```
pip install pyforest
```

```
Collecting pyforest
  Downloading pyforest-1.1.0.tar.gz (15 kB)
  Preparing metadata (setup.py) ... done
Building wheels for collected packages: pyforest
  Building wheel for pyforest (setup.py) ... done
  Created wheel for pyforest: filename=pyforest-1.1.0-py2.py3-none-any.whl size=14605 sha256=9b43e12f5de15b6a240e94b67b37cedc24accfbc74f
  Stored in directory: /root/.cache/pip/wheels/9e/7d/2c/5d2f5e62de376c386fd3bf5a8e5bd119ace6a9f48f49df6017
Successfully built pyforest
Installing collected packages: pyforest
Successfully installed pyforest-1.1.0
```

```
import pyforest
# use to import all lib at ones
```

reading data set with help of pandas library

from google.colab import drive drive.mount('/content/drive')

mounting drive to acess g drive files

```
df1 = pd.read_csv("/content/drive/MyDrive/Colab Notebooks/insurance.csv") #path
df = df1.copy() #df is variabel defining dataset
```

```
df
```

	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
...	...	...	...	...	...	...	...
1333	50	male	30.970	3	no	northwest	10600.54830
1334	18	female	31.920	0	no	northeast	2205.98080
1335	18	female	36.850	0	no	southeast	1629.83350
1336	21	female	25.800	0	no	southwest	2007.94500
1337	61	female	29.070	0	yes	northwest	29141.36030

1338 rows × 7 columns

```
list(df.columns),df.info() #no null value
```

```
<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
(['age', 'sex', 'bmi', 'children', 'smoker', 'region', 'charges'], None)
```

```
df.isnull().sum() #verified no null
```

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

```
df.describe().transpose()
```

	count	mean	std	min	25%	50%	75%
age	1338.0	39.207025	14.049960	18.0000	27.00000	39.000	51.000000
bmi	1338.0	30.663397	6.098187	15.9600	26.29625	30.400	34.693750
children	1338.0	1.094918	1.205493	0.0000	0.00000	1.000	2.000000
charges	1338.0	13270.422265	12110.011237	1121.8739	4740.28715	9382.033	16639.912515

```
df.describe(include="all").transpose()
```

	count	unique	top	freq	mean	std	min	25%
age	1338.0	NaN	NaN	NaN	39.207025	14.04996	18.0	27.0
sex	1338	2	male	676	NaN	NaN	NaN	NaN
bmi	1338.0	NaN	NaN	NaN	30.663397	6.098187	15.96	26.29625
children	1338.0	NaN	NaN	NaN	1.094918	1.205493	0.0	0.0
smoker	1338	2	no	1064	NaN	NaN	NaN	NaN
region	1338	4	southeast	364	NaN	NaN	NaN	NaN
charges	1338.0	NaN	NaN	NaN	13270.422265	12110.011237	1121.8739	4740.28715

```
##age - numerical/continous  col -in range 46
##sex - catogorical nomial  col - male ,female
##bmi -  body mass index (a measure of body fat based on height and weight)
#a person's weight in kilograms (or pounds) divided by the square of height in meters (or feet).
#A high BMI can indicate high body fatness.
#BMI screens for weight categories that may lead to health problems,
#but it does not diagnose the body fatness or health of an individual.

##children - number of children/dependents covered by the insurance
##smoker - individual is a smoker yes or no
##region - region in which the individual is located.
##charges - medical insurance cost for the individual. contionus column
```

```
#insurence calulated on many cahrecter
#health if low then high cahrges,good health moderate charges, best then low cahrges
#bmi , smoker, age,childrens covred,sex not much but impact,region as (cold,hot,grenary),income of person,etc
```

Age Distribution: The age of individuals in the dataset is distributed across various ranges. There is a concentration of younger individuals, but there is a wide age range.

BMI Distribution: BMI (Body Mass Index) shows a spread of values with some concentration in the center. It appears to be normally distributed.

Children: Most individuals do not have children, but there is a range of values, including some with multiple children.

Charges Distribution: The distribution of insurance charges is positively skewed, indicating that a few individuals have significantly higher charges.

```
**♉**
```

age column

```
df.age.value_counts()
range ,value_counts = df.age.max()-df.age.min() ,df.age.value_counts()
print( "in range :",range )
print(f'max is : ',(df.age.max())," low is : ",(df.age.min()),"\nand count is:\n",(df.age.value_counts()))
#thier is 69 and 68 count of 18 and 19 in range of 46.
#at least 22 records are repaeat and max 69 records.
#data have outlier.
```

```
in range : 46
max is :  64  low is :  18
and count is:
 18    69
19    68
50    29
51    29
47    29
46    29
45    29
20    29
48    29
52    29
22    28
49    28
54    28
53    28
21    28
26    28
24    28
25    28
28    28
27    28
23    28
43    27
29    27
30    27
41    27
42    27
44    27
31    27
40    27
32    26
33    26
56    26
34    26
55    26
57    26
37    25
59    25
58    25
36    25
38    25
35    25
39    25
61    23
60    23
63    23
62    23
64    22
Name: age, dtype: int64
```

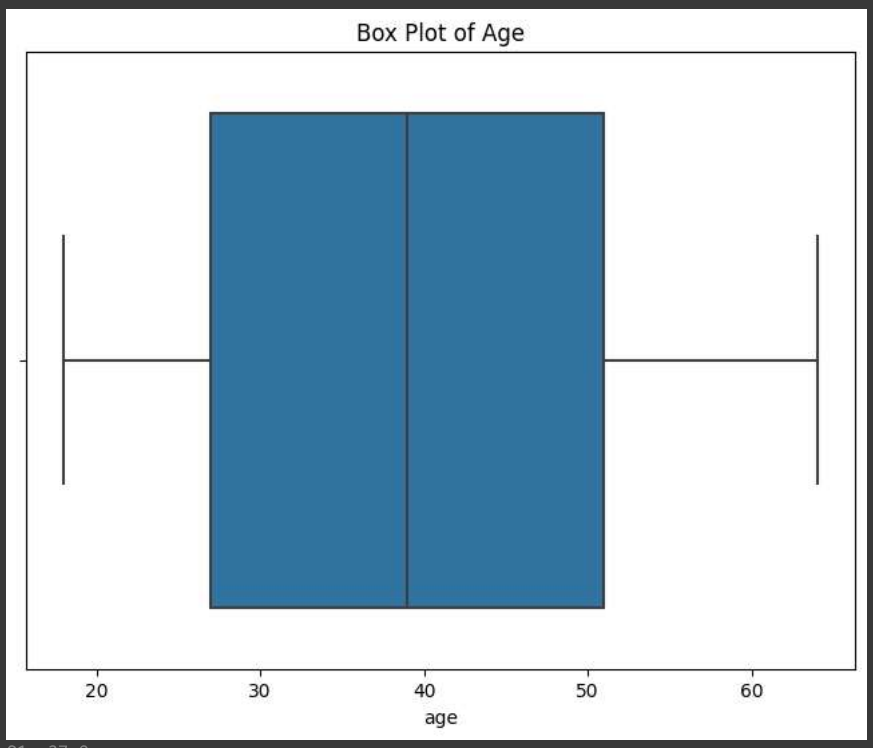
```
outlier = []
def outlier_detection(age):
  thr = 3
  std  = np.std(age)
  mean = np.mean(age)

  for x in age:
    z_score = (x - mean)/std
    if np.abs(z_score) > thr:
      outlier.append(x)
  return outlier
```

```

```

```
outlier_detection(df.age)
```

```
[]
```

```
q1 = df['age'].quantile(0.25)
q3 = df['age'].quantile(0.75)
iqr = q3 - q1
upper = q3 + 1.5 * iqr
lower = q1 - 1.5 * iqr
plt.figure(figsize=(8, 6))
sns.boxplot(x=df['age'])
plt.title('Box Plot of Age')
plt.show()
print(f'Q1: {q1}')
print(f'Q3: {q3}')
print(f'IQR: {iqr}')
print(f'Upper Bound: {upper}')
print(f'Lower Bound: {lower}')
outliers = df[(df['age'] < lower) | (df['age'] > upper)]
print(outliers)
```



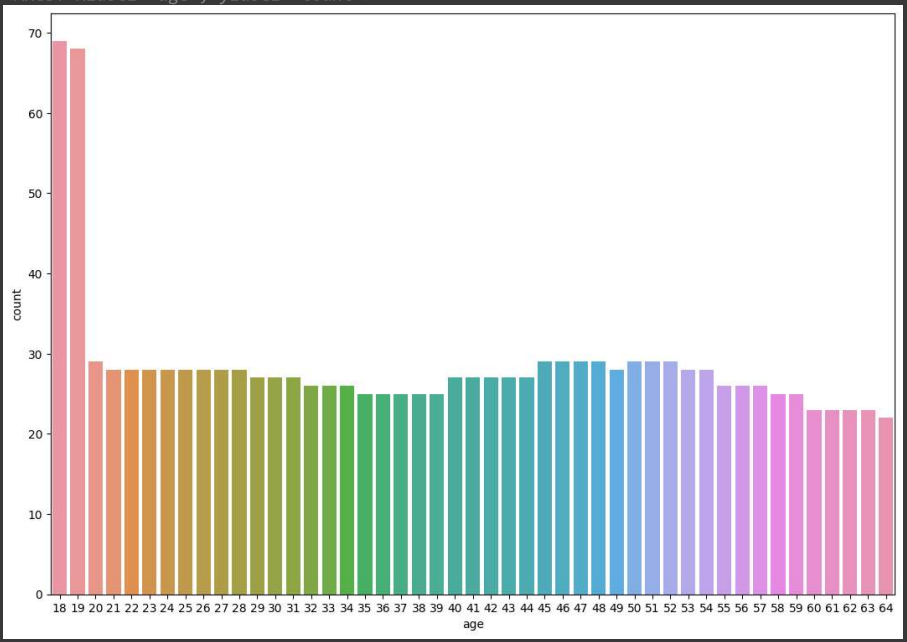

```
Q1: 27.0
Q3: 51.0
IQR: 24.0
Upper Bound: 87.0
Lower Bound: -9.0
Empty DataFrame
Columns: [age, sex, bmi, children, smoker, region, charges]
Index: []
```

```
df.age.max()-df.age.min()
```

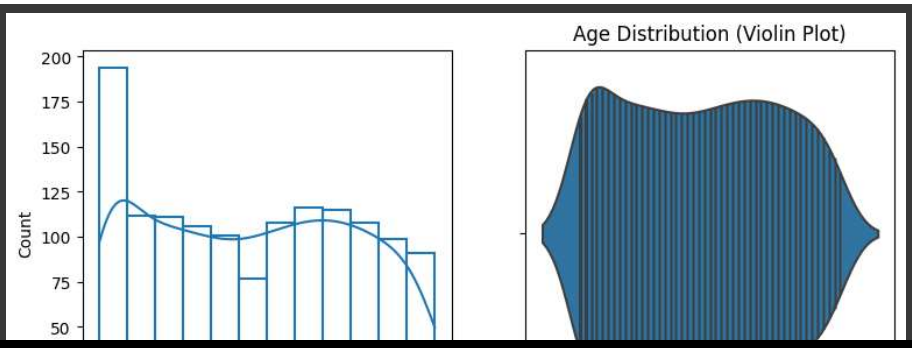
```
46
```

```
#countplot from seaborn for undestand  a auick by visvalize
#sns.histplot(data=df,x=df["age"])
plt.figure(figsize=(13,9))
sns.countplot(data=df,x=df["age"])
```

```
<Axes: xlabel='age', ylabel='count'>
```




```
plt.figure(figsize=(9,9))
plt.subplot(2, 2, 1)
sns.histplot(df.age,kde=True,fill=False)
plt.subplot(2, 2, 2)
sns.violinplot(x=df['age'], inner='stick')
plt.xlabel('Age')
plt.title('Age Distribution (Violin Plot)')
plt.show()
plt.figure(figsize=(8,6))
plt.subplot(2, 2, 1)
sns.boxplot(x=df['age'])
plt.show()
#uniform some what uniformally distributed box cox we can use to normalize
#no outlier detect
```

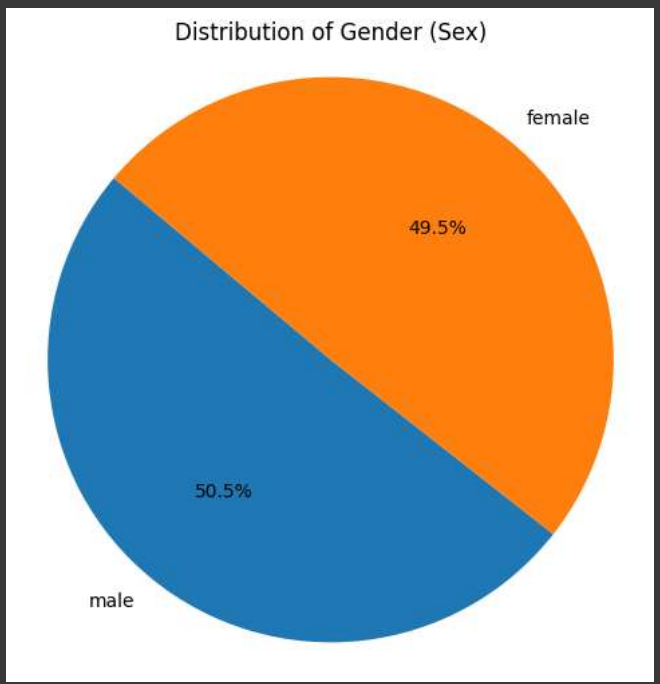



```
outlier = []
def outlier_detection(age):
  thr  = 3
  std  = np.std(df.age)
  mean = np.mean(df.age)

  for x in age:
    z_score = (x - mean)/std
    if np.abs(z_score) > thr:
      outlier.append(x)
  return outlier
outlier_detection(df.age)
#no outlier
```

```
[]
```

sex columns

```
sex_counts = df['sex'].value_counts()
plt.figure(figsize=(6, 6))
plt.pie(sex_counts.values, labels=sex_counts.index, autopct='%1.1f%%', startangle=140)
plt.title('Distribution of Gender (Sex)')
plt.axis('equal')
plt.show()
df.sex.value_counts()
#male domination by 1%  not much that impacted
```



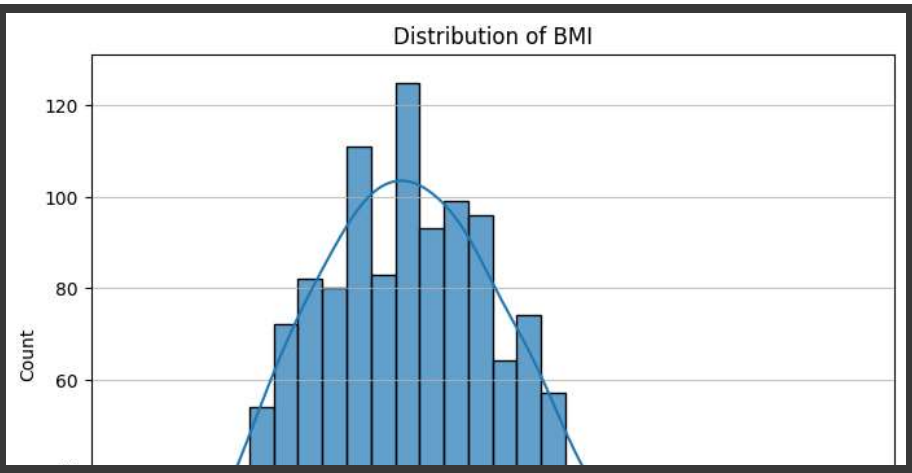

```
male      676
female    662
Name: sex, dtype: int64
```

BMI BODY MASS INDEX

```
df['bmi'].value_counts().unique(),df['bmi'].value_counts(),df.bmi.values
# Below 18 ,it's fall under underweight range.
#we can see unique values are underweight.
#18.5 to 24.9, it falls within the Healthy Weight range.
#If your BMI is 25.0 to 29.9, it falls within the overweight range.
#If your BMI is 30.0 or higher, it falls within the obese range.
```

```
(array([13,  9,  8,  7,  6,  5,  4,  3,  2,  1]),
 32.300    13
 28.310     9
 30.495     8
 30.875     8
 31.350     8
           ..
 46.200     1
 23.800     1
 44.770     1
 32.120     1
 30.970     1
 Name: bmi, Length: 548, dtype: int64,
 array([27.9 , 33.77, 33.  , ..., 36.85, 25.8 , 29.07]))
```

```
plt.figure(figsize=(8, 6))
sns.histplot(df['bmi'], bins=30, edgecolor='k', alpha=0.7,kde=True)
plt.title('Distribution of BMI')
plt.grid(axis='y', alpha=0.75)
plt.show()
plt.figure(figsize=(10, 6))
sns.kdeplot(data=df,x=df['bmi'])
#the curve show normal/berounlis ditribution
```



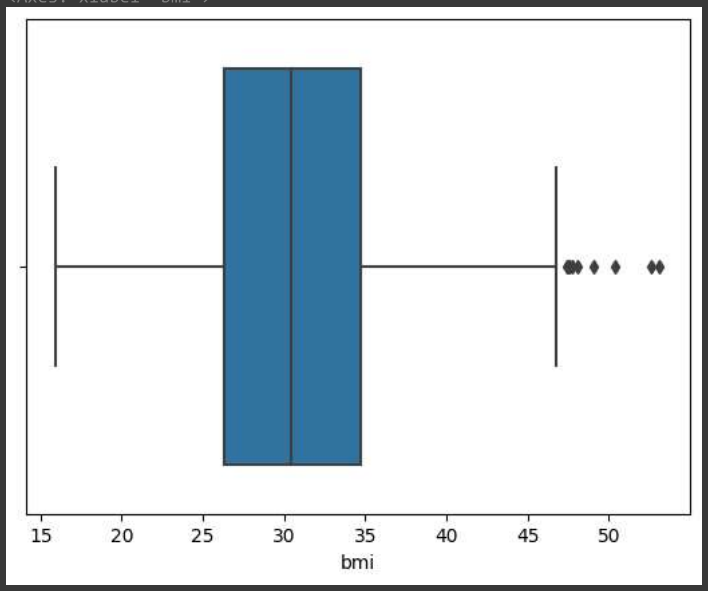
bmi_data = df['bmi'] mu = bmi_data.mean() sigma = bmi_data.std() x = np.linspace(mu - 3 * sigma, mu + 3 * sigma, 1000)

from scipy.stats import norm pdf_values = norm.pdf(x, loc=mu, scale=sigma)

plt.plot(x, pdf_values, label='Normal Distribution') plt.xlabel('X-axis') plt.ylabel('PDF Value') plt.title('Normal Distribution Curve') plt.legend() plt.grid(True) plt.show() print(mu,sigma)


```
sns.boxplot(data =df,x=df.bmi)
```

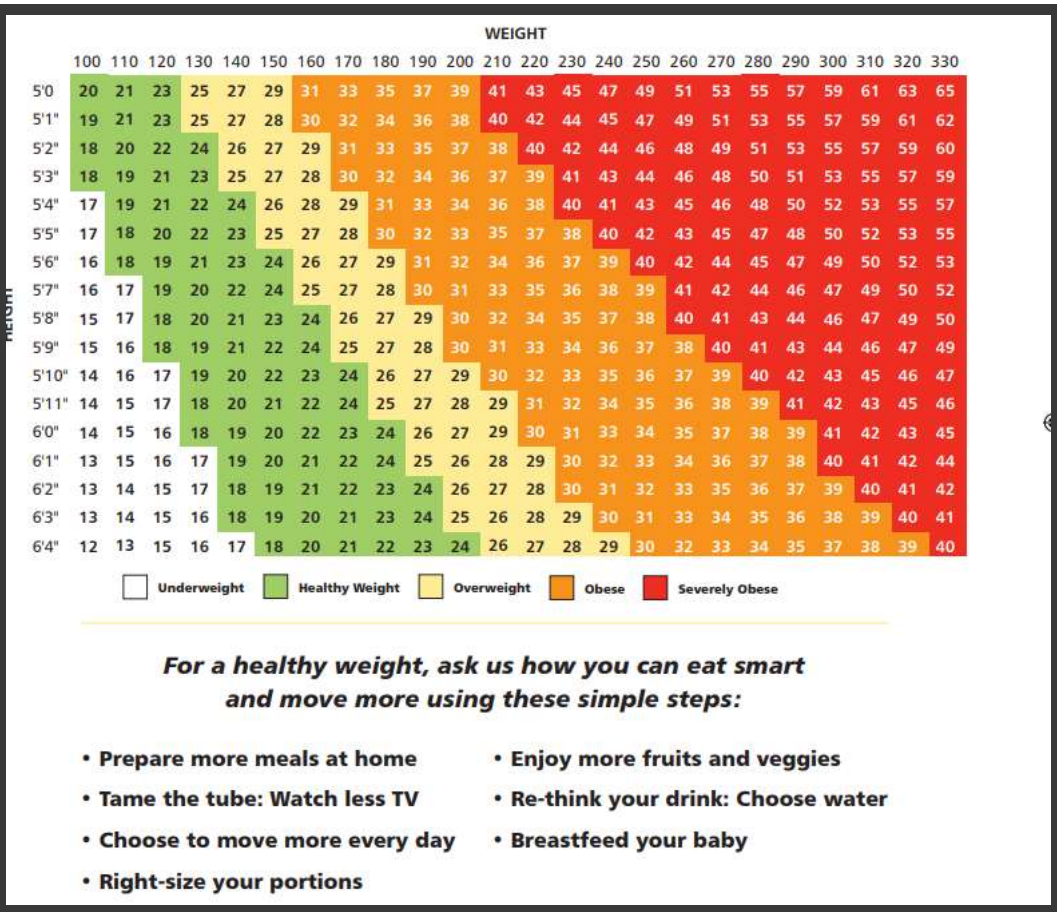
```
<Axes: xlabel='bmi'>
```




```
#as per math it is outlier but bmi index can be over 47 so with domain expertice i am not removing that
```

```
print(f'maxbmi is : ',(df.bmi.max()),"  low is : ",(df.bmi.min()),"\nand count is:\n",(df.bmi.value_counts()))
```

```
maxbmi is :  53.13  low is :  15.96
and count is:
 32.300    13
28.310     9
30.495     8
30.875     8
31.350     8
          ..
46.200     1
23.800     1
44.770     1
32.120     1
30.970     1
Name: bmi, Length: 548, dtype: int64
```




```
bins = [0, 18, 24.9, 29.9, float('inf')]
labels = ['Underweight', 'Healthy Weight', 'Overweight', 'Obese']
df['BMI Category'] = pd.cut(df['bmi'], bins=bins, labels=labels, right=False)
print(df[['bmi', 'BMI Category']])
```

```
          bmi    BMI Category
0      27.900      Overweight
1      33.770           Obese
2      33.000           Obese
3      22.705  Healthy Weight
4      28.880      Overweight
...       ...             ...
1333   30.970           Obese
1334   31.920           Obese
1335   36.850           Obese
1336   25.800      Overweight
1337   29.070      Overweight

[1338 rows x 2 columns]
```

```
df["bmi"].max()-df["bmi"].min()
```

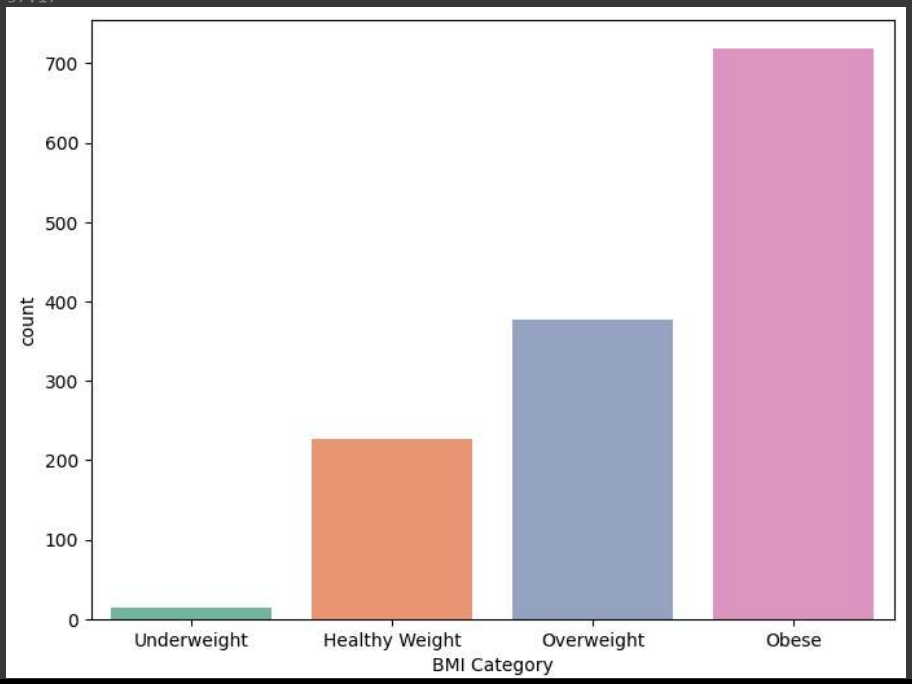
```
37.17
```

```
df["BMI Category"].value_counts()
```

```
Obese              719
Overweight         377
Healthy Weight     227
Underweight         15
Name: BMI Category, dtype: int64
```

```
plt.figure(figsize=(8, 6))
sns.countplot(data=df, x="BMI Category", palette="Set2")
df.bmi.max()-df.bmi.min()
```

37.17

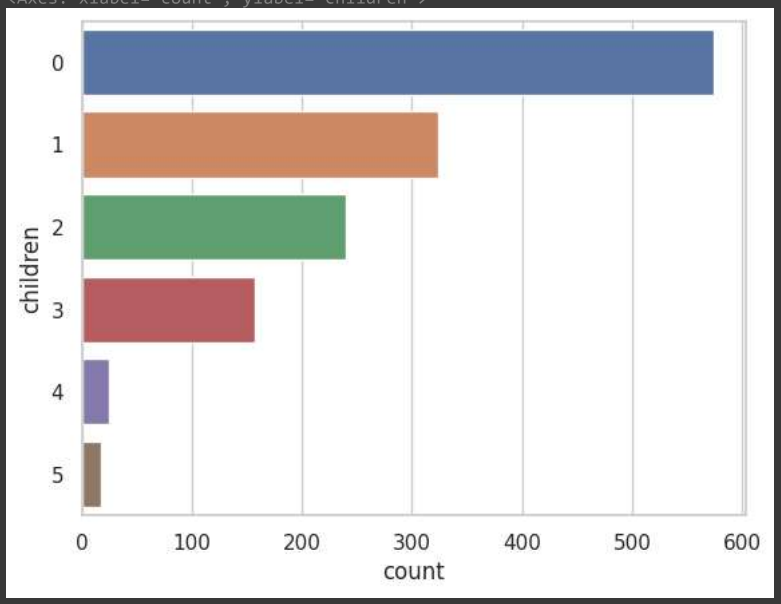



```
#to get % percenatge
sns.set(style="whitegrid")
percentage_data = df["BMI Category"].value_counts(normalize=True) * 100
plt.figure(figsize=(9, 7))
sns.barplot(x=percentage_data.index, y=percentage_data.values, palette="Set2")
plt.xticks(rotation=45)
plt.show()
print(df["BMI Category"].value_counts())
#observation - count of 'obese' ppl are higher than  'Underweight', 'Healthy Weight', 'Overweight'
# in range of 31.17
```

```
#Childrens coverd in insurence
```

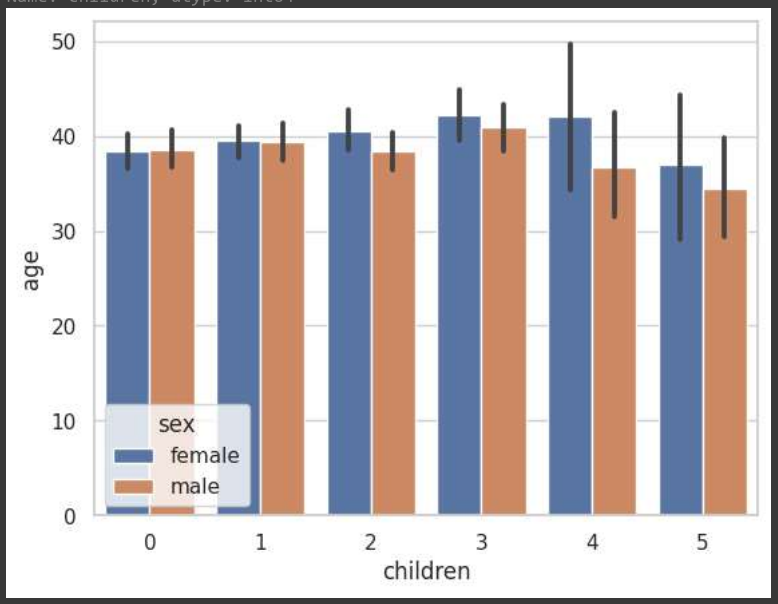
```
sns.countplot(data=df,y="children")#hue="sex"
#it's give how many childrens coverd in insurence policy
```

```
<Axes: xlabel='count', ylabel='children'>
```




```
sns.barplot(data=df,x="children",y='age',hue="sex")
df.children.value_counts()
```

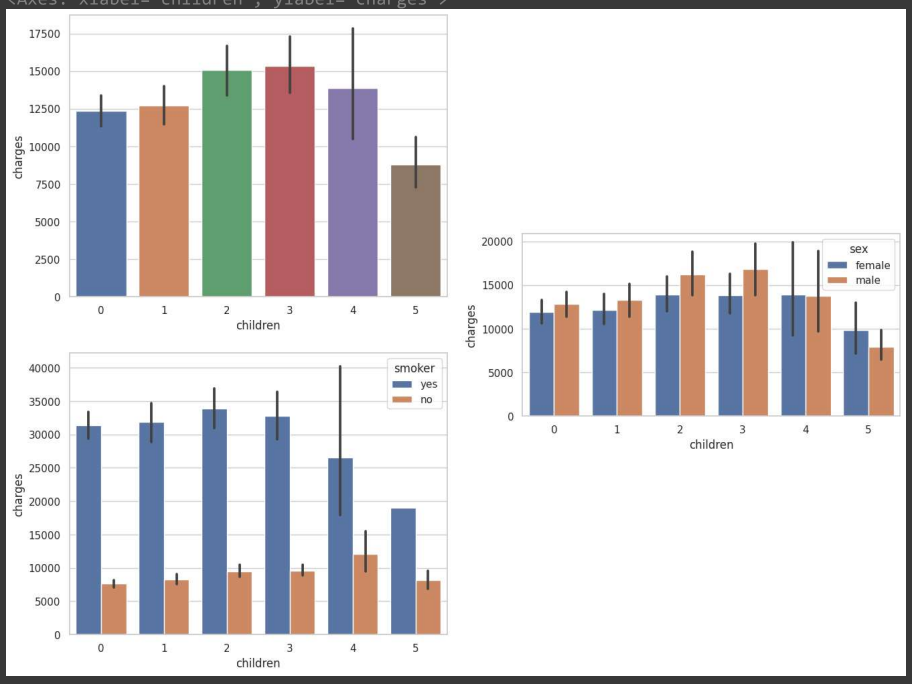
```
0    574
1    324
2    240
3    157
4     25
5     18
Name: children, dtype: int64
```




```
plt.figure(figsize=(16, 12))
plt.subplot(2,2,1)
sns.barplot(data=df,x="children",y="charges")
#for insurence poilcy  where the childrens are 2 3 and 4 have
```

```
# high insiurence cahrge wrt to other

plt.subplot(2,2,3)
sns.barplot(data=df,x="children",y='charges',hue="smoker")
plt.subplot(3,2,4)
sns.barplot(data=df,x="children",y='charges',hue="sex")
#error bar
```

```
<Axes: xlabel='children', ylabel='charges'>
```



smoker

```
df.bmi.values
```

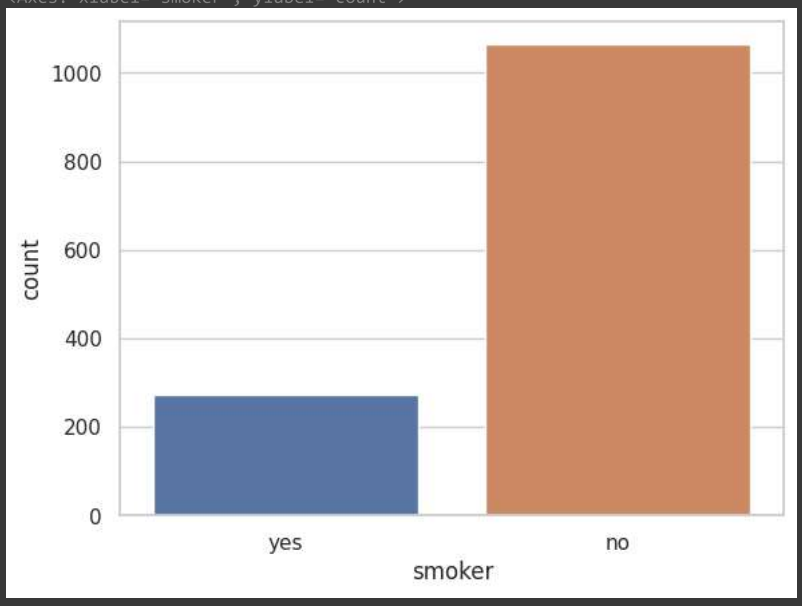
```
array([27.9 , 33.77, 33.  , ..., 36.85, 25.8 , 29.07])
```

```
df.smoker.value_counts()
```

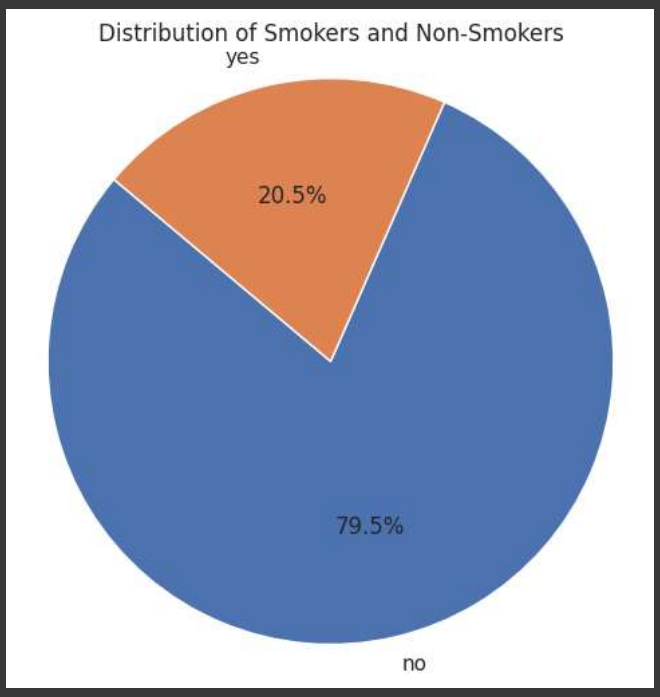
```
no     1064
yes     274
Name: smoker, dtype: int64
```

```
sns.countplot(x=df["smoker"])
#smoker are less so that will profit of an  insurence comapny they have less chance to get to pople claims their insurnce
#and comapny  will make profit .
#beacuse  due to smoking they will not grt any disese.
```

```
<Axes: xlabel='smoker', ylabel='count'>
```




```
smoker_counts = df['smoker'].value_counts()
plt.figure(figsize=(6, 6))
plt.pie(smoker_counts, labels=smoker_counts.index, autopct='%1.1f%%', startangle=140)
plt.title('Distribution of Smokers and Non-Smokers')
plt.axis('equal')
plt.show()
# for % distribtion
```

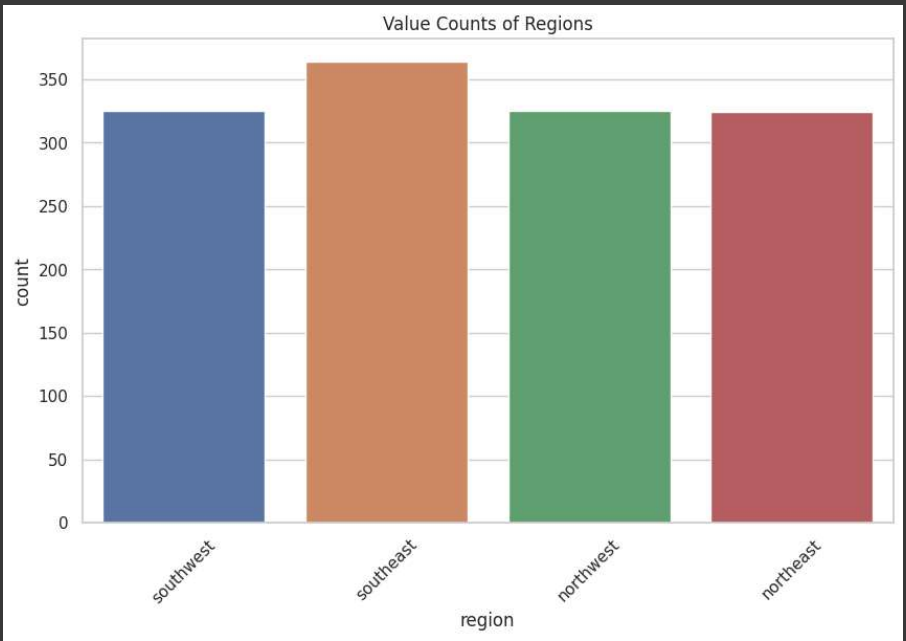


region

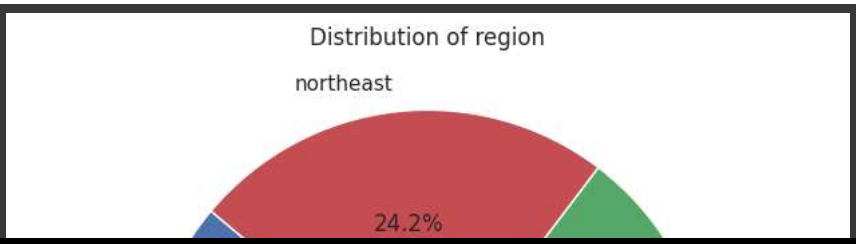
```
df.region.value_counts()
```

```
southeast    364
southwest    325
northwest    325
northeast    324
Name: region, dtype: int64
```

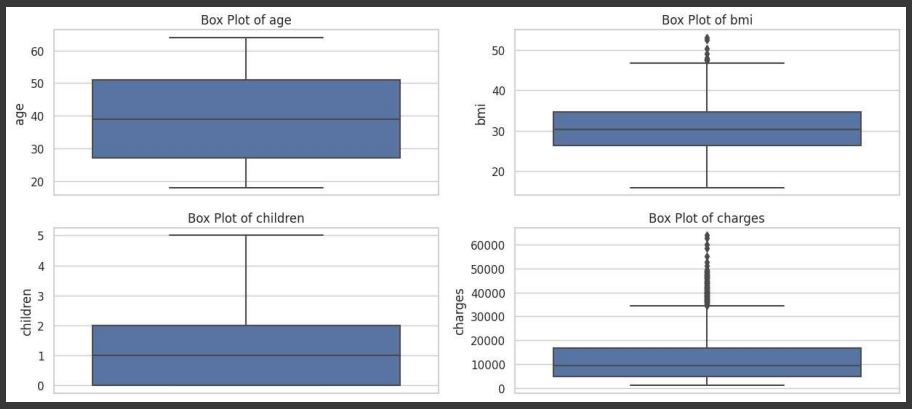
```
sns.set(style="whitegrid")
plt.figure(figsize=(10, 6))
sns.countplot(x=df.region)
plt.title('Value Counts of Regions')
plt.xticks(rotation=45)
plt.show()

#southeast ppl are more comapritvelly to other
```




```
smoker_counts = df['region'].value_counts()
plt.figure(figsize=(6, 6))
plt.pie(smoker_counts, labels=smoker_counts.index, autopct='%1.1f%%', startangle=140)
plt.title('Distribution of region\n ')
plt.axis('equal')
plt.show()
```



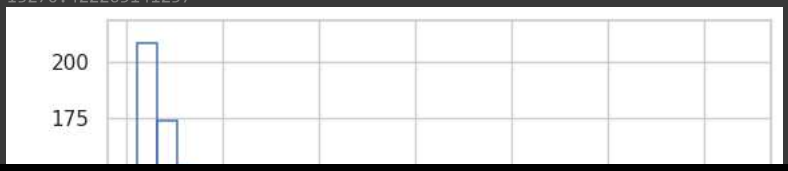

```
numeric_columns = df.select_dtypes(include=['number'])
plt.figure(figsize=(15, 10))
for i, column in enumerate(numeric_columns.columns):
    plt.subplot(3, 2, i + 1)
    sns.boxplot(data=df, y=column)
    plt.title(f'Box Plot of {column}')
#plt.tight_layout()
plt.show()
```



charges

```
sns.histplot(x=df.charges,fill=False,kde=True)
 #data is leptokurtica and right skwed box cox wil help to normalize
 #it shows more number of ppl have charges under near to 13000~
df.charges.mean()
```

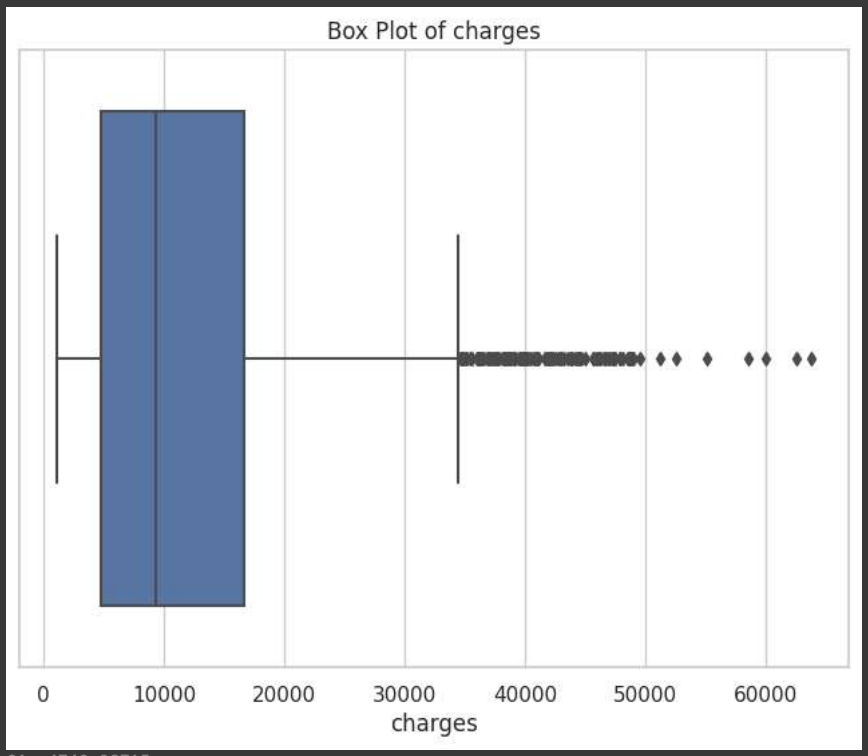
```
13270.422265141257
```




```
df.charges.max(),df.charges.min(),df.charges.max()-df.charges.min()
```

```
(63770.42801, 1121.8739, 62648.554110000005)
```

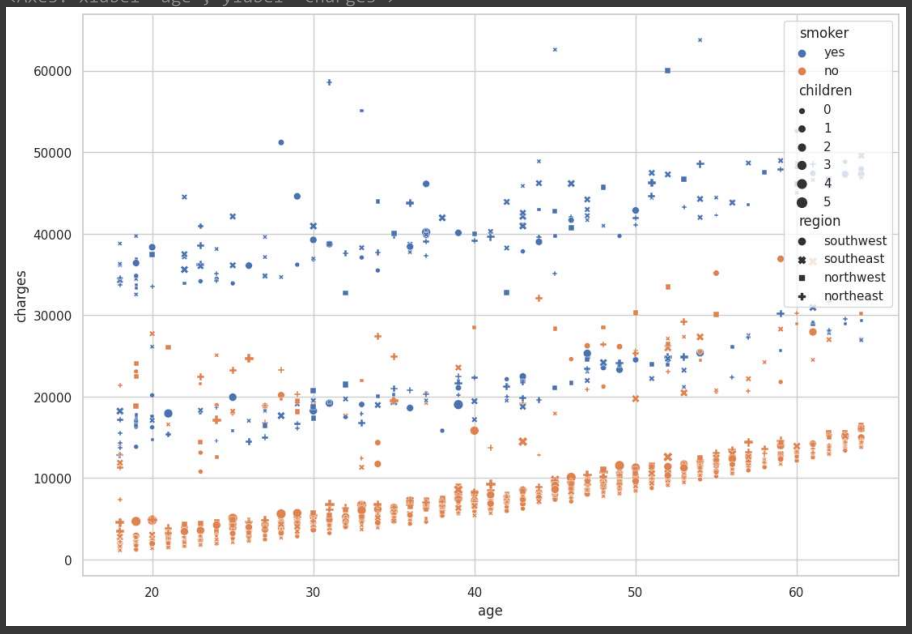
```
q1 = df['charges'].quantile(0.25)
q3 = df['charges'].quantile(0.75)
iqr = q3 - q1
upper = q3 + 1.5 * iqr
lower = q1 - 1.5 * iqr
plt.figure(figsize=(8, 6))
sns.boxplot(x=df['charges'])
plt.title('Box Plot of charges')
plt.show()
print(f'Q1: {q1}')
print(f'Q3: {q3}')
print(f'IQR: {iqr}')
print(f'Upper Bound: {upper}')
print(f'Lower Bound: {lower}')
outliers = df[(df['charges'] < lower) | (df['charges'] > upper)]
print(outliers.value_counts().sum())
# look like we found outlier
```




```
Q1: 4740.28715
Q3: 16639.912515
IQR: 11899.625365
Upper Bound: 34489.350562499996
Lower Bound: -13109.1508975
139
```

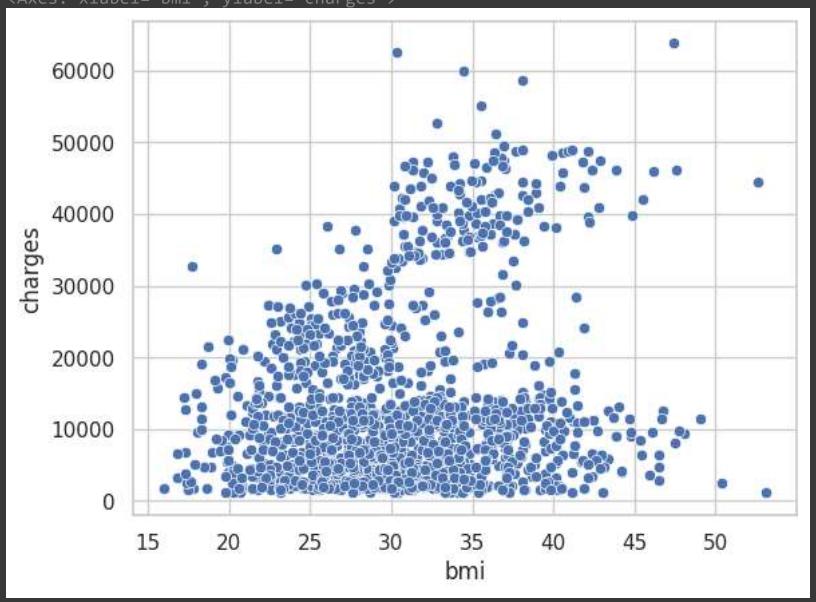
```
plt.figure(figsize = (13,9))
sns.scatterplot(x=df.age,y=df.charges,hue = df.smoker,size = df.children,style = df.region)
```

```
<Axes: xlabel='age', ylabel='charges'>
```



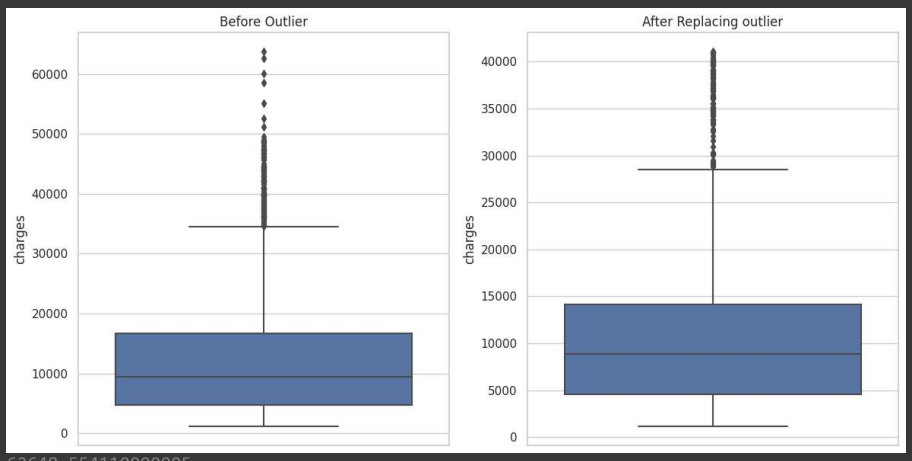

```
sns.scatterplot(x=df.bmi,y=df.charges)
```

```
<Axes: xlabel='bmi', ylabel='charges'>
```




```
upper_percentile = np.percentile(df['charges'], 95)
upper_bound = upper_percentile

filtered_data = df[(df['charges'] <= upper_bound)]
plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
sns.boxplot(data=df, y='charges')
plt.title('Before Outlier')
plt.subplot(1, 2, 2)
sns.boxplot(data=filtered_data, y='charges')
```

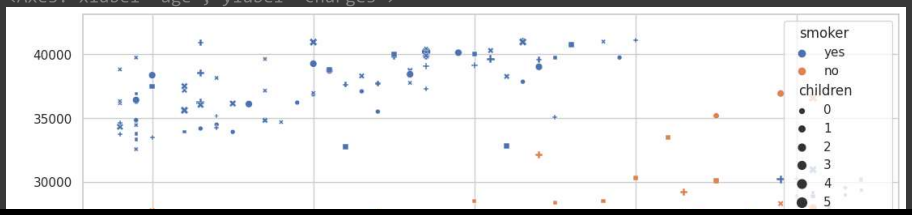
```
plt.title('After Replacing outlier')
plt.tight_layout()
plt.show()
print(df.charges.max()-df.charges.min())
df['charges'] = filtered_data['charges']
print(df.charges.max()-df.charges.min())
```




```
62648.554110000005
39975.28785
```

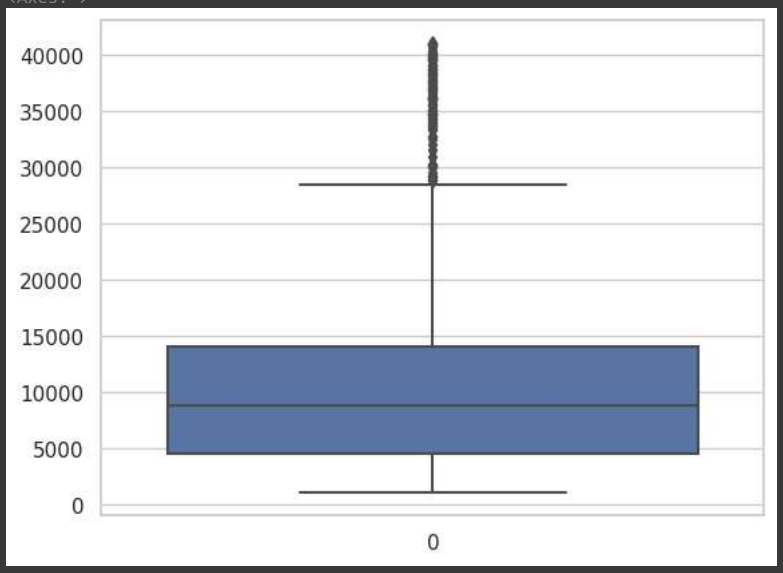
```
plt.figure(figsize = (13,9))
sns.scatterplot(x=df.age,y=df.charges,hue = df.smoker,size = df.children,style = df.region)
```

```
<Axes: xlabel='age', ylabel='charges'>
```

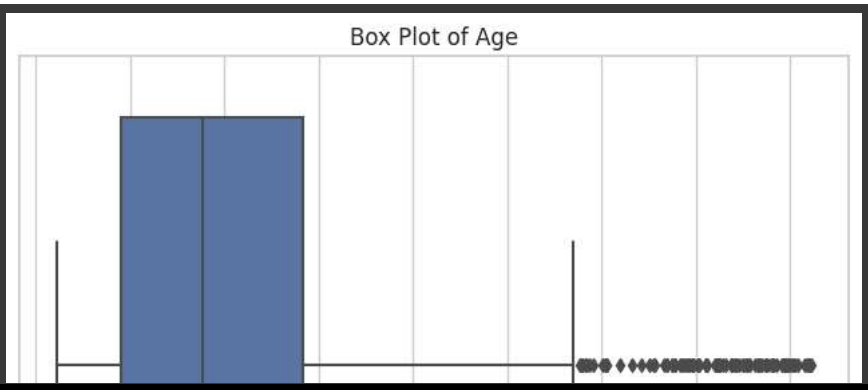



```
sns.boxplot(df.charges)
```

```
<Axes: >
```

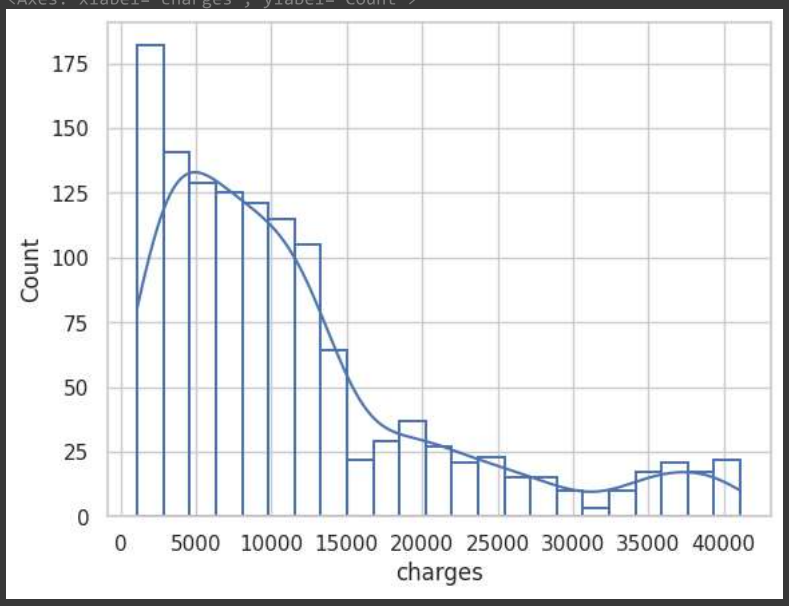



```
q1 = df['charges'].quantile(0.25)
q3 = df['charges'].quantile(0.75)
iqr = q3 - q1
upper = q3 + 1.5 * iqr
lower = q1 - 1.5 * iqr
plt.figure(figsize=(8, 6))
sns.boxplot(x=df['charges'])
plt.title('Box Plot of Age')
plt.show()
print(f'Q1: {q1}')
print(f'Q3: {q3}')
print(f'IQR: {iqr}')
print(f'Upper Bound: {upper}')
print(f'Lower Bound: {lower}')
outliers = df[(df['charges'] < lower) | (df['charges'] > upper)]
print(outliers.value_counts().sum())
# look like we found outlier
```




```
sns.histplot(x=df.charges,fill=False,kde=True)
```

```
<Axes: xlabel='charges', ylabel='Count'>
```

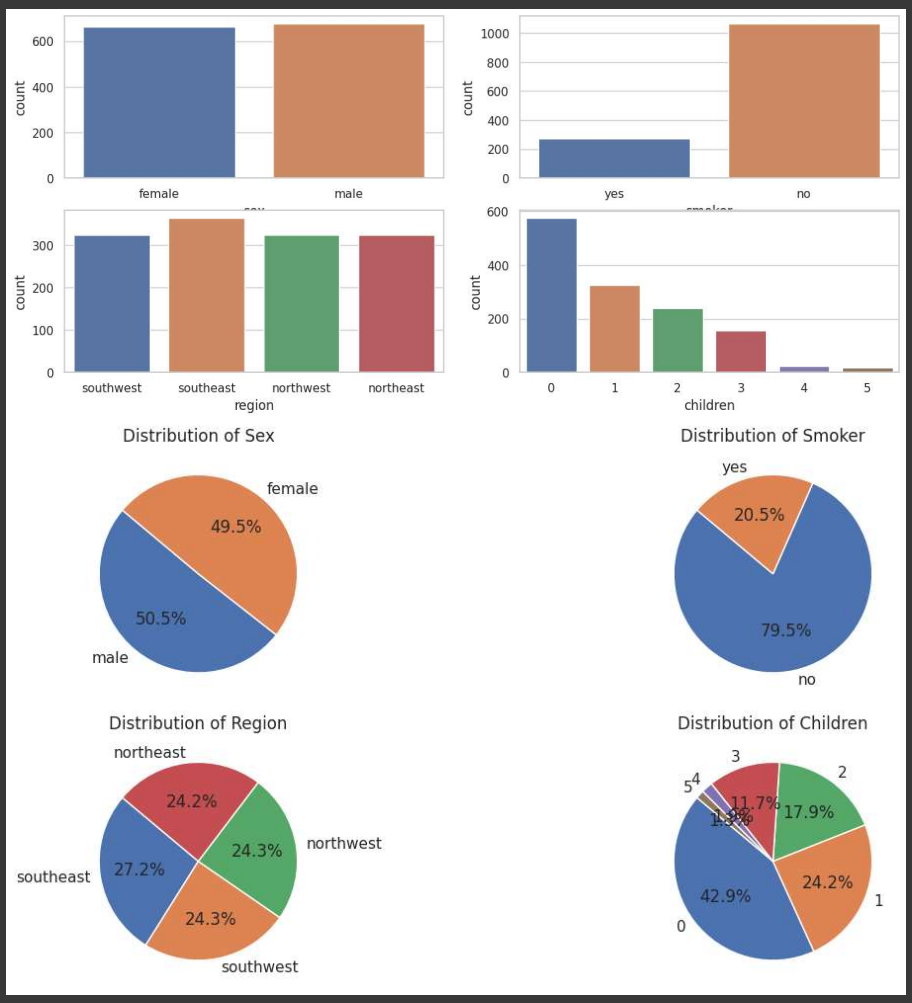


considering math for data science i am considering these as an oulier

in univarient anlaysis i see that their is an need to convert non -uniform data to uniform to feed data to to ml algo

```
plt.figure(figsize=(14, 6))
plt.subplot(2, 2, 1)
sns.countplot(data=df, x='sex')

plt.subplot(2, 2, 2)
sns.countplot(data=df, x='smoker')

plt.subplot(2, 2, 3)
sns.countplot(data=df, x='region')

plt.subplot(2, 2, 4)
sns.countplot(data=df, x='children')
plt.show()


fig, axes = plt.subplots(2, 2, figsize=(14, 6))
countplot_data = {
    'sex': df['sex'].value_counts(),
    'smoker': df['smoker'].value_counts(),
    'region': df['region'].value_counts(),
    'children': df['children'].value_counts()
}
# Convert countplots to pie charts
for (feature, data), ax in zip(countplot_data.items(), axes.ravel()):
```

```
    ax.pie(data, labels=data.index, autopct='%1.1f%%', startangle=140)
    ax.set_title(f'Distribution of {feature.capitalize()}')
plt.tight_layout()
plt.show()
```



duplicate_rows = df[df.duplicated(keep='first')] columns_to_check = ['age', 'sex', 'bmi', 'smoker', 'region'] duplicate_rows = df[df.duplicated(subset=columns_to_check, keep='first')] print("Duplicate Rows:") print(duplicate_rows) num_duplicates = duplicate_rows.shape[0] print(f"Number of Duplicate Rows: {num_duplicates}")

To remove duplicates based on specific columns

insurance_no_duplicates = df.drop_duplicates(subset=columns_to_check, keep='first')

if duplicate found frop andkeep frist

insurance_no_duplicates.info()

```
columns_to_check = ['age', 'sex', 'bmi', 'smoker', 'region']
df_no_duplicates = df.drop_duplicates(subset=columns_to_check, keep='first')
print(df_no_duplicates)
columns_to_check = ['age', 'sex', 'bmi', 'smoker', 'region']
df = df.drop_duplicates(subset=columns_to_check, keep='first')
```

```
        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
...     ...     ...     ...       ...    ...        ...         ...
1333     50    male  30.970         3     no  northwest  10600.54830
1334     18  female  31.920         0     no  northeast   2205.98080
1335     18  female  36.850         0     no  southeast   1629.83350
1336     21  female  25.800         0     no  southwest   2007.94500
1337     61  female  29.070         0    yes  northwest  29141.36030

          BMI Category
0         Overweight
1              Obese
2              Obese
3     Healthy Weight
4         Overweight
...               ...
1333           Obese
1334           Obese
1335           Obese
1336      Overweight
1337      Overweight

[1330 rows x 8 columns]
```

```
df
```

	age	sex	bmi	children	smoker	region	charges	BMI Category
0	19	female	27.900	0	yes	southwest	16884.92400	Overweight
1	18	male	33.770	1	no	southeast	1725.55230	Obese
2	28	male	33.000	3	no	southeast	4449.46200	Obese
3	33	male	22.705	0	no	northwest	21984.47061	Healthy Weight
4	32	male	28.880	0	no	northwest	3866.85520	Overweight
...	...	...	...	...	...	...	...	...
1333	50	male	30.970	3	no	northwest	10600.54830	Obese
1334	18	female	31.920	0	no	northeast	2205.98080	Obese
1335	18	female	36.850	0	no	southeast	1629.83350	Obese
1336	21	female	25.800	0	no	southwest	2007.94500	Overweight
1337	61	female	29.070	0	yes	northwest	29141.36030	Overweight

1330 rows × 8 columns

ට්***

lets's do multi- variet analysis to explore

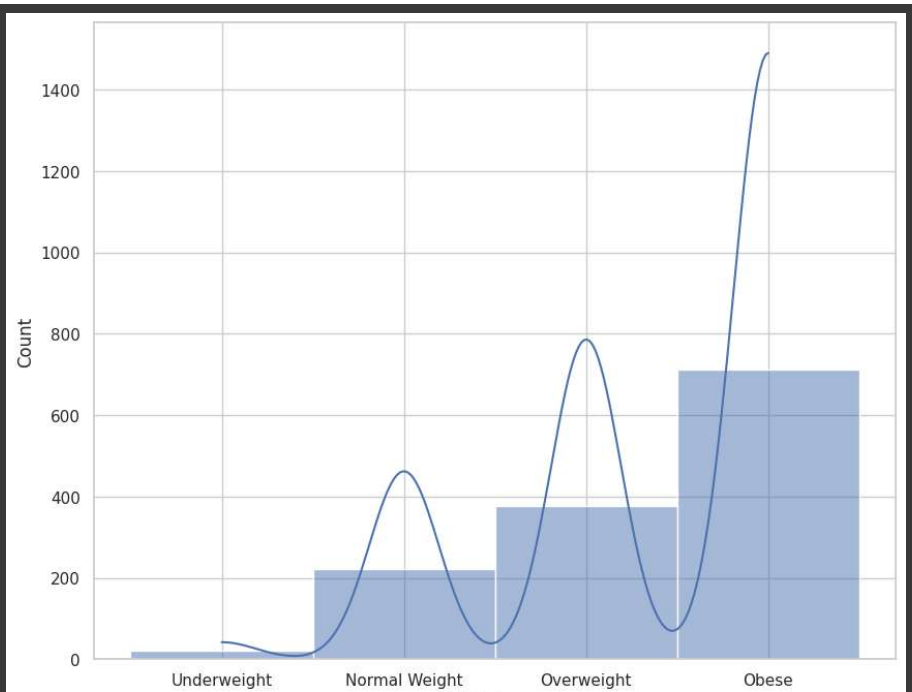
```
#body mass index (a measure of body fat based on height and weight)
#a person's weight in kilograms (or pounds) divided by the square of height in meters (or feet).
#A high BMI can indicate high body fatness.
```

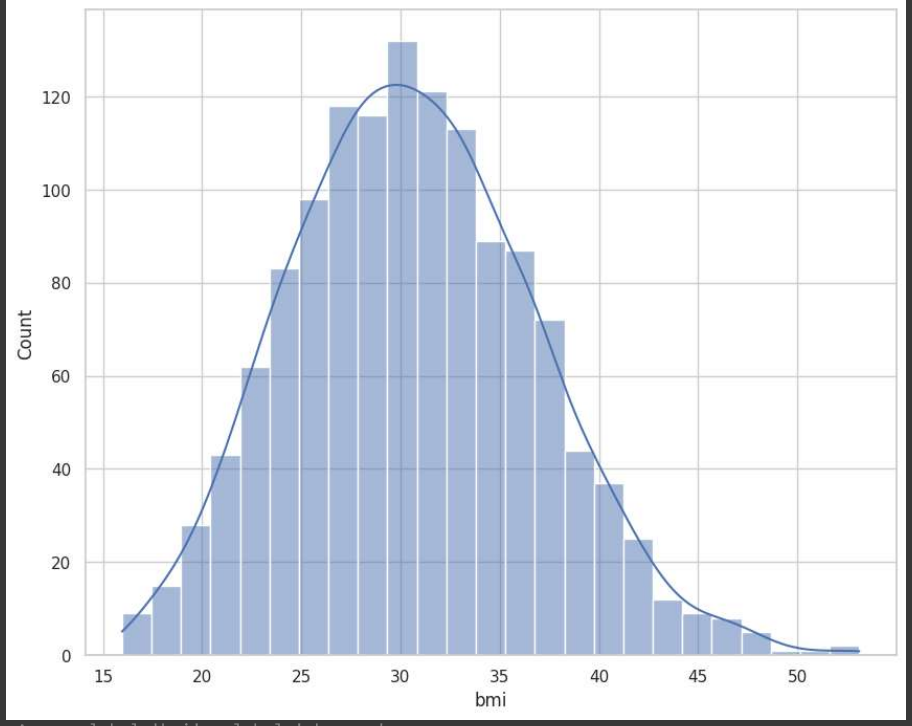
```
#BMI screens for weight categories that may lead to health problems,
#but it does not diagnose the body fatness or health of an individual.
#0, 18,underwight,,18- 24.9,healthy weight,24.9- 29.9,overwieght, above 29.9 obese
```

```
bmi_categories = ['Underweight', 'Normal Weight', 'Overweight', 'Obese']
bmi_ranges = [0, 18.5, 24.9, 29.9, float('inf')]
df['BMI Category'] = pd.cut(df['bmi'], bins=bmi_ranges, labels=bmi_categories, right=False)
```

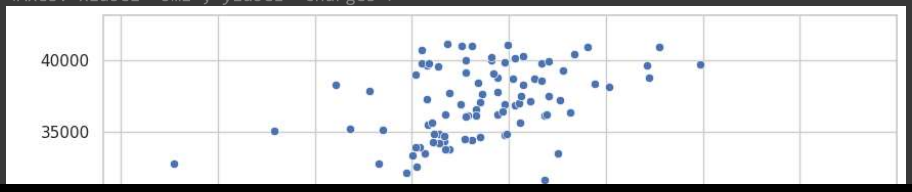
```
<ipython-input-56-25e8620380eb>:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
  df['BMI Category'] = pd.cut(df['bmi'], bins=bmi_ranges, labels=bmi_categories, right=F
```

```
plt.figure(figsize = (10,8))
sns.histplot(df["BMI Category"],kde=True)
plt.show()
plt.figure(figsize = (10,8))
sns.histplot(df["bmi"],kde=True)
plt.show()
plt.figure(figsize = (10,8))
sns.scatterplot(x=df.bmi,y=df.charges)
 #bmi index impacting as we see in catogory obese are more comparitivelly
#underweight and others.
#bmi count is normally ditributed.
```



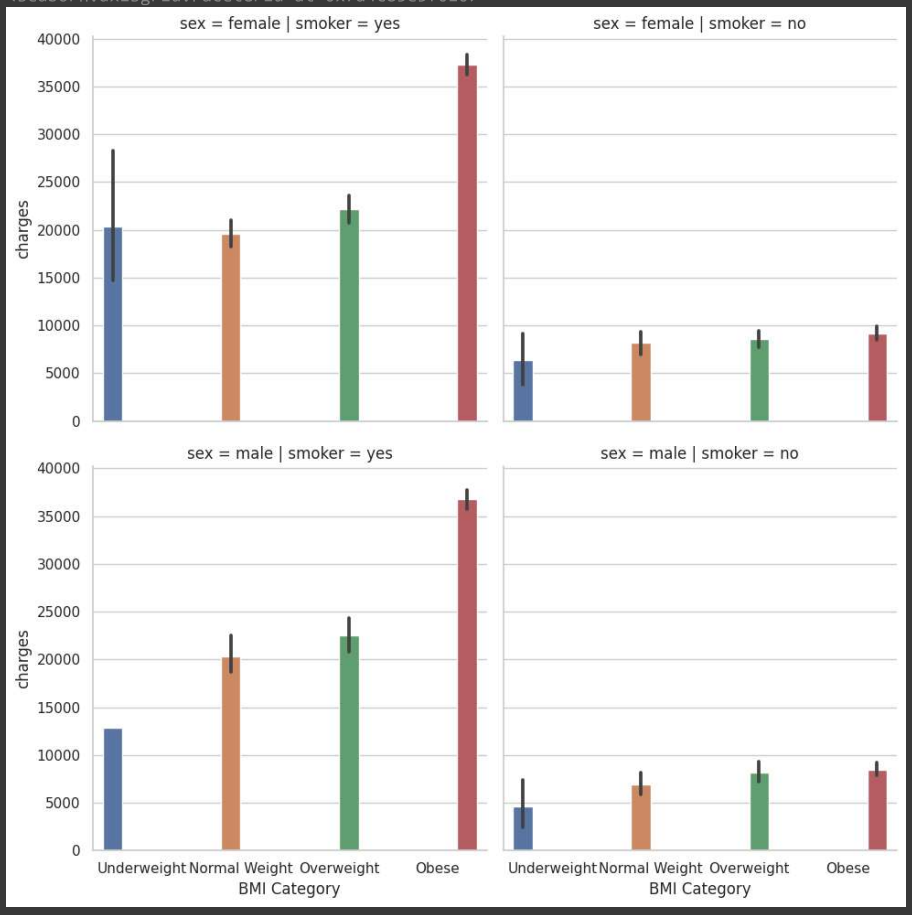



```
<Axes: xlabel='bmi', ylabel='charges'>
```



```
#smoker is lesss and male smoker is more compaaritivelly female
female_smoker_count = ((df["smoker"] == "yes") & (df["sex"] == "female")).sum()
male_smoker_count = ((df["smoker"] == "yes") & (df["sex"] == "male")).sum()
female_non_smoker_count = ((df["smoker"] == "no") & (df["sex"] == "female")).sum()
male_non_smoker_count = ((df["smoker"] == "no") & (df["sex"] == "male")).sum()
print("Female smokers count:", female_smoker_count)
print("Male smokers count:", male_smoker_count)
print("Female non-smokers count:", female_non_smoker_count)
print("Male nonsmokers count:", male_non_smoker_count)
sns.catplot(data=df,y="charges",x='BMI Category',hue="BMI Category",col="smoker",row="sex",kind='bar',orient="v")
#obese smoker are more comparitevilly others , where over all smoker with both gender are high.
```

```
Female smokers count: 115
Male smokers count: 159
Female non-smokers count: 542
Male nonsmokers count: 514
<seaborn.axisgrid.FacetGrid at 0x7a4c85e5f010>
```



g = sns.catplot(data=df,y="charges",x='BMI Category',hue="BMI Category",col="smoker",row="sex",kind='bar',orient="v", height=5, # Adjust the size of the plot as needed aspect=1.2 # Adjust the aspect ratio as needed)

g.map(sns.barplot, x=df["children"], order=[0, 1, 2, 3, 4, 5], hue_order=df['BMI Category'].unique()) g.set_titles("Smoker: {col_name} / Sex: {row_name}") g.set_xlabels("BMI Category") g.set_ylabels("Charges") plt.show()

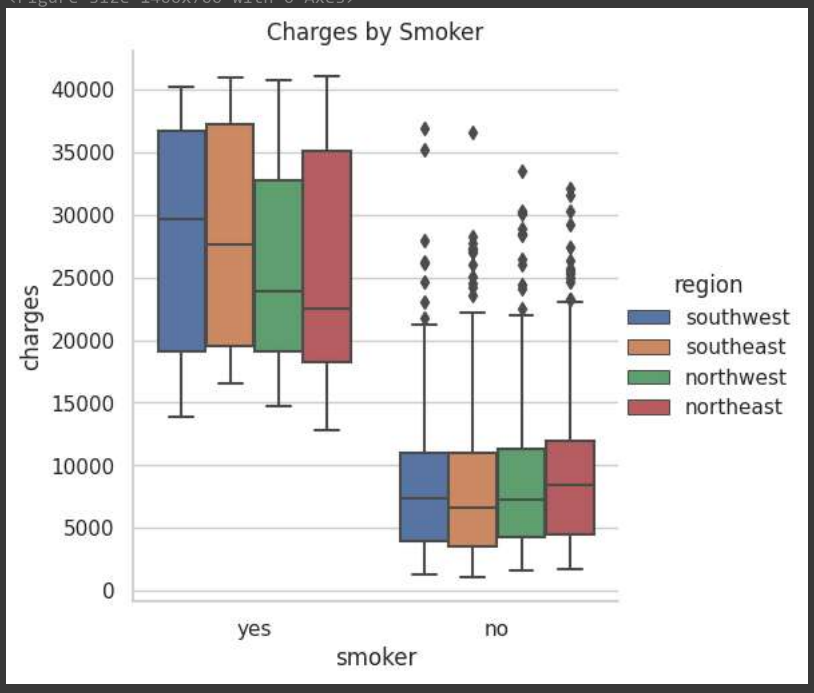
```
counts = df.groupby(['children', 'region', 'sex', 'smoker']).size().reset_index(name='count')
print("Counts for all combinations:")
print(counts)
plt.figure(figsize=(14, 6))
sns.catplot(data=counts, x='children', y='count', hue='region', col='sex', row='smoker',kind='bar')
plt.show()
# we can see how smoking habits differ between regions for both males and females with different numbers of children.
#It appears that there are more male smokers in most cases, but the difference may vary depending on the number of childre
#You can observe if there are any trends related to family size and smoking behavior.
```

```
Counts for all combinations:
    children     region     sex smoker  count
0          0  northeast  female     no     64
```

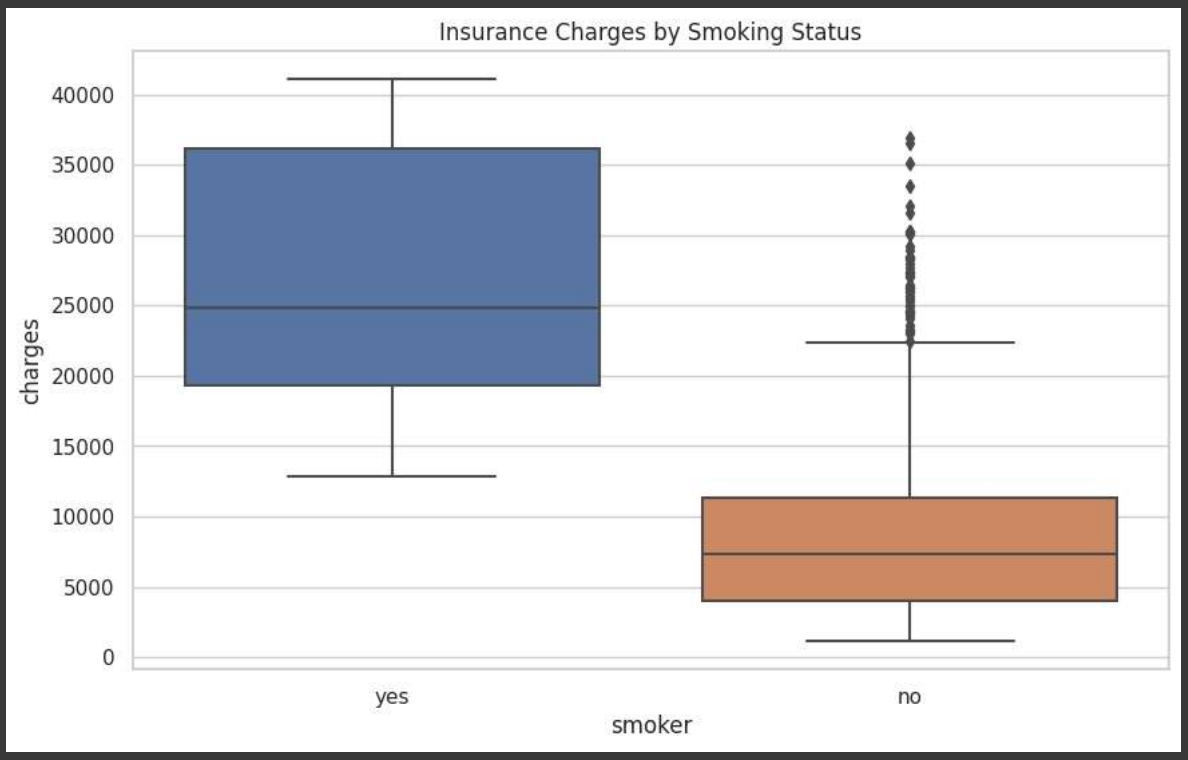
```
df['smoker_status'] = df['smoker'].apply(lambda x: 'smoker' if x == 'yes' else 'non-smoker')
plt.figure(figsize=(15,7))
sns.countplot(x='age', hue='smoker_status', data=df,orient="v")
plt.title('Count of Smokers and Non-Smokers by age')
plt.show()
sns.countplot(x='sex', hue='smoker_status', data=df,orient="v")
#if their is high number of non smoker then their is less chance
#the inscurnce claim get happen and insurence comapny is in profit
```

```
#lets see charges wrt smoker
plt.figure(figsize=(14,7))
sns.catplot(data=df, x='smoker', y='charges',kind="box",hue="region")
plt.title('Charges by Smoker')
plt.show()
#a s we can see the caharges of non smoker as we considering box plot
#their are some outlier llets see ehat beacuse of they from age or childrans
```

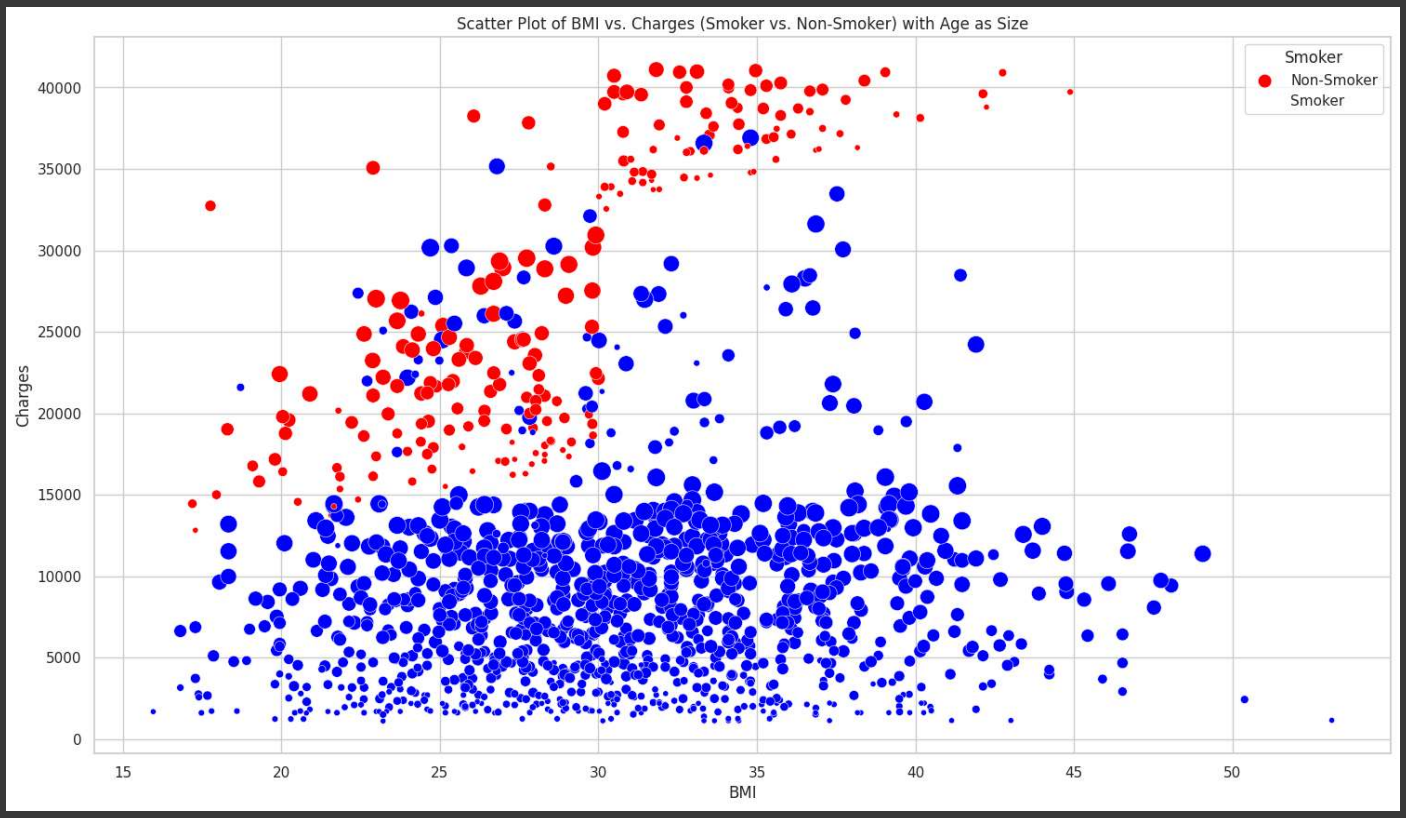
```
<Figure size 1400x700 with 0 Axes>
```




```
plt.figure(figsize=(10, 6))
sns.boxplot(data=df, x='smoker', y='charges')
plt.title('Insurance Charges by Smoking Status')
plt.show()
```

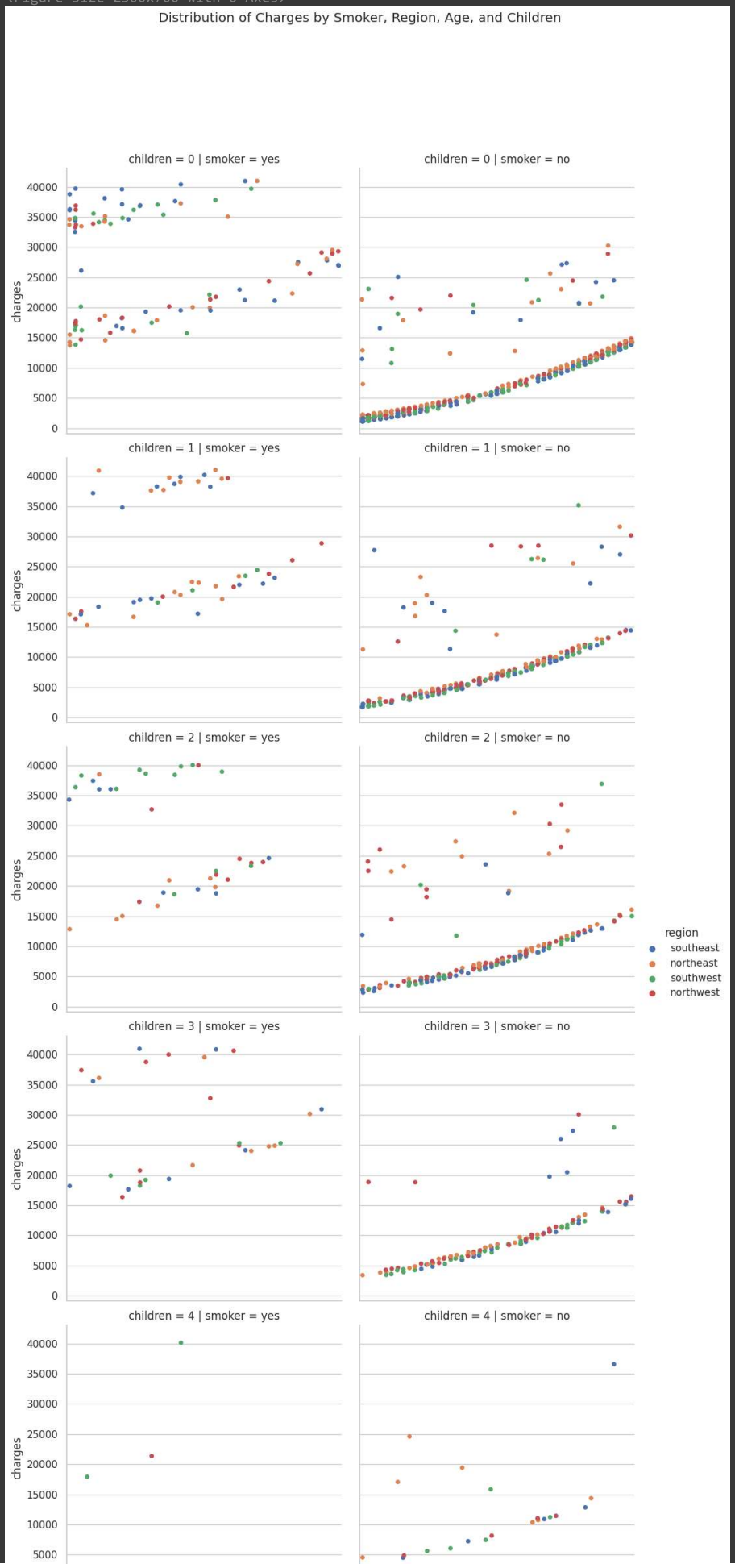


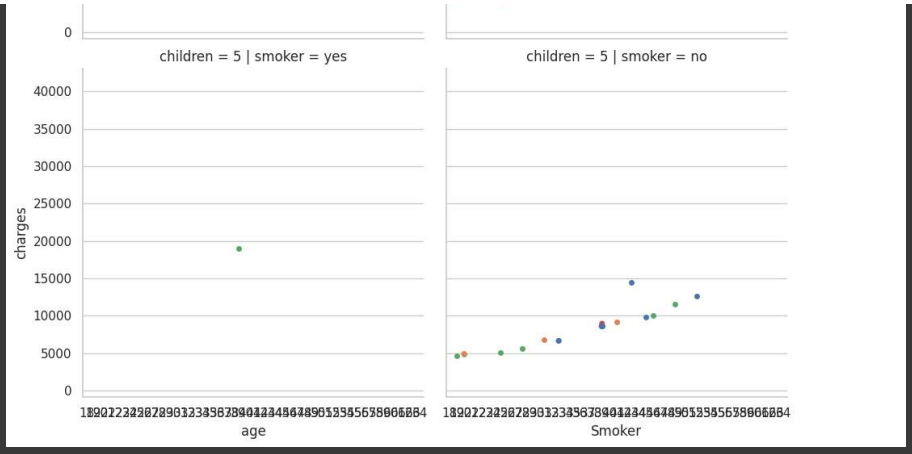
```
plt.figure(figsize=(18, 10))
sns.scatterplot(data=df, x='bmi', y='charges', hue='smoker', palette={'no': 'blue', 'yes': 'red'}, size='age', sizes=(20, 
plt.title('Scatter Plot of BMI vs. Charges (Smoker vs. Non-Smoker) with Age as Size')
plt.xlabel('BMI')
plt.ylabel('Charges')
plt.legend(title='Smoker', loc='upper right', labels=['Non-Smoker', 'Smoker'])
plt.show()
#proved bmi has paritial storng relatioship with charges
#beacuse their has an mixed charges of bmi indexes
```




```
plt.figure(figsize=(23,7))
sns.catplot(data=df, x='age', y='charges',  hue='region', col='smoker', row='children')
plt.xlabel('Smoker')
plt.ylabel('Charges')
plt.suptitle('Distribution of Charges by Smoker, Region, Age, and Children')
plt.subplots_adjust(top=0.9)
plt.show()

# i don't know why 0 childrens graph wasn't coming
# but it semms where  children  1 2 3 4  their are more charges # i want number and %
#where are for nonsmoker the charges are less and smoker is less .
```

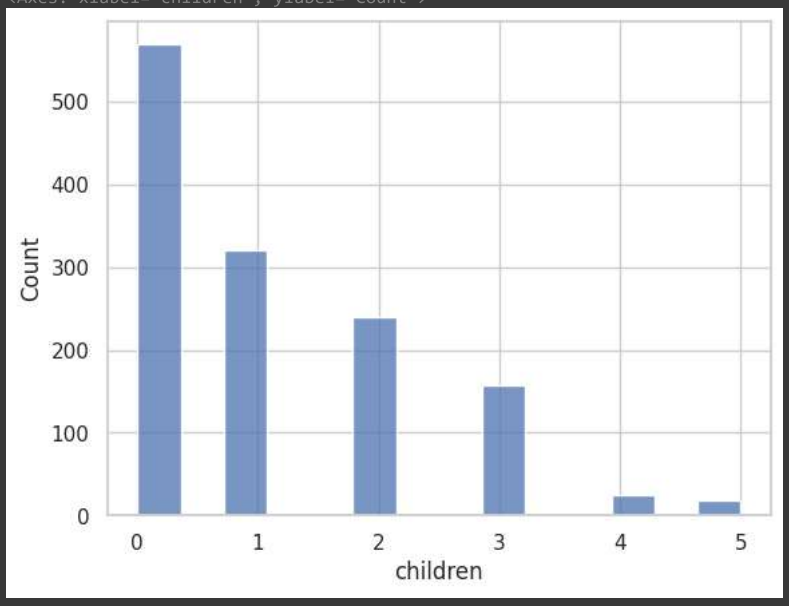
<Figure size 2300x700 with 0 Axes>




children = 5 | smoker = yes
children = 5 | smoker = no
charges
40000
35000
30000
25000
20000
15000
10000
5000
0
age
Smoker

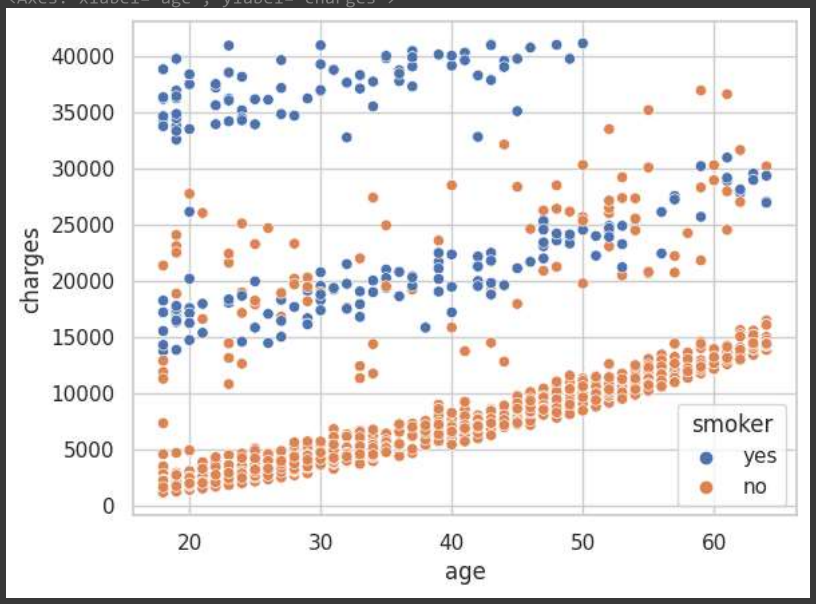
```
sns.histplot(df.children)
# as we can see here
```

```
<Axes: xlabel='children', ylabel='Count'>
```

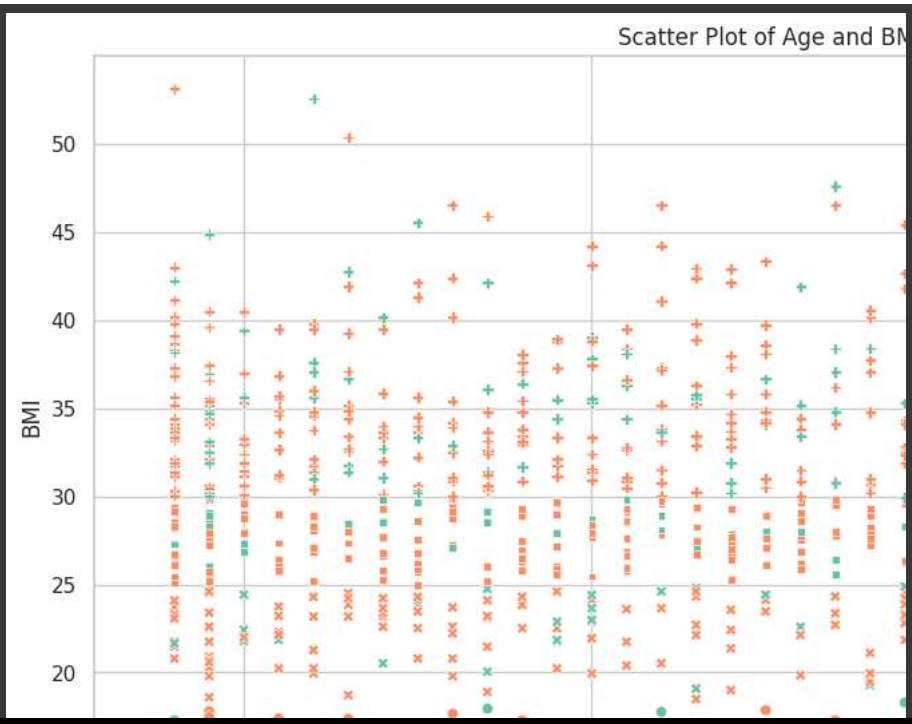



```
sns.scatterplot(x=df.age,y=df.charges,hue=df.smoker)
```

```
<Axes: xlabel='age', ylabel='charges'>
```




```
plt.figure(figsize=(17,7))
sns.scatterplot(data=df, x='age', y='bmi', hue='smoker', style='BMI Category', palette='Set2')
plt.xlabel('Age')
plt.ylabel('BMI')
plt.title('Scatter Plot of Age and BMI by Smoker Status and BMI Category')
plt.legend(title='Smoker', loc='upper right')
plt.show()
#in every age category we have smoker are their with health ,obese ,overweight and underweight is there
#underwights arre less smoker comapritve to other bmi catogries.
#but obese smoker are they will have high cahrges also they may be claim there insurence .
#but healthy and obese are more.
# as we see ornage is more it says smoker are less,with more obese in every age category
```

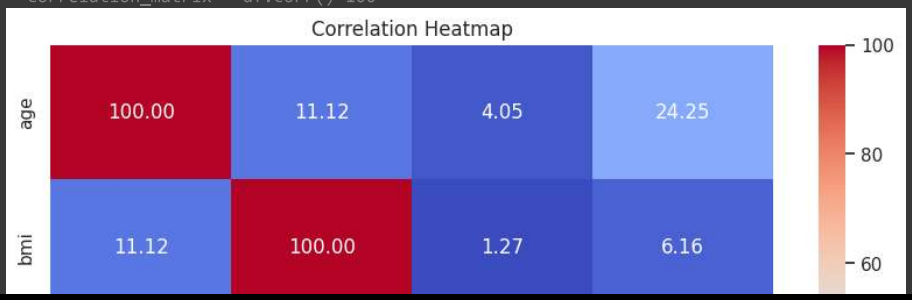



```
plt.figure(figsize=(23,7))
sns.catplot(data=df, x='bmi', y='charges',  hue='region', col='smoker', row='children')
plt.xlabel('Smoker')
plt.ylabel('Charges')
plt.suptitle('Distribution of Charges by Smoker, Region, Age, and Children')
plt.subplots_adjust(top=0.9)
plt.show()
# i don't know why 0 childrens graph wasn't coming
# but it semms the bmi is not that contributing as such
```

```
  plt.figure(figsize=(23,7))
  sns.catplot(data=df, x='BMI Category', y='charges',  hue='region', col='smoker', row='children')
  plt.xlabel('Smoker')
  plt.ylabel('Charges')
  plt.suptitle('Distribution of Charges by Smoker, Region, Age, and Children')
  plt.subplots_adjust(top=0.9)
  plt.show()
  # i don't know why 0 childrens graph wasn't coming
  # but it semms obese with smoking have high insurences charges.
```

```
correlation_matrix = df.corr()*100
print(correlation_matrix)
plt.figure(figsize=(10, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Heatmap')
plt.show()
```

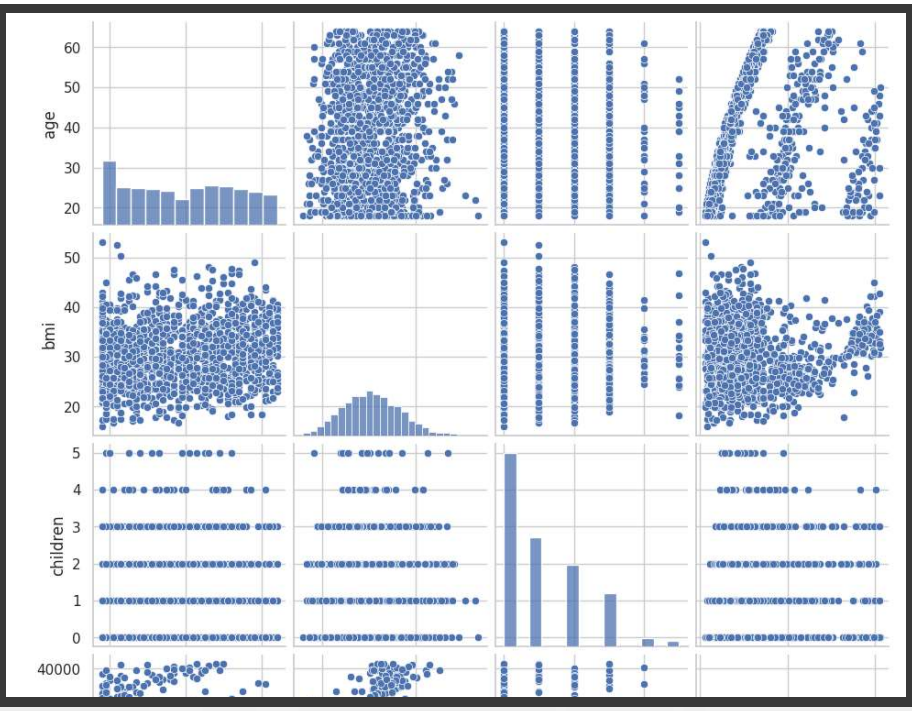
```
                 age         bmi    children     charges
age       100.000000   11.123110    4.046028   24.248702
bmi        11.123110  100.000000    1.273814    6.161066
children    4.046028    1.273814  100.000000    6.856504
charges    24.248702    6.161066    6.856504  100.000000
<ipython-input-69-ab18c3bbe774>:1: FutureWarning: The default value of numeric_only in D
  correlation_matrix = df.corr()*100
```




```
from scipy import stats
variance = df.var()
std_deviation = df.std()
print("Variance:")
print(variance)
print("\nStandard Deviation:")
print(std_deviation)
```

```
Variance:
age         1.966671e+02
bmi         3.731595e+01
children    1.457411e+00
charges     9.189288e+07
dtype: float64

Standard Deviation:
age           14.023805
bmi            6.108678
children       1.207233
charges     9586.077172
dtype: float64
<ipython-input-70-73f559868d1b>:2: FutureWarning: The default value of numeric_only in DataFrame.var is deprecated. In a future version,
  variance = df.var()
<ipython-input-70-73f559868d1b>:3: FutureWarning: The default value of numeric_only in DataFrame.std is deprecated. In a future version,
  std_deviation = df.std()
```

```
summary = df.describe()
# Visualization: Pairplot for numerical features
sns.pairplot(df[['age', 'bmi', 'children', 'charges']])
plt.show()
```



Age: Premiums are often lower for younger people because health risks increase with age. Health history: Pre-existing conditions and medical exams are taken into account. Coverage type: The type of health insurance policy is considered. Family medical history: Family health background is considered. Lifestyle: Lifestyle-related habits are considered. Gender: Premiums are different for men and women, but the difference is not significant. Smoking: Non-smokers generally pay lower premiums. smoking is come under term insurance.(life, health, auto, and long-term disability.) Other factors that may be considered include: Location , Current health status,Type of health insurance policy.

Age Distribution: The age of individuals in the dataset is distributed across various ranges. There is a concentration of younger individuals, but there is a wide age range.

BMI Distribution: BMI (Body Mass Index) shows a spread of values with some concentration in the center. It appears to be normally distributed.

Children: Most individuals do not have children, but there is a range of values, including some with multiple children.

Charges Distribution: The distribution of insurance charges is positively skewed, indicating that a few individuals have significantly higher charges.

Age Distribution: The dataset contains individuals of various ages, with the majority falling in the 20-60 age range. There's a relatively even distribution, with a slight peak in the early 20s.

BMI Distribution: Body Mass Index (BMI) varies across the dataset, but most individuals have BMIs in the 20-35 range. The distribution is relatively normal, with a few outliers on both ends.

Children Distribution: The number of children per individual varies, with many having no children or one child. There are fewer individuals with larger families.

Smoker Distribution: The dataset includes both smokers and non-smokers, but non-smokers are more prevalent.

Sex Distribution: There are nearly equal numbers of males and females in the dataset.

Region Distribution: The data spans multiple regions, with representation from the southeast, northeast, southwest, and northwest. The distribution across regions appears relatively even.

Charges Distribution: Medical charges vary significantly, with some individuals incurring very high charges. The distribution is right-skewed, with most charges on the lower end.

Charges vs. Age: There is a positive correlation between age and medical charges, indicating that older individuals tend to have higher medical costs.

Charges vs. BMI: There is a less pronounced but still positive correlation between BMI and medical charges, suggesting that individuals with higher BMIs tend to have somewhat higher medical costs.

Charges vs. Smoking: Smokers tend to have higher medical charges than non-smokers, which is expected since smoking can lead to various health issues.

Age vs. BMI: There is a moderate positive correlation between age and BMI, meaning that, on average, older individuals tend to have slightly higher BMIs.

Age vs. Number of Children: There is no clear correlation between age and the number of children an individual has. **

Children vs. Charges: The number of children doesn't show a strong correlation with medical charges. However, individuals with more children tend to have slightly lower medical costs on average.**

all columns are imp for ml algo to predict charges

```
#their are max 5 childrens and min is zero
df["children"].corr(df["charges"])*100
```

```
6.856504233469226
```

Age: Premiums are often lower for younger people because health risks increase with age. Health history: Pre-existing conditions and medical exams are taken into account. Coverage type: The type of health insurance policy is considered. Family medical history: Family health background is considered. Lifestyle: Lifestyle-related habits are considered. Gender: Premiums are different for men and women, but the difference is not significant. Smoking: Non-smokers generally pay lower premiums. Other factors that may be considered include: Location Current health status Type of health insurance policy

Smokers pay higher premiums than non-smokers. The amount depends on how often they smoke. Smokers may be offered reduced coverage or denied coverage altogether. Smokers may have limited policy options. The type and amount of tobacco used The smoker's age Some insurers charge a higher rate if the consumption level is above a particular threshold. Smoking can lead to complications like osteoporosis in pregnant women. If you don't inform your insurer that you have become a smoker and something happens during the policy term, it may appear like fraud to the insurance company. They might delay the payout until they can verify the claim request.

blog

Creating a blog for your insurance data analysis project can be an excellent way to share your insights and findings with a broader audience. Here's a brief outline of how you can structure your blog:

Title: Choose a catchy and informative title that represents the essence of your project.

Introduction:

- Start with an engaging introduction that explains the importance of insurance data analysis and why you conducted this project.

Project Overview:

- Provide a brief overview of your project, including its objectives, data sources, and methodologies used.

Data Exploration:

- Discuss the dataset you used, its structure, and the initial exploration you conducted. Include visualizations and statistics that highlight key insights.

Data Preprocessing:

- Explain the steps you took to clean and prepare the data for analysis. Mention any missing data handling, feature engineering, or encoding processes.

Data Analysis:

- Present your data analysis findings. This section can include:
  - Demographic insights: Age, gender distribution, region-wise analysis.
  - Insurance cost distribution and factors affecting costs.
  - Visualization of relationships between variables.
  - Highlight any interesting trends or patterns you observed.

Predictive Modeling (if applicable):

- Describe any predictive modeling you performed. Explain the machine learning algorithms used and discuss model accuracy and performance.

Customer Insights:

- Share any customer segmentation results and how these insights can be used for targeted marketing or services.

Risk Assessment:

- Discuss your findings related to risk assessment, and if possible, suggest strategies to mitigate risks.

Conclusion:

- Summarize the key takeaways from your project. What are the most important insights you gained from the data?

Future Scope:

- Discuss potential future directions for the project, as mentioned in a previous response.

Data Visualization:

- Include relevant data visualizations, charts, and graphs to make your findings more accessible.

References:

- Cite any sources or references you used during the project.

Call to Action:

- Encourage readers to share their thoughts, questions, or insights. Provide links to relevant resources or your contact information.

Promotion and Sharing:

- Share your blog on social media, professional networks, and relevant online communities to reach a wider audience.

Feedback and Comments:

- Encourage readers to leave comments and provide feedback. Engaging with your audience can lead to valuable discussions and future collaboration opportunities.

Remember to keep your blog reader-friendly by using simple language, breaking down complex concepts, and focusing on the most interesting and relevant aspects of your project. Regularly updating the blog with new insights and findings can also keep your audience engaged and informed about the latest developments in your insurance data analysis project.

feature scope

Predictive Modeling:

Develop predictive models to estimate insurance costs for new customers based on their demographics, medical history, and other relevant factors. Implement machine learning algorithms like regression, decision trees, or neural networks to improve the accuracy of predictions.
Customer Segmentation:

Segment customers into different groups based on their characteristics and insurance usage. This can help in targeted marketing and personalized services. Risk Assessment:

Improve risk assessment models to identify high-risk individuals and propose preventive measures to reduce the risk. Customer Retention:

Develop strategies to retain existing customers, such as personalized offers, improved services, and proactive communication. Data Security and Privacy:

Focus on enhancing data security and privacy measures, especially if the project involves sensitive customer information. Mobile Applications:

Develop a mobile application that allows customers to manage their insurance policies, submit claims, and access their data on the go. Automation and Chatbots:

Implement chatbots and automation for customer support and claims processing, which can improve efficiency and reduce operational costs. Data Visualization:

Create interactive dashboards and data visualizations to provide stakeholders with real-time insights into the insurance business's performance. Compliance and Regulations:

Stay updated with insurance regulations and ensure compliance with data protection laws (e.g., GDPR) to avoid legal issues. Customer Feedback Analysis:

Analyze customer feedback to identify areas for improvement in customer service and product offerings. Integration with IoT and Wearables:

Explore how data from IoT devices and wearables can be integrated into the insurance model for better risk assessment and more accurate pricing. Market Expansion:

Consider expanding the market by offering new insurance products or targeting different geographic regions. Partnerships and Alliances:

Collaborate with healthcare providers or other businesses to offer bundled services or discounts for customers. Fraud Detection:

Improve fraud detection mechanisms to reduce insurance fraud. Cost Reduction:

Explore cost reduction strategies, such as optimizing internal operations, claims processing, and underwriting. Ecosystem Development:

Build a comprehensive ecosystem around the insurance business, including partnerships with healthcare providers, financial institutions, and others. The future scope of your project can be highly dependent on the goals and resources available to your organization. Consider the specific needs of your target customers and stakeholders to determine which direction to pursue. Additionally, staying informed about industry trends and technological advancements is crucial for long-term success in the insurance domain.

To gain insights from the data you've provided, we can perform some exploratory data analysis (EDA). Here are some insights you can derive from the data:

Data Summary:

You have records for individuals with features such as age, sex, BMI, children, smoker status, region, and their insurance charges. Descriptive Statistics:

Calculate basic statistics like mean, median, and standard deviation for numerical columns (age, BMI, children, and charges). Distributions:

Plot histograms for numerical columns to visualize the distribution of age, BMI, children, and charges. Create bar plots for categorical variables like sex, smoker status, and region to understand their distributions. Correlations:

Calculate correlations between numerical variables, especially the relationship between age and charges. You can use scatter plots to visualize this relationship. Insurance Charges Analysis:

Analyze the distribution of insurance charges to understand how charges are spread across the dataset. Calculate average insurance charges for smokers vs. non-smokers. Calculate the average insurance charges for different regions. Age and BMI Insights:

Analyze how age and BMI are distributed within the dataset. Explore if there's a relationship between age and BMI. Smoker Analysis:

Investigate how smoking status affects insurance charges. Compare the distribution of charges for smokers and non-smokers. Children Analysis:

Analyze how the number of children (dependents) affects insurance charges. Region Analysis:

Explore if there are differences in charges across different regions. Gender Analysis:

Analyze whether gender has an impact on insurance charges. Multi-variable Analysis:

Consider multi-variable analysis to understand more complex relationships. For example, you can analyze how smoking, age, and BMI interact to affect insurance charges. Outlier Detection:

Identify and handle any outliers in the data that might affect your analysis. Data Visualization:

Create plots and visualizations to help understand the data better. For instance, use box plots, violin plots, and pair plots for insights. Statistical Tests:

Perform statistical tests to determine if observed differences (e.g., charges between smokers and non-smokers) are statistically significant. Machine Learning:

If you want to predict insurance charges or understand which features have the most impact on charges, you can build regression models. Please let me know if you'd like to explore any specific aspect of the data or if you have any particular questions in mind.

```
correlation_matrix = df.corr()
print(correlation_matrix)

# Perform a t-test
t_stat, p_value = stats.ttest_ind(df['charges'][df['smoker'] == 'yes'], df['charges'][df['smoker'] == 'no'])
print("T-statistic:", t_stat)
print("P-value:", p_value)
```

```
                age       bmi  children   charges
age       1.000000  0.111231  0.040460  0.242487
bmi       0.111231  1.000000  0.012738  0.061611
children  0.040460  0.012738  1.000000  0.068565
charges   0.242487  0.061611  0.068565  1.000000
T-statistic: nan
P-value: nan
<ipython-input-73-bc750d286314>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version
  correlation_matrix = df.corr()
```

```
model = sm.OLS(df['charges'], df[[ 'age', 'bmi', 'children']])

# Get the model summary
results = model.fit()
print(results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                charges   R-squared (uncentered):                   nan
Model:                            OLS   Adj. R-squared (uncentered):              nan
Method:                 Least Squares   F-statistic:                              nan
Date:                Fri, 27 Oct 2023   Prob (F-statistic):                       nan
Time:                        03:07:29   Log-Likelihood:                          nan
No. Observations:                1330   AIC:                                     nan
Df Residuals:                    1327   BIC:                                     nan
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
age              nan        nan        nan        nan         nan         nan
bmi              nan        nan        nan        nan         nan         nan
children         nan        nan        nan        nan         nan         nan
==============================================================================
Omnibus:                          nan   Durbin-Watson:                     nan
Prob(Omnibus):                    nan   Jarque-Bera (JB):                  nan
Skew:                             nan   Prob(JB):                          nan
Kurtosis:                         nan   Cond. No.                         42.2
==============================================================================

Notes:
[1] R² is computed without centering (uncentered) since the model does not contain a cor
[2] Standard Errors assume that the covariance matrix of the errors is correctly specifi
```

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
X = df.drop(['charges','BMI Category','smoker_status'], axis=1)
y = df['charges']
categorical_cols = ['smoker', 'region','sex',]
preprocessor = ColumnTransformer(
    transformers=[
        ('cat', OneHotEncoder(), categorical_cols)
    ],
```

```
    remainder='passthrough'  # Include numeric columns as is
)

model = Pipeline([
    ('preprocessor', preprocessor),
    ('model', LinearRegression())
])
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
model.fit(X_train, y_train)
score = model.score(X_test, y_test)
print("R-squared score:", score)
```

Dealing with outliers in your dataset depends on the nature of your analysis and the context of your project. Here are some common approaches to handle outliers:

1. **Data Transformation:** One way to handle outliers is by applying data transformations. Common transformations include taking the logarithm, square root, or cube root of the data. This can make the distribution more symmetric and help mitigate the impact of outliers.
2. **Winsorizing:** Winsorizing is a method where you replace extreme values with less extreme values. For example, you could replace values above the upper bound with the upper bound value and values below the lower bound with the lower bound value.
3. **Removing Outliers:** You can choose to remove outliers from your dataset. However, this should be done with caution, as it can result in a loss of information. Removing outliers may be more appropriate when the outliers are due to data entry errors or are irrelevant to the analysis.
4. **Robust Statistical Methods:** Some statistical methods and machine learning algorithms are inherently robust to outliers. For example, robust regression methods can minimize the influence of outliers on model parameters.
5. **Binning or Categorization:** You can categorize or bin the data, converting it into a categorical variable. This can help preserve information about the presence of outliers.
6. **Modeling Techniques:** Use modeling techniques that are robust to outliers, such as decision trees or random forests. These models are less sensitive to extreme values.
7. **Analyze Separately:** In some cases, it might make sense to analyze the data both with and without outliers to understand the impact on your results.
8. **Domain Knowledge:** Consult with domain experts to determine if outliers are meaningful and whether they should be treated differently.

The choice of method should be based on the goals of your analysis and a clear understanding of the data and its context. It's important to document and justify the approach you choose when dealing with outliers in your data analysis.

```

```

pip install pandas-profiling

import pandas as pd from pandas_profiling import ProfileReport

Load your dataset

df = pd.read_csv('insurance.csv')

Create a profile report

profile = ProfileReport(df)

To generate an HTML report, you can use the following line:

profile.to_file("insurance_report.html")

To display the report in the Jupyter Notebook or Python script:

profile.to_widgets()

Understand the context Choose an appropriate view Eliminate confusion Draw attention Think like a designer Tell a story Conclusions Sources You can use a template for data analysis in PowerPoint. For example, you can use the Data Analysis for Business template. The basic steps in the analytic process are: Identify issues Determine the availability of suitable data Decide on which methods are appropriate for answering the questions of interest Apply the methods and evaluate Summarize and communicate the results

Creating a PowerPoint presentation for a data analysis project involves summarizing and visually representing your findings and insights. Here's a suggested outline for your presentation:

Slide 1: Title

- Title of the Presentation
- Your Name
- Date

Slide 2: Introduction

- Briefly introduce the dataset and its source
- Mention the objectives and goals of your analysis

Slide 3: Data Overview

- Describe the dataset's size, structure, and data types
- Highlight any initial data preprocessing steps

Slide 4: Data Visualization

- Include a few key visualizations (e.g., histograms, scatter plots, bar charts) that illustrate the dataset
- Provide insights from these visualizations

Slide 5: Descriptive Statistics

- Display summary statistics (mean, median, standard deviation, etc.) for key variables
- Interpret these statistics

Slide 6: Data Preprocessing

- Discuss any data cleaning or preprocessing steps performed
- Explain how missing values and outliers were handled

Slide 7: Exploratory Data Analysis (EDA)

- Present insights gained through EDA, including relationships between variables
- Use visualizations to support your findings

Slide 8: Hypothesis Testing

- Describe the hypotheses you tested
- Present the results of your hypothesis tests
- Explain the significance of the findings

Slide 9: Feature Engineering

- Discuss any feature engineering or transformation done
- Explain the rationale behind these changes

Slide 10: Machine Learning Model

- Describe the machine learning algorithm used
- Explain the model's purpose (e.g., prediction, classification)
- Mention model evaluation metrics

Slide 11: Model Results

- Present the model's performance (accuracy, F1-score, etc.)
- Discuss any insights or patterns learned from the model

Slide 12: Data Insights

- Summarize the main insights and key findings from your analysis
- Include any actionable recommendations

Slide 13: Limitations

- Discuss any limitations of your analysis or dataset

Slide 14: Future Work

- Suggest possible future work or extensions of your analysis

Slide 15: Conclusion

- Summarize the overall findings and their implications

Slide 16: Questions

- Invite questions from the audience

Slide 17: Thank You

- Express your gratitude for the audience's attention

Slide 18: References

- Cite any data sources, libraries, or references you used

Remember to keep the presentation concise, use visuals effectively, and focus on conveying the most important insights. Practice your presentation several times to ensure clarity and smooth delivery.

```
data_encoded = pd.get_dummies(df, columns=['sex', 'region','smoker'])
df = pd.concat([df, data_encoded], axis=1)
#df = df.drop(columns=['gender', 'region'])
```

```
df
```

	age	sex	bmi	children	smoker	region	charges	BMI Category	smoker_status	age	...	BMI Category	smoker_status	sex_fema
0	19	female	27.900	0	yes	southwest	16884.92400	Overweight	smoker	19	...	Overweight	smoker	
1	18	male	33.770	1	no	southeast	1725.55230	Obese	non-smoker	18	...	Obese	non-smoker	
2	28	male	33.000	3	no	southeast	4449.46200	Obese	non-smoker	28	...	Obese	non-smoker	
3	33	male	22.705	0	no	northwest	21984.47061	Normal Weight	non-smoker	33	...	Normal Weight	non-smoker	
4	32	male	28.880	0	no	northwest	3866.85520	Overweight	non-smoker	32	...	Overweight	non-smoker	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1333	50	male	30.970	3	no	northwest	10600.54830	Obese	non-smoker	50	...	Obese	non-smoker	