Name: Priansh Madan

Section: E Batch: E4

Roll Number: 58

DVA Practical 6

Aim:- Write a program to perform Chi-square Test statistical test using user defined functions. Use smoking.CSV dataset data set to perform above tests using user defined functions. Verify the results obtained with standard functions.

Plot the Heatmap to visualize the results obtained through the test

In [ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats as stats
import matplotlib.pyplot as plt
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
sns.set(style="darkgrid")
from scipy.stats import chi2,chi2_contingency
```

In [ ]:
```
df=pd.read_csv('Smoking_2.csv')
```

In [ ]:
```
df.head()
```

Out[ ]:

	Unnamed: 0	gender	age	marital_status	highest_qualification	nationality	ethnicity	gross_incor
0	1	Male	38	Divorced	No Qualification	British	White	2,600 to 5,2
1	2	Female	42	Single	No Qualification	British	White	Under 2,6
2	3	Male	40	Married	Degree	English	White	28,600 36,4
3	4	Female	40	Married	Degree	English	White	10,400 15,6
4	5	Female	39	Married	GCSE/O Level	British	White	2,600 to 5,2

In [ ]:
```
df.isna().sum()
```

Out[ ]:
```
Unnamed: 0                 0
gender                     0
age                        0
marital_status             0
highest_qualification      0
nationality                0
ethnicity                  0
gross_income               0
region                     0
smoke                      0
amt_weekends            1270
amt_weekdays            1270
type                    1270
dtype: int64
```

In [ ]:
```
df.describe()
```

Out[ ]:

	Unnamed: 0	age	amt_weekends	amt_weekdays
count	1691.000000	1691.000000	421.000000	421.000000
mean	846.000000	49.836192	16.410926	13.750594
std	488.293969	18.736851	9.892988	9.388292
min	1.000000	16.000000	0.000000	0.000000
25%	423.500000	34.000000	10.000000	7.000000
50%	846.000000	48.000000	15.000000	12.000000
75%	1268.500000	65.500000	20.000000	20.000000
max	1691.000000	97.000000	60.000000	55.000000

In [ ]:
```
```

EDA

In [ ]:
```
print ("Rows     : " ,df.shape[0])
print ("Columns  : " , df.shape[1])
print ("\nFeatures : \n", df.columns.tolist())
print ("\nMissing values :  ", df.isnull().sum().values.sum())
print ("\nUnique values :  \n", df.nunique())
```

```
Rows     :  1691
Columns  :  13

Features :
 ['Unnamed: 0', 'gender', 'age', 'marital_status', 'highest_qualification', 'natio
nality', 'ethnicity', 'gross_income', 'region', 'smoke', 'amt_weekends', 'amt_week
days', 'type']

Missing values :   3810

Unique values :
 Unnamed: 0               1691
gender                       2
age                         79
marital_status               5
highest_qualification        8
nationality                  8
ethnicity                    7
gross_income                10
region                       7
smoke                        2
amt_weekends                24
amt_weekdays                24
type                         4
dtype: int64
```

In [ ]:
```
df.gender=df['gender'].astype("category")
df.marital_status=df['marital_status'].astype("category")
df.highest_qualification=df['highest_qualification'].astype("category")
df.nationality=df['nationality'].astype("category")
df.ethnicity=df['ethnicity'].astype("category")
df.gross_income=df['gross_income'].astype("category")
df.region=df['region'].astype("category")
df.smoke=df['smoke'].astype("category")
df.type=df['type'].astype("category")
```

In [ ]:
```
df.describe(include='category')
```

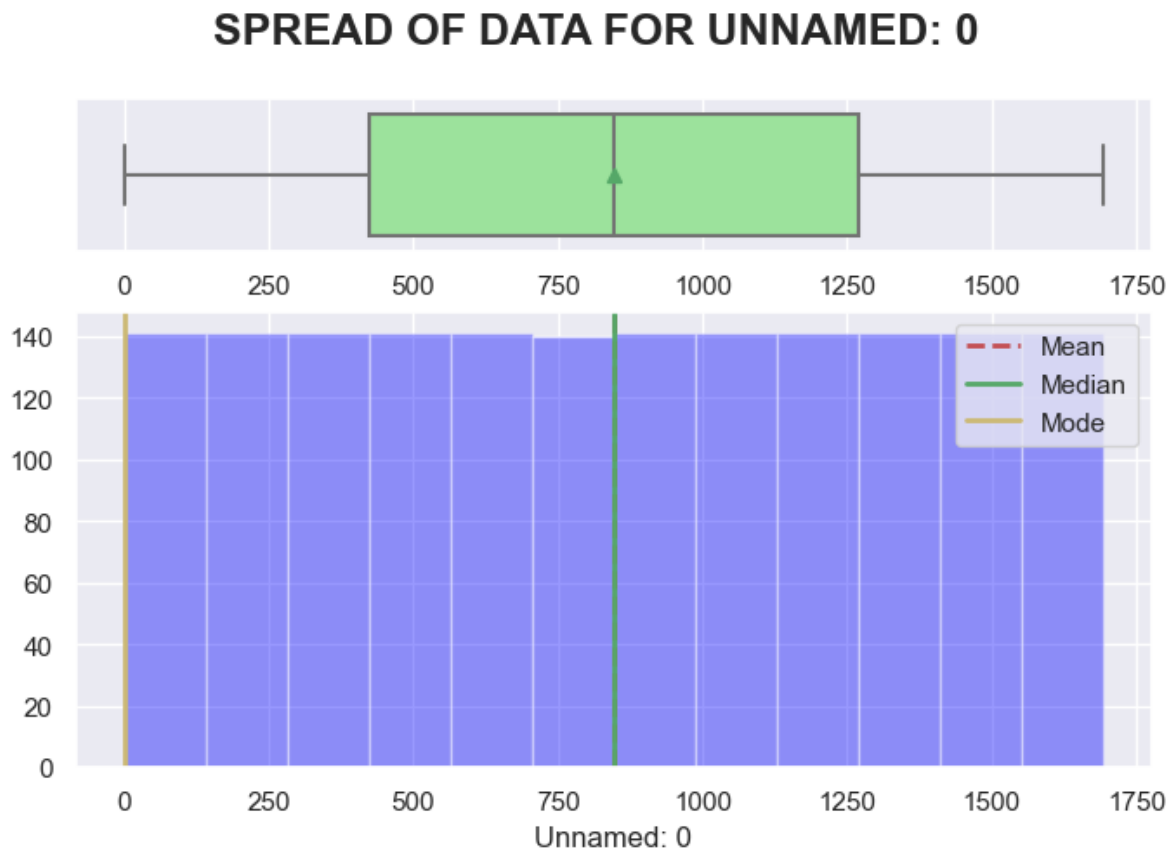
Out[ ]:

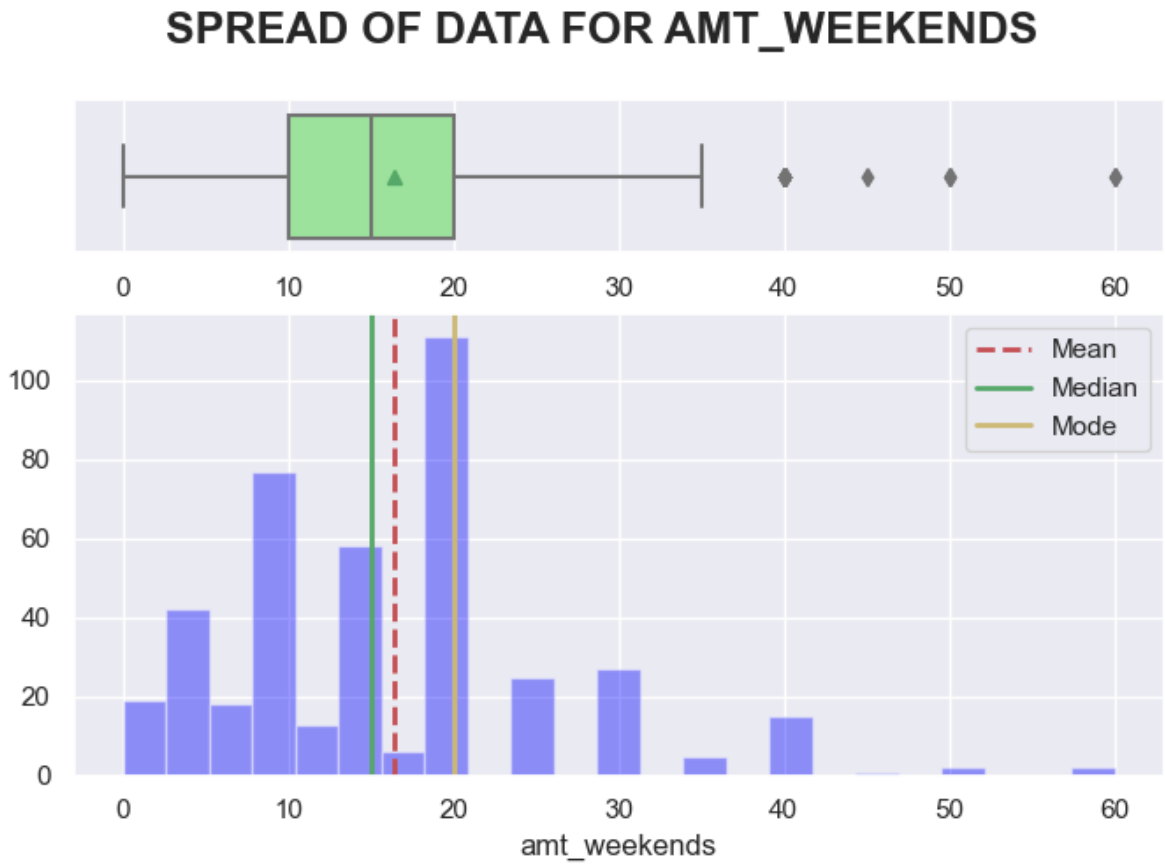
	gender	marital_status	highest_qualification	nationality	ethnicity	gross_income	regio
count	1691	1691	1691	1691	1691	1691	169
unique	2	5	8	8	7	10	
top	Female	Married	No Qualification	English	White	5,200 to 10,400	Midland & Eas Angli
freq	965	812	586	833	1560	396	44

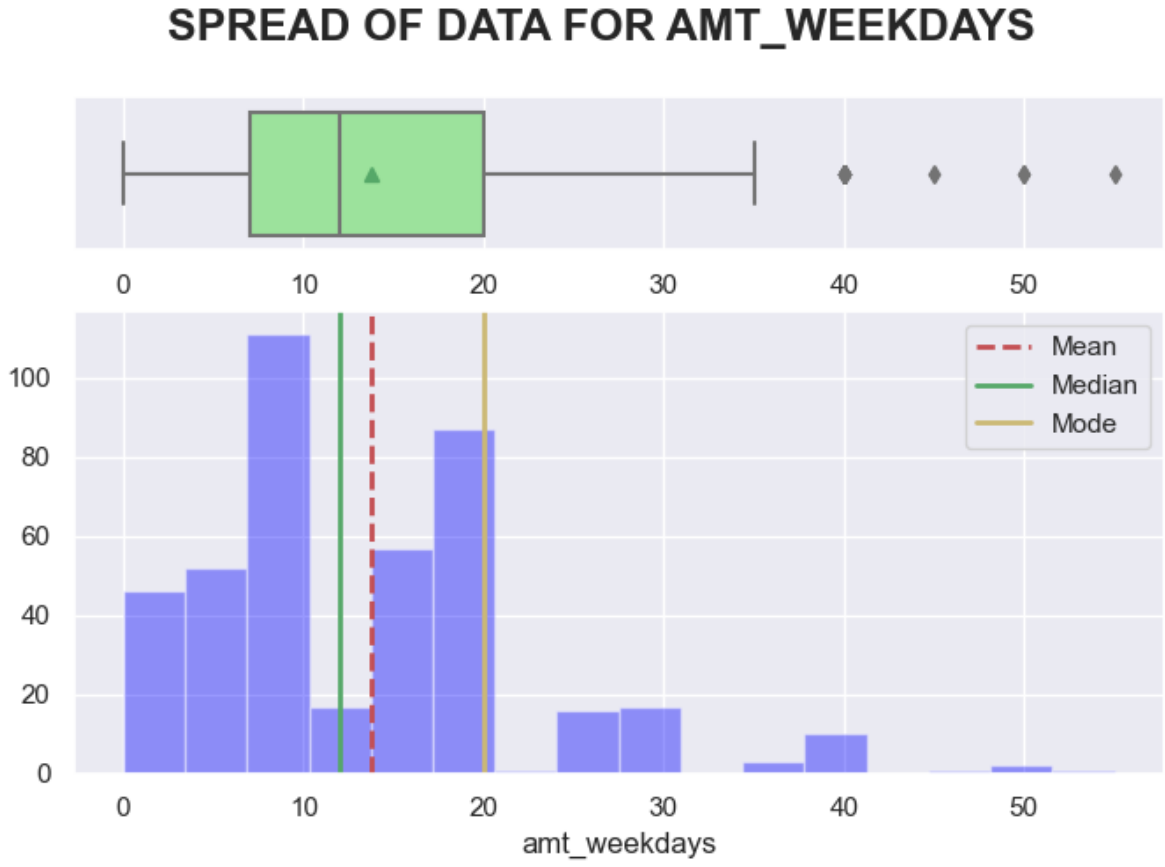
In [ ]:
```
def dist_box(data):

    Name=data.name.upper()
    fig,(ax_box,ax_dis)  =plt.subplots(2,1,gridspec_kw = {"height_ratios": (.25, .
    mean=data.mean()
    median=data.median()
    mode=data.mode().tolist()[0]
    fig.suptitle("SPREAD OF DATA FOR "+ Name  , fontsize=18, fontweight='bold')
    sns.boxplot(x=data,showmeans=True, orient='h',color="lightgreen",ax=ax_box)
    ax_box.set(xlabel='')
    sns.distplot(data,kde=False,color='blue',ax=ax_dis)
    ax_dis.axvline(mean, color='r', linestyle='--',linewidth=2)
```

```
    ax_dis.axvline(median, color='g', linestyle='-',linewidth=2)
    ax_dis.axvline(mode, color='y', linestyle='-',linewidth=2)
    plt.legend({'Mean':mean,'Median':median,'Mode':mode})
```

In [ ]:
```
list_col=  df.select_dtypes([np.number]).columns
for i in range(len(list_col)):
    dist_box(df[list_col[i]])
```

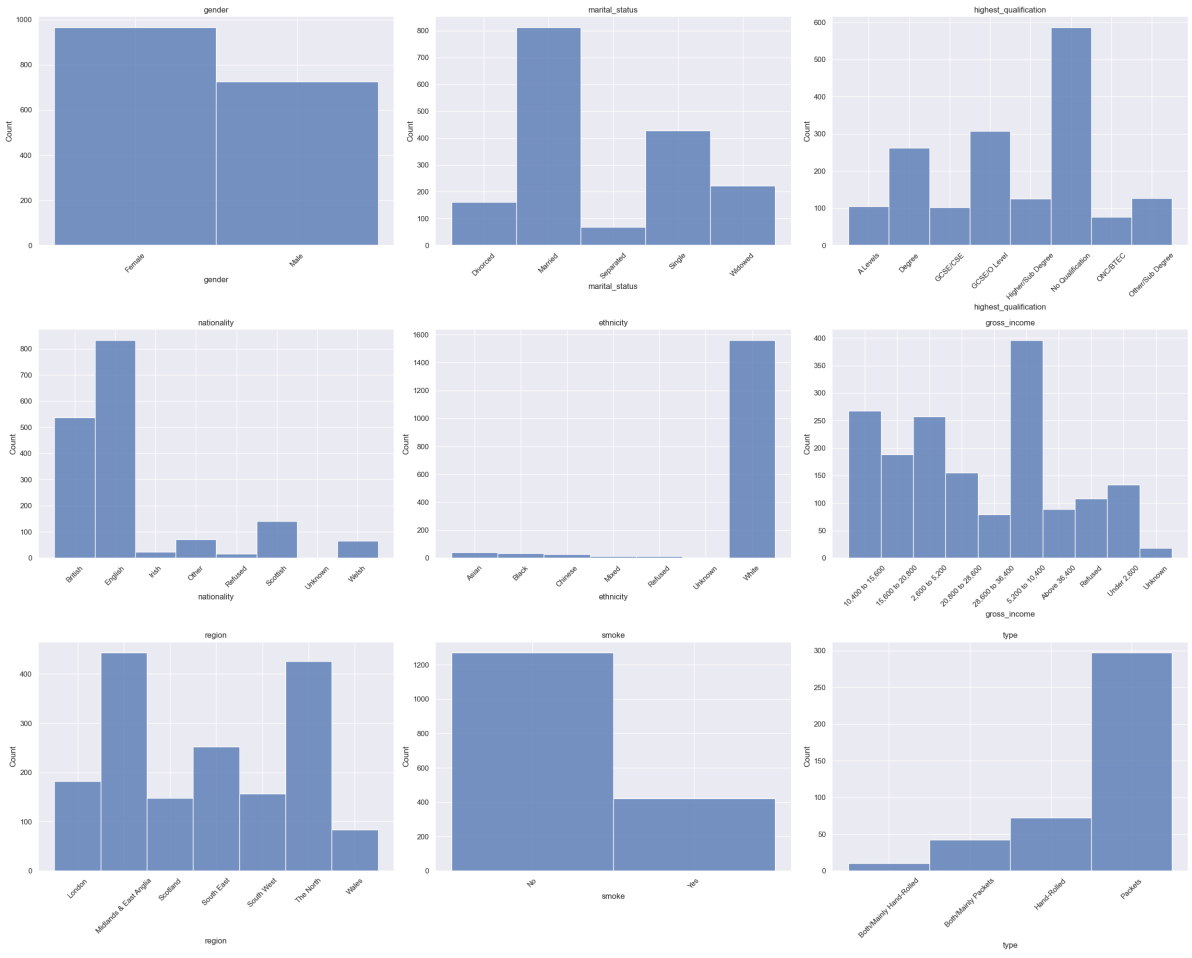



SPREAD OF DATA FOR AMT_WEEKENDS
0
10
20
30
40
50
60
Mean
Median
Mode
100
80
60
40
20
0
amt_weekends

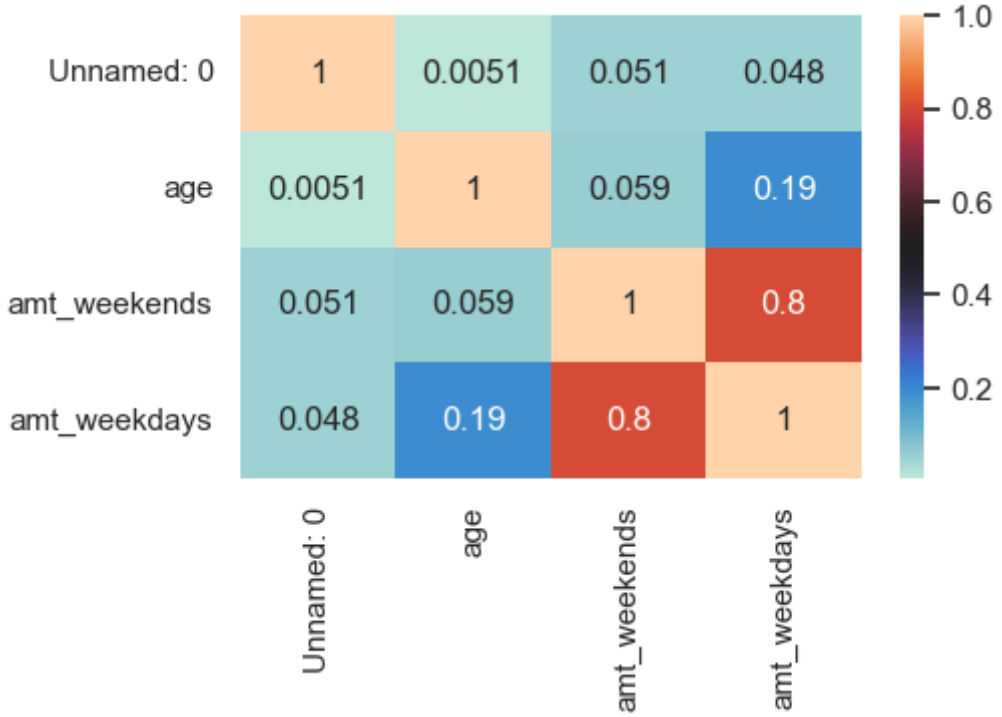

SPREAD OF DATA FOR AMT_WEEKDAYS
0
10
20
30
40
50
Mean
Median
Mode
100
80
60
40
20
0
amt_weekdays

In [ ]:
```
def bar_perc(plot, feature):
    total = len(feature)
    for p in plot.patches:
        percentage = '{:.1f}%'.format(100 * p.get_height()/total)
        x = p.get_x() + p.get_width() / 2 - 0.05
        y = p.get_y() + p.get_height()
        plot.annotate(percentage, (x, y), size = 12)
```

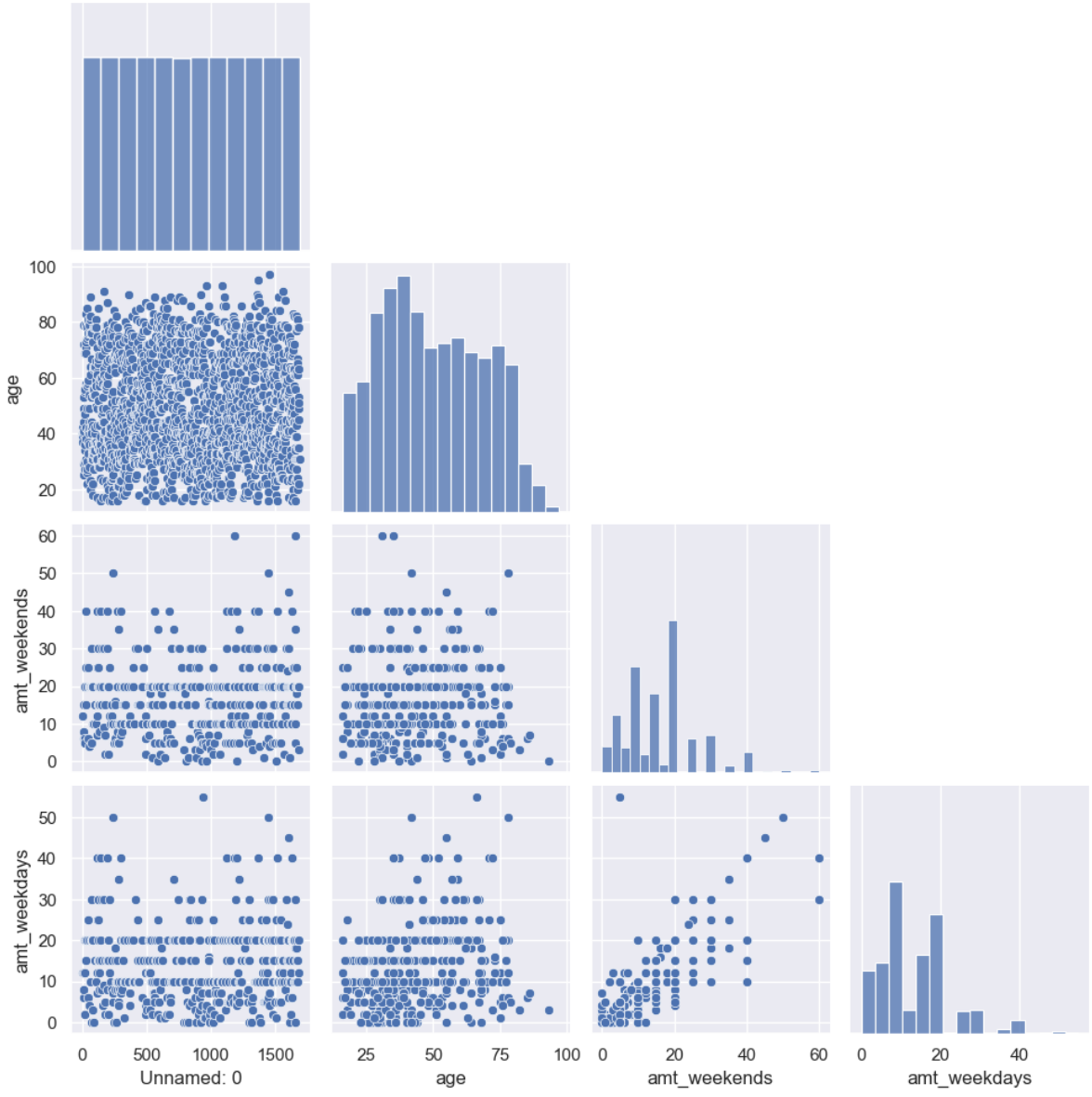
In [ ]:
```
list_col=  ['gender', 'marital_status', 'highest_qualification','nationality', 'eth
plt.figure(figsize=(25, 20))
for i in range(len(list_col)):
    plt.subplot(3,3,i+1)
    plt.title(list_col[i])
    sns.histplot(data=df,x=df[list_col[i]])
    sns.set(font_scale=1)
    plt.xticks(rotation=45)

plt.tight_layout()
plt.show()
```



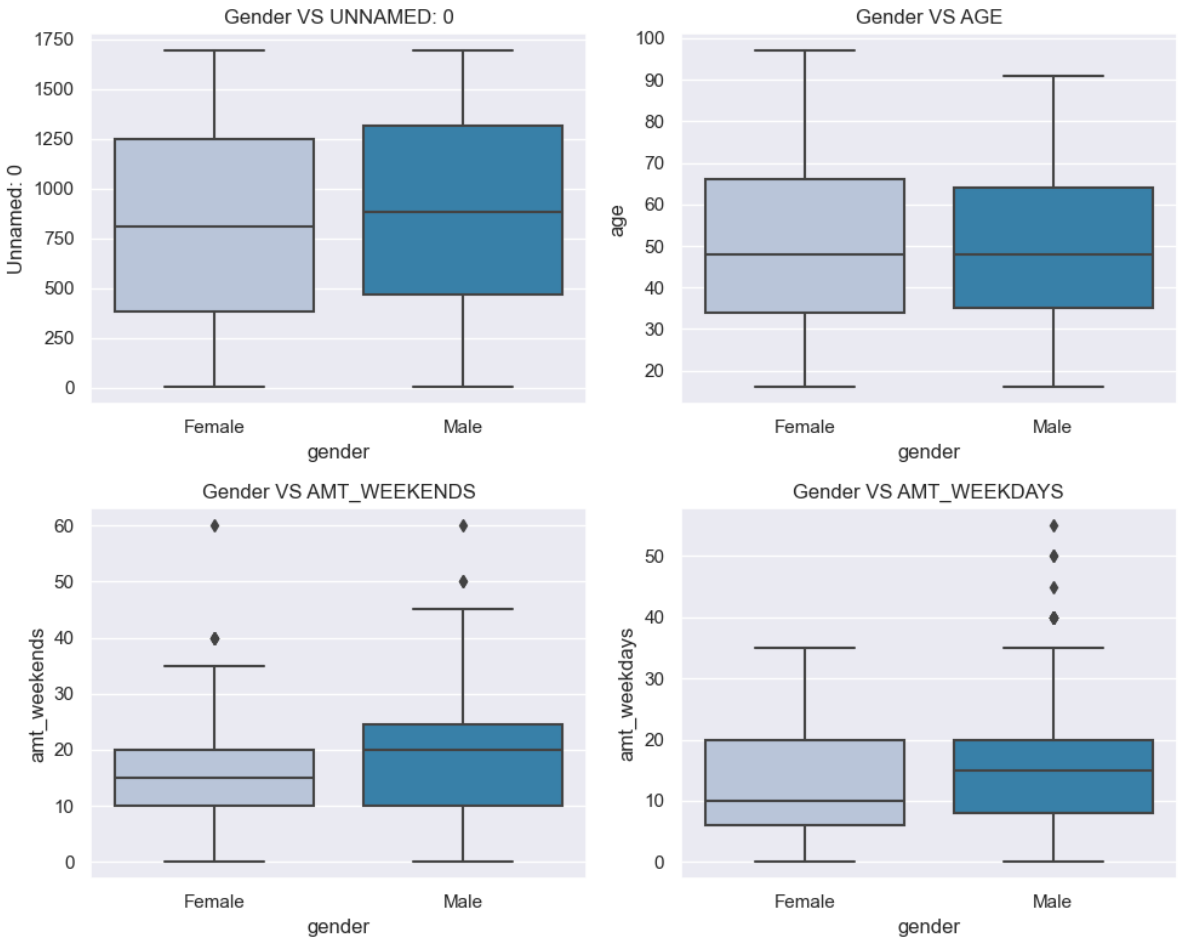
In [ ]:
```
plt.figure(figsize=(5,3))
sns.heatmap(df.corr(),annot=True ,cmap="icefire" )
plt.show()
```



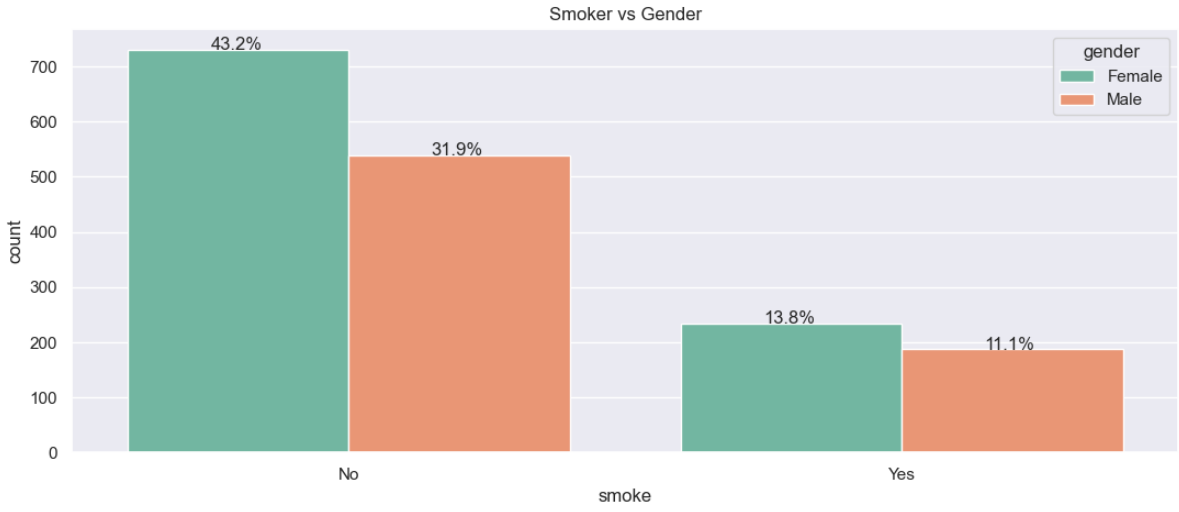
In [ ]:
```
sns.pairplot(data=df , corner=True)
plt.show()
```



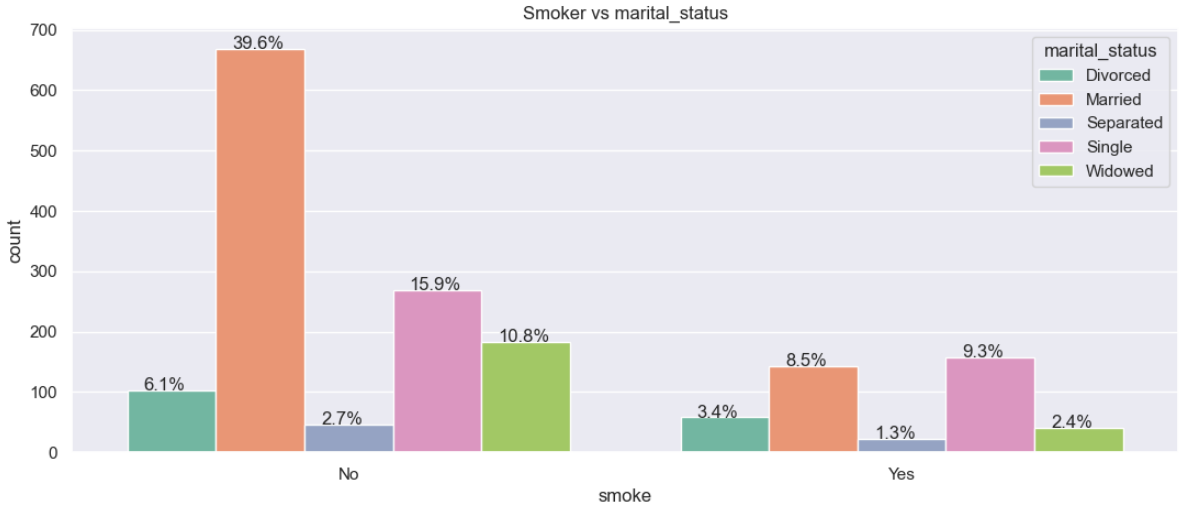
In [ ]:
```
fig1, axes1 =plt.subplots(2,2,figsize=(10, 8))

list_col=  df.select_dtypes([np.number]).columns
for i in range(len(list_col)):
    row=i//2
    col=i%2
    ax=axes1[row,col]
    sns.boxplot(y=df[list_col[i]],x=df['gender'],ax=ax,palette="PuBu", orient='v')

plt.tight_layout()
plt.show()
```



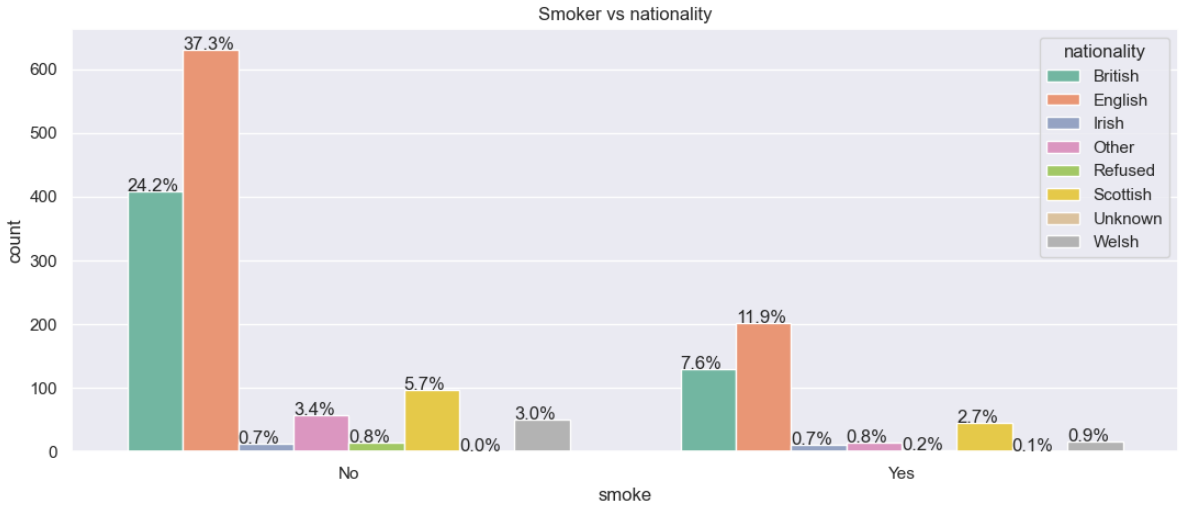
In [ ]:
```
plt.figure(figsize=(13,5))
ax=sns.countplot(x='smoke',hue='gender',data=df,palette='Set2')
bar_perc(ax,df['gender'])
ax.set(title="Smoker vs Gender");
```



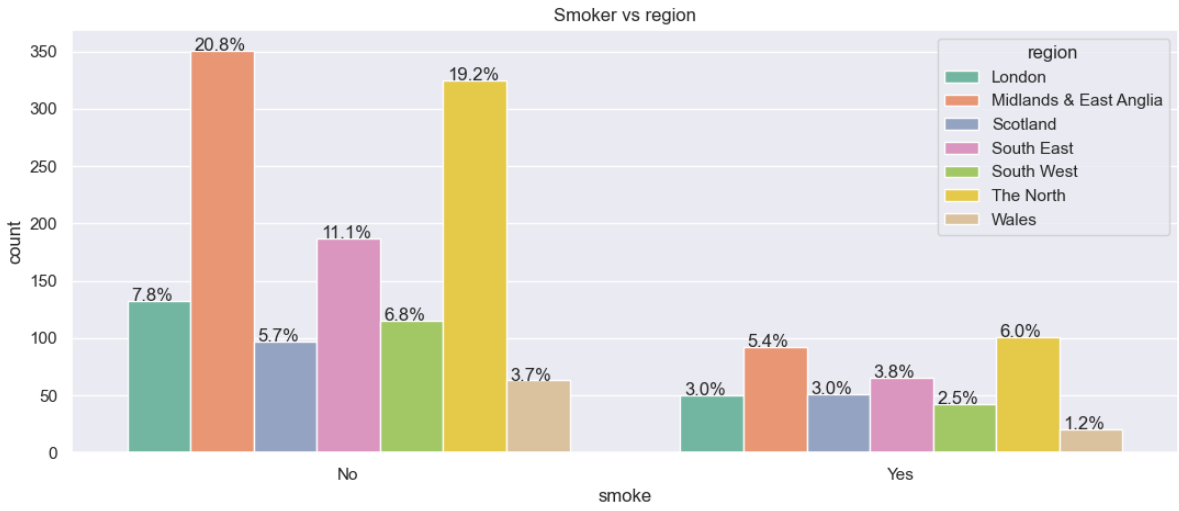
In [ ]:
```
plt.figure(figsize=(13,5))
ax=sns.countplot(x='smoke',hue='marital_status',data=df,palette='Set2')
bar_perc(ax,df['marital_status'])
ax.set(title="Smoker vs marital_status");
```



In [ ]:
```
plt.figure(figsize=(13,5))
ax=sns.countplot(x='smoke',hue='nationality',data=df,palette='Set2')
bar_perc(ax,df['nationality'])
ax.set(title="Smoker vs nationality");
```



In [ ]:
```
plt.figure(figsize=(13,5))
ax=sns.countplot(x='smoke',hue='region',data=df,palette='Set2')
bar_perc(ax,df['region'])
ax.set(title="Smoker vs region");
```



Chi Square Test

In [ ]:
```
contingency_table=pd.crosstab(df["gender"],df["smoke"])
print('contingency_table :-\n',contingency_table)
```

```
contingency_table :-
 smoke     No  Yes
gender
Female   731  234
Male     539  187
```

In [ ]:
```
Observed_Values = contingency_table.values
print("Observed Values :-\n",Observed_Values)
```

```
Observed Values :-
 [[731 234]
 [539 187]]
```

In [ ]:
```
b=stats.chi2_contingency(contingency_table)
Expected_Values = b[3]
print("Expected Values :-\n",Expected_Values)
```

```
Expected Values :-
 [[724.74866943 240.25133057]
 [545.25133057 180.74866943]]
```

In [ ]:
```
no_of_rows=len(contingency_table.iloc[0:2,0])
no_of_columns=len(contingency_table.iloc[0,0:2])
df=(no_of_rows-1)*(no_of_columns-1)
print("Degree of Freedom:-",df)
```

```
Degree of Freedom:- 1
```

In [ ]:
```
alpha=0.05
```

In [ ]:
```
chi_square=sum([(o-e)**2./e for o,e in zip(Observed_Values,Expected_Values)])
chi_square_statistic=chi_square[0]+chi_square[1]
print("chi-square statistic:-",chi_square_statistic)
```

```
chi-square statistic:- 0.5044591484173093
```

In [ ]:
```
#Critical Value
critical_value=chi2.ppf(q=1-alpha,df=df)
print('critical_value:',critical_value)
```

```
critical_value: 3.841458820694124
```

In [ ]:
```
#p-value
p_value=1-chi2.cdf(x=chi_square_statistic,df=df)
print('p-value:',p_value)
```

```
p-value: 0.4775473364084263
```

In [ ]:
```
print('Significance level: ',alpha)
print('Degree of Freedom: ',df)
print('chi-square statistic:',chi_square_statistic)
print('critical_value:',critical_value)
print('p-value:',p_value)
```

```
Significance level:  0.05
Degree of Freedom:  1
chi-square statistic: 0.5044591484173093
critical_value: 3.841458820694124
p-value: 0.4775473364084263
```

Define null and alternative hypothesis

H0:μ1=μ2 There are not any difference in smoking between Female and Male

H1:μ1<μ2 or μ1>μ2 There are any difference in smoking between Female and Male

If P values is less than alpha reject the null hypothesis. α = 0.05

Identify gender is norminal varialble. So The Chi-Squre test is one of the good options to prove. The Chi-Square test is a statistical procedure for determining the difference between observed and expected data. This test can also be used to determine whether it correlates to the categorical variables in our data. It helps to find out whether a difference between two categorical variables is due to chance or a relationship between them.

In [ ]:
```
if chi_square_statistic>=critical_value:
    print("Reject H0,there are not any difference in smoking between Female and Ma
else:
    print("Retain H0,there are not any difference in smoking between Female and Ma

if p_value<=alpha:
    print("Reject H0,there are not any difference in smoking between Female and Ma
else:
    print("Retain H0,there are not any difference in smoking between Female and Ma
```

```
Retain H0,there are not any difference in smoking between Female and Male
Retain H0,there are not any difference in smoking between Female and Male
```

Define null and alternative hypothesis

H0:μ1=μ2 There are not any difference in income between smoker and non-smoker

H1:μ1<μ2 or μ1>μ2 There are any difference in income between smoker and non-smoker

If P values is less than alpha reject the null hypothesis. α = 0.05

Identify income is ordinal varialbles. So The Chi-Squre test is one of the good options to prove. The Chi-Square test is a statistical procedure for determining the difference between observed and expected data. This test can also be used to determine whether it correlates to the categorical variables in our data. It helps to find out whether a difference between two categorical variables is due to chance or a relationship between them.

In [ ]:
```
smoking=pd.read_csv('Smoking_2.csv')
contingency_table=pd.crosstab(smoking["gross_income"],smoking["smoke"])
print('contingency_table :-\n',contingency_table)
```

```
contingency_table :-
 smoke              No  Yes
gross_income
10,400 to 15,600  185   83
15,600 to 20,800  143   45
2,600 to 5,200    193   64
20,800 to 28,600  117   38
28,600 to 36,400   70    9
5,200 to 10,400   289  107
Above 36,400       74   15
Refused            87   21
Under 2,600        97   36
Unknown            15    3
```

In [ ]:
```
Observed_Values = contingency_table.values
print("Observed Values :-\n",Observed_Values)
```

```
Observed Values :-
 [[185  83]
 [143  45]
 [193  64]
 [117  38]
 [ 70   9]
 [289 107]
 [ 74  15]
 [ 87  21]
 [ 97  36]
 [ 15   3]]
```

In [ ]:
```
b=stats.chi2_contingency(contingency_table)
Expected_Values = b[3]
print("Expected Values :-\n",Expected_Values)
```

```
Expected Values :-
 [[201.27735068  66.72264932]
 [141.19455943  46.80544057]
 [193.01596688  63.98403312]
 [116.41040804  38.58959196]
 [ 59.33175636  19.66824364]
 [297.40981668  98.59018332]
 [ 66.84210526  22.15789474]
 [ 81.11176818  26.88823182]
 [ 99.88764045  33.11235955]
 [ 13.51862803   4.48137197]]
```

In [ ]:
```
no_of_rows=len(contingency_table.iloc[0:11,0])
no_of_columns=len(contingency_table.iloc[0,0:2])
df=(no_of_rows-1)*(no_of_columns-1)
print("Degree of Freedom:-",df)
```

```
Degree of Freedom:- 9
```

In [ ]:
```
alpha=0.05
```

In [ ]:
```
chi_square=sum([(o-e)**2./e for o,e in zip(Observed_Values,Expected_Values)])
chi_square_statistic=chi_square[0]+chi_square[1]
print("chi-square statistic:-",chi_square_statistic)
```

```
chi-square statistic:- 19.835003487043213
```

In [ ]:
```
#critical_value
critical_value=chi2.ppf(q=1-alpha,df=df)
print('critical_value:',critical_value)
```

```
critical_value: 16.918977604620448
```

In [ ]:
```
#p-value
p_value=1-chi2.cdf(x=chi_square_statistic,df=df)
print('p-value:',p_value)
```

```
p-value: 0.018958411214945903
```

In [ ]:
```
print('Significance level: ',alpha)
print('Degree of Freedom: ',df)
print('chi-square statistic:',chi_square_statistic)
print('critical_value:',critical_value)
print('p-value:',p_value)
```

```
Significance level:  0.05
Degree of Freedom:  9
chi-square statistic: 19.835003487043213
critical_value: 16.918977604620448
p-value: 0.018958411214945903
```

In [ ]:
```
if chi_square_statistic>=critical_value:
    print("Reject H0,there are not any difference in income between smoker and non
else:
    print("Retain H0,there are not any difference in income between smoker and non

if p_value<=alpha:
    print("Reject H0,there are not any difference in income between smoker and non
else:
    print("Retain H0,there are not any difference in income between smoker and non
```

```
Reject H0,there are not any difference in income between smoker and non-smoker
Reject H0,there are not any difference in income between smoker and non-smoker
```