scratchpad-6

May 3, 2023

[4]:
```
#Importing packages
import numpy as np
import pandas as pd

# Importing matplotlib and seaborn for graphs
import matplotlib.pyplot as plt
import seaborn as sns
sns.set(style='whitegrid')

import warnings
warnings.filterwarnings('ignore')

from scipy import stats
from scipy.stats import kstest
import statsmodels.api as sm

# Importing Date & Time util modules
from dateutil.parser import parse

import statistics
from scipy.stats import norm
```

[18]:
```
df=pd.read_csv("https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/
↪001/293/original/walmart_data.csv?1641285094")
```

Missing Value - Calculator

[6]:
```
def missingValue(df):
    #Identifying Missing data. Already verified above. To be sure again␣
↪checking.
    total_null = df.isnull().sum().sort_values(ascending = False)
    percent = ((df.isnull().sum()/df.isnull().count())*100).
↪sort_values(ascending = False)
    print("Total records = ", df.shape[0])

    md = pd.concat([total_null,percent.round(2)],axis=1,keys=['Total␣
↪Missing','In Percent'])
```

```
    return md
```

Categorical Variable Analysis Bar plot - Frequency of feature in percentage Pie Chart

[7]:
```
# Frequency of each feature in percentage.
def cat_analysis(df, colnames, nrows=2,mcols=2,width=20,height=30,␣
 ↪sortbyindex=False):
    fig , ax = plt.subplots(nrows,mcols,figsize=(width,height))
    fig.set_facecolor(color = 'lightgrey')
    string = "Frequency of "
    rows = 0
    for colname in colnames:
        count = (df[colname].value_counts(normalize=True)*100)
        string += colname + ' in (%)'
        if sortbyindex:
                count = count.sort_index()
        count.plot.bar(color=sns.color_palette("Paired"),ax=ax[rows][0])
        ax[rows][0].set_ylabel(string, fontsize=14,family = "Comic Sans MS")
        ax[rows][0].set_xlabel(colname, fontsize=14,family = "Comic Sans MS")
        count.plot.pie(colors = sns.color_palette("Paired"),autopct='%0.0f%%',
                       textprops={'fontsize': 14,'family':"Comic Sans␣
 ↪MS"},ax=ax[rows][1])
        string = "Frequency of "
        rows += 1
```

Frequency Graph in percentage

[8]:
```
# Frequency of each feature in percentage.
def bar_plot_percentage(df, colnames, sortbyindex=False):
    fig = plt.figure(figsize=(32, 36))
    fig.set_facecolor("lightgrey")
    string = "Frequency of "
    for colname in colnames:
        plt.subplot(5,2,colnames.index(colname)+1)
        count = (df[colname].value_counts(normalize=True)*100)
        string += colname + ' in (%)'
        if sortbyindex:
                count = count.sort_index()
        count.plot.bar(color=sns.color_palette('Paired'))
        plt.xticks(rotation = 70,fontsize=14,family="Comic Sans MS")
        plt.yticks(fontsize=14,family="Comic Sans MS")
        plt.ylabel(string, fontsize=14,family = "Comic Sans MS")
        plt.xlabel(colname, fontsize=14,family = "Comic Sans MS")
        string = "Frequency of "
```

Bi-Varainte Analysis for Numerical and Categorical variables

[9]:
```
def num_cat_bi(df,col_cat,col_num,nrows=1,mcols=2,width=15,height=6):
    fig , ax = plt.subplots(nrows,mcols,figsize=(width,height),squeeze=False)
    sns.set(style='white')
    fig.set_facecolor("lightgrey")
    rows = 0
    i = 0
    while rows < nrows:
        sns.boxplot(x = col_cat[i],y = col_num, data =␣
↪df,ax=ax[rows][0],palette="Paired")
        ax[rows][0].set_xlabel(col_cat[i], fontweight="bold",fontsize=14,family␣
↪= "Comic Sans MS")
        ax[rows][0].set_ylabel(col_num,fontweight="bold", fontsize=14,family =␣
↪"Comic Sans MS")
        i += 1
        sns.boxplot(x = col_cat[i],y = col_num, data =␣
↪df,ax=ax[rows][1],palette="Paired")
        ax[rows][1].set_xlabel(col_cat[i], fontweight="bold",fontsize=14,family␣
↪= "Comic Sans MS")
        ax[rows][1].set_ylabel(col_num,fontweight="bold", fontsize=14,family =␣
↪"Comic Sans MS")
        i += 1
        rows += 1
    plt.show()
```

Distribution plot based on the Male and Female

[10]:
```
def bar_M_vs_F(colname):
    fig = plt.figure(figsize=(16,6))

    male = retail_data_v1[retail_data_v1["Gender"]=='M'][colname].
↪value_counts().reset_index()
    male["percentage"]  = (male[colname]*100/male[colname].sum())
    male["legends"]        = "Male"


    female = retail_data_v1[retail_data_v1["Gender"]=='F'][colname].
↪value_counts().reset_index()
    female["percentage"] = (female[colname]*100/female[colname].sum())
    female["legends"]    = "Female"

    m_f_status = pd.concat([female,male],axis=0)

    ax = sns.
↪barplot("index","percentage",data=m_f_status,hue="legends",palette="Blues_d")
    plt.xlabel(colname)
    fig.set_facecolor("white")
    plt.title(colname + "percentage in data with respect to churn status")
```

Multi-Varainte Analysis for Numerical and Categorical variables

[11]:
```
def␣
↪num_cat_bi_grpby(df,colname,category,groupby,nrows=1,mcols=2,width=18,height=6):
↪
    fig , ax = plt.subplots(nrows,mcols,figsize=(width,height),squeeze=False)
    sns.set(style='white')
    fig.set_facecolor("lightgrey")
    rows = 0
    for var in colname:
        sns.boxplot(x = category,y = var,hue=groupby, data =␣
↪df,ax=ax[rows][0],palette="Set3")
        sns.
↪lineplot(x=df[category],y=df[var],ax=ax[rows][1],hue=df[groupby],palette="bright")
        ax[rows][0].set_ylabel(var, fontweight="bold",fontsize=14,family =␣
↪"Comic Sans MS")
        ax[rows][0].set_xlabel(category,fontweight="bold", fontsize=14,family =␣
↪"Comic Sans MS")
        ax[rows][0].legend(loc='lower right')
        ax[rows][1].set_ylabel(var, fontweight="bold",fontsize=14,family =␣
↪"Comic Sans MS")
        ax[rows][1].set_xlabel(category,fontweight="bold", fontsize=14,family =␣
↪"Comic Sans MS")
        rows += 1
    plt.show()
```

Function for Booststrapping technique to calculate the CI

[12]:
```
def bootstrapping(sample1,sample2,smp_siz=500,itr_size=5000,confidence_level=0.
↪95,no_of_tails=2):

    smp1_means_m = np.empty(itr_size)
    smp2_means_m = np.empty(itr_size)
    for i in range(itr_size):
        smp1_n = np.empty(smp_siz)
        smp2_n = np.empty(smp_siz)
        smp1_n = np.random.choice(sample1, size = smp_siz,replace=True)
        smp2_n = np.random.choice(sample2, size = smp_siz,replace=True)
        smp1_means_m[i] = np.mean(smp1_n)
        smp2_means_m[i] = np.mean(smp2_n)

    #Calcualte the Z-Critical value
    alpha = (1 - confidence_level)/no_of_tails
    z_critical = stats.norm.ppf(1 - alpha)

    # Calculate the mean, standard deviation & standard Error of sampling␣
↪distribution of a sample mean
```

```
    mean1  = np.mean(smp1_means_m)
    sigma1 = statistics.stdev(smp1_means_m)
    sem1   = stats.sem(smp1_means_m)

    lower_limit1 = mean1 - (z_critical * sigma1)
    upper_limit1 = mean1 + (z_critical * sigma1)

    # Calculate the mean, standard deviation & standard Error of sampling␣
↪distribution of a sample mean
    mean2  = np.mean(smp2_means_m)
    sigma2 = statistics.stdev(smp2_means_m)
    sem2   = stats.sem(smp2_means_m)

    lower_limit2 = mean2 - (z_critical * sigma2)
    upper_limit2 = mean2 + (z_critical * sigma2)

    fig, ax = plt.subplots(figsize=(14,6))
    sns.set_style("darkgrid")

    sns.kdeplot(data=smp1_means_m,color="#467821",fill=True,linewidth=2)
    sns.kdeplot(data=smp2_means_m,color='#e5ae38',fill=True,linewidth=2)

    label_mean1=("  (Males) :  {:.2f}".format(mean1))
    label_ult1=("Lower Limit(M):  {:.2f}\nUpper Limit(M):   {:.2f}".
↪format(lower_limit1,upper_limit1))
    label_mean2=("  (Females):  {:.2f}".format(mean2))
    label_ult2=("Lower Limit(F):  {:.2f}\nUpper Limit(F):   {:.2f}".
↪format(lower_limit2,upper_limit2))

    plt.title(f"Sample Size: {smp_siz}, Male Avg: {np.round(mean1, 2)}, Male␣
↪SME: {np.round(sem1,2)},Female Avg:{np.round(mean2, 2)}, Female SME: {np.
↪round(sem2,2)}",
              fontsize=14,family = "Comic Sans MS")
    plt.xlabel('Purchase')
    plt.axvline(mean1, color = 'y', linestyle = 'solid', linewidth =␣
↪2,label=label_mean1)
    plt.axvline(upper_limit1, color = 'r', linestyle = 'solid', linewidth =␣
↪2,label=label_ult1)
    plt.axvline(lower_limit1, color = 'r', linestyle = 'solid', linewidth = 2)
    plt.axvline(mean2, color = 'b', linestyle = 'dashdot', linewidth =␣
↪2,label=label_mean2)
    plt.axvline(upper_limit2, color = '#56B4E9', linestyle = 'dashdot',␣
↪linewidth = 2,label=label_ult2)
    plt.axvline(lower_limit2, color = '#56B4E9', linestyle = 'dashdot',␣
↪linewidth = 2)
    plt.legend(loc='upper right')
```

```
    plt.show()

    return smp1_means_m,smp2_means_m ,np.round(lower_limit1,2),np.
 ↪round(upper_limit1,2),np.round(lower_limit2,2),np.round(upper_limit2,2)
```

[13]:
```
def␣
 ↪bootstrapping_m_vs_um(sample1,sample2,smp_siz=500,itr_size=5000,confidence_level=0.
 ↪95,no_of_tails=2):

    smp1_means_m = np.empty(itr_size)
    smp2_means_m = np.empty(itr_size)
    for i in range(itr_size):
        smp1_n = np.empty(smp_siz)
        smp2_n = np.empty(smp_siz)
        smp1_n = np.random.choice(sample1, size = smp_siz,replace=True)
        smp2_n = np.random.choice(sample2, size = smp_siz,replace=True)
        smp1_means_m[i] = np.mean(smp1_n)
        smp2_means_m[i] = np.mean(smp2_n)

#     std_dev1 = np.std(sample1)
#     std_err1 = np.std(sample1,ddof=1)/np.sqrt(smp_siz)
#     std_dev2 = np.std(sample2)
#     std_err2 = np.std(sample2,ddof=1)/np.sqrt(smp_siz)

    #Calcualte the Z-Critical value
    alpha = (1 - confidence_level)/no_of_tails
    z_critical = stats.norm.ppf(1 - alpha)

    # Calculate the mean, standard deviation & standard Error of sampling␣
 ↪distribution of a sample mean
    mean1  = np.mean(smp1_means_m)
    sigma1 = statistics.stdev(smp1_means_m)
    sem1   = stats.sem(smp1_means_m)

    lower_limit1 = mean1 - (z_critical * sigma1)
    upper_limit1 = mean1 + (z_critical * sigma1)

    # Calculate the mean, standard deviation & standard Error of sampling␣
 ↪distribution of a sample mean
    mean2  = np.mean(smp2_means_m)
    sigma2 = statistics.stdev(smp2_means_m)
    sem2   = stats.sem(smp2_means_m)

#     print(smp_siz,std_dev1,std_err1,sem1)
#     print(smp_siz,std_dev2,std_err2,sem2)
```

```
    lower_limit2 = mean2 - (z_critical * sigma2)
    upper_limit2 = mean2 + (z_critical * sigma2)

    fig, ax = plt.subplots(figsize=(14,6))
    sns.set_style("darkgrid")

    sns.kdeplot(data=smp1_means_m,color="#467821",fill=True,linewidth=2)
    sns.kdeplot(data=smp2_means_m,color='#e5ae38',fill=True,linewidth=2)

    label_mean1=("  (Married) :  {:.2f}".format(mean1))
    label_ult1=("Lower Limit(M):  {:.2f}\nUpper Limit(M):   {:.2f}".
↪format(lower_limit1,upper_limit1))
    label_mean2=("  (Unmarried):  {:.2f}".format(mean2))
    label_ult2=("Lower Limit(F):  {:.2f}\nUpper Limit(F):   {:.2f}".
↪format(lower_limit2,upper_limit2))

    plt.title(f"Sample Size: {smp_siz}, Married Avg: {np.round(mean1, 2)},␣
↪Married SME: {np.round(sem1,2)},Unmarried Avg:{np.round(mean2, 2)},␣
↪Unmarried SME: {np.round(sem2,2)}",
              fontsize=14,family = "Comic Sans MS")
    plt.xlabel('Purchase')
    plt.axvline(mean1, color = 'y', linestyle = 'solid', linewidth =␣
↪2,label=label_mean1)
    plt.axvline(upper_limit1, color = 'r', linestyle = 'solid', linewidth =␣
↪2,label=label_ult1)
    plt.axvline(lower_limit1, color = 'r', linestyle = 'solid', linewidth = 2)
    plt.axvline(mean2, color = 'b', linestyle = 'dashdot', linewidth =␣
↪2,label=label_mean2)
    plt.axvline(upper_limit2, color = '#56B4E9', linestyle = 'dashdot',␣
↪linewidth = 2,label=label_ult2)
    plt.axvline(lower_limit2, color = '#56B4E9', linestyle = 'dashdot',␣
↪linewidth = 2)
    plt.legend(loc='upper right')

    plt.show()

    return smp1_means_m,smp2_means_m ,np.round(lower_limit1,2),np.
↪round(upper_limit1,2),np.round(lower_limit2,2),np.round(upper_limit2,2)
```

[14]:
```
def bootstrapping_age(sample,smp_siz=500,itr_size=5000,confidence_level=0.
↪95,no_of_tails=2):

    smp_means_m = np.empty(itr_size)
    for i in range(itr_size):
        smp_n = np.empty(smp_siz)
        smp_n = np.random.choice(sample, size = smp_siz,replace=True)
```

```
        smp_means_m[i] = np.mean(smp_n)

    #Calcualte the Z-Critical value
    alpha = (1 - confidence_level)/no_of_tails
    z_critical = stats.norm.ppf(1 - alpha)

    # Calculate the mean, standard deviation & standard Error of sampling␣
↪distribution of a sample mean
    mean  = np.mean(smp_means_m)
    sigma = statistics.stdev(smp_means_m)
    sem   = stats.sem(smp_means_m)

    lower_limit = mean - (z_critical * sigma)
    upper_limit = mean + (z_critical * sigma)

    fig, ax = plt.subplots(figsize=(14,6))
    sns.set_style("darkgrid")

    sns.kdeplot(data=smp_means_m,color="#7A68A6",fill=True,linewidth=2)

    label_mean=(" :  {:.2f}".format(mean))
    label_ult=("Lower Limit:  {:.2f}\nUpper Limit:   {:.2f}".
↪format(lower_limit,upper_limit))

    plt.title(f"Sample Size: {smp_siz},Mean:{np.round(mean,2)}, SME:{np.
↪round(sem,2)}",fontsize=14,family="Comic Sans MS")
    plt.xlabel('Purchase')
    plt.axvline(mean, color = 'y', linestyle = 'solid', linewidth =␣
↪2,label=label_mean)
    plt.axvline(upper_limit, color = 'r', linestyle = 'solid', linewidth =␣
↪2,label=label_ult)
    plt.axvline(lower_limit, color = 'r', linestyle = 'solid', linewidth = 2)
    plt.legend(loc='upper right')

    plt.show()

    return smp_means_m ,np.round(lower_limit,2),np.round(upper_limit,2)
```

Exploratory Data Analysis

Loading and inspecting the Dataset

Loading the csv file

[21]:
```
retail_data = pd.read_csv("https://d2beiqkhq929f0.cloudfront.net/public_assets/
↪assets/000/001/293/original/walmart_data.csv?1641285094")
```

[22]:
```
retail_data.head()
```

[22]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status	Product_Category	Purchase
0	1000001	P00069042	F	0-17	10	A	2	0	3	8370
1	1000001	P00248942	F	0-17	10	A	2	0	1	15200
2	1000001	P00087842	F	0-17	10	A	2	0	12	1422
3	1000001	P00085442	F	0-17	10	A	2	0	12	1057
4	1000002	P00285442	M	55+	16	C	4+	0	8	7969

Checking Shape and Column names

[23]:
```
retail_data.shape
```

[23]:
```
(550068, 10)
```

[24]:
```
retail_data.columns
```

[24]:
```
Index(['User_ID', 'Product_ID', 'Gender', 'Age', 'Occupation', 'City_Category',
       'Stay_In_Current_City_Years', 'Marital_Status', 'Product_Category',
       'Purchase'],
      dtype='object')
```

Validating Duplicate Records

[25]:
```
retail_data.duplicated().sum()
```

[25]:
```
0
```

Inference Missing Data Analysis

[26]:
```
missingValue(retail_data).head(5)
```

```
Total records =  550068
```

[26]:

	Total Missing	In Percent
User_ID	0	0.0
Product_ID	0	0.0
Gender	0	0.0
Age	0	0.0
Occupation	0	0.0

Unique values (counts) for each Feature

[28]:
```
retail_data.nunique()
```

[28]:
```
User_ID                       5891
Product_ID                    3631
Gender                           2
Age                              7
Occupation                      21
City_Category                    3
Stay_In_Current_City_Years       5
Marital_Status                   2
Product_Category                20
Purchase                     18105
dtype: int64
```

The total number of records exceeds five million but the UserIDs are only 5891, meaning customers have visited multiple times in order to buy products.

A deep dive into User ID Based on 5891 user IDs, how many are married, male or female, or the age of the users.

[29]:
```
retail_data.groupby(['Gender'])['User_ID'].nunique()
```

[29]:
```
Gender
F    1666
M    4225
Name: User_ID, dtype: int64
```

[30]:
```
print("Females are ", 1666/5891)
print("Females are ", 4225/5891)
```

```
Females are  0.2828042777117637
Females are  0.7171957222882362
```

The percentage of women customers is only 28%

Around 72% of customers are male

[31]:
```
retail_data.groupby(['Age'])['User_ID'].nunique()
```

[31]:
```
Age
0-17      218
18-25    1069
26-35    2053
36-45    1167
46-50     531
51-55     481
55+       372
Name: User_ID, dtype: int64
```

[32]:
```
retail_data.groupby(['City_Category'])['User_ID'].nunique()
```

[32]:
```
City_Category
A    1045
B    1707
C    3139
Name: User_ID, dtype: int64
```

[33]:
```
retail_data.groupby(['Stay_In_Current_City_Years'])['User_ID'].nunique()
```

[33]:
```
Stay_In_Current_City_Years
0      772
1     2086
2     1145
3      979
4+     909
Name: User_ID, dtype: int64
```

[34]:
```
retail_data.groupby(['Marital_Status'])['User_ID'].nunique()
```

[34]:
```
Marital_Status
0    3417
1    2474
Name: User_ID, dtype: int64
```

Unique values (names) are checked for each Features

[35]:
```
colname =␣
  ↪['Gender','Age','City_Category','Stay_In_Current_City_Years','Marital_Status','Product_Cate
for col in colname:
    print("\nUnique values of ",col," are : ",list(retail_data[col].unique()))
```

```
Unique values of  Gender  are :  ['F', 'M']

Unique values of  Age  are :  ['0-17', '55+', '26-35', '46-50', '51-55',
'36-45', '18-25']

Unique values of  City_Category  are :  ['A', 'C', 'B']

Unique values of  Stay_In_Current_City_Years  are :  ['2', '4+', '3', '1', '0']

Unique values of  Marital_Status  are :  [0, 1]

Unique values of  Product_Category  are :  [3, 1, 12, 8, 5, 4, 2, 6, 14, 11, 13,
15, 7, 16, 18, 10, 17, 9, 20, 19]

Unique values of  Occupation  are :  [10, 16, 15, 7, 20, 9, 1, 12, 17, 0, 3, 4,
11, 8, 19, 2, 18, 5, 14, 13, 6]
```

All the values looks good.

DataType Validation

[36]:
```
retail_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 550068 entries, 0 to 550067
Data columns (total 10 columns):
 #   Column                      Non-Null Count   Dtype
---  ------                      --------------   -----
 0   User_ID                     550068 non-null  int64
 1   Product_ID                  550068 non-null  object
 2   Gender                      550068 non-null  object
 3   Age                         550068 non-null  object
 4   Occupation                  550068 non-null  int64
 5   City_Category               550068 non-null  object
 6   Stay_In_Current_City_Years  550068 non-null  object
 7   Marital_Status              550068 non-null  int64
 8   Product_Category            550068 non-null  int64
 9   Purchase                    550068 non-null  int64
dtypes: int64(5), object(5)
memory usage: 42.0+ MB
```

'User_ID','Product_ID','Gender', 'Age','City_Category','Marital_Status' are categorical variables. As a result, we need to change the datatype to category.

[37]:
```
cols = ['User_ID','Product_ID','Gender', 'Age','City_Category','Marital_Status']
for col_name in cols:
    retail_data[col_name] = retail_data[col_name].astype("category")
```

[38]:
```
retail_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 550068 entries, 0 to 550067
Data columns (total 10 columns):
 #   Column                      Non-Null Count   Dtype
---  ------                      --------------   -----
 0   User_ID                     550068 non-null  category
 1   Product_ID                  550068 non-null  category
 2   Gender                      550068 non-null  category
 3   Age                         550068 non-null  category
 4   Occupation                  550068 non-null  int64
 5   City_Category               550068 non-null  category
 6   Stay_In_Current_City_Years  550068 non-null  object
 7   Marital_Status              550068 non-null  category
 8   Product_Category            550068 non-null  int64
 9   Purchase                    550068 non-null  int64
dtypes: category(6), int64(3), object(1)
```

```
memory usage: 21.3+ MB
```

Basic Statistics Analysis - count, min, max, and mean

[39]:
```
retail_data.describe().T
```

[39]:
```
                     count          mean           std    min     25%     50%  \
Occupation        550068.0      8.076707      6.522660    0.0     2.0     7.0
Product_Category  550068.0      5.404270      3.936211    1.0     1.0     5.0
Purchase          550068.0   9263.968713   5023.065394   12.0  5823.0  8047.0

                     75%      max
Occupation          14.0     20.0
Product_Category     8.0     20.0
Purchase         12054.0  23961.0
```

[40]:
```
retail_data.describe(include=['object','category']).T
```

[40]:
```
                            count unique        top    freq
User_ID                    550068   5891    1001680    1026
Product_ID                 550068   3631  P00265242    1880
Gender                     550068      2          M  414259
Age                        550068      7      26-35  219587
City_Category              550068      3          B  231173
Stay_In_Current_City_Years 550068      5          1  193821
Marital_Status             550068      2          0  324731
```

[41]:
```
retail_data.groupby(['Gender'])['Purchase'].describe()
```

[41]:
```
            count          mean          std    min     25%     50%      75%  \
Gender
F        135809.0  8734.565765  4767.233289   12.0  5433.0  7914.0  11400.0
M        414259.0  9437.526040  5092.186210   12.0  5863.0  8098.0  12454.0

             max
Gender
F        23959.0
M        23961.0
```

[42]:
```
retail_data.groupby(['Marital_Status'])['Purchase'].describe()
```

[42]:
```
                    count          mean          std    min     25%     50%  \
Marital_Status
0                324731.0  9265.907619  5027.347859   12.0  5605.0  8044.0
1                225337.0  9261.174574  5016.897378   12.0  5843.0  8051.0

                    75%      max
Marital_Status
```

```
0               12061.0  23961.0
1               12042.0  23961.0
```

[43]:
```
retail_data.groupby(['Age'])['Purchase'].describe()
```

[43]:
```
          count         mean          std   min     25%     50%      75%  \
Age
0-17    15102.0  8933.464640  5111.114046  12.0  5328.0  7986.0  11874.0
18-25   99660.0  9169.663606  5034.321997  12.0  5415.0  8027.0  12028.0
26-35  219587.0  9252.690633  5010.527303  12.0  5475.0  8030.0  12047.0
36-45  110013.0  9331.350695  5022.923879  12.0  5876.0  8061.0  12107.0
46-50   45701.0  9208.625697  4967.216367  12.0  5888.0  8036.0  11997.0
51-55   38501.0  9534.808031  5087.368080  12.0  6017.0  8130.0  12462.0
55+     21504.0  9336.280459  5011.493996  12.0  6018.0  8105.5  11932.0

           max
Age
0-17   23955.0
18-25  23958.0
26-35  23961.0
36-45  23960.0
46-50  23960.0
51-55  23960.0
55+    23960.0
```

[44]:
```
retail_data.groupby(['City_Category'])['Purchase'].describe()
```

[44]:
```
                  count         mean          std   min     25%     50%  \
City_Category
A              147720.0  8911.939216  4892.115238  12.0  5403.0  7931.0
B              231173.0  9151.300563  4955.496566  12.0  5460.0  8005.0
C              171175.0  9719.920993  5189.465121  12.0  6031.5  8585.0

                   75%      max
City_Category
A              11786.0  23961.0
B              11986.0  23960.0
C              13197.0  23961.0
```

[45]:
```
retail_data.groupby(['City_Category'])['User_ID'].nunique()
```

[45]:
```
City_Category
A    1045
B    1707
C    3139
Name: User_ID, dtype: int64
```

There are more single people than married people.

Most mall customers are between the ages of 26 and 35.

The majority of our customers come from city category B but customers come from City category C spent more as mean is 9719.

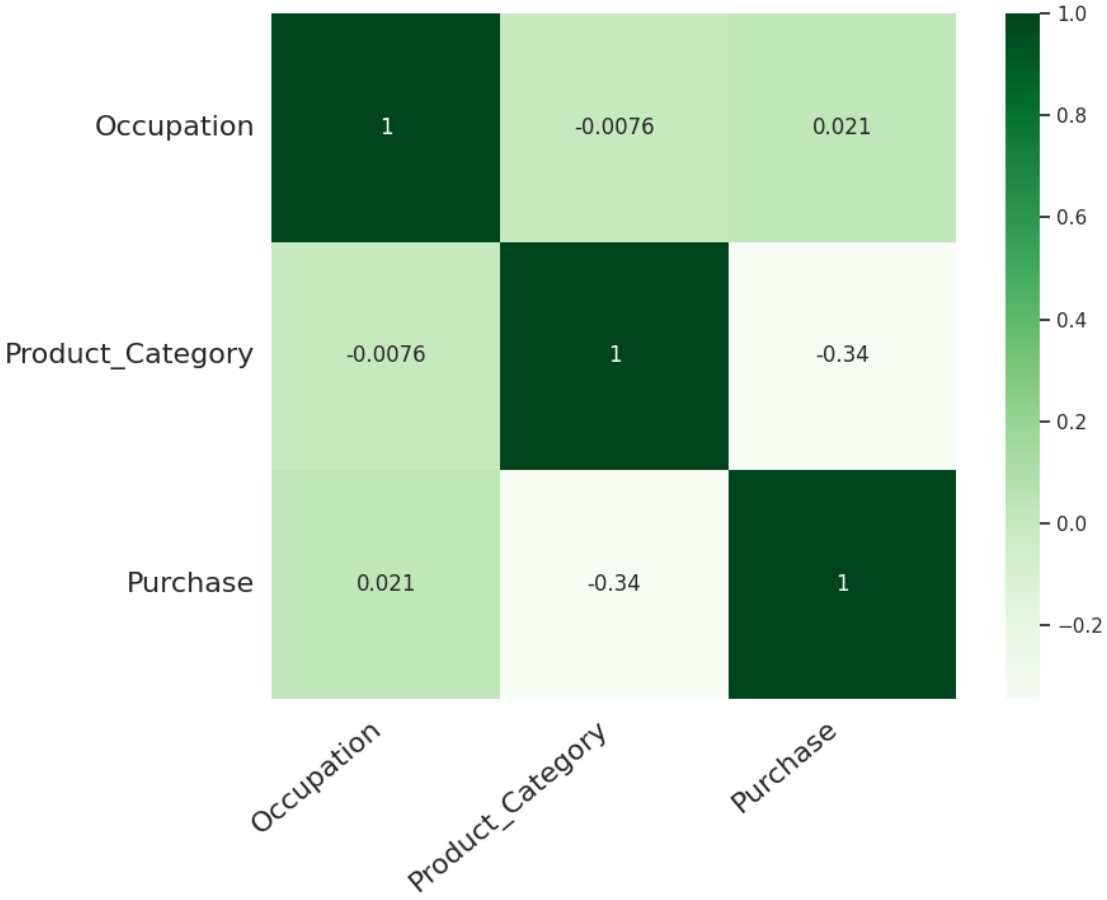
Male customers tend to spend more than female customers, as the mean is higher for male customers.

The majority of users come from City Category C, but more people from City Category B tend to purchase, which suggests the same users visit the mall multiple times in City Category B.

Correlation Analysis

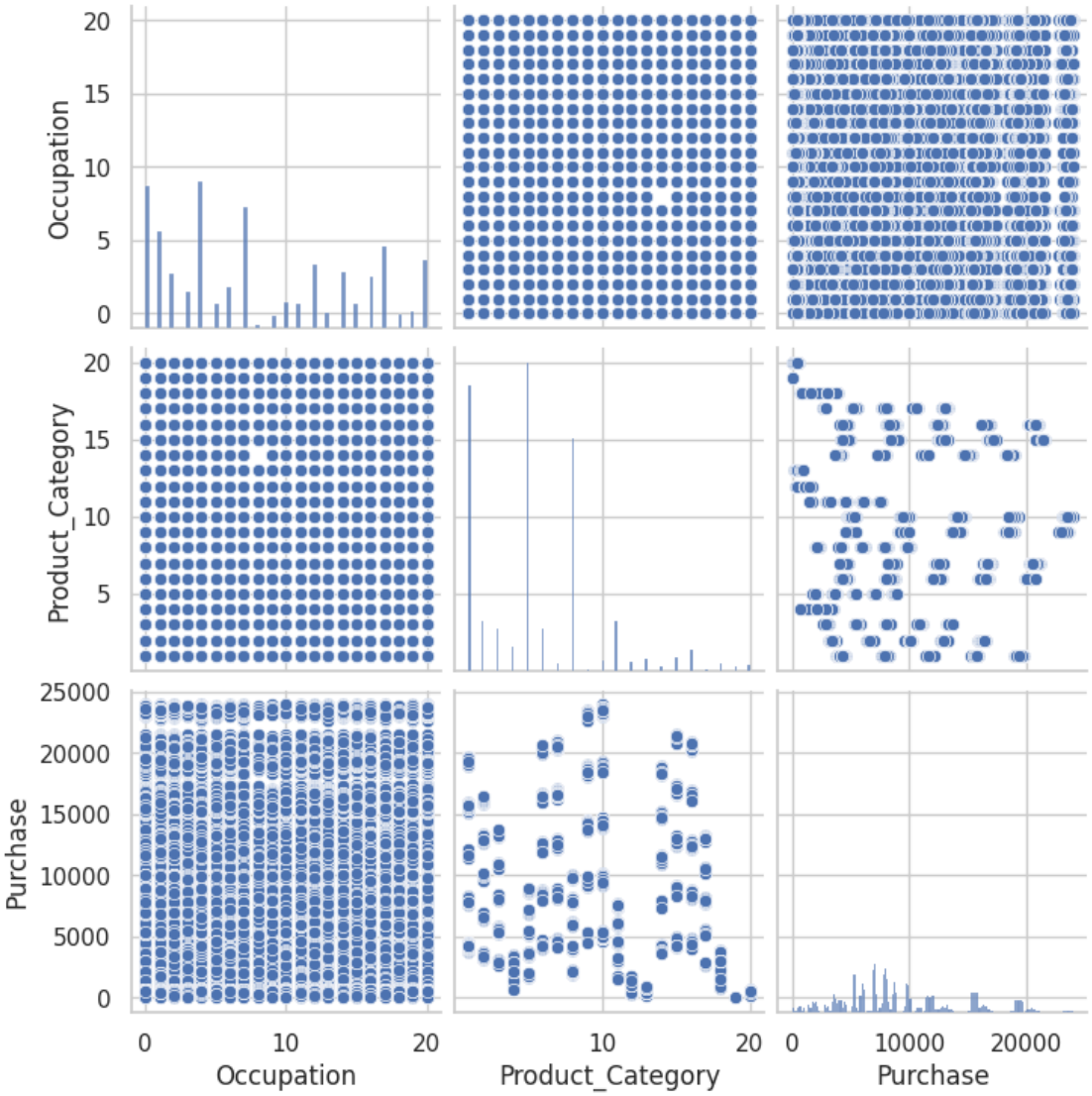
[46]:
```
plt.figure(figsize = (10, 7))
ax = sns.heatmap(retail_data.corr(),
            annot=True,cmap='Greens',square=True)

ax.set_xticklabels(
    ax.get_xticklabels(),
    rotation=40,fontsize=16,family = "Comic Sans MS",
    horizontalalignment='right')

ax.set_yticklabels(
    ax.get_yticklabels(),
    rotation=0,fontsize=16,family = "Comic Sans MS",
    horizontalalignment='right')

plt.show()
```

```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```



[47]:
```
sns.pairplot(retail_data)
```

[47]: <seaborn.axisgrid.PairGrid at 0x7f4ddb40d5d0>



[ ]:

Mostly features are categorical and not much correlation can be observed from above graphs.

Numerical Variables Categorial variables Numerical Variables - Outlier detection

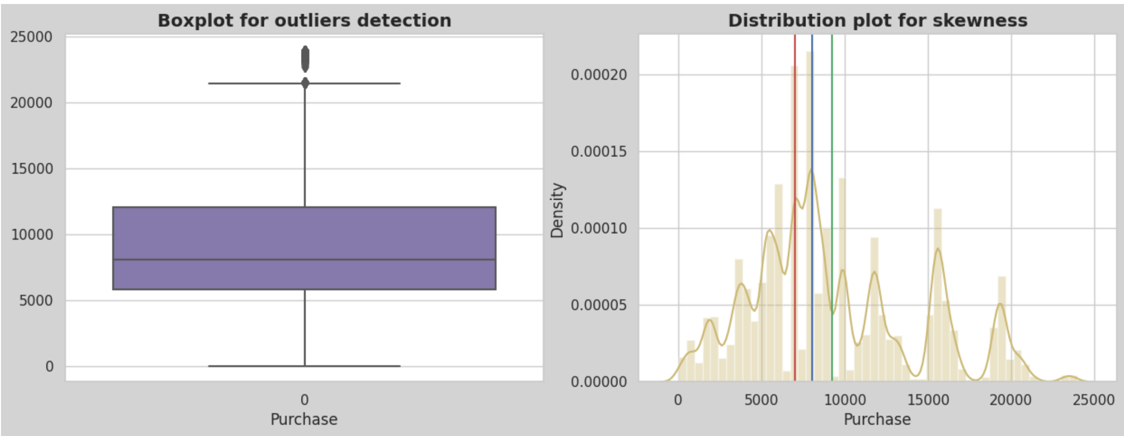
[48]:
```
retail_data.columns
```

[48]:
```
Index(['User_ID', 'Product_ID', 'Gender', 'Age', 'Occupation', 'City_Category',
       'Stay_In_Current_City_Years', 'Marital_Status', 'Product_Category',
       'Purchase'],
      dtype='object')
```

[49]:

```
# Visualizing our dependent variable for Outliers and Skewness
fig = plt.figure(figsize=(15,5))
fig.set_facecolor("lightgrey")

plt.subplot(1,2,1)
sns.boxplot(retail_data["Purchase"],color='m')
plt.title("Boxplot for outliers detection", fontweight="bold",fontsize=14)
plt.xlabel('Purchase', fontsize=12,family = "Comic Sans MS")

plt.subplot(1,2,2)
sns.distplot(retail_data["Purchase"],color='y')

plt.title("Distribution plot for skewness", fontweight="bold",fontsize=14)
plt.ylabel('Density', fontsize=12,family = "Comic Sans MS")
plt.xlabel('Purchase', fontsize=12,family = "Comic Sans MS")
plt.axvline(retail_data["Purchase"].mean(),color="g")
plt.axvline(retail_data["Purchase"].median(),color="b")
plt.axvline(retail_data["Purchase"].mode()[0],color="r")

plt.show()
```

```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```



Above graphs ;ooks like "right-skewed distribution" which means the mass of the distribution is concentrated on the left of the figure.

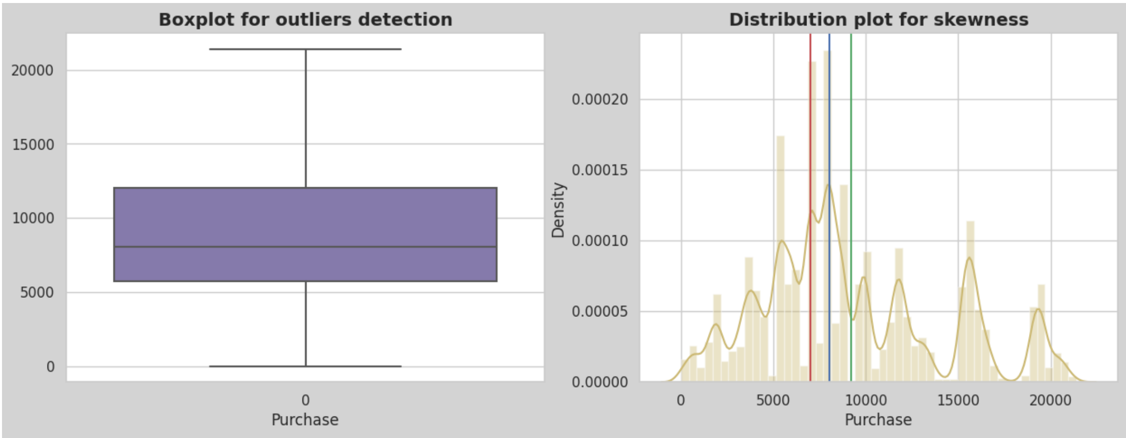
Majority of Customers purchase within the 5,000 - 20,000 range.

Handling outliers¶

[50]:
```
retail_data_v1 = retail_data.copy()
```

[51]:
```
#Outlier Treatment: Remove top 5% & bottom 1% of the Column Outlier values
Q3 = retail_data_v1['Purchase'].quantile(0.75)
Q1 = retail_data_v1['Purchase'].quantile(0.25)
IQR = Q3-Q1
retail_data_v1 = retail_data_v1[(retail_data_v1['Purchase'] > Q1 - 1.5*IQR) &
 ↪(retail_data_v1['Purchase'] < Q3 + 1.5*IQR)]

# Visualizing our dependent variable for Outliers and Skewness
fig = plt.figure(figsize=(15,5))
fig.set_facecolor("lightgrey")

plt.subplot(1,2,1)
sns.boxplot(retail_data_v1["Purchase"],color='m')
plt.title("Boxplot for outliers detection", fontweight="bold",fontsize=14)
plt.xlabel('Purchase', fontsize=12,family = "Comic Sans MS")

plt.subplot(1,2,2)
sns.distplot(retail_data_v1["Purchase"],color='y')

plt.title("Distribution plot for skewness", fontweight="bold",fontsize=14)
plt.ylabel('Density', fontsize=12,family = "Comic Sans MS")
plt.xlabel('Purchase', fontsize=12,family = "Comic Sans MS")
plt.axvline(retail_data_v1["Purchase"].mean(),color="g")
plt.axvline(retail_data_v1["Purchase"].median(),color="b")
plt.axvline(retail_data_v1["Purchase"].mode()[0],color="r")

plt.show()
```

```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```



Categorical variable Uni-variante Analysis

[52]:
```
cat_colnames =␣
↪['Gender','Age','City_Category','Stay_In_Current_City_Years','Marital_Status']
cat_analysis(retail_data_v1,cat_colnames,5,2,12,32)
```

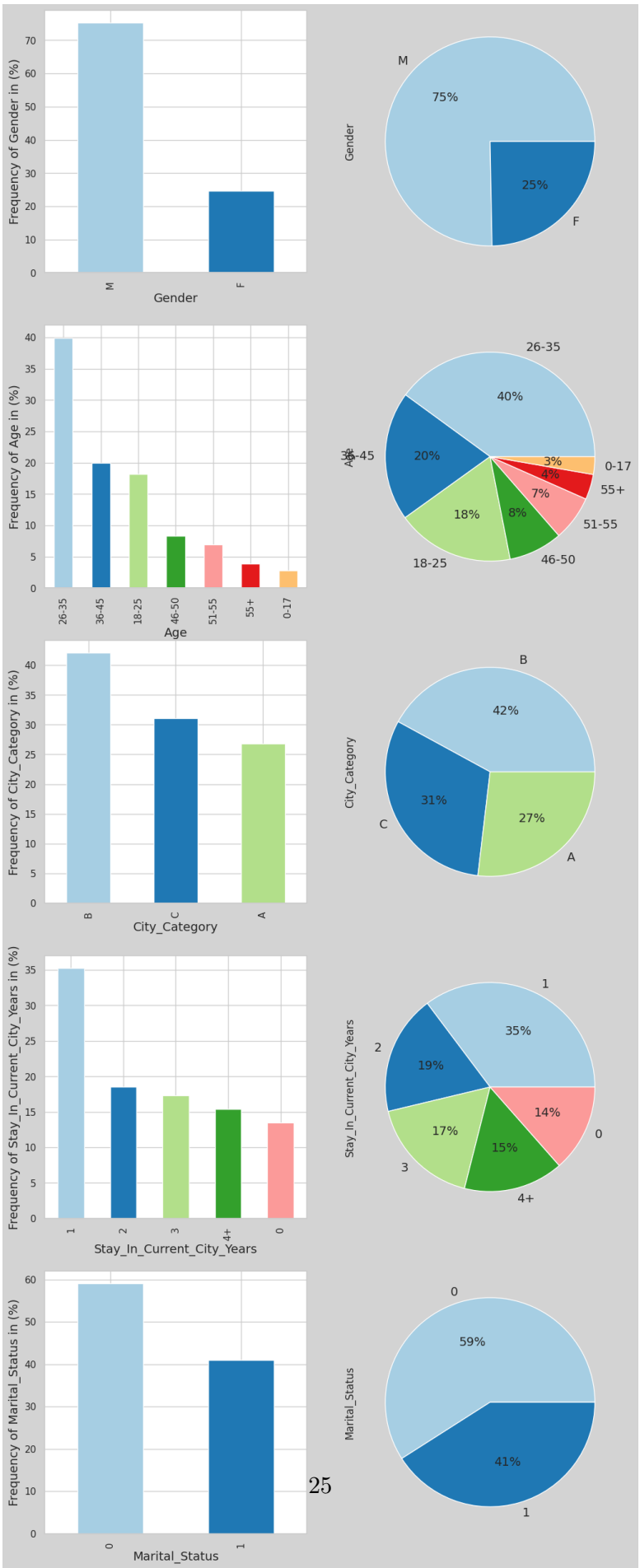
```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```

```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```

```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```

```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```

```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```


Frequency of Gender in (%)
Gender
M
F
75%
25%
Frequency of Age in (%)
Age
26-35
36-45
18-25
46-50
51-55
55+
0-17
40%
20%
18%
8%
7%
4%
3%
Frequency of City_Category in (%)
City_Category
B
C
A
42%
31%
27%
Frequency of Stay_In_Current_City_Years in (%)
Stay_In_Current_City_Years
1
2
3
4+
0
35%
19%
17%
15%
14%
Frequency of Marital_Status in (%)
Marital_Status
0
1
59%
41%

Males clearly purchase more than females. 75% of men and only 25% of women purchase products.

60% of purchases are made by people between the ages of 26 and 45

City Category B accounts for 42%, City Category C 31%, and City Category A represents 27% of all customer purchases.

[53]:
```
retail_data.columns
```

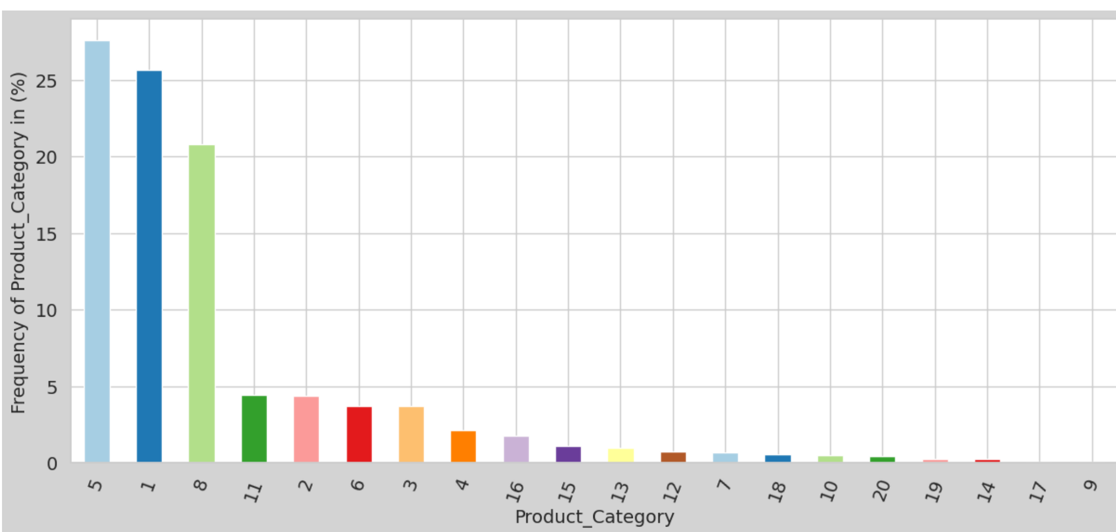
[53]:
```
Index(['User_ID', 'Product_ID', 'Gender', 'Age', 'Occupation', 'City_Category',
       'Stay_In_Current_City_Years', 'Marital_Status', 'Product_Category',
       'Purchase'],
      dtype='object')
```

[54]:
```
bar_plot_percentage(retail_data_v1,['Product_Category'])
```

```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```

```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```

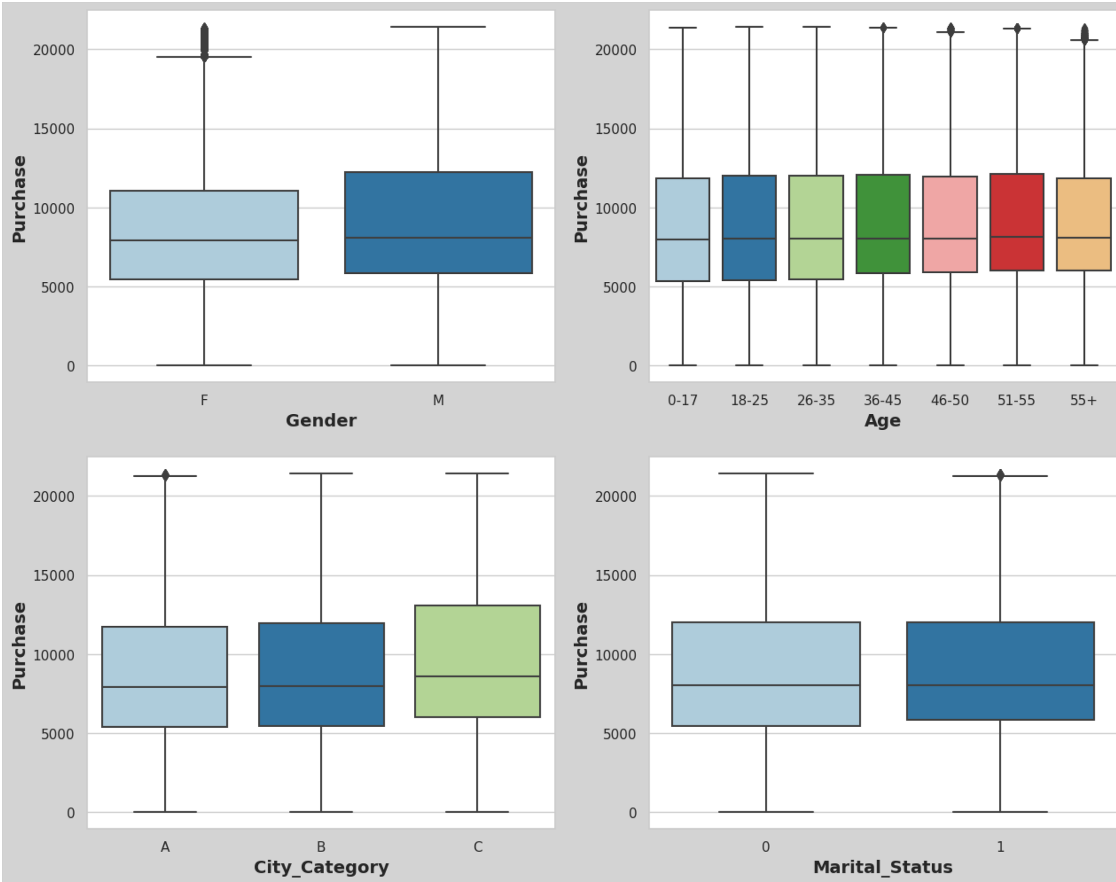
```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```



Product Category 5,1 & 8 are the products that customers buy the most.

[55]:
```
col_cat = ['Gender', 'Age','City_Category','Marital_Status']
num_cat_bi(retail_data_v1,col_cat,'Purchase',2,2,15,12)
```

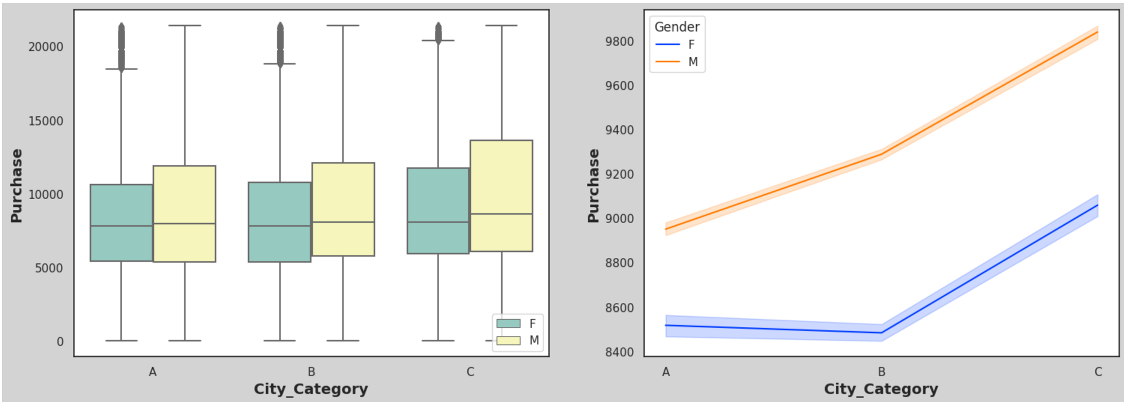
```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```



[56]:

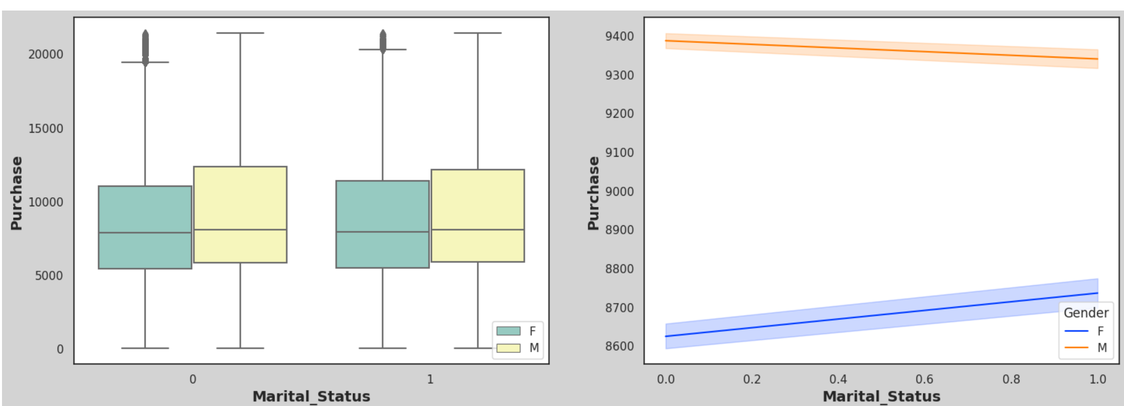
```
col_num = [ 'Purchase']
num_cat_bi_grpby(retail_data_v1,col_num,"City_Category",'Gender')
```

```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```




```
[57]: col_num = [ 'Purchase']
num_cat_bi_grpby(retail_data_v1,col_num,"Marital_Status",'Gender')
```

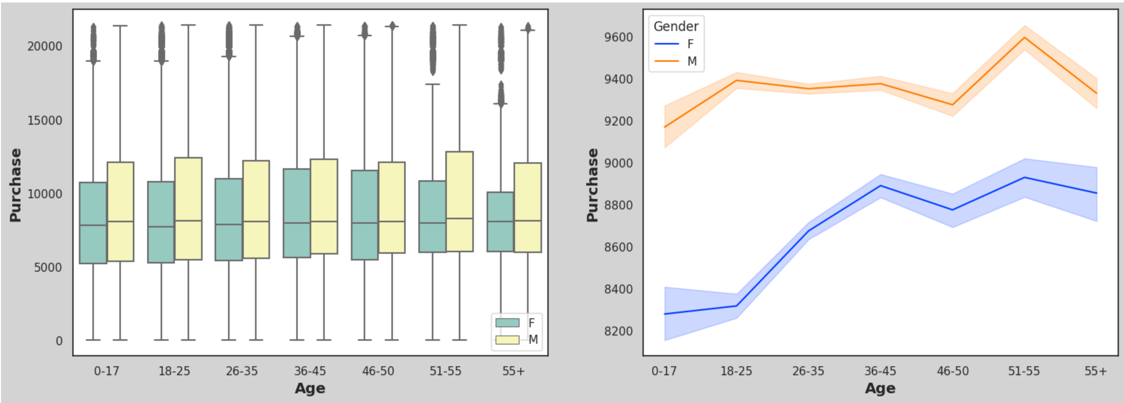
```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```




```
[58]: col_num = [ 'Purchase']
num_cat_bi_grpby(retail_data_v1,col_num,"Age",'Gender')
```

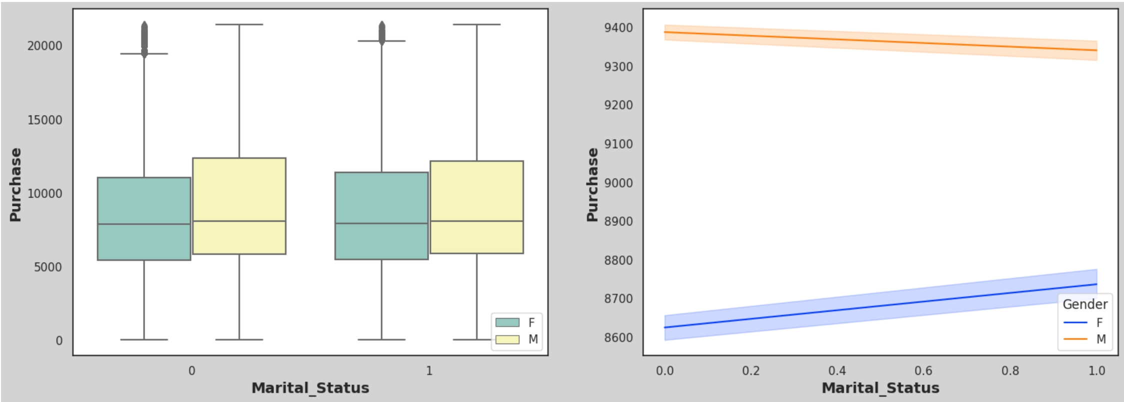
```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```

```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```



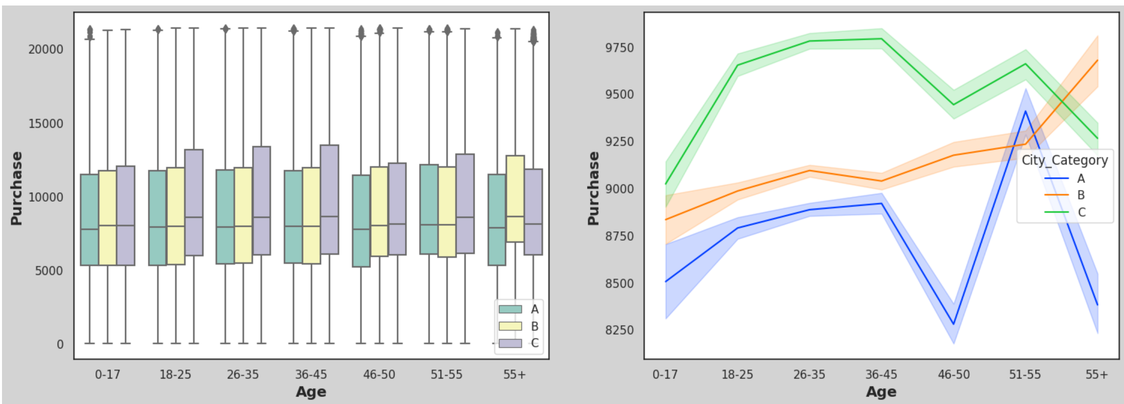
[59]:
```
col_num = [ 'Purchase']
num_cat_bi_grpby(retail_data_v1,col_num,"Marital_Status",'Gender')
```

```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```



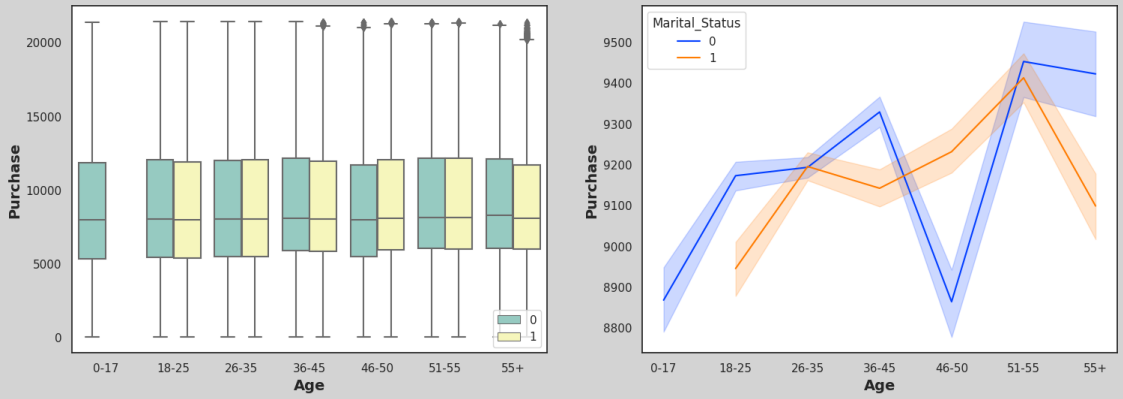
[60]:
```
col_num = [ 'Purchase']
num_cat_bi_grpby(retail_data_v1,col_num,"Age",'City_Category')
```

```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```



[61]:
```
col_num = [ 'Purchase']
num_cat_bi_grpby(retail_data_v1,col_num,"Age",'Marital_Status')
```

```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```



Purchases are high in city category C

Purchase is the same for all age groups

Most of the customers are 55+ and live in city category B

City category C has more customers between the ages of 18 and 45.

[62]:
```
bar_M_vs_F('Product_Category')
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-62-c9ccf65193a2> in <cell line: 1>()
----> 1 bar_M_vs_F('Product_Category')

<ipython-input-10-3e0bec5d4a6d> in bar_M_vs_F(colname)
     13     m_f_status = pd.concat([female,male],axis=0)
     14
---> 15     ax = sns.
 ↪barplot("index","percentage",data=m_f_status,hue="legends",palette="Blues_d")
     16     plt.xlabel(colname)
     17     fig.set_facecolor("white")

TypeError: barplot() got multiple values for argument 'data'
```

```
<Figure size 1600x600 with 0 Axes>
```

[ ]:
```
bar_M_vs_F('City_Category')
```

[ ]:
```
bar_M_vs_F('Age')
```

[ ]:
```
Product 5 and 8 is common among females.

In City Category C, there are slightly more female customers.
```

Product 5 and 8 is common among females.

In City Category C, there are slightly more female customers.

[ ]:
```
print(retail_data_v1.groupby(['Gender','City_Category'])['User_ID'].count())
```

[ ]:
```
fig = plt.figure(figsize=(25,10))
fig.set_facecolor("lightgrey")
sns.set(style='dark')
sns.displot(x= 'Purchase',data=retail_data_v1,hue='Gender',bins=25)
plt.show()
```

[ ]:
```
retail_data_v1.sample(500,replace=True).groupby(['Gender'])['Purchase'].
↪describe()
```

[ ]:
```
Even the sample mean shows that males spend more than females.
```

[63]:
```
retail_data_v1.groupby(['Gender'])['Purchase'].describe()
```

[63]:
```
            count          mean           std    min     25%     50%      75%  \
Gender
F       135220.0  8671.049039  4679.058483   12.0  5429.0  7906.0  11064.0
M       412171.0  9367.724355  5009.234088   12.0  5852.0  8089.0  12247.0

             max
Gender
F        21398.0
M        21399.0
```

[64]:
```
retail_data_smp_male = retail_data_v1[retail_data_v1['Gender'] ==␣
↪'M']['Purchase']
retail_data_smp_female = retail_data_v1[retail_data_v1['Gender'] ==␣
↪'F']['Purchase']
```

Given the sample size of 5.4 Million data for customer purhase history with 1.3M Females and 4.1 Males

Assumptions

Randomization: The data must be sampled randomly such that every member in a population has an equal probability of being selected to be in the sample.

Independence: The sample values must be independent of each other.

The 10% Condition: When the sample is drawn without replacement, the sample size should be no larger than 10% of the population.

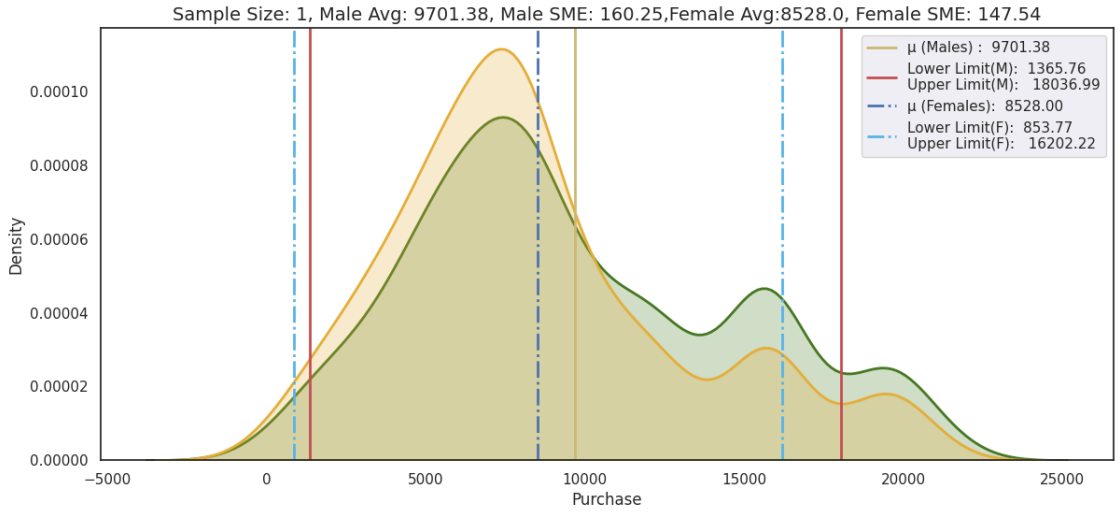
Large Sample Condition: The sample size needs to be sufficiently large.

CLT Analysis for mean purchase with confidence 90% - Based on Gender

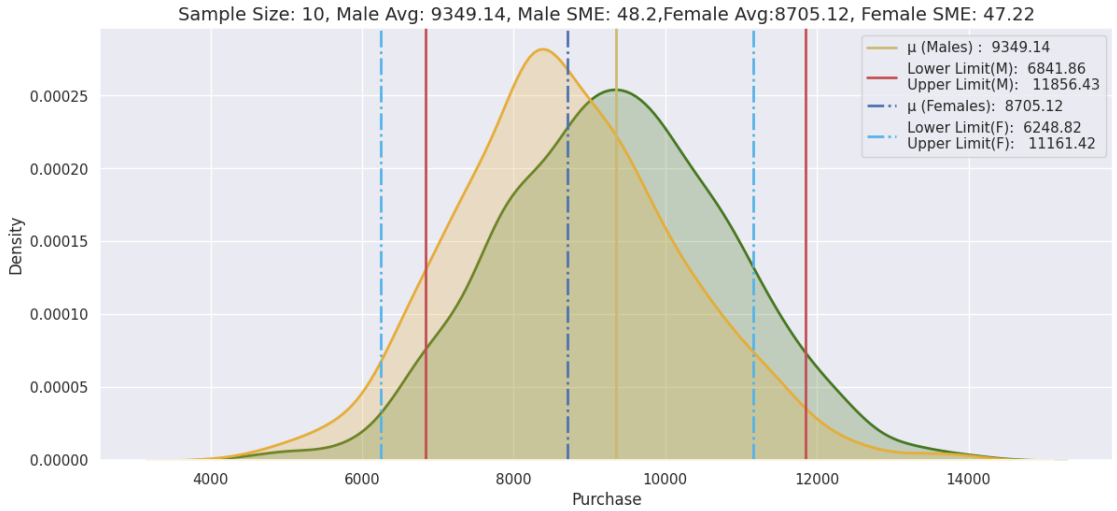
Analysis of the true mean of purchase values by gender with a 90% confidence

[65]:
```
itr_size = 1000
size_list = [1, 10, 30, 300, 1000, 100000]
ci = 0.90

array = np.empty((0,7))

for smp_siz in size_list:
    m_avg, f_avg, ll_m, ul_m, ll_f, ul_f =␣
 ↪bootstrapping(retail_data_smp_male,retail_data_smp_female,smp_siz,itr_size,ci)

    array = np.append(array, np.array([['M', ll_m, ul_m, smp_siz, ([ll_m,ul_m])␣
 ↪,(ul_m-ll_m),90]]), axis=0)
    array = np.append(array, np.array([['F', ll_f, ul_f, smp_siz, ([ll_f,ul_f])␣
 ↪,(ul_f-ll_f),90]]), axis=0)

overlap = pd.DataFrame(array, columns =␣
 ↪['Gender','Lower_limit','Upper_limit','Sample_Size','CI','Range','Confidence_pct'])
print()
```

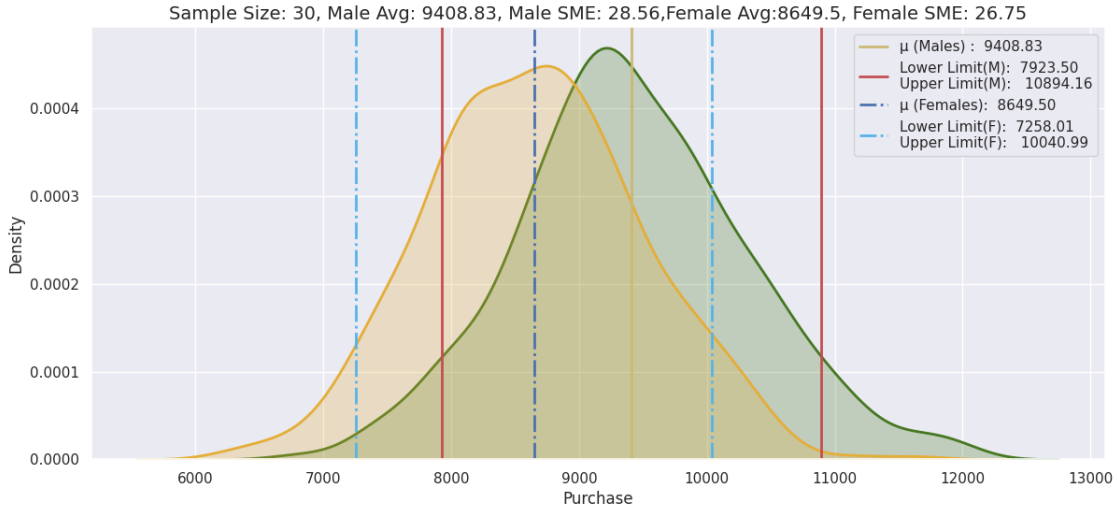
```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```



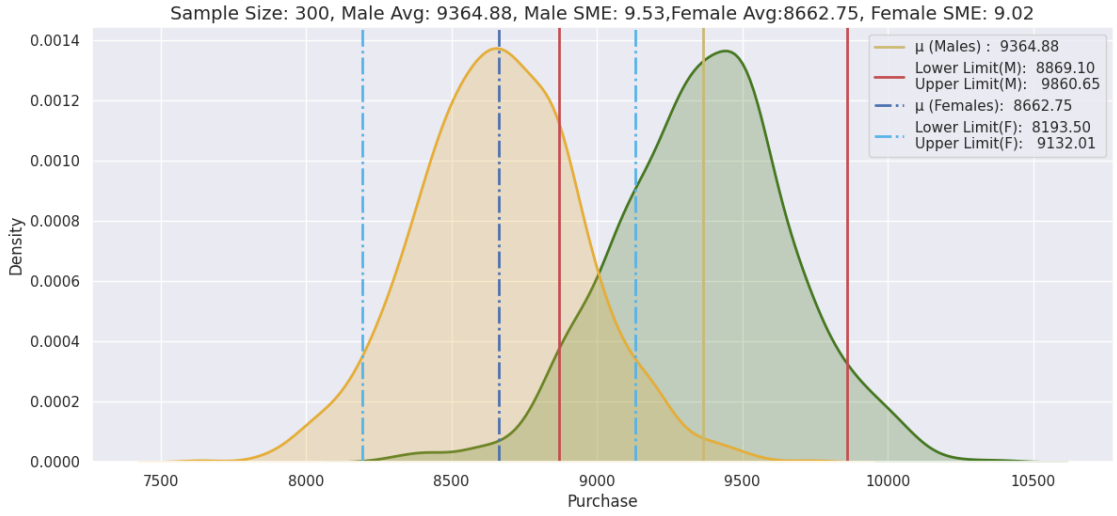

```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```



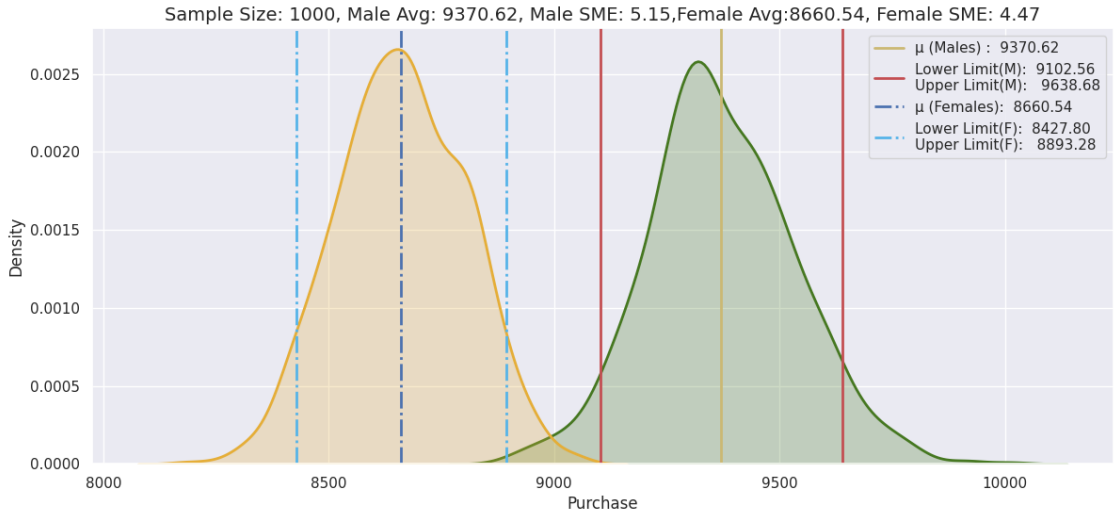

```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```



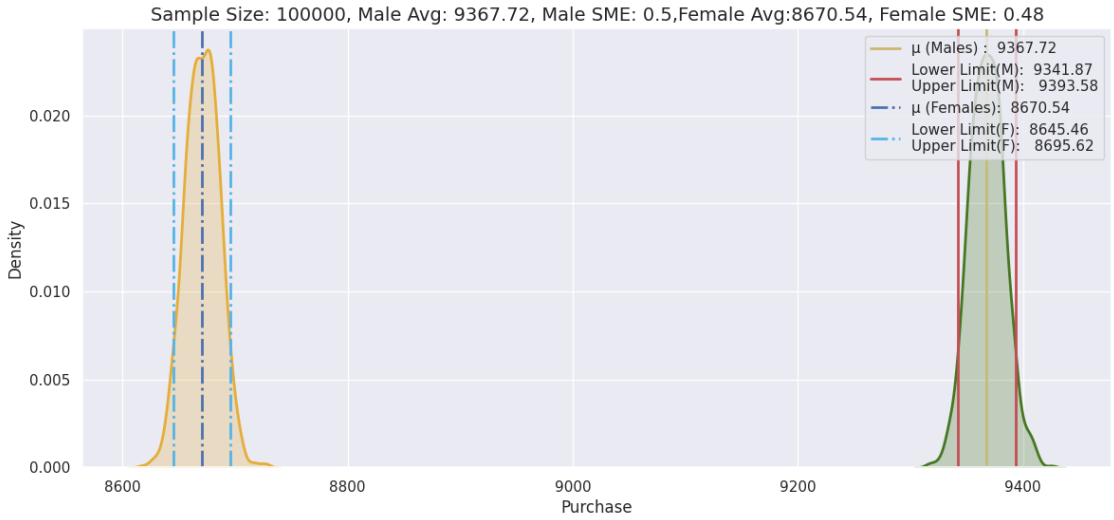

```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```




```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```




```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```



[66]:
```
overlap.loc[(overlap['Gender'] == 'F') & (overlap['Sample_Size'] >= 300)]
```

[66]:

	Gender	Lower_limit	Upper_limit	Sample_Size	CI	Range	Confidence_pct
7	F	8193.5	9132.01	300	[8193.5, 9132.01]	938.51	90
9	F	8427.8	8893.28	1000	[8427.8, 8893.28]	465.48	90
11	F	8645.46	8695.62	100000	[8645.46, 8695.62]	50.16	90

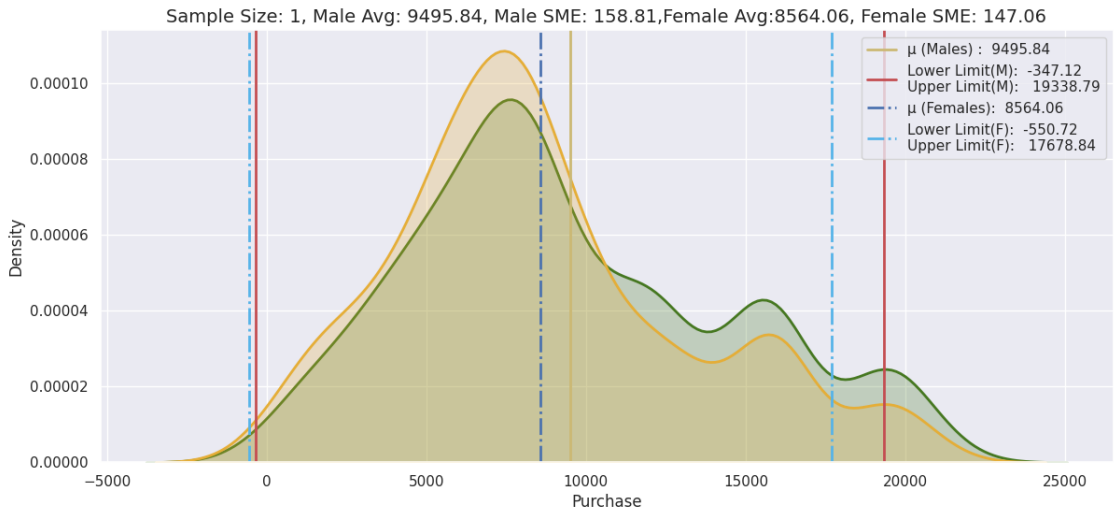
[ ]:
```
As the sample size increases, the two groups start to become distinct

With increasing sample size, Standard error of the mean in the samples␣
 ↪decreases.

For sample size 100000 is 0.49

For Female (sample size 100000) range for mean purchase with confidence␣
 ↪interval 90% is [8645.68, 8696.14]

For Male range for mean purchase with confidence interval 90% is [9341.03, 9393.
 ↪94]
```

CLT Analysis for mean purchase with confidence 95% - Based on Gender

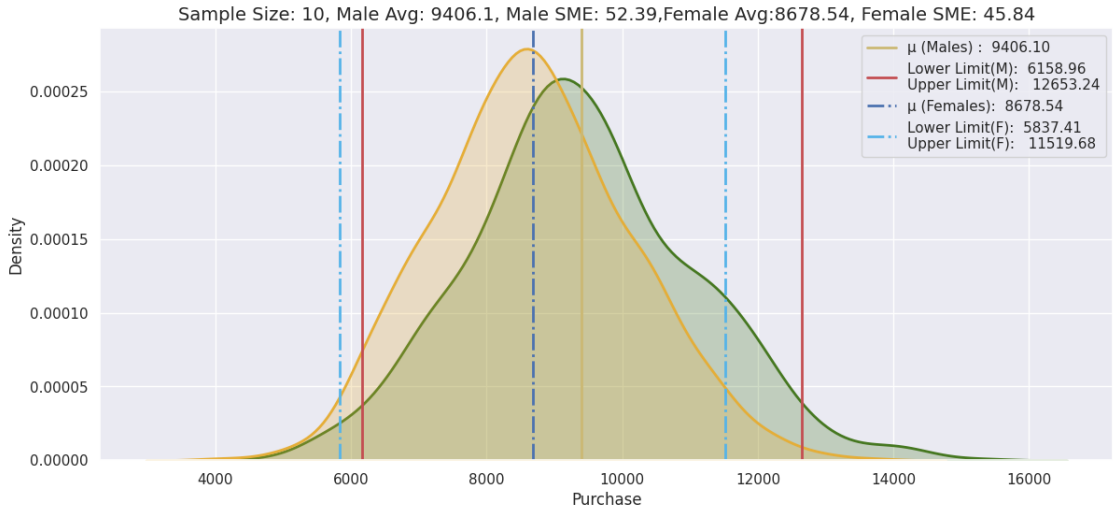
Analysis of the true mean of purchase values by gender with a 95% confidence

[67]:
```
itr_size = 1000
size_list = [1, 10, 30, 300, 1000, 100000]
ci = 0.95

array = np.empty((0,7))

for smp_siz in size_list:
    m_avg, f_avg, ll_m, ul_m, ll_f, ul_f =␣
 ↪bootstrapping(retail_data_smp_male,retail_data_smp_female,smp_siz,itr_size,ci)

    array = np.append(array, np.array([['M', ll_m, ul_m, smp_siz, ([ll_m,ul_m])␣
 ↪,(ul_m-ll_m),95]]), axis=0)
    array = np.append(array, np.array([['F', ll_f, ul_f, smp_siz, ([ll_f,ul_f])␣
 ↪,(ul_f-ll_f),95]]), axis=0)

overlap_95 = pd.DataFrame(array, columns =␣
 ↪['Gender','Lower_limit','Upper_limit','Sample_Size','CI','Range','Confidence_pct'])
overlap = pd.concat([overlap, overlap_95], axis=0)
```

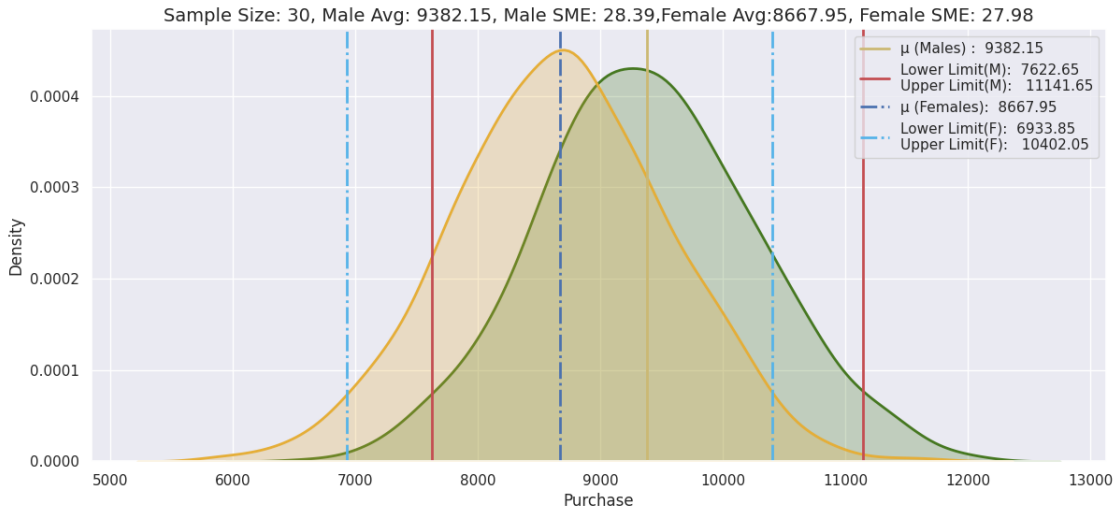
```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```



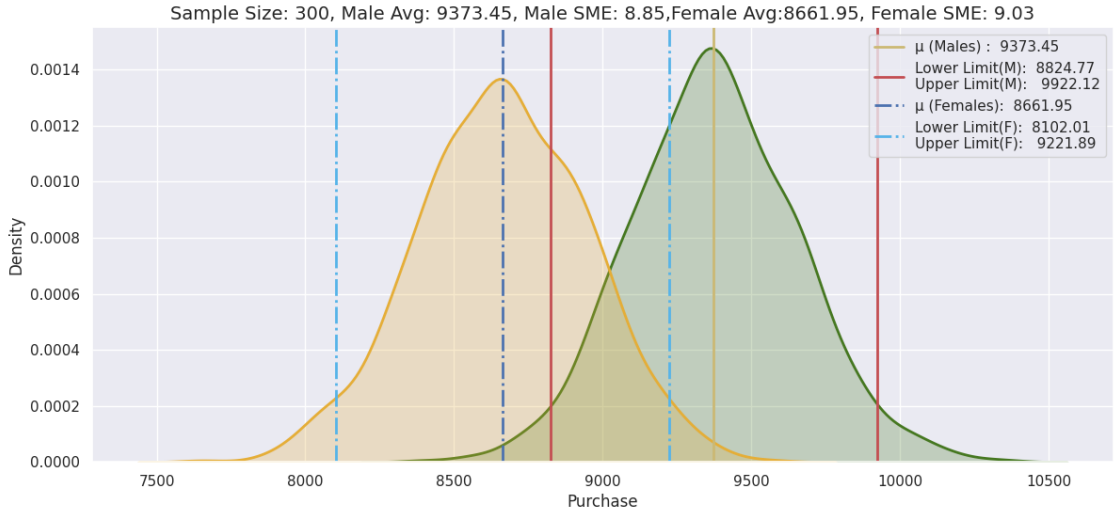

```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```



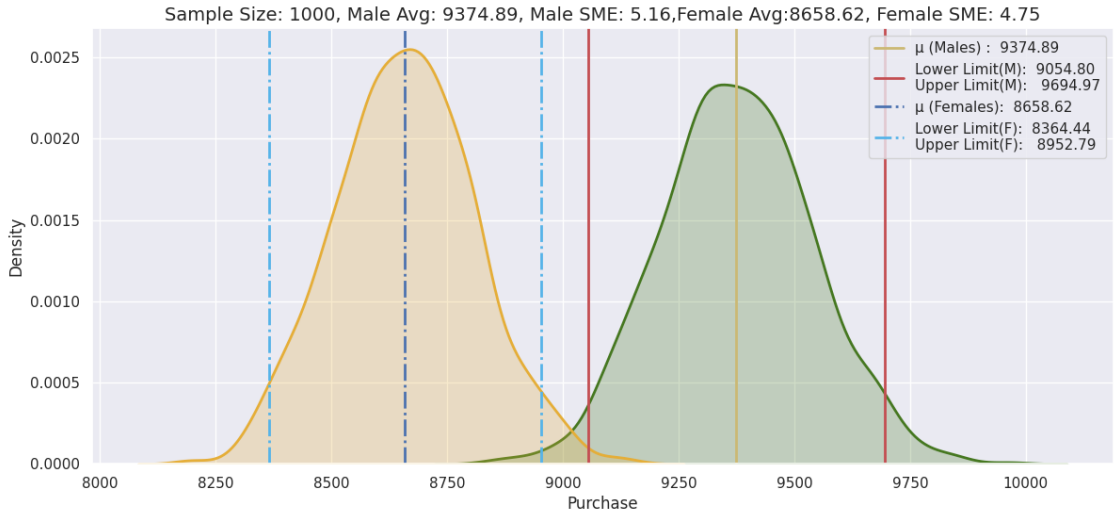

```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```



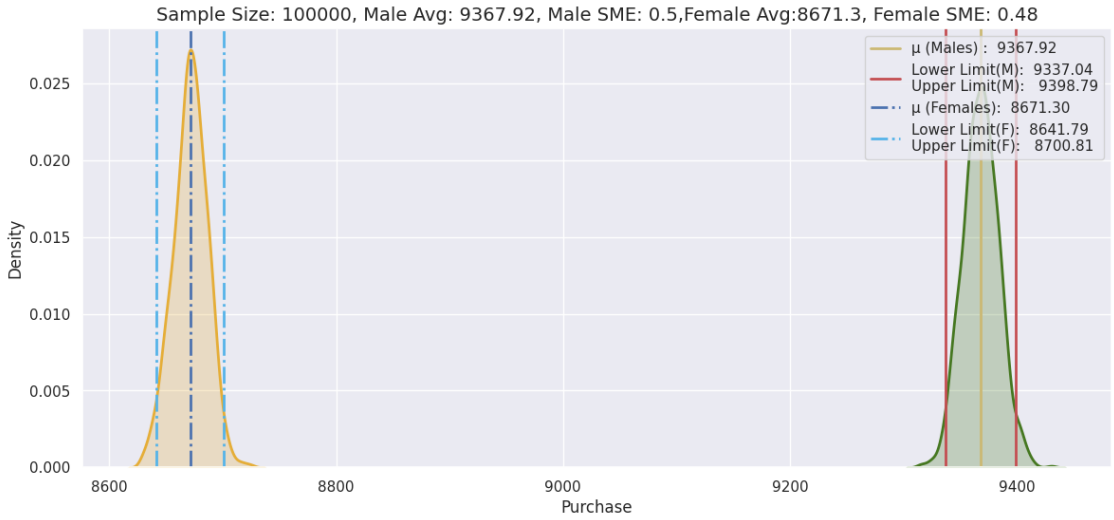

```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```




```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```




```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```



Using confidence interval 95%, the mean purchase value by gender shows a similar pattern to that found with confidence interval 90%-

As the sample size increases, the Male and female groups start to become distinct

With increasing sample size, Standard error of the mean in the samples decreases. For sample size 100000 is 0.47

For Female (sample size 100000) range for mean purchase with confidence interval 90% is [8642.58, 8701.58]

For Male range for mean purchase with confidence interval 95% is [9336.23, 9397.53]

Overlappings are increasing with a confidence interval of 95%. Due to the increasing CI, we consider higher ranges within which the actual population might fall, so that both mean purchase are more likely to fall within the same range.

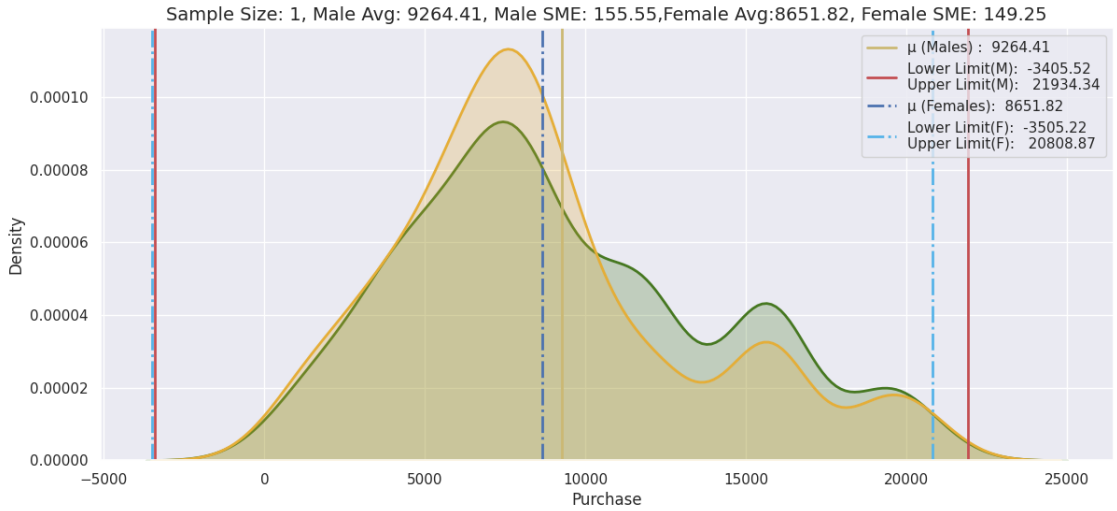
CLT Analysis for mean purchase with confidence 99% - Based on Gender

Analysis of the true mean of purchase values by gender with a 99% confidence.

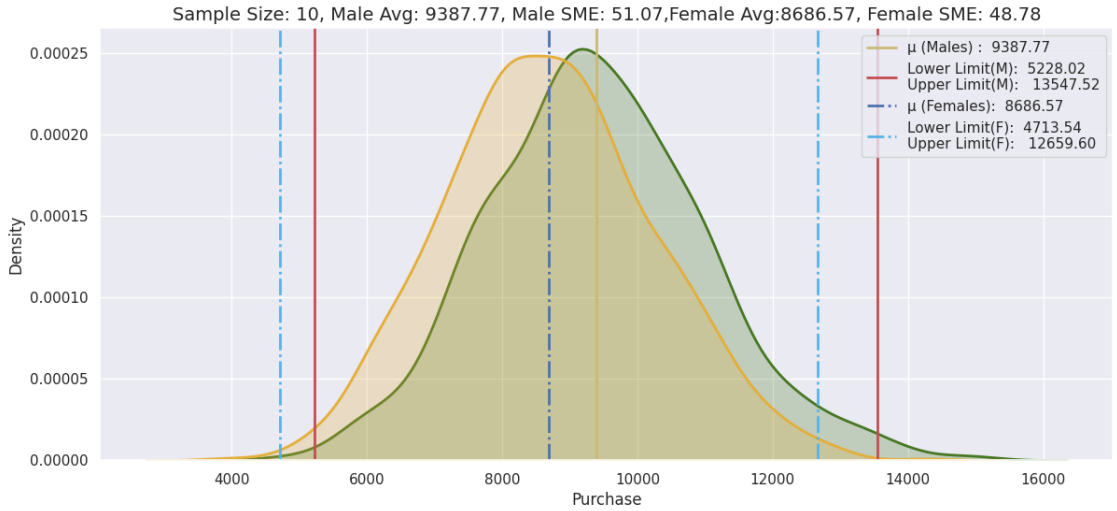
[68]:
```
itr_size = 1000
size_list = [1, 10, 30, 300, 1000, 100000]
ci = 0.99

array = np.empty((0,7))

for smp_siz in size_list:
    m_avg, f_avg, ll_m, ul_m, ll_f, ul_f =␣
 ↪bootstrapping(retail_data_smp_male,retail_data_smp_female,smp_siz,itr_size,ci)

    array = np.append(array, np.array([['M', ll_m, ul_m, smp_siz, ([ll_m,ul_m])␣
 ↪,(ul_m-ll_m),99]]), axis=0)
```

```
    array = np.append(array, np.array([['F', ll_f, ul_f, smp_siz, ([ll_f,ul_f])␣
 ↪,(ul_f-ll_f),99]]), axis=0)

overlap_99 = pd.DataFrame(array, columns =␣
 ↪['Gender','Lower_limit','Upper_limit','Sample_Size','CI','Range','Confidence_pct'])
overlap = pd.concat([overlap, overlap_99], axis=0)
```

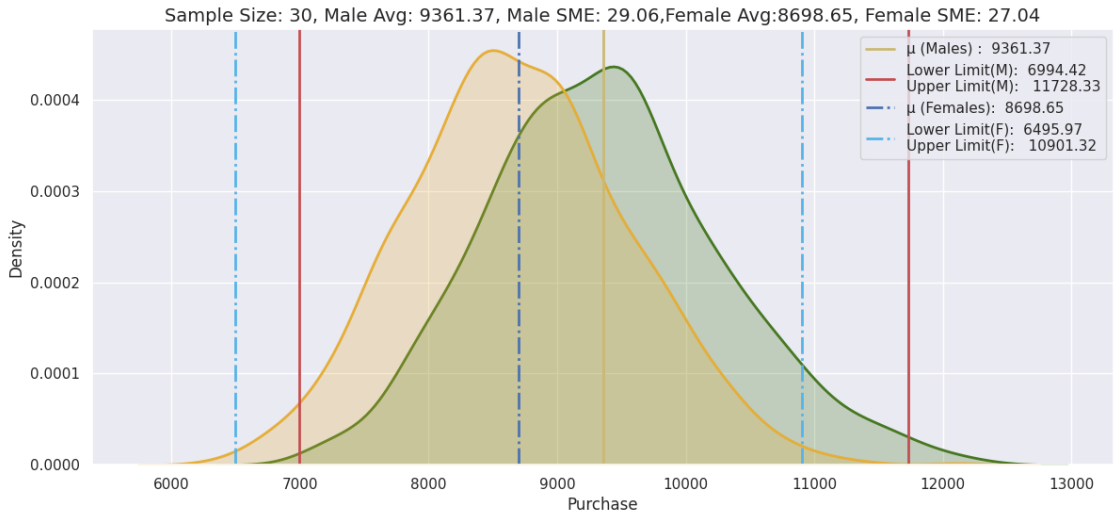
```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```



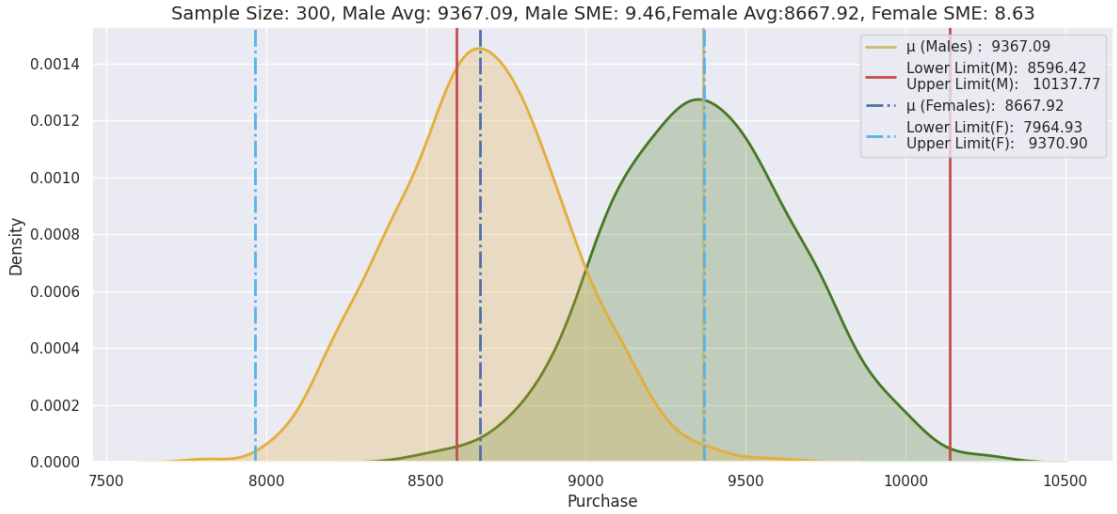

```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```



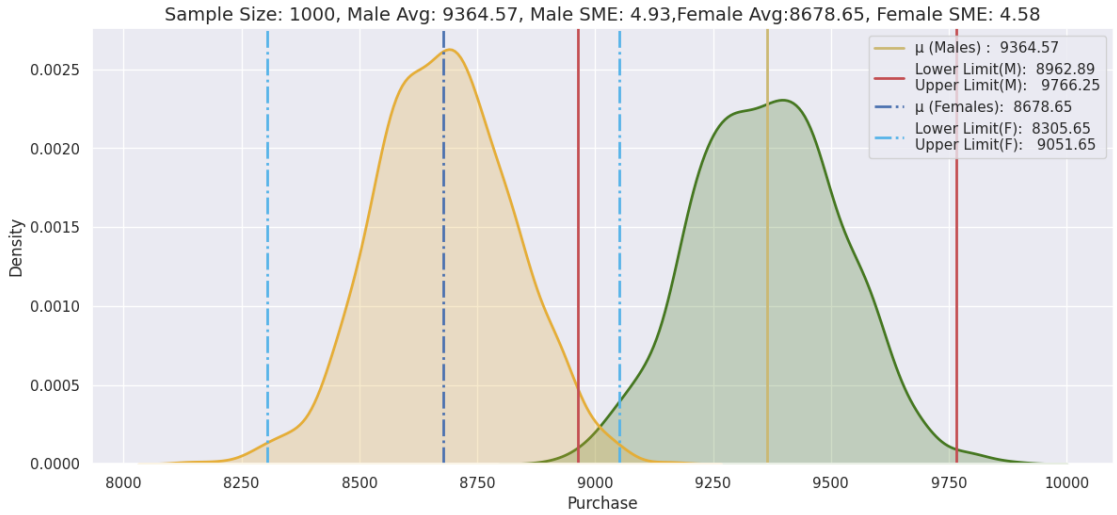

```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```



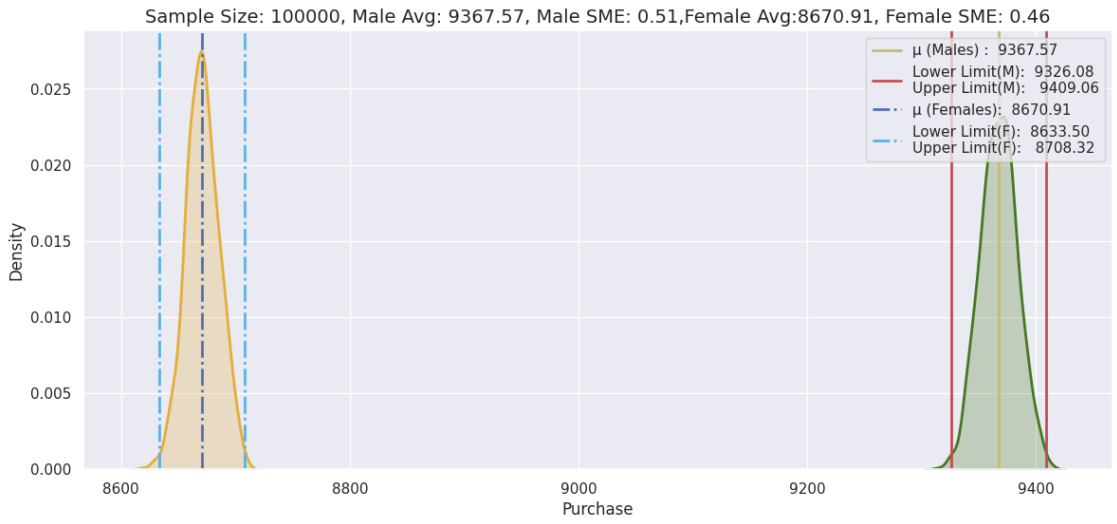

```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```




```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```




```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```



Using confidence interval 99%, the mean purchase value by gender shows a similar pattern to that found with confidence interval 90% & 95%-

As the sample size increases, the Male and female groups start to become distinct

With increasing sample size, Standard error of the mean in the samples decreases. For sample size 100000 is 0.45

For Female (sample size 100000) range for mean purchase with confidence interval 99% is [8634.54, 8707.85]

For Male range for mean purchase with confidence interval 90% is [9328.03, 9409.07]

When the confidence percentage increases, the spread, that is the difference between the upper and lower limits, also increases. For Female Confidence percent as [90,95,99] have difference between the upper & lower limits as [50.46,59,73.31]

Recommendations In light of the fact that females spend less than males on average, management needs to focus on their specific needs differently. Adding some additional offers for women can increase their spending on Black Friday.

Calculate Confidence Interval (CI) - to estimate the mean weight of the expenses by married and unmarried customers.

CLT Analysis for mean purchase with confidence 99% - Based on Marital Status

Analysis of the true mean of purchase values by marital Status with a 99% confidence.

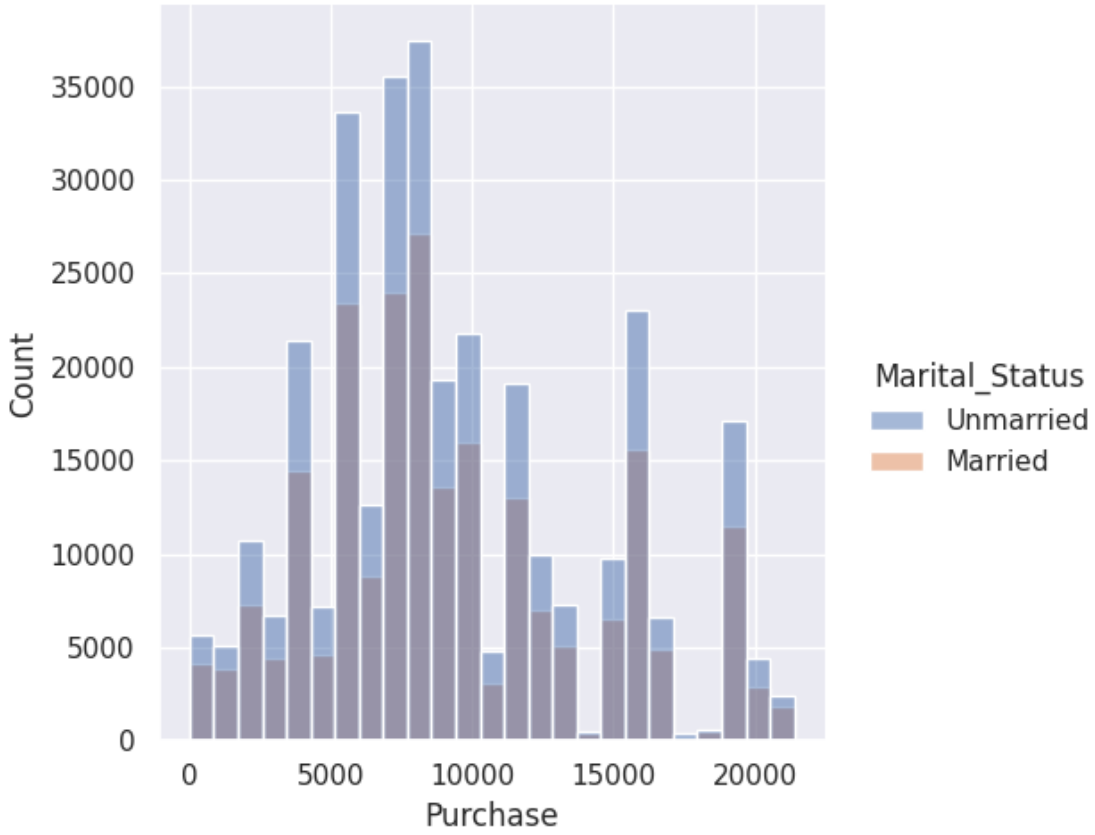
[69]:
```
retail_data_v1['Marital_Status'].replace(to_replace = 0, value = 'Unmarried',␣
↪inplace = True)
retail_data_v1['Marital_Status'].replace(to_replace = 1, value = 'Married',␣
↪inplace = True)
```

[70]:
```
retail_data_v1.sample(500,replace=True).groupby(['Marital_Status'])['Purchase'].
↪describe()
```

[70]:

Marital_Status	count	mean	std	min	25%	50%	75%	max
Unmarried	292.0	9243.910959	4763.750081	25.0	5901.0	8137.0	12138.75	20896.0
Married	208.0	8453.264423	4947.596040	216.0	5251.0	7915.5	11634.75	20658.0

[71]:
```
sns.displot(data = retail_data_v1, x = 'Purchase', hue = 'Marital_Status',bins␣
↪= 25)
plt.show()
```

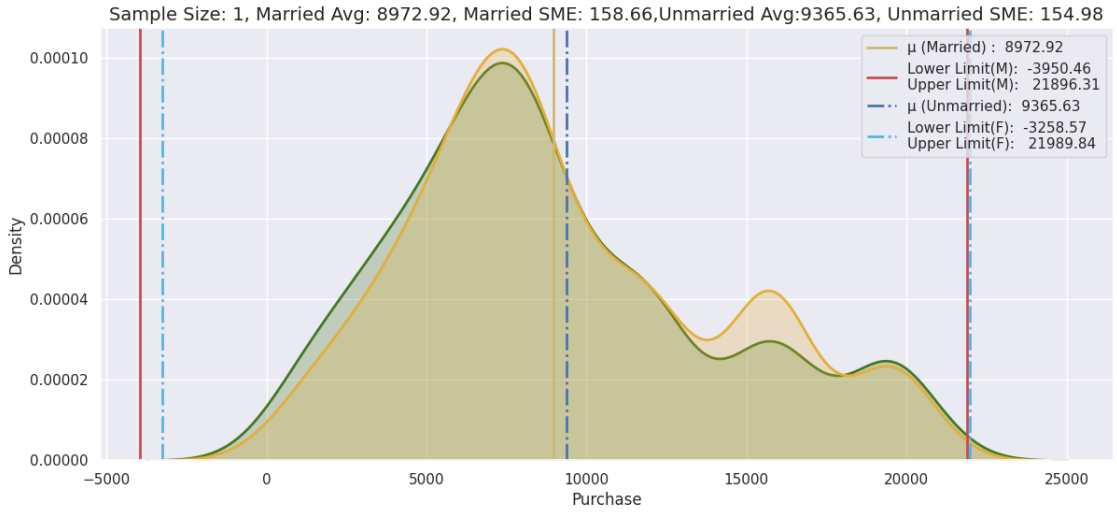


[72]:
```
retail_data_smp_married = retail_data_v1[retail_data_v1['Marital_Status'] ==␣
↪'Married']['Purchase']
```

```
retail_data_smp_unmarried = retail_data_v1[retail_data_v1['Marital_Status'] ==␣
 ↪'Unmarried']['Purchase']
```

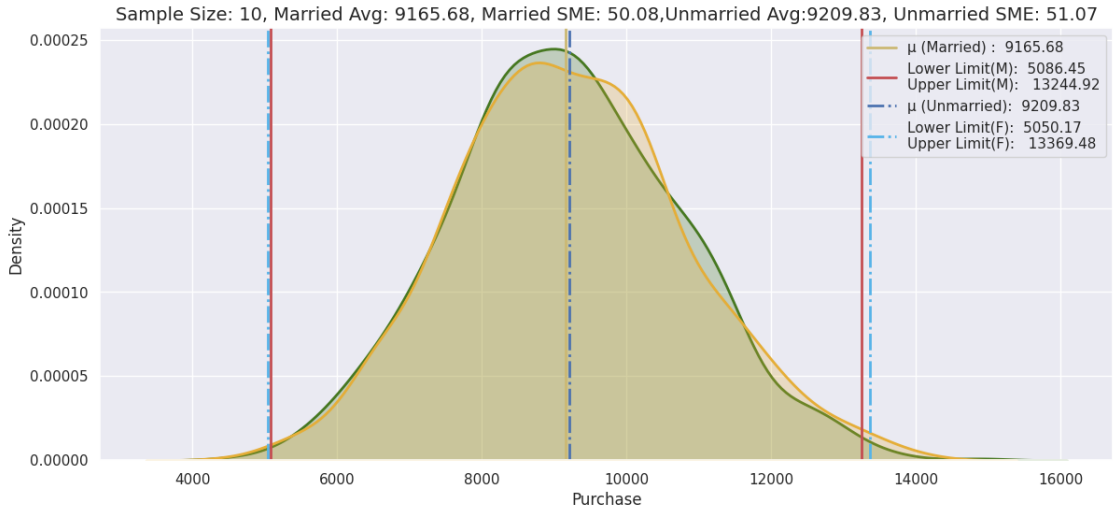
[73]:
```
itr_size = 1000
size_list = [1, 10, 30, 300, 1000, 100000]
ci = 0.99

array = np.empty((0,7))

for smp_siz in size_list:
    m_avg, f_avg, ll_m, ul_m, ll_u, ul_u =␣
 ↪bootstrapping_m_vs_um(retail_data_smp_married,retail_data_smp_unmarried,smp_siz,itr_size,ci

    array = np.append(array, np.array([['Married', ll_m, ul_m, smp_siz,␣
 ↪([ll_m,ul_m]) ,(ul_m-ll_m),99]]), axis=0)
    array = np.append(array, np.array([['Unmarried', ll_u, ul_u, smp_siz,␣
 ↪([ll_u,ul_u]) ,(ul_u-ll_u),99]]), axis=0)

overlap = pd.DataFrame(array, columns =␣
 ↪['Marital_Status','Lower_limit','Upper_limit','Sample_Size','CI','Range','Confidence_pct'])
```

```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```

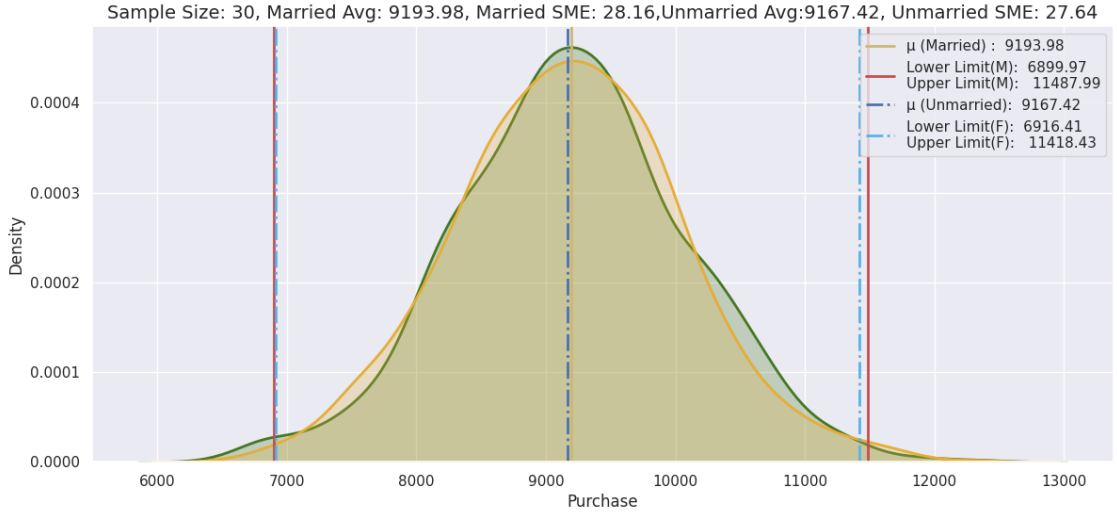



```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```

```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```

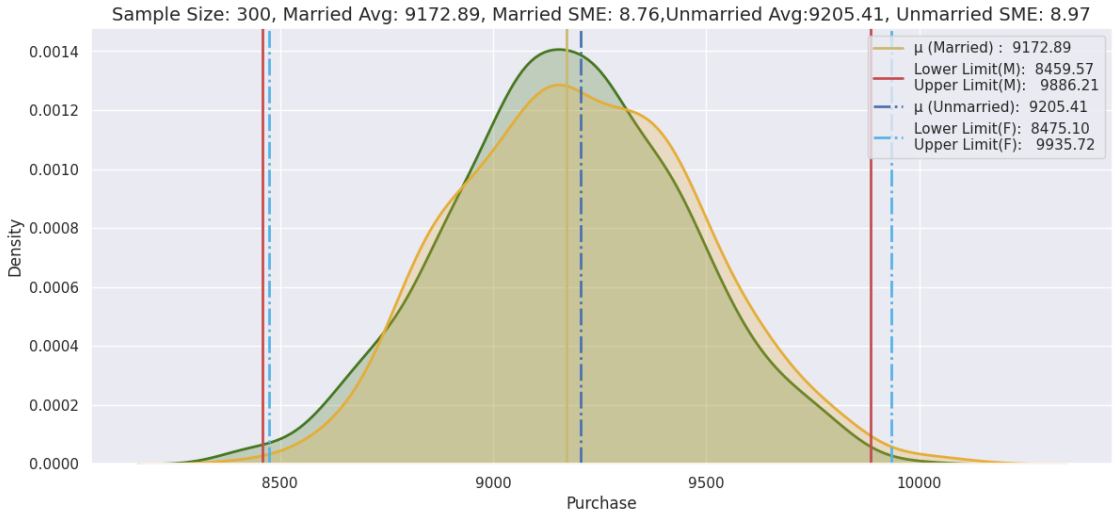



```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```

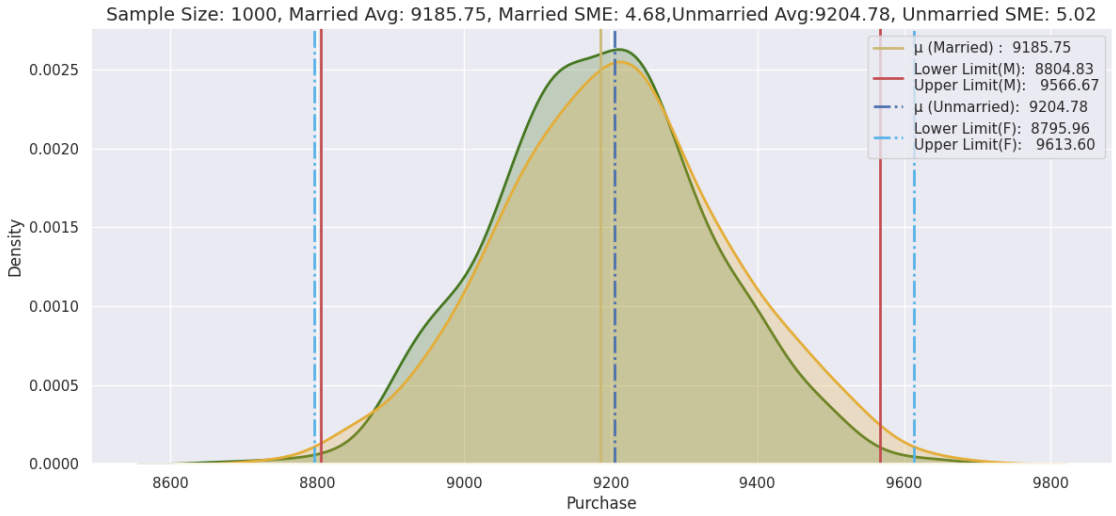



```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```

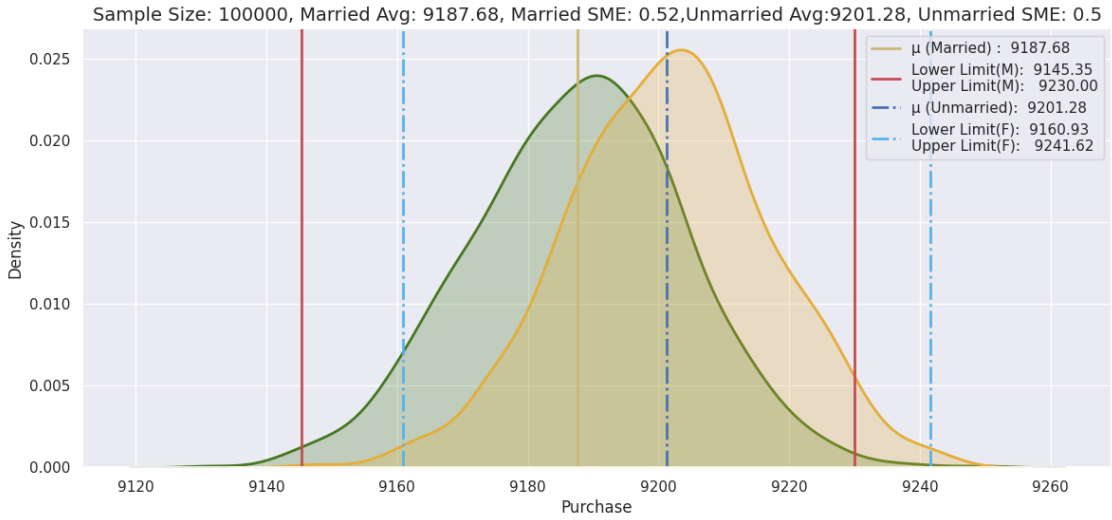
```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```




```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```




```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```



[74]:
```
overlap.head()
```

[74]:

	Marital_Status	Lower_limit	Upper_limit	Sample_Size	CI	Range	Confidence_pct
0	Married	-3950.46	21896.31	1	[-3950.46, 21896.31]	25846.77	99
1	Unmarried	-3258.57	21989.84	1	[-3258.57, 21989.84]	25248.41	99
2	Married	5086.45	13244.92	10	[5086.45, 13244.92]	8158.47	99
3	Unmarried	5050.17	13369.48	10	[5050.17, 13369.48]	8319.31	99
4	Married	6899.97	11487.99	30	[6899.97, 11487.99]	4588.02	99

[75]:
```
overlap.loc[(overlap['Marital_Status'] == 'Married') & (overlap['Sample_Size']
↪>= 300)]
```

[75]:

	Marital_Status	Lower_limit	Upper_limit	Sample_Size	CI	Range	Confidence_pct
6	Married	8459.57	9886.21	300	[8459.57, 9886.21]	1426.64	99
8	Married	8804.83	9566.67	1000	[8804.83, 9566.67]	761.84	99
10	Married	9145.35	9230.0	100000	[9145.35, 9230.0]	84.65	99

[76]:
```
overlap.loc[(overlap['Marital_Status'] == 'Unmarried') &␣
 ↪(overlap['Sample_Size'] >= 300)]
```

[76]:
```
   Marital_Status Lower_limit Upper_limit Sample_Size                 CI  \
7       Unmarried      8475.1     9935.72         300   [8475.1, 9935.72]
9       Unmarried     8795.96      9613.6        1000   [8795.96, 9613.6]
11      Unmarried     9160.93     9241.62      100000  [9160.93, 9241.62]

      Range Confidence_pct
7   1460.62             99
9    817.64             99
11    80.69             99
```

[ ]:
```
Overlapping is evident for married vs single customer spend even when more␣
 ↪samples are analyzed, which indicates that customers spend the same␣
 ↪regardless of whether they are single or married.

For Unmarried customer (sample size 100000) range for mean purchase with␣
 ↪confidence interval 99% is [9162.0, 9241.98]

For married customer range for mean purchase with confidence interval 90% is␣
 ↪[9148.09, 9227.05]
```

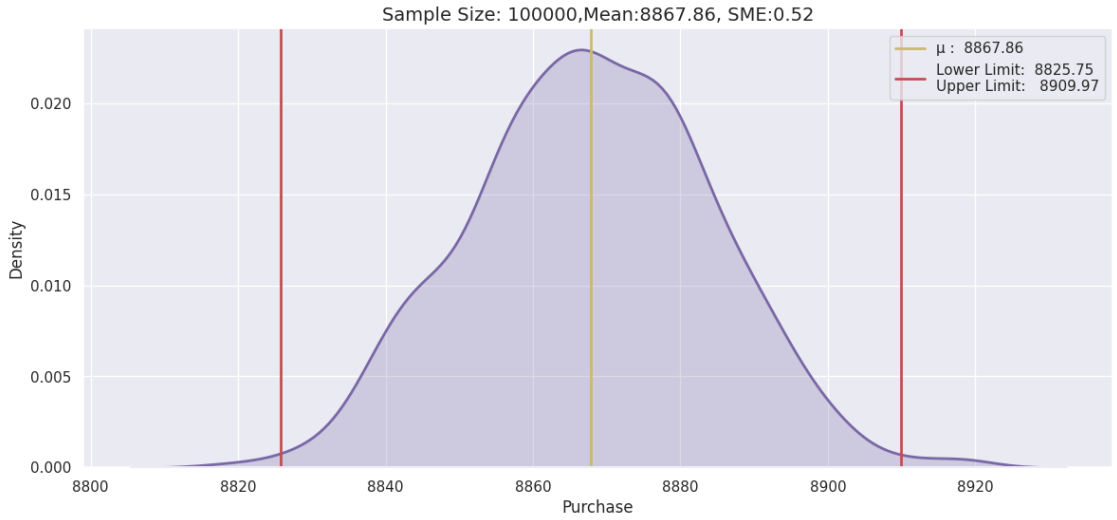
CLT Analysis for mean purchase with confidence 99% - Based on Age Group

Analysis of the true mean of purchase values by Age Group with a 99% confidence.

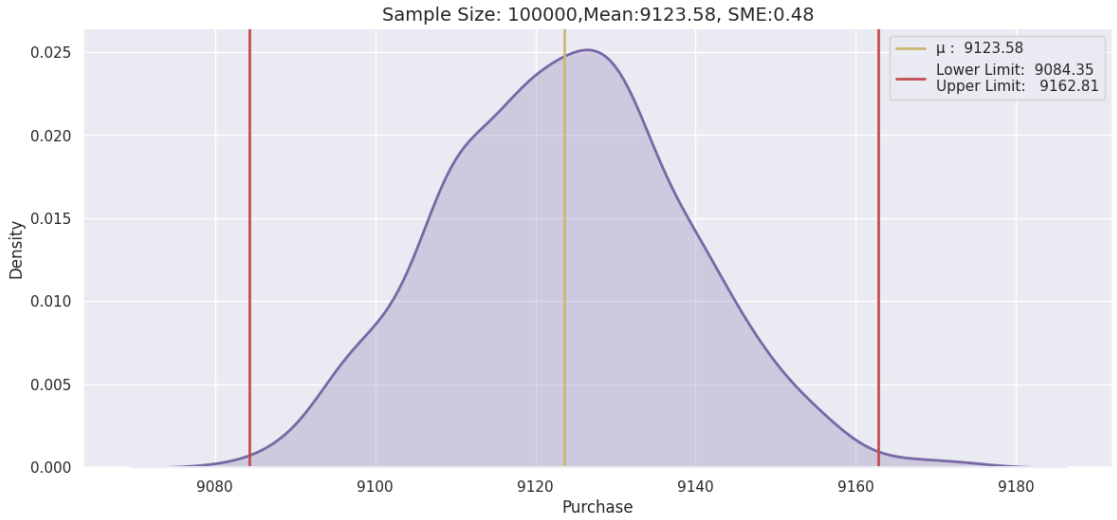
[77]:
```
itr_size = 1000
smp_size = 1000
ci = 0.99
age_list =['0-17', '18-25', '26-35', '36-45', '46-50', '51-55', '55+']

array = np.empty((0,8))

for age in age_list:
    mean, ll_m, ul_m = bootstrapping_age(retail_data_v1[retail_data_v1['Age']␣
 ↪== age]['Purchase'],smp_siz,itr_size,ci)

    array = np.append(array, np.array([[age,np.round(mean,2), ll_m, ul_m,␣
 ↪smp_siz, ([ll_m,ul_m]) ,(ul_m-ll_m),99]]), axis=0)

age_data = pd.DataFrame(array, columns =␣
 ↪['Age_Group','Mean','Lower_limit','Upper_limit','Sample_Size','CI','Range','Confidence_pct']
```

```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```

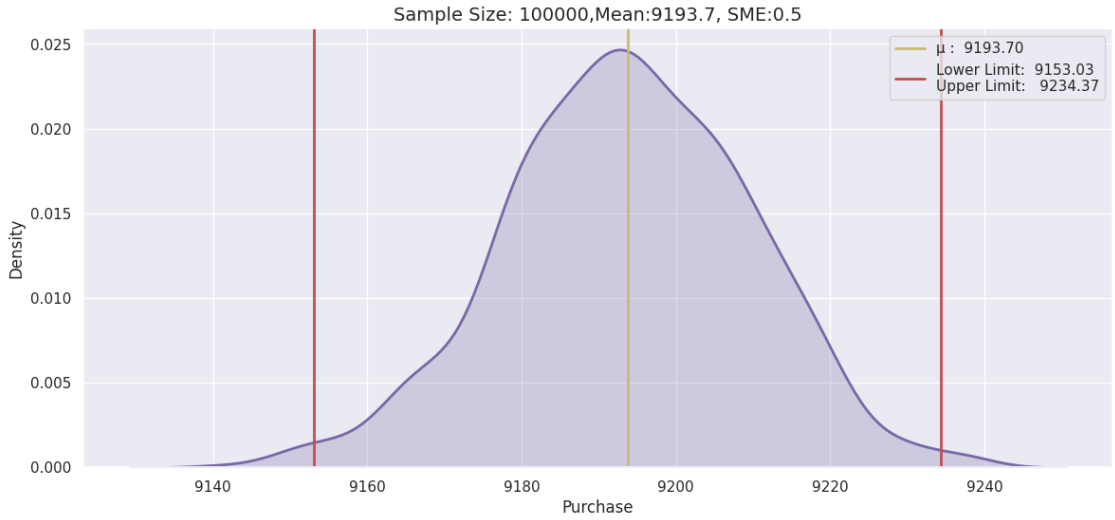
```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```



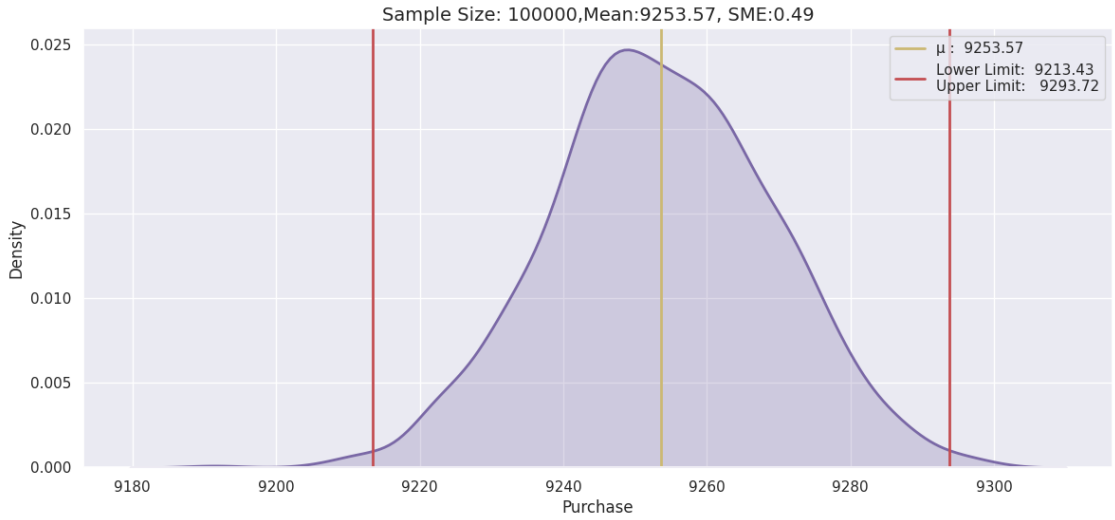

```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```



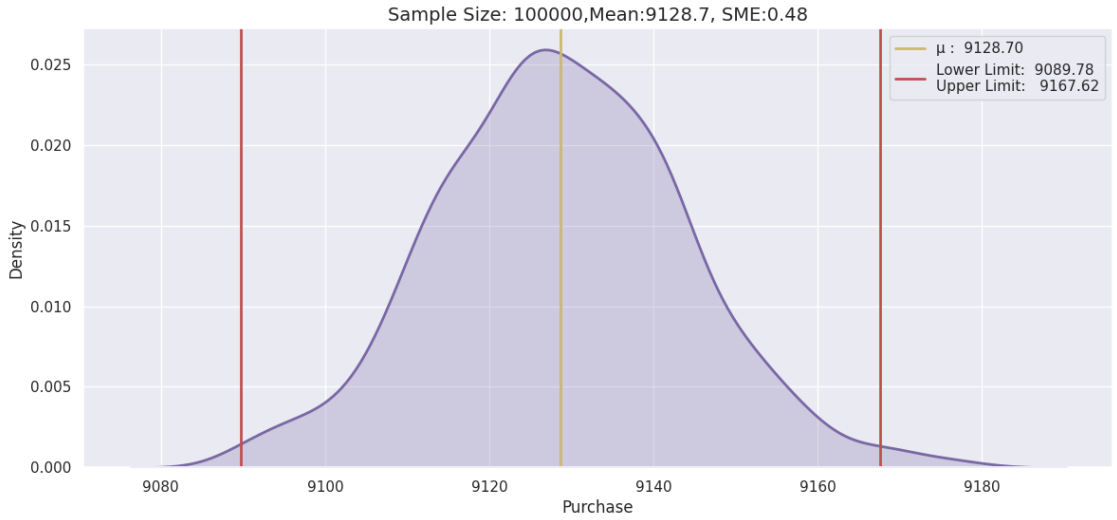

```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```



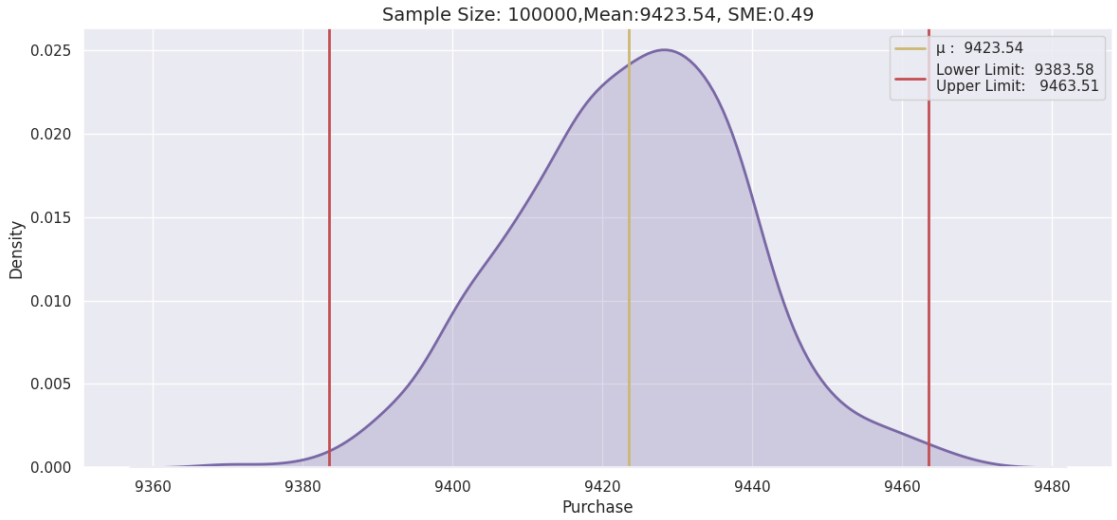

```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```



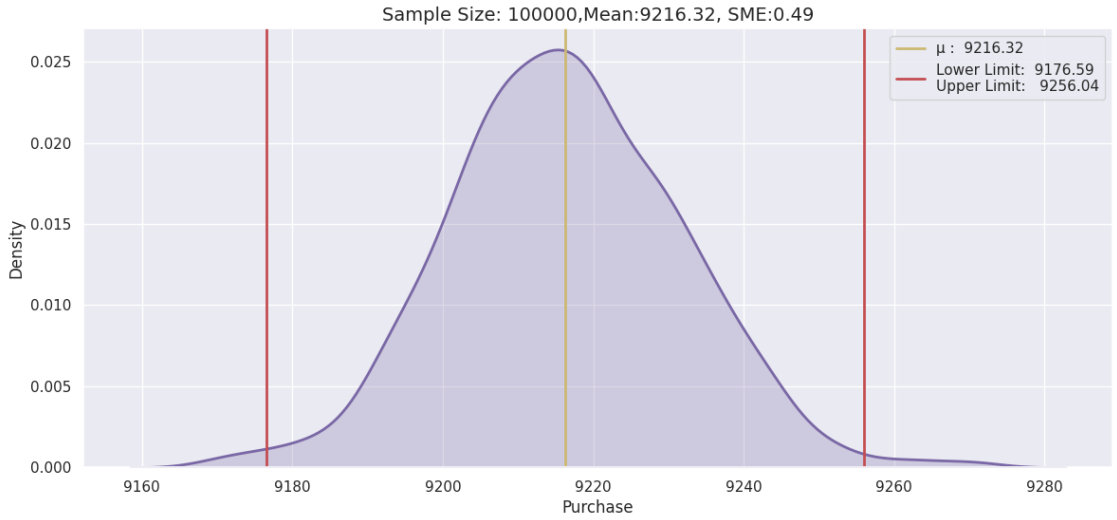

```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```




```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```




```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```



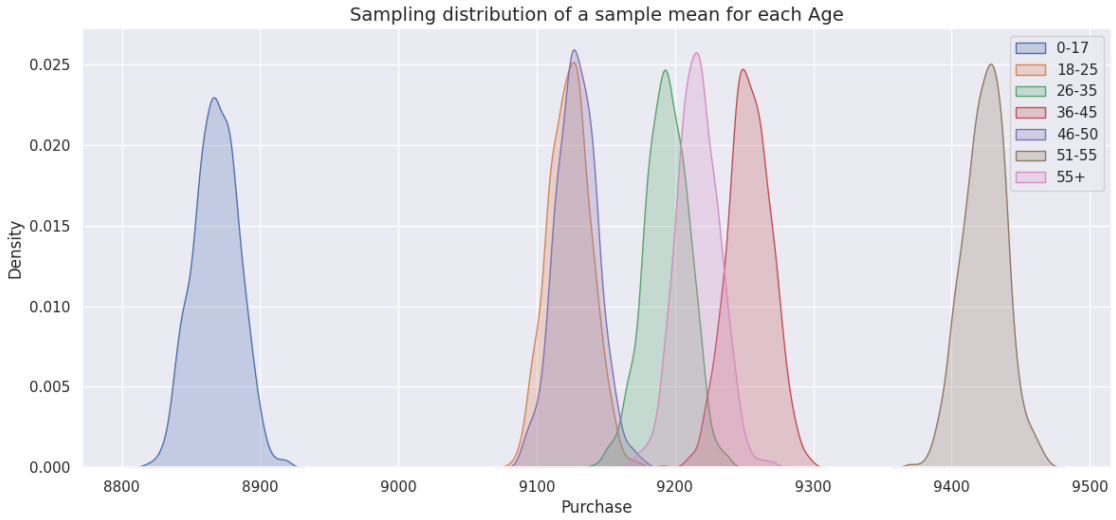
Checking the Sampling distribution of a sample mean for each Age Group

[78]:
```
age_dict = {}
age_list = ['0-17', '18-25', '26-35', '36-45', '46-50', '51-55', '55+']
for i in range(len(age_data)):
    age_dict[age_list[i]] = age_data.loc[i, "Mean"]
```

[79]:
```
fig, ax = plt.subplots(figsize=(14,6))
sns.set_style("darkgrid")
for label_val in age_dict.keys():
    sns.kdeplot(age_dict[label_val], shade = True, label = label_val)

plt.title("Sampling distribution of a sample mean for each␣
 ↪Age",fontsize=14,family="Comic Sans MS")
plt.xlabel('Purchase')
plt.legend(loc='upper right')
```

[79]:
```
<matplotlib.legend.Legend at 0x7f4dd0264e80>
```

```
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
WARNING:matplotlib.font_manager:findfont: Font family 'Comic Sans MS' not found.
```



Spending by Age_group 0-17 is low compared to other age groups.

Customers in Age_group 51-55 spend the most between 9381.9 and 9463.7

Recommendations

Management should come-up with some games in the mall to attract more younger generation will can help them to increase the sale.

The management should have some offers on kids (0-17 years) in order to increase sales.

In order to attract more young shoppers, they can offer some games for the younger generation.

The majority of our customers come from city category B but customers come from City category C spent more as mean is 9719.

The majority of users come from City Category C, but more people from City Category B tend to purchase, which suggests the same users visit the mall multiple times in City Category B.

Majority of Customers purchase within the 5,000 - 20,000 range.

Males clearly purchase more than females. 75% of men and only 25% of women purchase products.

Most mall customers are between the ages of 26 and 35.60% of purchases are made by people between the ages of 26 and 45

City Category B accounts for 42%, City Category C 31%, and City Category A represents 27% of all customer purchases.Purchases are high in city category C

Most mall customers are between the ages of 26 and 35.City category C has more customers between the ages of 18 and 45.

In City Category C, there are slightly more female customers.

Product 5 and 8 is common among females.

As the sample size increases, the two groups start to become distinct. With increasing sample size, Standard error of the mean in the samples decreases. For sample size 100000 is 0.49 with confidence is 90%.

Overlappings are increasing with a confidence interval of 95%. Due to the increasing CI, we consider higher ranges within which the actual population might fall, so that both mean purchase are more likely to fall within the same range.

Using confidence interval 99%, the mean purchase value by gender shows a similar pattern to that found with confidence interval 90% & 95%

For Female (sample size 100000) range for mean purchase with confidence interval 99% is [8634.54, 8707.85]

For Male range for mean purchase with confidence interval 99% is [9328.03, 9409.07]

When the confidence percentage increases, the spread, that is the difference between the upper and lower limits, also increases. For Female Confidence percent as [90,95,99] have difference between the upper & lower limits as [50.46,59,73.31]

Overlapping is evident for married vs single customer spend even when more samples are analyzed, which indicates that customers spend the same regardless of whether they are single or married.

Spending by Age_group 0-17 is low compared to other age groups.

Customers in Age_group 51-55 spend the most between 9381.9 and 9463.7

Recommendations

In light of the fact that females spend less than males on average, management needs to focus on their specific needs differently. Adding some additional offers for women can increase their spending on Black Friday.

Management should come-up with some games in the mall to attract more younger generation will can help them to increase the sale.

The management should have some offers on kids (0-17 years) in order to increase sales.

In order to attract more young shoppers, they can offer some games for the younger generation.