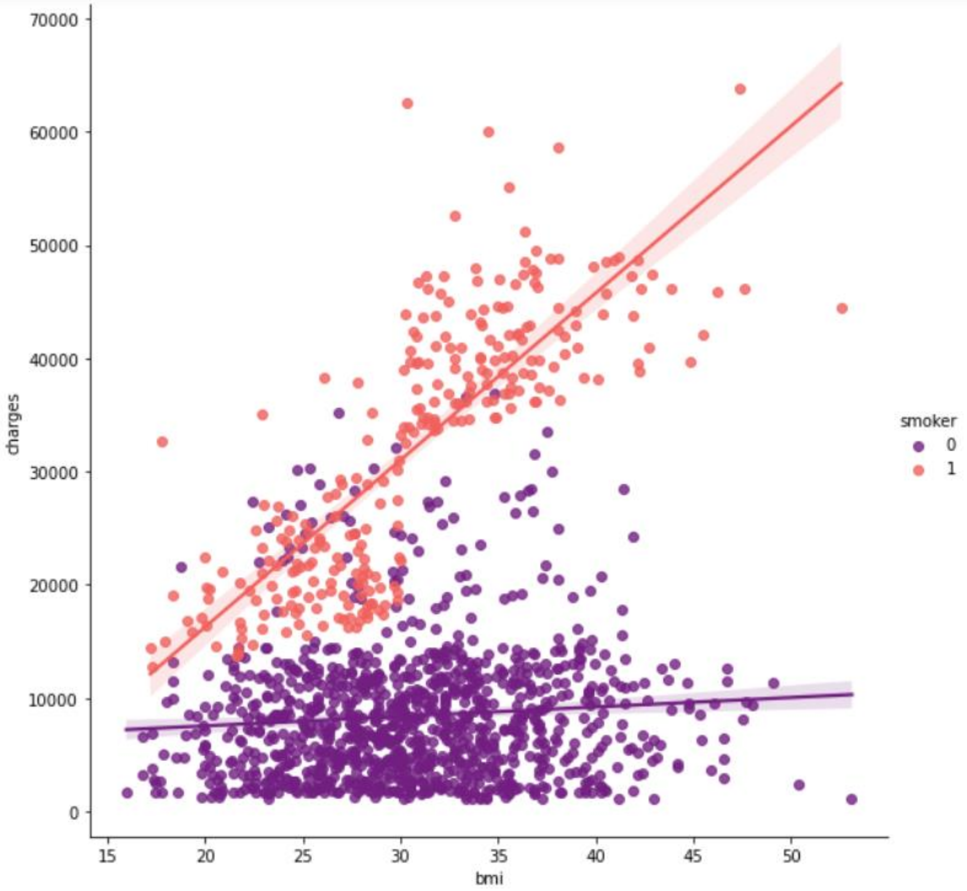
Patients with BMI above 30 spend more on treatment.



```
In [32]: g = sns.jointplot(x="bmi", y="charges", data = data,kind="kde", color="r")
         g.plot_joint(pl.scatter, c="w", s=30, linewidth=1, marker="+")
         g.ax_joint.collections[0].set_alpha(0)
         g.set_axis_labels("$BMI$", "$CHARGES$")
         ax.set_title('Distribution of bmi and charges')

Out[32]: Text(0.5, 1.0, 'Distribution of bmi and charges')
```

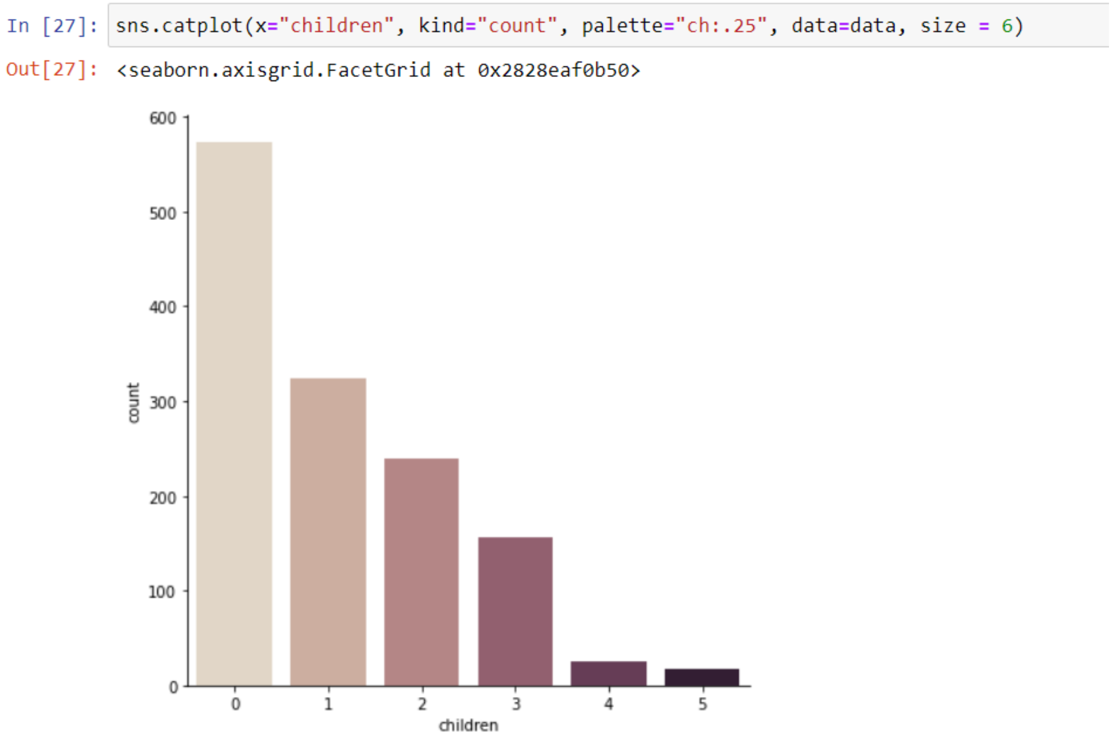


```
In [26]: pl.figure(figsize=(10,6))
         ax = sns.scatterplot(x='bmi',y='charges',data=data,palette='magma',hue='smoker')
         ax.set_title('Scatter plot of charges and bmi')

         sns.lmplot(x="bmi", y="charges", hue="smoker", data=data, palette = 'magma', size = 8)

Out[26]: <seaborn.axisgrid.FacetGrid at 0x2828edf5070>
```



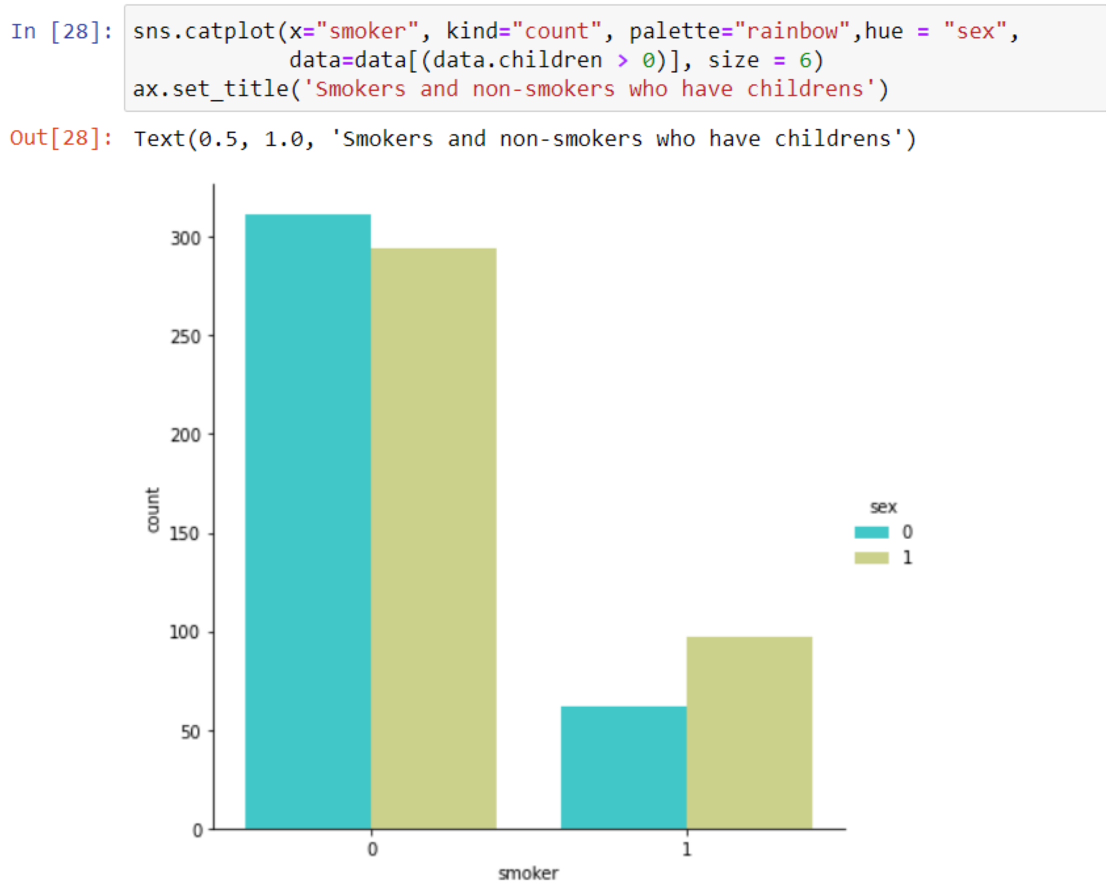
Let's pay attention to children.

First, let's see how many children our patients have.



```
In [27]: sns.catplot(x="children", kind="count", palette="ch:.25", data=data, size = 6)
Out[27]: <seaborn.axisgrid.FacetGrid at 0x2828eaf0b50>
```

Most patients do not have children. And the least number of them have 5.

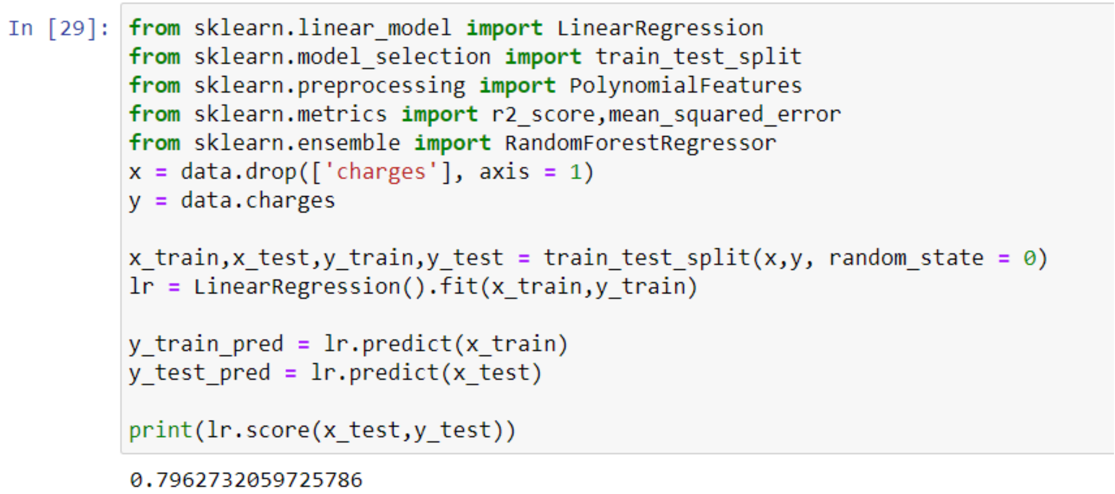


```
In [28]: sns.catplot(x="smoker", kind="count", palette="rainbow",hue = "sex",
                        data=data[(data.children > 0)], size = 6)
         ax.set_title('Smokers and non-smokers who have childrens')

Out[28]: Text(0.5, 1.0, 'Smokers and non-smokers who have childrens')
```

The above graph shows that that non-smoking parents are much more.

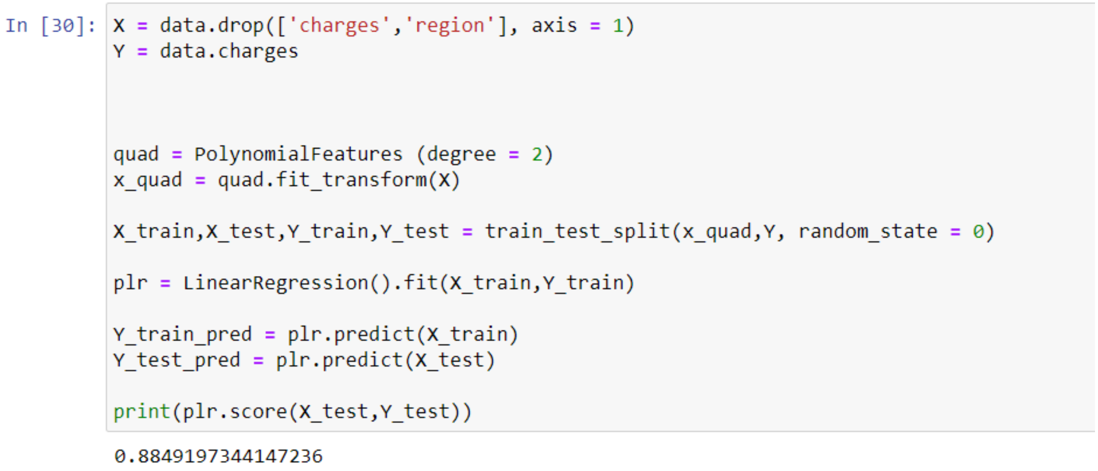
And now we are going to predict the cost of treatment.

By using the regular linear regression, we get nearly 79.62% accuracy.



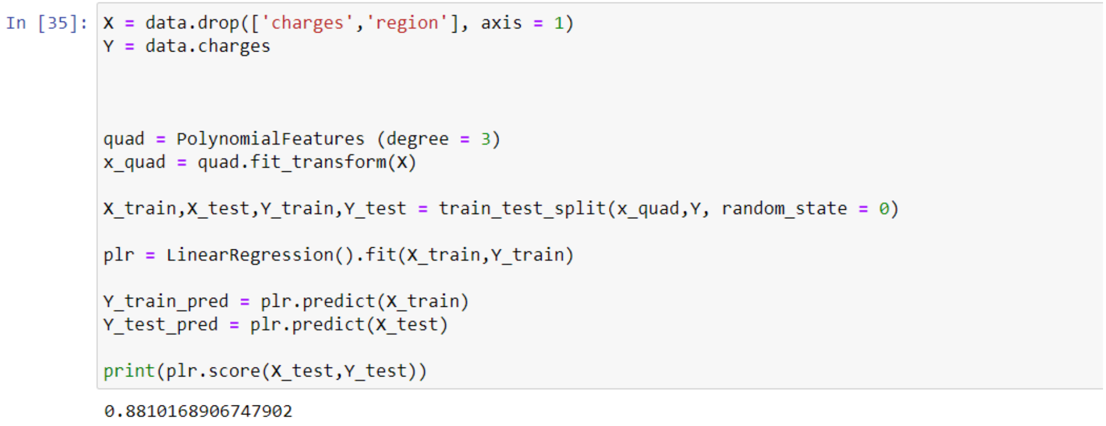
```
In [29]: from sklearn.linear_model import LinearRegression
         from sklearn.model_selection import train_test_split
         from sklearn.preprocessing import PolynomialFeatures
         from sklearn.metrics import r2_score,mean_squared_error
         from sklearn.ensemble import RandomForestRegressor
         x = data.drop(['charges'], axis = 1)
         y = data.charges

         x_train,x_test,y_train,y_test = train_test_split(x,y, random_state = 0)
         lr = LinearRegression().fit(x_train,y_train)

         y_train_pred = lr.predict(x_train)
         y_test_pred = lr.predict(x_test)

         print(lr.score(x_test,y_test))

         0.7962732059725786
```

But the accuracy percentage has increased to 88.49% by using polynomial kernel with degree 2.



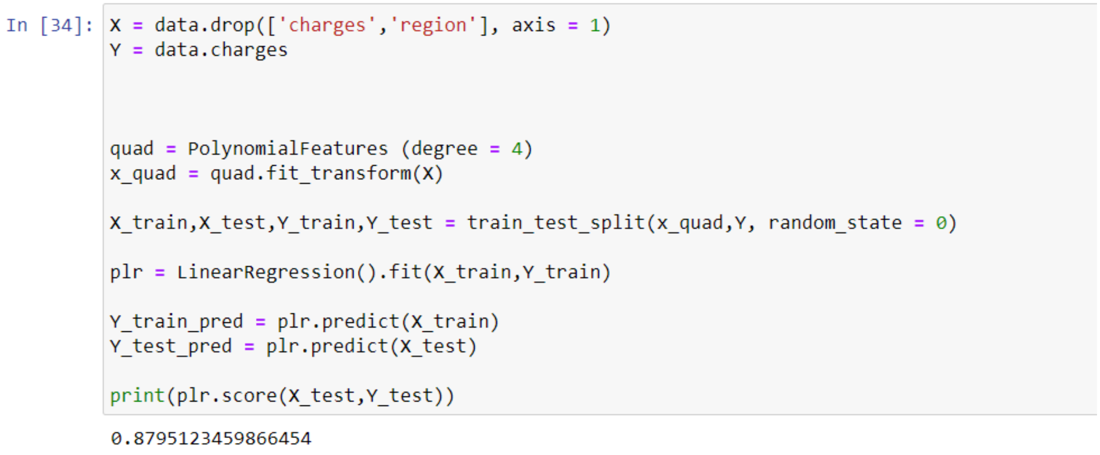
```
In [30]: X = data.drop(['charges','region'], axis = 1)
         Y = data.charges



         quad = PolynomialFeatures (degree = 2)
         x_quad = quad.fit_transform(X)

         X_train,X_test,Y_train,Y_test = train_test_split(x_quad,Y, random_state = 0)

         plr = LinearRegression().fit(X_train,Y_train)

         Y_train_pred = plr.predict(X_train)
         Y_test_pred = plr.predict(X_test)

         print(plr.score(X_test,Y_test))

         0.8849197344147236
```

With degree 3 again the accuracy is lowered to 88.10%



```
In [35]: X = data.drop(['charges','region'], axis = 1)
         Y = data.charges



         quad = PolynomialFeatures (degree = 3)
         x_quad = quad.fit_transform(X)

         X_train,X_test,Y_train,Y_test = train_test_split(x_quad,Y, random_state = 0)

         plr = LinearRegression().fit(X_train,Y_train)

         Y_train_pred = plr.predict(X_train)
         Y_test_pred = plr.predict(X_test)

         print(plr.score(X_test,Y_test))

         0.8810168906747902
```

With degree as 4, it is 87.95%.



```
In [34]: X = data.drop(['charges','region'], axis = 1)
         Y = data.charges



         quad = PolynomialFeatures (degree = 4)
         x_quad = quad.fit_transform(X)

         X_train,X_test,Y_train,Y_test = train_test_split(x_quad,Y, random_state = 0)

         plr = LinearRegression().fit(X_train,Y_train)

         Y_train_pred = plr.predict(X_train)
         Y_test_pred = plr.predict(X_test)

         print(plr.score(X_test,Y_test))

         0.8795123459866454
```

Code and dataset drive link:Click here for python code file and csv file