**Statistical Learning Graded assignment**

**1. Import the necessary libraries**

*import numpy as np*

*import pandas as pd*

*from matplotlib import pyplot as plt*

*import seaborn as sns*

*import scipy.stats as stats*

*from statsmodels.stats.proportion import proportions\_ztest*

Scipy.stats and statsmodels.stats.proportion are the libraries that contain functions to carry out t test and z test respectively

Matplotlib.pyplot and seaborn are used to plot graphs

Pandas is used to work on the dataframe and it contains all the functions to process the dataframes.

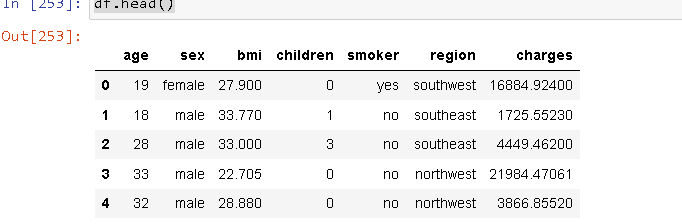
Numpy contains mathematical functions.

**2. Read the data as a data frame (3 marks)**

*df=pd.read\_csv(r'C:\Users\hp\Desktop\insurance.csv')*

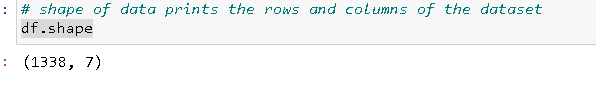
*df.head()*

We use pandas library to upload the dataset. A preliminary process head() can be carried out to have a first view of the dataframe.



**3. Perform basic EDA which should include the following and print out your insights at every step. ( 27 marks)**

**a. Shape of the data**

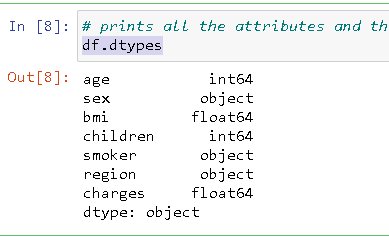
*df.shape*

Shape function returns the rows and columns of the dataframe.

Inference: The current dataset contains 1338 rows (entries) and 7 columns (variables)

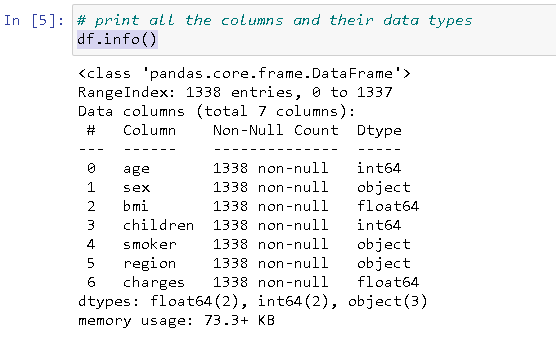
**b. Data type of each attribute**

*df.dtypes*



‘dtypes’ returns list of attributes and their data-types. However we use info() function more often as it shows attributes, their data types as well as the number of non-null entries they have .

*df.info()*

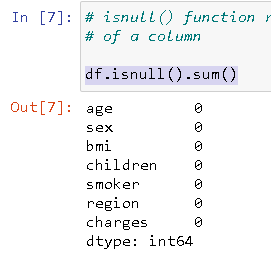
**

Info() function returns list of columns, count of non-null entries in all columns and the data-types of all the variables.

Inference: age and children variables are integer in nature (children acts more like a categorical variable than a numerical variable). Charges and bmi are float variables and, region, smoker and sex are character variables (object variables)

**c. Checking the presence of missing values**

*df.isnull().sum()*

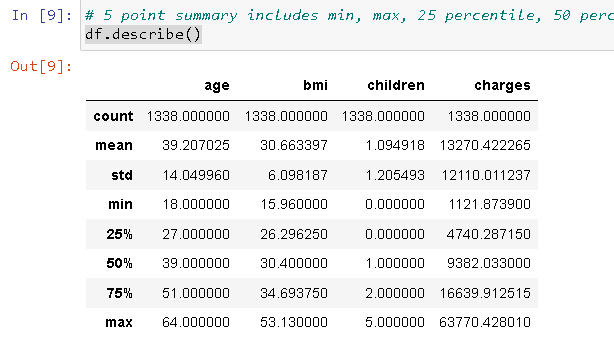
**

Isnull() function returns 'true' value to NaN values and appending sum() to it, returns the total number of null values for each column.

Inference: The current has no missing values.

**d. 5 point summary of numerical attributes**

*df.describe()*

**

*plt.figure(figsize= (20,15))*

*plt.subplot(3,3,1)*

*sns.boxplot(df.age)*

*plt.xlabel('charges')*

*plt.subplot(3,3,2)*

*sns.boxplot(df.bmi)*

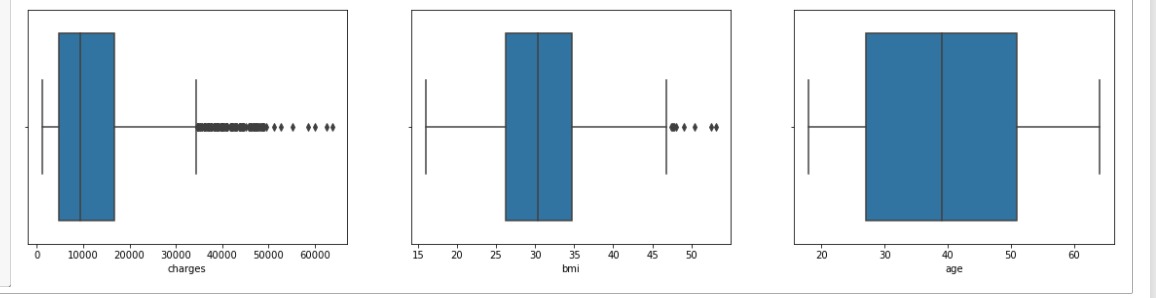
*plt.xlabel('bmi')*

*plt.subplot(3,3,3)*

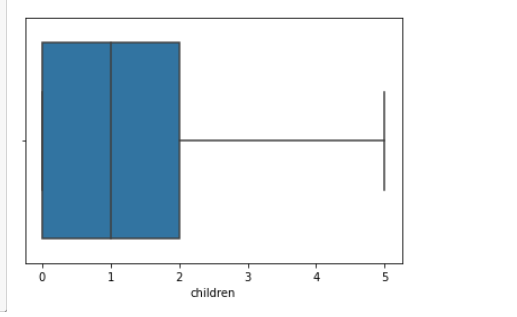
*sns.boxplot(df.age)*

*plt.xlabel('age')*

*plt.show()*

**

*sns.boxplot(df.children)*



5 point summary includes min, max, 25 percentile, 50 percentile (median), 75 percentile. These are the subset points described by describe() function.

We have also used boxplots, which show the 5 point summaries along with showing the outliers.

Children column functions as a categorical variable rather than numerical variable.

Inference: (min, 25%ile, 50%ile, 75%ile, max) for age: (18, 27, 39, 51, 64)

(min, 25%ile, 50%ile, 75%ile, max) for charges: (1121.87, 4740.28, 9382.03, 16639.91, 63770.42)

(min, 25%ile, 50%ile, 75%ile, max) for bmi: (15.96, 26.29, 30.40, 34.69, 53.13)

(min, 25%ile, 50%ile, 75%ile, max) for children: (0, 0, 1, 2, 5)

**e. Distribution of ‘bmi’, ‘age’ and ‘charges’ columns.**

*plt.figure(figsize= (20,15))*

*plt.subplot(3,3,1)*

*plt.hist(df.age, edgecolor = 'white', alpha = 0.7)*

*plt.xlabel('age')*

*plt.subplot(3,3,2)*

*plt.hist(df.bmi, edgecolor='white', bins=30)*

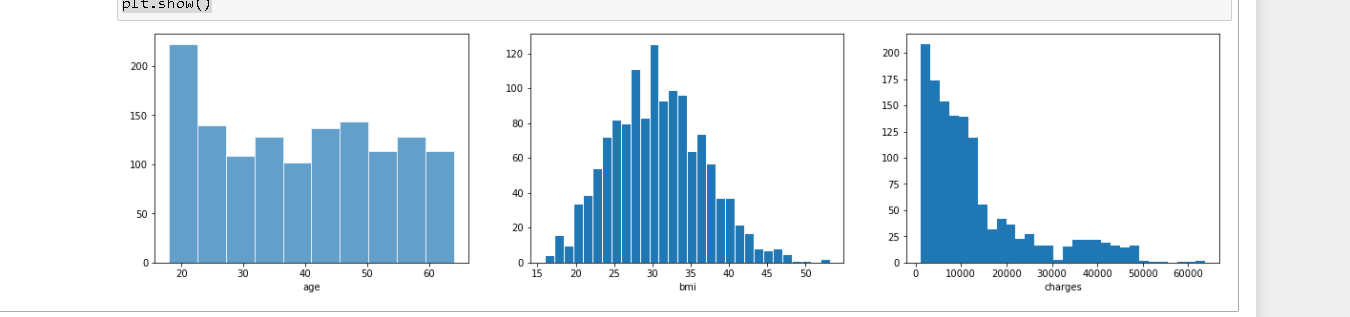
*plt.xlabel('bmi')*

*plt.subplot(3,3,3)*

*plt.hist(df.charges, bins=30)*

*plt.xlabel('charges')*

*plt.show()*

**

We can see that age and charges have measurable right skew. It is further superimposed when we plot distribution using seaborn, which shows the normal fitting curve as well.

*plt.figure(figsize= (20,15))*

*plt.subplot(3,3,1)*

*sns.distplot(df.age, bins=50)*

*plt.xlabel('age')*

*plt.subplot(3,3,2)*

*sns.distplot(df.bmi)*

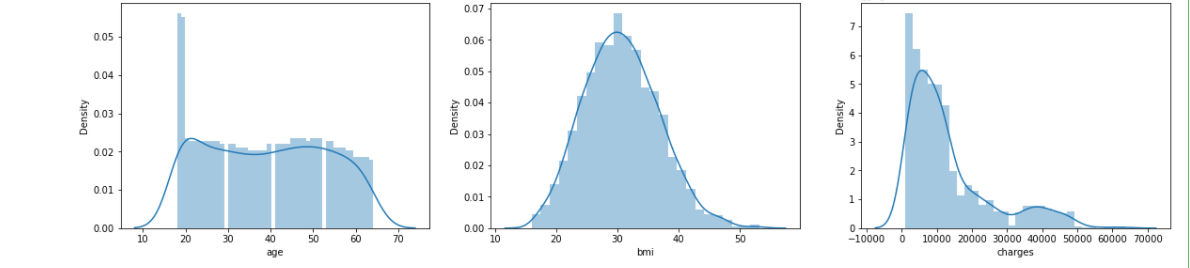
*plt.xlabel('bmi')*

*plt.subplot(3,3,3)*

*sns.distplot(df.charges)*

*plt.xlabel('charges')*

*plt.show()*

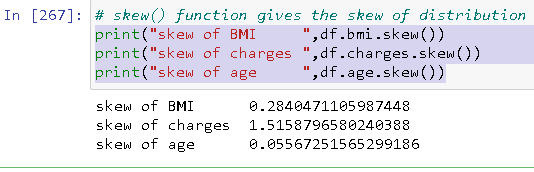
**

**f. Measure of skewness of ‘bmi’, ‘age’ and ‘charges’ columns**

*print("skew of BMI ",df.bmi.skew())*

*print("skew of charges ",df.charges.skew())*

*print("skew of age ",df.age.skew())*

**

As expected from the previous distribution plots, we get a measurable right skewness for bmi and charges. Skewness in age is insignificant.

**g. Checking the presence of outliers in ‘bmi’, ‘age’ and ‘charges columns**

*plt.figure(figsize= (20,15))*

*plt.subplot(3,3,1)*

*sns.boxplot(df.age)*

*plt.xlabel('charges')*

*plt.subplot(3,3,2)*

*sns.boxplot(df.bmi)*

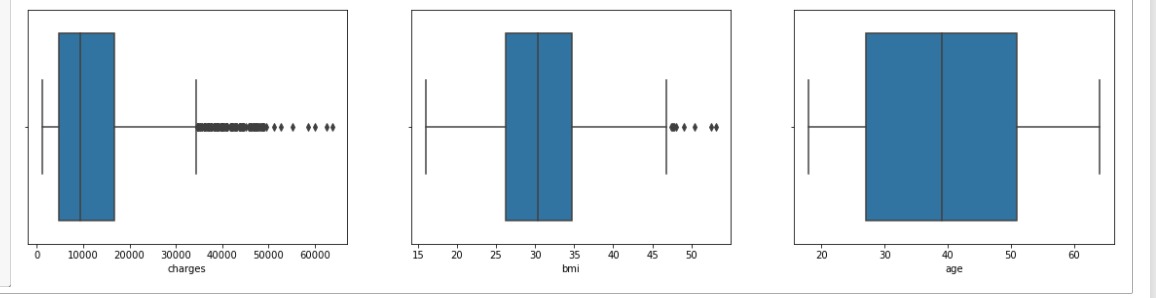
*plt.xlabel('bmi')*

*plt.subplot(3,3,3)*

*sns.boxplot(df.age)*

*plt.xlabel('age')*

*plt.show()*

**

Box plots are very efficient in spotting the outliers. They visibly separate the outliers if they fall beyond the interquartile range. We can calculate the number of outliers using the interquartile range as well using the following code.

*bmiQ1 = df.bmi.quantile(0.25)*

*bmiQ3 = df.bmi.quantile(0.75)*

*bmiIQR = bmiQ3 - bmiQ1*

*chargesQ1 = df.charges.quantile(0.25)*

*chargesQ3 = df.charges.quantile(0.75)*

*chargesIQR = chargesQ3 - chargesQ1*

*ageQ1 = df.age.quantile(0.25)*

*ageQ3 = df.age.quantile(0.75)*

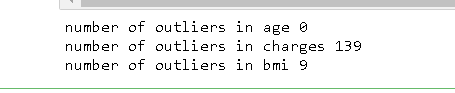
*ageIQR = ageQ3 - ageQ1*

Determination of IQR for age, charges and bmi variables.

*print("number of outliers in age", ((df.age < (ageQ1 - 1.5 \* ageIQR)) | (df.age > (ageQ3 + 1.5 \* ageIQR))).sum())*

*print("number of outliers in charges",((df.charges < (chargesQ1 - 1.5 \* chargesIQR)) | (df.charges > (chargesQ3 + 1.5 \* chargesIQR))).sum())*

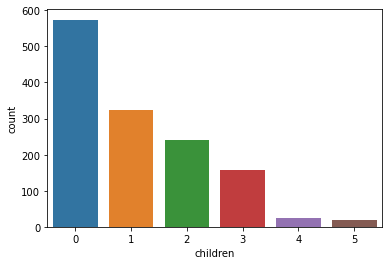
*print("number of outliers in bmi",((df.bmi < (bmiQ1 - 1.5 \* bmiIQR)) | (df.bmi > (bmiQ3 + 1.5 \* bmiIQR))).sum())*

**

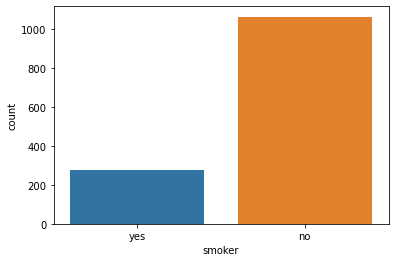
In the above codes, we have counted those values which fall beyond the IQR for their respective quantiles. As expected by seeing the boxplots, charges has highest number of outliers followed by BMI and age has no outliers.

**h. Distribution of categorical columns (include children)**

*sns.countplot(df.children)*

**

*sns.countplot(df.smoker)*

**

*sns.countplot(df.sex)*

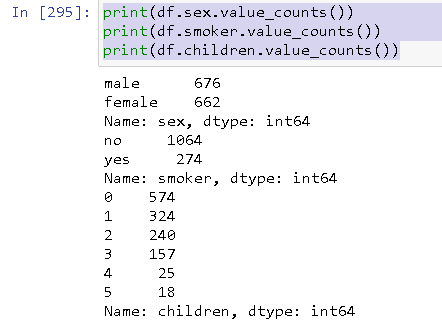
**

Count plots are very useful in plotting the categorical variables with their frequencies. We can easily see the value counts by using value\_count function as well.

*print(df.sex.value\_counts())*

*print(df.smoker.value\_counts())*

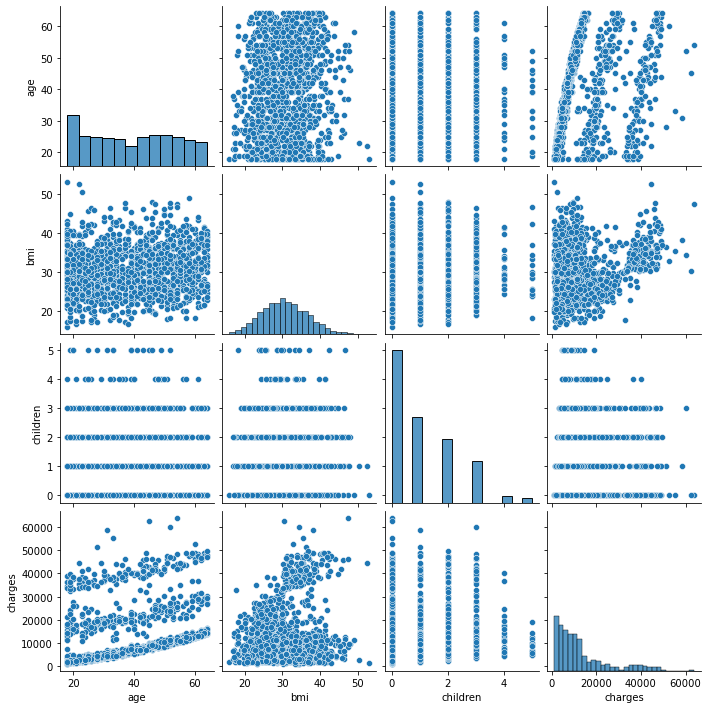
*print(df.children.value\_counts())*

**

Inference: We can conclude from this data that the entries are evenly taken from both the genders. Smokers are around 22% of the total population. And number of individuals decrease with increasing children (which means that people prefer having fewer kids)

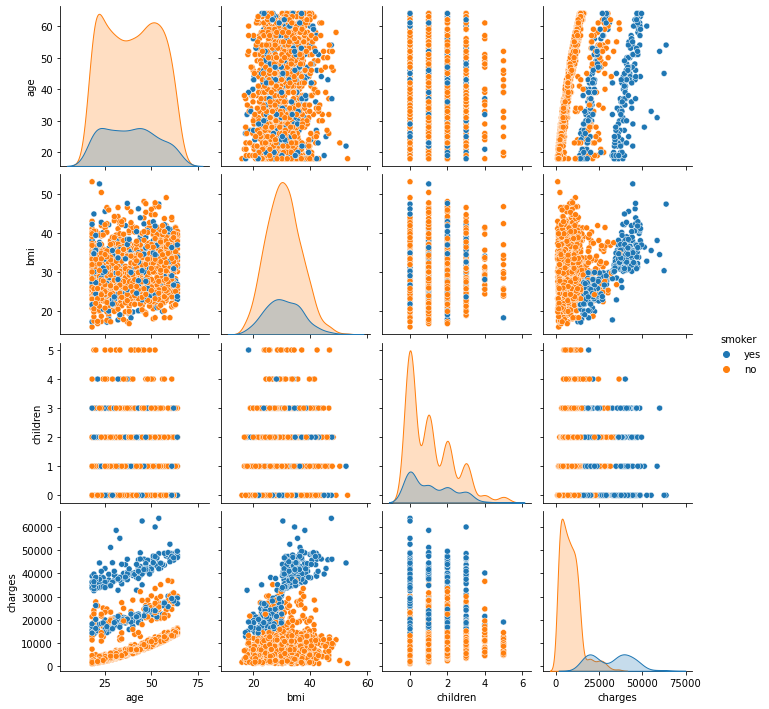
**i. Pair plot that includes all the columns of the data frame**

*sns.pairplot(df)*

**

categorical variables cannot be plotted on pairplots, at best we can add categorical variables as hue (as following)

*sns.pairplot(df, hue='smoker')*

**

To plot pairplots of the categorical variables too, we need to encode them as numerical variables. We are encoding the categorical variables in the following codes

*df['regionid'] =1*

*df.loc[(df['region']=='southwest'), 'regionid'] =1*

*df.loc[(df['region']=='northwest'), 'regionid'] =2*

*df.loc[(df['region']=='southeast'), 'regionid'] =3*

*df.loc[(df['region']=='northeast'), 'regionid'] =4*

*df['sexid'] =1*

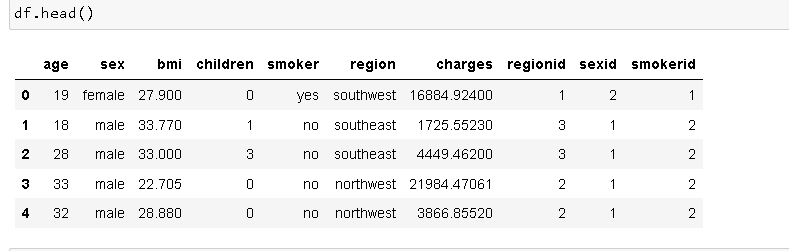
*df.loc[(df['sex']=='male'), 'sexid'] =1*

*df.loc[(df['sex']=='female'), 'sexid'] =2*

*df['smokerid'] =1*

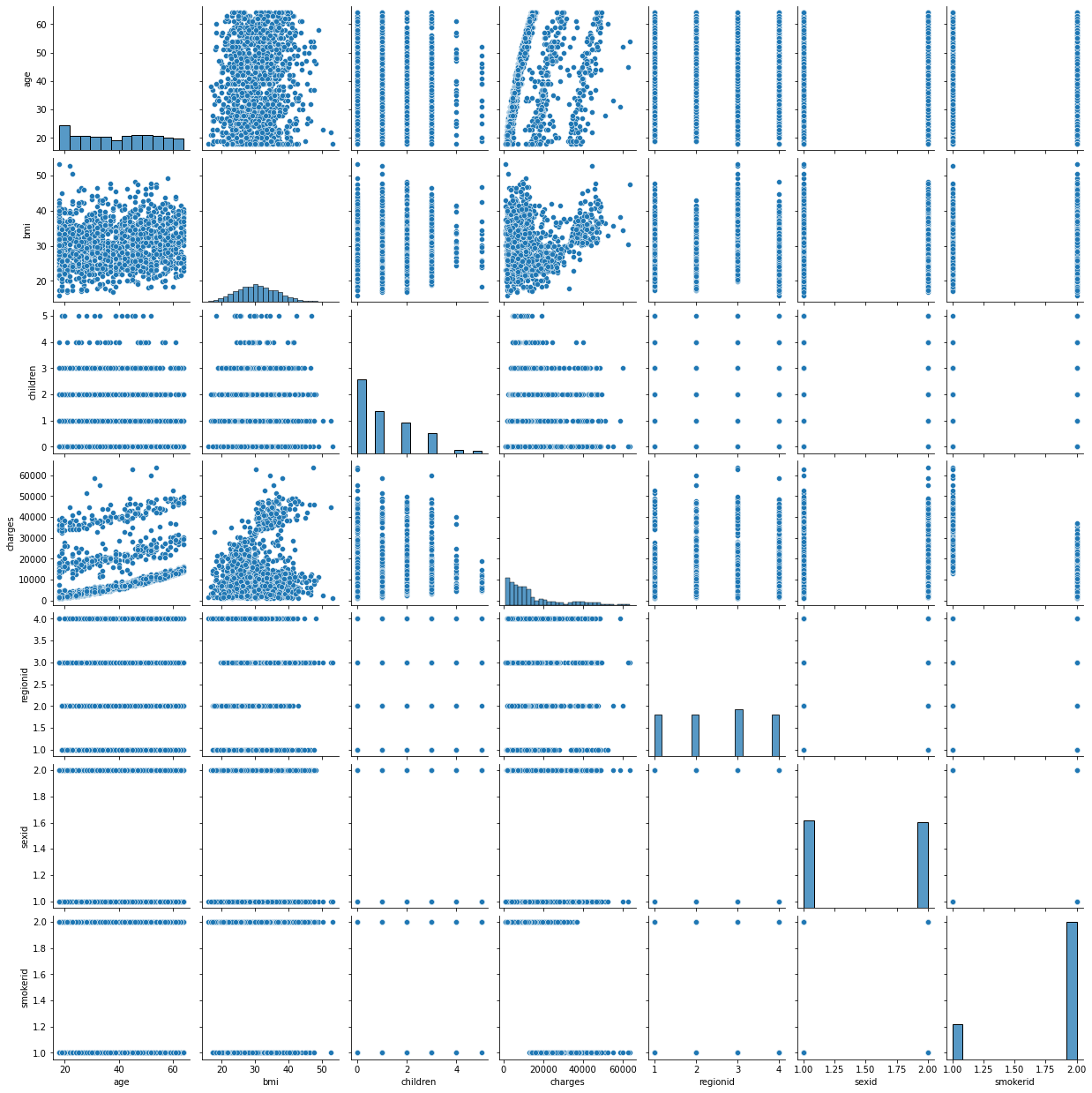
*df.loc[(df['smoker']=='yes'), 'smokerid'] =1*

*df.loc[(df['smoker']=='no'), 'smokerid'] =2*

**

We can see that three variables regionid, sexid and smokerid have been created pertaining to region, sex and smoker columns, these can now be used for pairplots.

*sns.pairplot(df)*

**

Inference: It can be concluded from the pair plots that bmi, charges and age all three have positive relation to each other. Non smokers have lower charges compared to the smokers.

**# 4. Answer the following questions with statistical evidence (20 marks)**

**a) Do charges of people who smoke differ significantly from the people who don't?**

*smokers=df[df['smoker']== 'yes']*

*nonsmokers=df[df['smoker']=='no']*

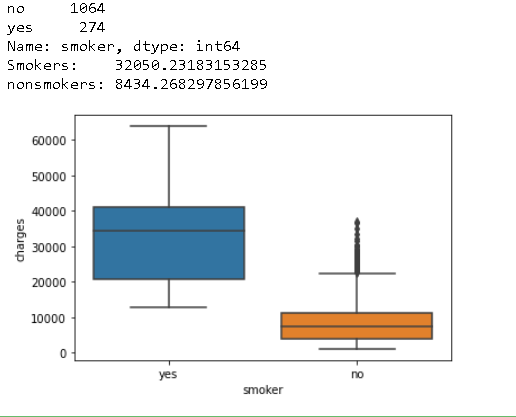
In the above code we are creating two separate dataframes on the basis of smoking habits of an individual.

*print(df.smoker.value\_counts())*

*sns.boxplot(x=df.smoker, y=df.charges)*

*print('Smokers: ',smokers.charges.mean())*

*print('nonsmokers:',nonsmokers.charges.mean())*

**

From the plot and the means we can clealy see that there is a significant difference in the charges of smokers and nonsmokers. Thus we can expect that same is reflected in the t-test. Since the two data points are unrelated, we take them to be independent

Name of test: two-sample t-test

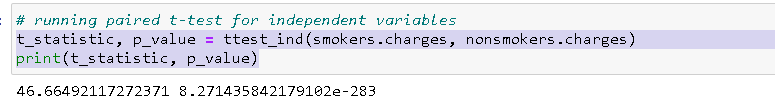
Null hypothesis: both smokers and nonsmokers have same charges

Alternate hypothesis: charges of people who smoke differ from those who don't

This test assumes the two groups have the same variance...

*t\_statistic, p\_value = ttest\_ind(smokers.charges, nonsmokers.charges)*

*print(t\_statistic, p\_value)*

**

Inference: (p\_value < 0.05) => alternate hypothesis:charges of people who smoke differ from those who don't. We have already seen that charges for smokers are around 3.7 times higher than those who don't smoke

**b) Does bmi of males differ significantly from that of females?**

*males=df[df['sex']== 'male']*

*females=df[df['sex']=='female']*

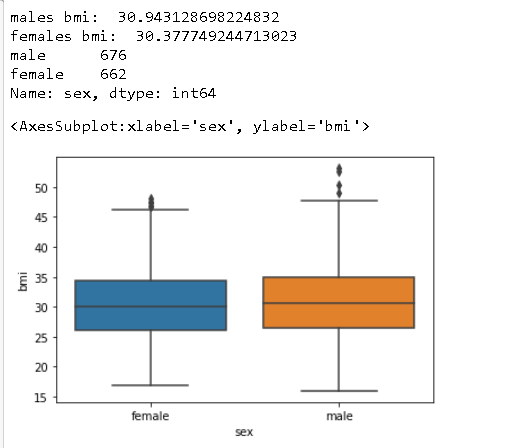
Creating two separate dataset based on the sex of an individual..

*print("males bmi: ",males.bmi.mean())*

*print("females bmi: ",females.bmi.mean())*

*print(df.sex.value\_counts())*

*sns.boxplot(x=df.sex, y=df.bmi)*

**

From the graph, we can see that mean bmi for both the sexes are not significantly different

Name of test: two-sample t-test

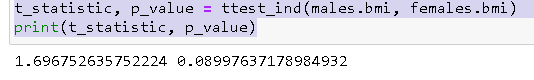
Null hypothesis: both males and females have same bmi

Alternate hypothesis: bmi of males differ from females

This test assumes the two groups have the same variance...

*t\_statistic, p\_value = ttest\_ind(males.bmi, females.bmi)*

*print(t\_statistic, p\_value)*

**

Inference: (p\_value > 0.05) => fail to reject the null hypothesis. We had already seen that mean bmi for both the sexes are very close.

**c) Is the proportion of smokers significantly different in different genders?**

*femalesmokers=df[df['sex']=='female'].smoker.value\_counts()[1]*

*malesmokers=df[df['sex']=='male'].smoker.value\_counts()[1]*

*numbermales=df.sex.value\_counts()[0]*

*numberfemales=df.sex.value\_counts()[1]*

Creating new separate data for female smokers and male smokers.

*print(df.sex.value\_counts())*

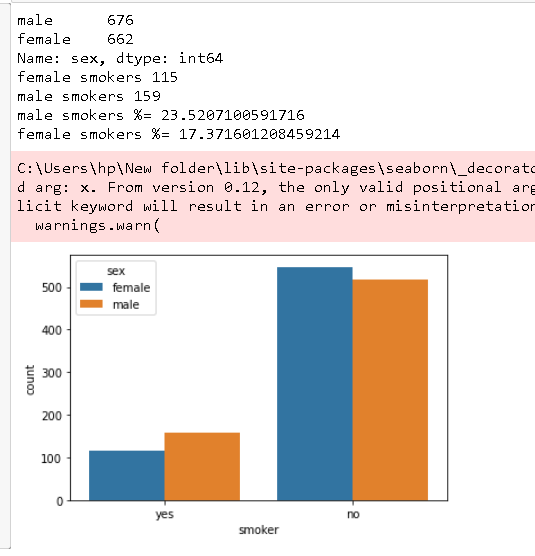
*print('female smokers',femalesmokers)*

*print('male smokers',malesmokers)*

*sns.countplot(df.smoker,hue = df.sex)*

*print('male smokers %=', malesmokers\*100/numbermales)*

*print('female smokers %=', femalesmokers\*100/numberfemales)*

**

We can see from the graph that number of male smokers is higher than female smokers

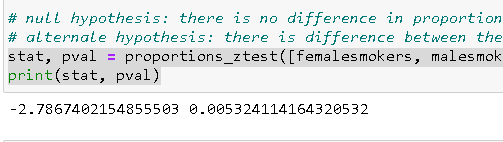
Name of test: Z-test of proportions

Null hypothesis: there is no difference in proportion among male and females with respect to smoking habit

Alternale hypothesis: there is difference between the proportion of males smoking versus females smoking

*stat, pval = proportions\_ztest([femalesmokers, malesmokers] , [numberfemales, numbermales])*

*print(stat, pval)*

**

Inference: p<0.05: we reject the null, i.e., alternate hypothesis stands. One sex has higher proportion of smokers than the others. We can safely conclude that males have higher proportion of smokers compared to females

**d) Is the distribution of bmi across women with no children, one child and two children, the same ?**

*women\_0= df[(df['sex'] =='female') & (df['children'] ==0) ]*

*women\_1= df[(df['sex'] =='female') & (df['children'] ==1) ]*

*women\_2= df[(df['sex'] =='female') & (df['children'] ==2) ]*

In the above code we create separate datasets for women with no children, 1 children and 2 children.

*print(women\_0.bmi.mean())*

*print(women\_1.bmi.mean())*

*print(women\_2.bmi.mean())*

*plt.figure(figsize= (20,15))*

*plt.subplot(3,3,1)*

*sns.distplot(women\_0.bmi)*

*plt.xlabel('women\_0')*

*plt.subplot(3,3,2)*

*sns.distplot(women\_1.bmi)*

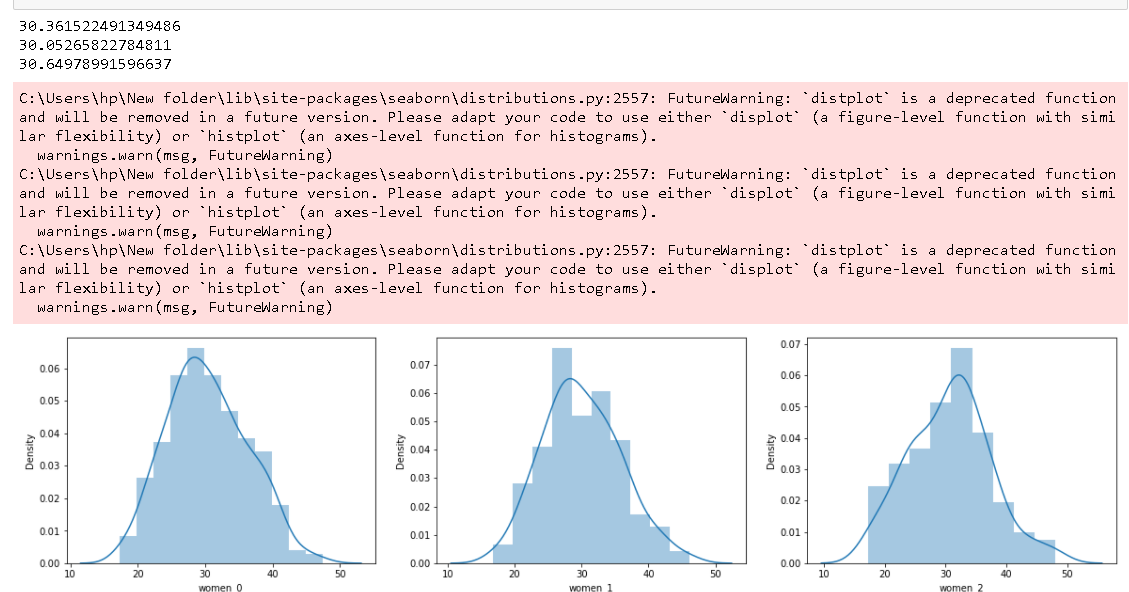
*plt.xlabel('women\_1')*

*plt.subplot(3,3,3)*

*sns.distplot(women\_2.bmi)*

*plt.xlabel('women\_2')*

*plt.show()*

** mean for bmi of women in all the categories is similar

*v1, v2, v3 = np.var(women\_0.bmi) , np.var(women\_1.bmi), np.var(women\_2.bmi)*

*stddev1, stddev2, stddev3= statistics.stdev(women\_0.bmi) , statistics.stdev(women\_1.bmi), statistics.stdev(women\_2.bmi)*

*print("women\_0: variance",v1,"women\_1: variance",v2,"women\_2: variance",v3)*

*print("women\_0: std dev",stddev1,"women\_1: std dev",stddev2,"women\_2: std dev",stddev3)*

**

different variance and stddev seem to be present in three samples, but we have to check if they are significant

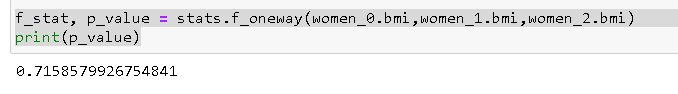
Name of test: one way-anova test

Null hypothesis: women in all three categories have similar bmi

Alternate hypothesis: bmi of women is dependent on the number of kids they have

*f\_stat, p\_value = stats.f\_oneway(women\_0.bmi,women\_1.bmi,women\_2.bmi)*

*print(p\_value)*

**

Inference: p\_value>0.05: we fail to reject the null hypothesis,i.e. bmi of women is not dependent on number of kids they have.