In [1]:

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:

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

```
Mounted at /content/drive
```

In [3]:

```
import pandas as pd
df=pd.read_csv("/content/drive/MyDrive/Scaler Business Case Studies/Walmart Case Study/walmart_data.csv")
print(df.shape)
df.head()
```

```
(550068, 10)
```

Out[3]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status	Product_Categor
0	1000001	P00069042	F	0-17	10	A	2	0	
1	1000001	P00248942	F	0-17	10	A	2	0	
2	1000001	P00087842	F	0-17	10	A	2	0	1
3	1000001	P00085442	F	0-17	10	A	2	0	1
4	1000002	P00285442	M	55+	16	C	4+	0	

In [ ]:

```
df.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
```

In [ ]:

```
df.describe()
```

Out[ ]:

	User_ID	Occupation	Marital_Status	Product_Category	Purchase
count	5.500680e+05	550068.000000	550068.000000	550068.000000	550068.000000
mean	1.003029e+06	8.076707	0.409653	5.404270	9263.968713
std	1.727592e+03	6.522660	0.491770	3.936211	5023.065394
min	1.000001e+06	0.000000	0.000000	1.000000	12.000000
25%	1.001516e+06	2.000000	0.000000	1.000000	5823.000000
50%	1.003077e+06	7.000000	0.000000	5.000000	8047.000000
75%	1.004478e+06	14.000000	1.000000	8.000000	12054.000000
max	1.006040e+06	20.000000	1.000000	20.000000	23961.000000

In [ ]:

```
df.isnull().any()
```

Out[ ]:

```
User_ID                     False
Product_ID                  False
Gender                      False
Age                         False
Occupation                  False
City_Category               False
Stay_In_Current_City_Years  False
Marital_Status              False
Product_Category            False
Purchase                    False
dtype: bool
```

In [ ]:

```
df.dtypes
```

Out[ ]:

```
User_ID                      int64
Product_ID                  object
Gender                      object
Age                         object
Occupation                   int64
City_Category               object
Stay_In_Current_City_Years  object
Marital_Status               int64
Product_Category             int64
Purchase                     int64
dtype: object
```

In [4]:

```
cols = ['Occupation', 'Marital_Status', 'Product_Category']
df[cols] = df[cols].astype('object')
```

In [ ]:

```
df.describe(include='all')
```

Out[ ]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status	Pr
count	5.500680e+05	550068	550068	550068	550068.0	550068	550068	550068.0	
unique	NaN	3631	2	7	21.0	3	5	2.0	
top	NaN	P00265242	M	26-35	4.0	B	1	0.0	
freq	NaN	1880	414259	219587	72308.0	231173	193821	324731.0	

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status	Pr
mean	1.003029e+06	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
std	1.727592e+03	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
min	1.000001e+06	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
25%	1.001516e+06	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
50%	1.003077e+06	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
75%	1.004478e+06	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
max	1.006040e+06	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

In [ ]:

```
df['User_ID'].value_counts().shape
```

Out[ ]:

```
(5891,)
```

In [ ]:

```
df['Product_Category'].value_counts()
```

Out[ ]:

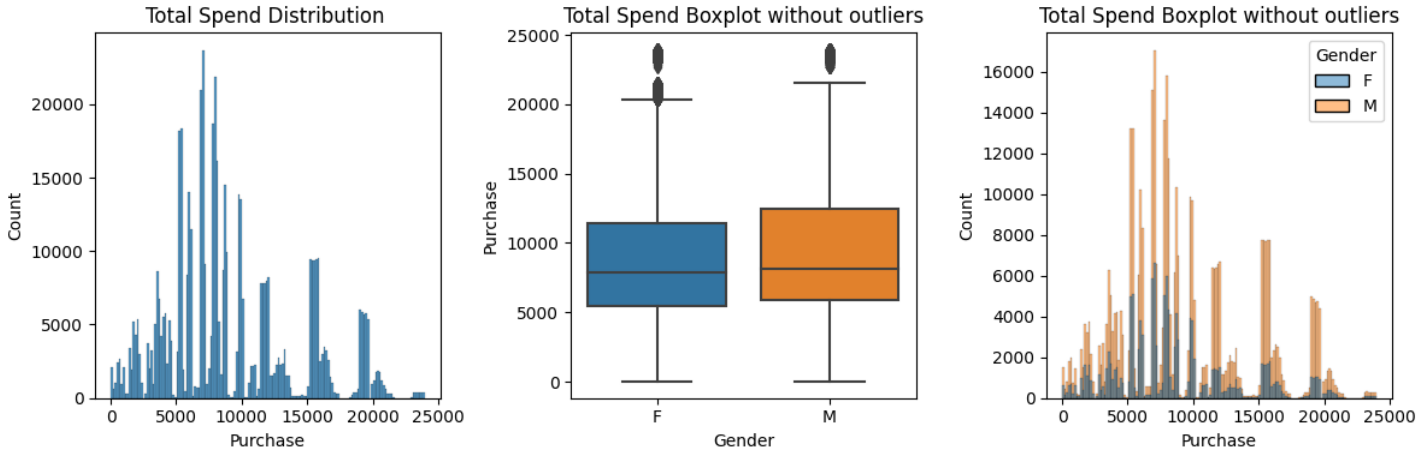
```
5     150933
1     140378
8     113925
11     24287
2      23864
6      20466
3      20213
4      11753
16      9828
15      6290
13      5549
10      5125
12      3947
7       3721
18      3125
20      2550
19      1603
14      1523
17       578
9        410
Name: Product_Category, dtype: int64
```

1. **Total 5891 unique customers.**
2. **Total 3631 unique products.**
3. **Most sold product is P00265242, sold 1880 times. Increase the inventory of these products.**
4. **There are no null values in the dataset.**
5. **Cheapest product cost 12 , mean cost of a product is 9263.96, and max cost of a product is 23961.**
6. **Total 5,50,068 products sold.**
7. **Total 20 unique product categories**
8. **Total 20 differnent types of occupations.**
9. **Product Category 5,1,8 are sold the most. They should be kept in high visibility area of Walmart stores.**

In [14]:

```
#Purchase

fig = plt.figure(figsize=(12, 4))

plt.subplot(1,3,1)
sns.histplot(df['Purchase'])
plt.title("Total Spend Distribution ")
```

```
plt.subplot(1,3,2)
sns.boxplot(data=df,y='Purchase',x='Gender')
plt.title("Total Spend Boxplot without outliers")

plt.subplot(1,3,3)
sns.histplot(data=df,x='Purchase',hue='Gender')
plt.title("Total Spend Boxplot without outliers")


plt.tight_layout()
```



In [25]:

```
def summ_stats(df,var):
  print("Summary stats for {0} are == \n ".format(var))
  print('#'*20)
  print("min Value is = ",df['Purchase'].quantile(0))
  print("25th Percentile value is = ",df['Purchase'].quantile(.25))
  print("Median value is = ",df['Purchase'].quantile(.5))
  print("Mean value is = ",round(df['Purchase'].mean()))
  print("75th Percentile value is = ",df['Purchase'].quantile(.75))
  ul=df['Purchase'].quantile(.75)+((df['Purchase'].quantile(.75)-df['Purchase'].quantile
(.25))*1.5)
  print('Upper outlier value is =',ul)
  print("99th Percentile value is = ",df['Purchase'].quantile(.99))
  print("Max value is = ",df['Purchase'].quantile(1))
  print('*'*20)
summ_stats(df,"Purchase Amount")
summ_stats(df.loc[df['Gender']=='M'],"Purchase Amount for Male")
summ_stats(df.loc[df['Gender']=='F'],"Purchase Amount for Female")
```

```
Summary stats for Purchase Amount are ==

####################
min Value is =  12.0
25th Percentile value is =  5823.0
Median value is =  8047.0
Mean value is =  9264
75th Percentile value is =  12054.0
Upper outlier value is = 21400.5
99th Percentile value is =  20665.0
Max value is =  23961.0
********************
Summary stats for Purchase Amount for Male are ==

####################
min Value is =  12.0
25th Percentile value is =  5863.0
Median value is =  8098.0
Mean value is =  9438
75th Percentile value is =  12454.0
Upper outlier value is = 22340.5
99th Percentile value is =  20682.0
Max value is =  23961.0
********************
Summary stats for Purchase Amount for Female are ==
```

```
####################
min Value is =  12.0
25th Percentile value is =  5433.0
Median value is =  7914.0
Mean value is =  8735
75th Percentile value is =  11400.0
Upper outlier value is = 20350.5
99th Percentile value is =  20613.0
Max value is =  23959.0
********************
```

Q2. Are women spending more money per transaction than men? Why or Why not? (10 Points)

Ans

Mean,Median purchase amount for males is = 9438 and 8098 respectively.
Mean,Median purchase amount for females is = 8735 and 7914 respectively.

Clearly men on average are spending per transaction more then women. This could be due to number of reasons.

1. Men prefer expensive products.
2. Maybe, men are paid more than women hence spend more.
3. Men are targetted with campaigns and advertisement of products with high price.
4. Women makw more sensible/conservative choices while buying products.

In [ ]:

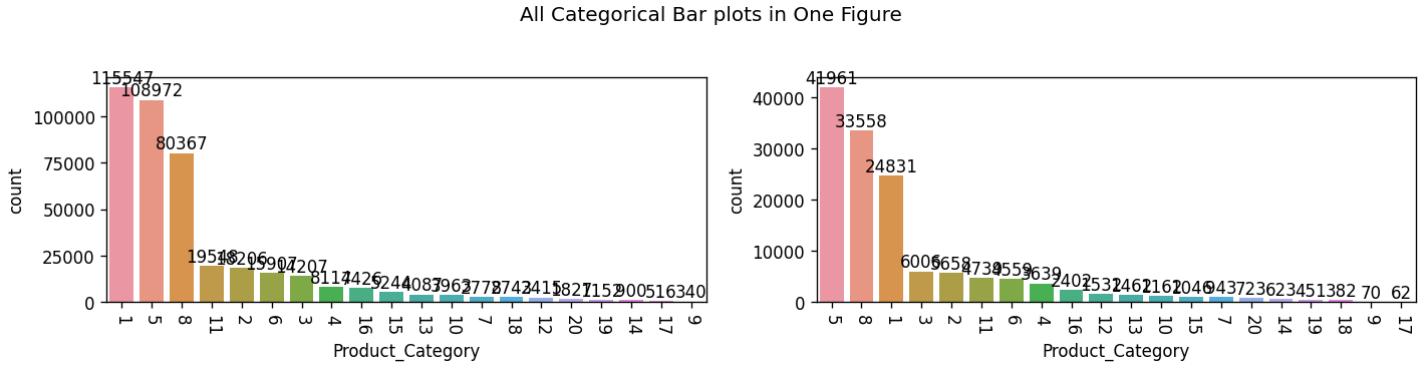
```
#Product
print("No of unique products sold are == ",df['Product_ID'].nunique())
print("No of unique products bought by Males are == ",df.loc[df['Gender']=='M']['Product_
ID'].nunique())
print("No of unique products bought by Females are ==  ",df.loc[df['Gender']=='F']['Produ
ct_ID'].nunique())

print()
print("{0} is the Most popular product bought {1} times ".format(df['Product_ID'].value_c
ounts().index[0],df['Product_ID'].value_counts()[0]))
print("{0} is the Most bought product by males bought {1} times ".format(df.loc[df['Gende
r']=='M']['Product_ID'].value_counts().index[0],df.loc[df['Gender']=='M']['Product_ID'].
value_counts()[0]))
print("{0} is the Most bought product by females bought {1} times ".format(df.loc[df['Gen
der']=='F']['Product_ID'].value_counts().index[0],df.loc[df['Gender']=='F']['Product_ID'
].value_counts()[0]))
```

```
No of unique products sold are ==  3631
No of unique products bought by Males are ==  3588
No of unique products bought by Females are ==   3367

P00265242 is the Most bought product by 1880 times
P00265242 is the Most bought product by males 1372 times
P00265242 is the Most bought product by females 508 times
```

In [ ]:

```
catcols=['M','F']
_, axes = plt.subplots(nrows=1, ncols=2,sharex=False,sharey=False, figsize=(12, 3),dpi=1
20)

for ax,gen in zip(axes.ravel(),catcols):
    sns.countplot(x='Product_Category', data=df.loc[df['Gender']==gen],ax=ax,order=df.lo
c[df['Gender']==gen]['Product_Category'].value_counts().index)
    ax.tick_params(axis ='x', rotation = -90)

    for label in ax.containers:
```

```
        ax.bar_label(label)

plt.suptitle('All Categorical Bar plots in One Figure', verticalalignment='bottom', fontsize=12)
plt.tight_layout()
plt.show()
```



Product Category Insight

1. **Product Category 1 is most bought category by Male while 5 is the most bought category by females.**

Value Counts

In [104]:

```
df.columns
```

Out[104]:

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

In [106]:

```
print(np.round(df['Gender'].value_counts(normalize=True)*100))
print(np.round(df['City_Category'].value_counts(normalize = True) * 100))
print(np.round(df['Marital_Status'].value_counts(normalize = True) * 100))
print(np.round(df['Age'].value_counts(normalize = True) * 100))
```

```
M    75.0
F    25.0
Name: Gender, dtype: float64
B    42.0
C    31.0
A    27.0
Name: City_Category, dtype: float64
0    59.0
1    41.0
Name: Marital_Status, dtype: float64
26-35    40.0
36-45    20.0
18-25    18.0
46-50     8.0
51-55     7.0
55+       4.0
0-17      3.0
Name: Age, dtype: float64
```

In [108]:

```
print(np.round(df['Stay_In_Current_City_Years'].value_counts(normalize = True) * 100))
print(np.round(df['Occupation'].value_counts(normalize = True) * 100))
print(np.round(df['Product_Category'].value_counts(normalize = True) * 100))
```

```
1     35.0
2     19.0
3     17.0
4+    15.0
0     14.0
Name: Stay_In_Current_City_Years, dtype: float64
4     13.0
0     13.0
7     11.0
1      9.0
17     7.0
20     6.0
12     6.0
14     5.0
2      5.0
16     5.0
6      4.0
3      3.0
10     2.0
5      2.0
15     2.0
11     2.0
19     2.0
13     1.0
18     1.0
9      1.0
8      0.0
Name: Occupation, dtype: float64
5     27.0
1     26.0
8     21.0
11     4.0
2      4.0
6      4.0
3      4.0
4      2.0
16     2.0
15     1.0
13     1.0
10     1.0
12     1.0
7      1.0
18     1.0
20     0.0
19     0.0
14     0.0
17     0.0
9      0.0
Name: Product_Category, dtype: float64
```

Insights from Value Count

1. **75% transactions on Black friday are done by males.**
2. **Most no (42%) transactions are done in City B.**
3. **59% transactions are done by single people.**
4. **Most no (40%) transactions are done BY people in age group 26-35.**
5. **35% transactions are done by people who have lived for one year in the city.**
6. **People with occupation 4(13%),0(13%),7(13%) transact the most on Black friday.**
7. **Product category 5 (27%),1 (26%),8 (21%) are sold the most during Black Friday.**

Per User Statistics

In [27]:

```
first_rows = df.groupby('User_ID',as_index=False).first()
```

```
second_row=df.groupby('User_ID',as_index=False).agg({'Product_Category':'nunique','Purchase':'sum'})
second_row.rename(columns={'Product_Category':'no_of_products','Purchase':'total_spend'},inplace=True)

df_cust=pd.merge(first_rows,second_row,how='left',on='User_ID')
df_cust.drop(['Occupation','Product_Category','Product_ID','Purchase'],axis=1,inplace=True)
df_cust.head()
```

Out[27]:

	User_ID	Gender	Age	City_Category	Stay_In_Current_City_Years	Marital_Status	no_of_products	total_spend
0	1000001	F	0-17	A	2	0	11	334093
1	1000002	M	55+	C	4+	0	6	810472
2	1000003	M	26-35	A	3	0	6	341635
3	1000004	M	46-50	B	2	1	2	206468
4	1000005	M	26-35	A	1	1	12	821001

In [ ]:

```
df_cust.describe(include='all')
```

Out[ ]:

	User_ID	Gender	Age	City_Category	Stay_In_Current_City_Years	Marital_Status	no_of_products	total_spend
count	5.891000e+03	5891	5891	5891	5891	5891.000000	5891.000000	5.891000e+03
unique	NaN	2	7	3	5	NaN	NaN	NaN
top	NaN	M	26-35	C	1	NaN	NaN	NaN
freq	NaN	4225	2053	3139	2086	NaN	NaN	NaN
mean	1.003025e+06	NaN	NaN	NaN	NaN	0.419963	9.638771	8.650166e+05
std	1.743379e+03	NaN	NaN	NaN	NaN	0.493594	3.595397	9.436445e+05
min	1.000001e+06	NaN	NaN	NaN	NaN	0.000000	1.000000	4.668100e+04
25%	1.001518e+06	NaN	NaN	NaN	NaN	0.000000	7.000000	2.376780e+05
50%	1.003026e+06	NaN	NaN	NaN	NaN	0.000000	9.000000	5.212130e+05
75%	1.004532e+06	NaN	NaN	NaN	NaN	1.000000	12.000000	1.119250e+06
max	1.006040e+06	NaN	NaN	NaN	NaN	1.000000	19.000000	1.053691e+07

In [109]:

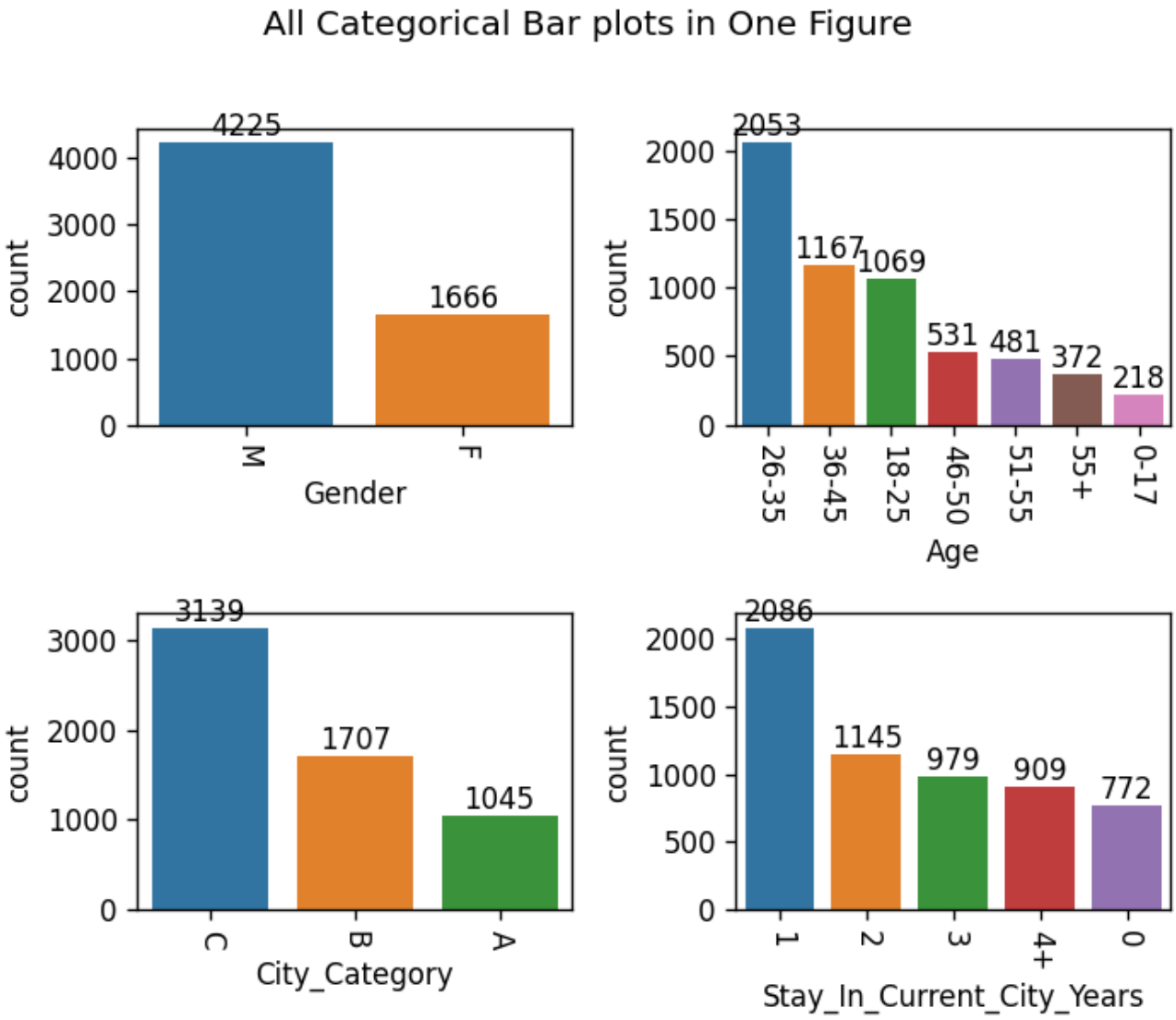
```
print(np.round(df_cust['Marital_Status'].value_counts(normalize = True) * 100))
```

```
0    58.0
1    42.0
Name: Marital_Status, dtype: float64
```

In [ ]:

```
catcols=['Gender','Age','City_Category','Stay_In_Current_City_Years']
_, axes = plt.subplots(nrows=2, ncols=2,sharex=False,sharey=False, figsize=(6, 5),dpi=120)

for ax,j in zip(axes.ravel(),catcols):
    sns.countplot(x=j, data=df_cust, ax=ax,order=df_cust[j].value_counts().index)
    ax.tick_params(axis ='x', rotation = -90)

    for label in ax.containers:
```

```
        ax.bar_label(label)

plt.suptitle('All Categorical Bar plots in One Figure', verticalalignment='bottom', fontsize=12)
plt.tight_layout()
plt.show()
```

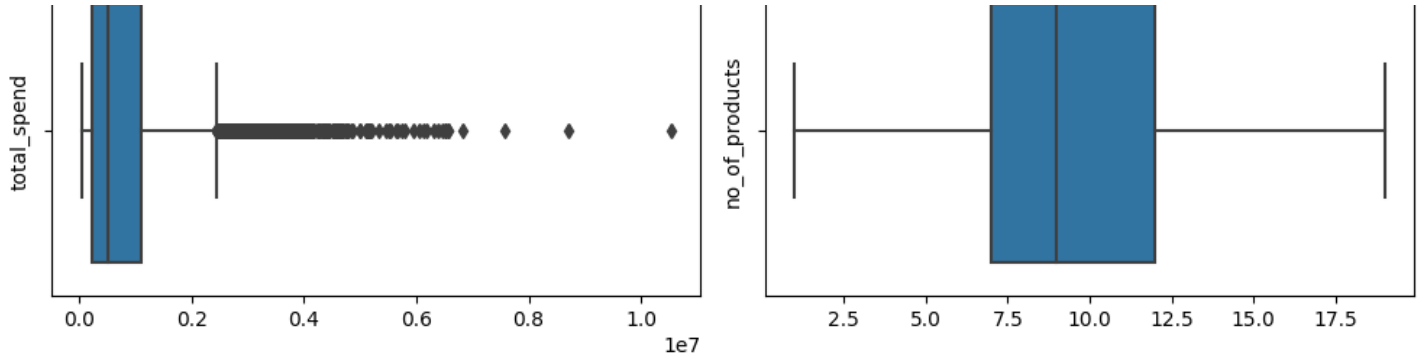


Insights on Categorical Variable

1. **4225 (72%) people are Male while 1666 (28%) customers are Female.**
2. **3417 (58%) customers are single while 2474 (42%) are married.**
3. **Majoity of customers fall in 26-25 age group followed by 36-45 group. Walmart Can include more products for this age group. Target more advertisement towards this age group.**
4. **Least no of customers in age group 0-17 and 46 and above. Can take meausres to improve their purchase behaviour by introducing new products for this age group.**
5. **Majority customers belong to C city category and least to city A. City C customers should be of prime focus as they generate most of the revenue.**
6. **Send discount offers to the people who have been staying less than a year in the city to increase their sales.**

In [ ]:

```
numcols=['total_spend','no_of_products']
fig, ax = plt.subplots(nrows=1, ncols=2, figsize=(10,3))
for ax, feat in zip(ax.ravel(),numcols):
    sns.boxplot(x=feat, data=df_cust, ax=ax)
    ax.set_xlabel('')
    ax.set_ylabel(feat)
fig.tight_layout()
```



In [35]:

```
def summ_stats(df,var,var2="dataset"):
  print("Summary stats for {0} are == \n ".format(var2))
  print('#'*20)
  print("min Value is = ",df[var].quantile(0))
  print("25th Percentile value is = ",df[var].quantile(.25))
  print("Median value is = ",df[var].quantile(.5))
  print("Mean value is = ",round(df[var].mean()))
  print("75th Percentile value is = ",df[var].quantile(.75))
  ul=df[var].quantile(.75)+((df[var].quantile(.75)-df[var].quantile(.25))*1.5)
  print('Upper outlier value is =',ul)
  print("99th Percentile value is = ",df[var].quantile(.99))
  print("Max value is = ",df[var].quantile(1))
  print('*'*20)

summ_stats(df_cust,"total_spend")
summ_stats(df_cust.loc[df['Gender']=='M'],"total_spend","Purchase Amount of Male")
summ_stats(df_cust.loc[df['Gender']=='F'],"total_spend","Purchase Amount of Female")
```

```
Summary stats for  dataset are ==

####################
min Value is =  46681.0
25th Percentile value is =  237678.0
Median value is =  521213.0
Mean value is =  865017
75th Percentile value is =  1119249.5
Upper outlier value is = 2441606.75
99th Percentile value is =  4418794.200000001
Max value is =  10536909.0
********************
Summary stats for Purchase Amount of Male are ==

####################
min Value is =  46681.0
25th Percentile value is =  241548.0
Median value is =  523983.0
Mean value is =  867220
75th Percentile value is =  1122962.0
Upper outlier value is = 2445083.0
99th Percentile value is =  4408730.400000005
Max value is =  7577756.0
********************
Summary stats for Purchase Amount of Female are ==

####################
min Value is =  52371.0
25th Percentile value is =  230187.0
Median value is =  519347.0
Mean value is =  857744
75th Percentile value is =  1096184.75
Upper outlier value is = 2395181.375
99th Percentile value is =  4494850.819999986
Max value is =  10536909.0
********************
```

In [37]:

```
def summ_stats(df,var,var2="dataset"):
  print("Summary stats for {0} are == \n ".format(var2))
  print('#'*20)
  print("min Value is = ",df[var].quantile(0))
  print("25th Percentile value is = ",df[var].quantile(.25))
  print("Median value is = ",df[var].quantile(.5))
  print("Mean value is = ",round(df[var].mean()))
  print("75th Percentile value is = ",df[var].quantile(.75))
  ul=df[var].quantile(.75)+((df[var].quantile(.75)-df[var].quantile(.25))*1.5)
  print('Upper outlier value is =',ul)
  print("99th Percentile value is = ",df[var].quantile(.99))
  print("Max value is = ",df[var].quantile(1))
  print('*'*20)

summ_stats(df_cust,"no_of_products")
summ_stats(df_cust.loc[df_cust['Gender']=='M'],"no_of_products","Purchase Amount of Male"
)
summ_stats(df_cust.loc[df_cust['Gender']=='F'],"no_of_products","Purchase Amount of Femal
e")
```

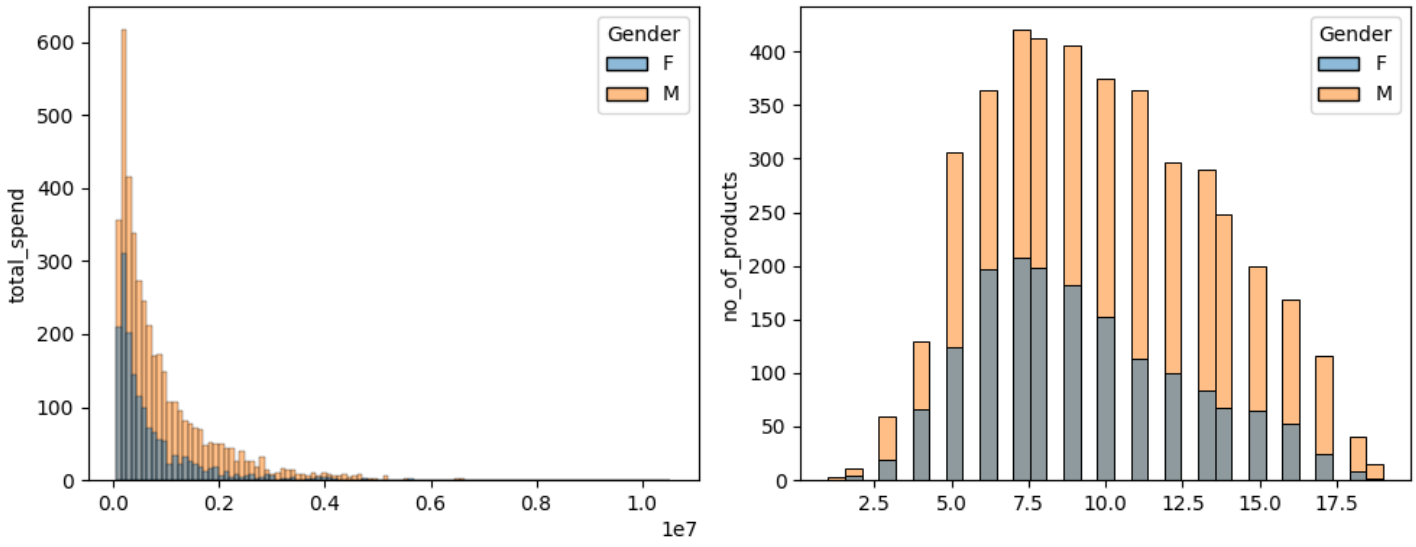
```
Summary stats for dataset are ==

####################
min Value is =  1.0
25th Percentile value is =  7.0
Median value is =  9.0
Mean value is =  10
75th Percentile value is =  12.0
Upper outlier value is = 19.5
99th Percentile value is =  18.0
Max value is =  19.0
********************
Summary stats for Purchase Amount of Male are ==

####################
min Value is =  1.0
25th Percentile value is =  7.0
Median value is =  10.0
Mean value is =  10
75th Percentile value is =  13.0
Upper outlier value is = 22.0
99th Percentile value is =  18.0
Max value is =  19.0
********************
Summary stats for Purchase Amount of Female are ==

####################
min Value is =  2.0
25th Percentile value is =  7.0
Median value is =  9.0
Mean value is =  9
75th Percentile value is =  11.0
Upper outlier value is = 17.0
99th Percentile value is =  17.0
Max value is =  19.0
********************
```

Insight on total_spend and Total products per user on Black friday

1. **Total spend follows a log normal distribution.**
2. **There are outliers present in the dataset for the total spend by each customer during black friday sales.**
3. **Median value for total spend on black friday per user for male and female is 5,23,983 and 5,19,347 respectively.**
4. **Median value for total products bought per user for male and female is 10 and 9 respectively.**

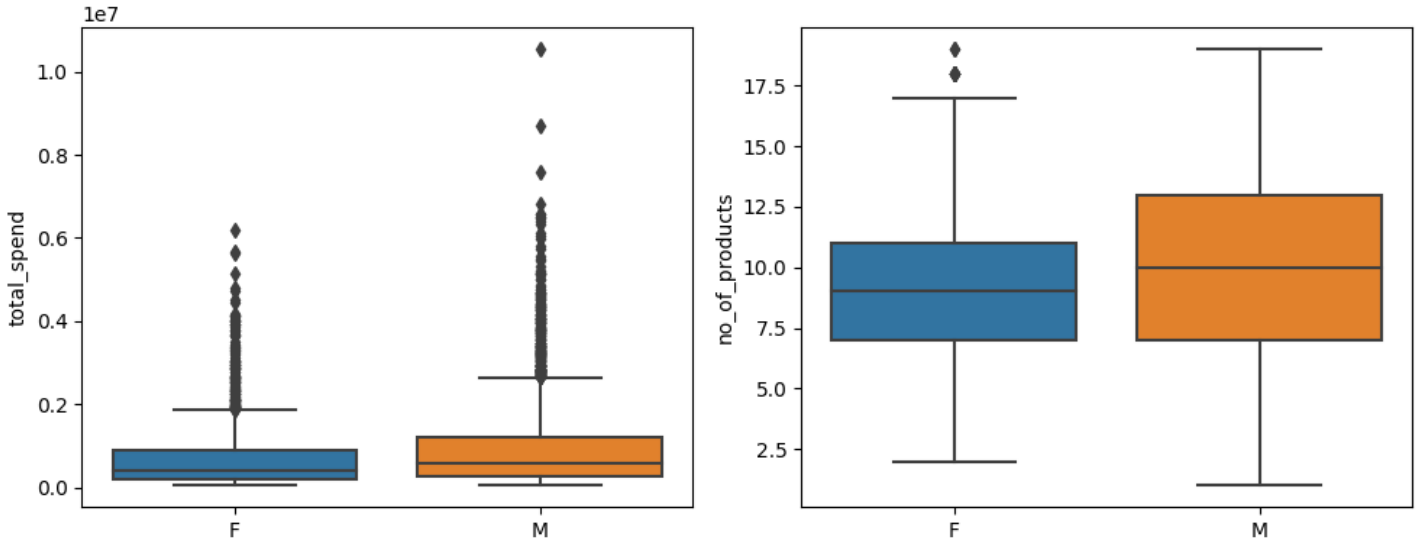
In [31]:

```
numcols=['total_spend','no_of_products']

fig, ax = plt.subplots(nrows=1, ncols=2, figsize=(10, 4))
for ax, feat in zip(ax.ravel(),numcols):
    sns.histplot(hue='Gender', x=feat, data=df_cust, ax=ax)
    ax.set_xlabel('')
    ax.set_ylabel(feat)
fig.tight_layout();
```



In [ ]:

```
numcols=['total_spend','no_of_products']

fig, ax = plt.subplots(nrows=1, ncols=2, figsize=(10, 4))
for ax, feat in zip(ax.ravel(),numcols):
    sns.boxplot(x='Gender', y=feat, data=df_cust, ax=ax)
    ax.set_xlabel('')
    ax.set_ylabel(feat)
fig.tight_layout();
```



In [40]:

```
df_male=df_cust.loc[df_cust['Gender']=='M','total_spend']
df_female=df_cust.loc[df_cust['Gender']=='F','total_spend']
print(df_male.shape)
print(df_female.shape)
```

```
(4225,)
(1666,)
```

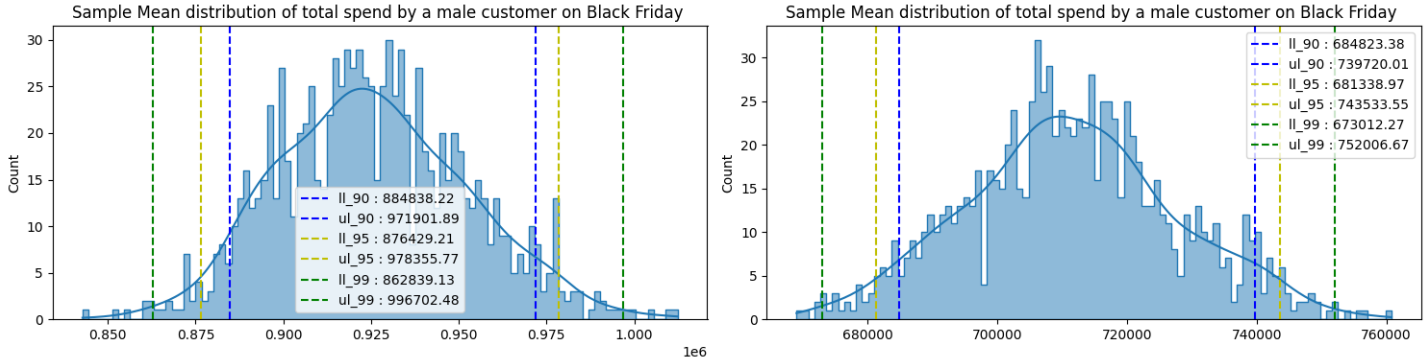
Q. Confidence intervals and distribution of the mean of the expenses by female and male customers ?

In [69]:

```
sample_mean_male=[]
sample_mean_female=[]
sample_size=1000
for i in range(1000):
  sample_mean_male.append(df_male.sample(sample_size).mean())
  sample_mean_female.append(df_female.sample(sample_size).mean())

print("Actual population mean for total spend per Male is  ",df_male.mean())
print("Approximate population mean for total spend per Male is  ",np.mean(sample_mean_mal
e))
print("Actual population mean for total spend per FeMale is  ",df_female.mean())
print("Approximate population mean for total spend per Female is  ",np.mean(sample_mean_f
emale))
print()


fig = plt.figure(figsize=(15, 4))

plt.subplot(1,2,1)
sns.histplot(sample_mean_male, kde = True, bins = 100, fill = True, element = 'step')

ll95 = np.percentile(sample_mean_male, 2.5)
ul95 = np.percentile(sample_mean_male, 97.5)
ll90 = np.percentile(sample_mean_male, 5)
ul90 = np.percentile(sample_mean_male, 95)
ll99 = np.percentile(sample_mean_male, 0.5)
ul99 = np.percentile(sample_mean_male, 99.5)

plt.axvline(ul90, label = f'll_90 : {round(ll90, 2)}', linestyle = '--', color = 'b')
plt.axvline(ll90, label = f'ul_90 : {round(ul90, 2)}', linestyle = '--', color = 'b')
plt.axvline(ul95, label = f'll_95 : {round(ll95, 2)}', linestyle = '--', color = 'y')
plt.axvline(ll95, label = f'ul_95 : {round(ul95, 2)}', linestyle = '--', color = 'y')
plt.axvline(ul99, label = f'll_99 : {round(ll99, 2)}', linestyle = '--', color = 'g')
plt.axvline(ll99, label = f'ul_99 : {round(ul99, 2)}', linestyle = '--', color = 'g')



plt.legend()
print("90 CI for Average Spending by a male customer on black friday is ({0},{1}) ".forma
t(ll90,ul90))
print("95 CI for Average Spending by a male customer on black friday is ({0},{1}) ".forma
t(ll95,ul95))
print("99 CI for Average Spending by a male customer on black friday is ({0},{1}) ".forma
t(ll99,ul99))
print("std devaiation of sample means is = ",np.std(sample_mean_male))
print()

plt.title("Sample Mean distribution of total spend by a male customer on Black Friday ")

plt.subplot(1,2,2)
sns.histplot(sample_mean_female, kde = True, bins = 100, fill = True, element = 'step')
ll95 = np.percentile(sample_mean_female, 2.5)
ul95 = np.percentile(sample_mean_female, 97.5)
ll90 = np.percentile(sample_mean_female, 5)
ul90 = np.percentile(sample_mean_female, 95)
ll99 = np.percentile(sample_mean_female, 0.5)
ul99 = np.percentile(sample_mean_female, 99.5)

plt.axvline(ul90, label = f'll_90 : {round(ll90, 2)}', linestyle = '--', color = 'b')
plt.axvline(ll90, label = f'ul_90 : {round(ul90, 2)}', linestyle = '--', color = 'b')
plt.axvline(ul95, label = f'll_95 : {round(ll95, 2)}', linestyle = '--', color = 'y')
plt.axvline(ll95, label = f'ul_95 : {round(ul95, 2)}', linestyle = '--', color = 'y')
plt.axvline(ul99, label = f'll_99 : {round(ll99, 2)}', linestyle = '--', color = 'g')
plt.axvline(ll99, label = f'ul_99 : {round(ul99, 2)}', linestyle = '--', color = 'g')
print()

print("90 CI for Average Spending by a female customer on black friday is ({0},{1}) ".for
mat(ll90,ul90))
print("95 CI for Average Spending by a female customer on black friday is ({0},{1}) ".for
mat(ll95,ul95))
print("99 CI for Average Spending by a female customer on black friday is ({0},{1}) ".for
```

```
mat(ll99,ul99))
print("std devaiation of sample means is = ",np.std(sample_mean_female))
plt.title("Sample Mean distribution of total spend by a male customer on Black Friday ")
plt.legend()

plt.tight_layout()
```

```
Actual population mean for total spend per Male is   925344.4023668639
Approximate population mean for total spend per Male is   925479.106617
Actual population mean for total spend per FeMale is   712024.3949579832
Approximate population mean for total spend per Female is   711445.6914319999

90 CI for Average Spending by a male customer on black friday is (884838.2244000001,97190
1.8884500001)
95 CI for Average Spending by a male customer on black friday is (876429.212925,978355.77
0525)
99 CI for Average Spending by a male customer on black friday is (862839.1272799999,99670
2.4834899999)
std devaiation of sample means is =  26933.125088257213


90 CI for Average Spending by a female customer on black friday is (684823.3811,739720.00
71500001)
95 CI for Average Spending by a female customer on black friday is (681338.9657500001,743
533.550075)
99 CI for Average Spending by a female customer on black friday is (673012.26824,752006.6
697750001)
std devaiation of sample means is =  16145.729653041702
```



With 95 % confidence we can say that average spending by a male customer on black friday will be in the range (876429,978355)

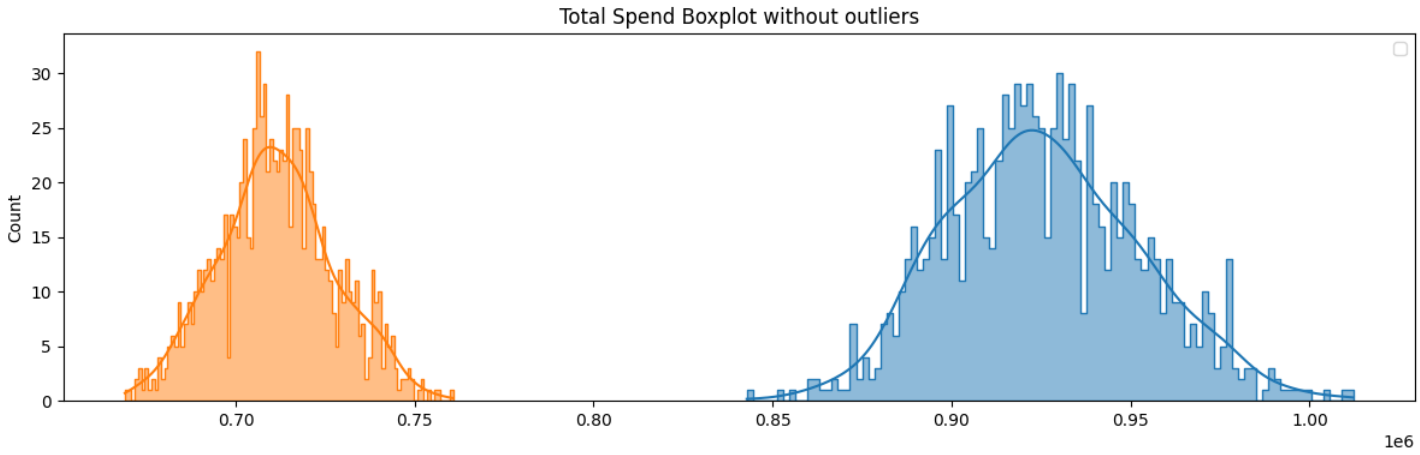
Q Are confidence intervals of average male and female spending overlapping? How can Walmart leverage this conclusion to make changes or improvements?

In [70]:

```
fig = plt.figure(figsize=(12, 4))

sns.histplot(sample_mean_male, kde = True, bins = 100, fill = True, element = 'step')

ll95 = np.percentile(sample_mean_male, 2.5)
ul95 = np.percentile(sample_mean_male, 97.5)
ll90 = np.percentile(sample_mean_male, 5)
ul90 = np.percentile(sample_mean_male, 95)
ll99 = np.percentile(sample_mean_male, 0.5)
ul99 = np.percentile(sample_mean_male, 99.5)


sns.histplot(sample_mean_female, kde = True, bins = 100, fill = True, element = 'step')
ll95 = np.percentile(sample_mean_female, 2.5)
ul95 = np.percentile(sample_mean_female, 97.5)
ll90 = np.percentile(sample_mean_female, 5)
ul90 = np.percentile(sample_mean_female, 95)
```

```
ll99 = np.percentile(sample_mean_female, 0.5)
ul99 = np.percentile(sample_mean_female, 99.5)


plt.legend()
plt.tight_layout()
```

```
WARNING:matplotlib.legend:No artists with labels found to put in legend.  Note that artis
ts whose label start with an underscore are ignored when legend() is called with no argum
ent.
```



Ans.

1. Confidence interval of average male and female customer spendings are not overlapping.
2. We can conclude that on average males customer spend
 more than a female customer.
3. Male customers are more likely to buy expensive products in comparison to female customers.
4. Walmart should target ads with expensive products to male customers. Female customers should be targetted with economical products.

Q Results when the same activity is performed for Married vs Unmarried (10 Points)

In [71]:

```
df_cust['Marital_Status'].value_counts()
```

Out[71]:

```
0    3417
1    2474
Name: Marital_Status, dtype: int64
```

In [74]:

```
df_marital=df_cust.loc[df_cust['Marital_Status']==1,'total_spend']
df_single=df_cust.loc[df_cust['Marital_Status']==0,'total_spend']
print(df_marital.shape)
print(df_single.shape)
```

```
(2474,)
(3417,)
```

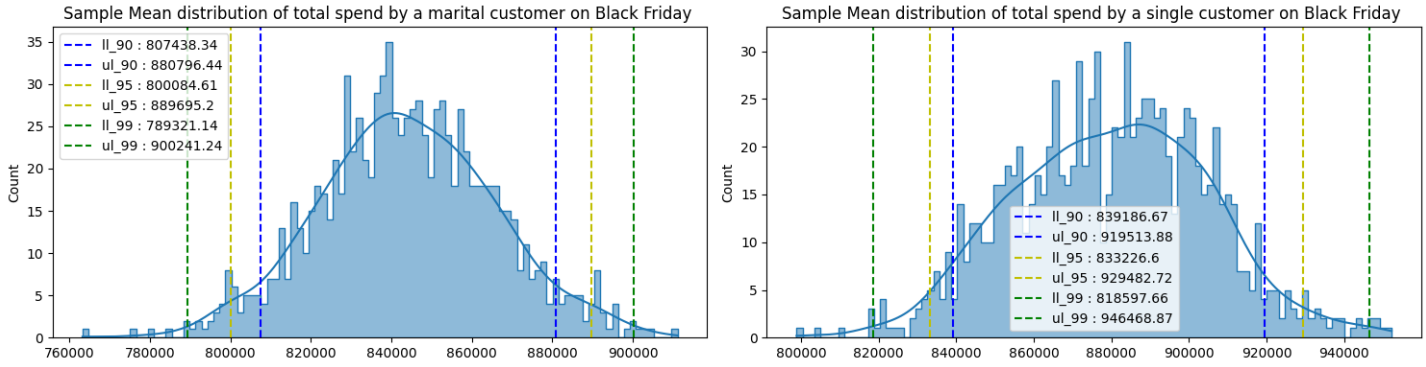
In [76]:

```
sample_mean_marital=[]
sample_mean_single=[]
sample_size=1000
for i in range(1000):
  sample_mean_marital.append(df_marital.sample(sample_size).mean())
  sample_mean_single.append(df_single.sample(sample_size).mean())
```

```
print("Actual population mean for total spend for a married customer is  ",df_marital.mea
n())
print("Approximate population mean for total spend for a married customer is  ",np.mean(s
ample_mean_marital))
print("Actual population mean for total spend for a single customer is  ",df_single.mean(
))
print("Approximate population mean for total spend for a single customer is  ",np.mean(s
ample_mean_single))
print()


fig = plt.figure(figsize=(15, 4))

plt.subplot(1,2,1)
sns.histplot(sample_mean_marital, kde = True, bins = 100, fill = True, element = 'step')

ll95 = np.percentile(sample_mean_marital, 2.5)
ul95 = np.percentile(sample_mean_marital, 97.5)
ll90 = np.percentile(sample_mean_marital, 5)
ul90 = np.percentile(sample_mean_marital, 95)
ll99 = np.percentile(sample_mean_marital, 0.5)
ul99 = np.percentile(sample_mean_marital, 99.5)

plt.axvline(ul90, label = f'll_90 : {round(ll90, 2)}', linestyle = '--', color = 'b')
plt.axvline(ll90, label = f'ul_90 : {round(ul90, 2)}', linestyle = '--', color = 'b')
plt.axvline(ul95, label = f'll_95 : {round(ll95, 2)}', linestyle = '--', color = 'y')
plt.axvline(ll95, label = f'ul_95 : {round(ul95, 2)}', linestyle = '--', color = 'y')
plt.axvline(ul99, label = f'll_99 : {round(ll99, 2)}', linestyle = '--', color = 'g')
plt.axvline(ll99, label = f'ul_99 : {round(ul99, 2)}', linestyle = '--', color = 'g')



plt.legend()
print("90 CI for Average Spending by a marital customer on black friday is ({0},{1}) ".fo
rmat(ll90,ul90))
print("95 CI for Average Spending by a marital customer on black friday is ({0},{1}) ".fo
rmat(ll95,ul95))
print("99 CI for Average Spending by a marital customer on black friday is ({0},{1}) ".fo
rmat(ll99,ul99))
print("std devaiation of sample means is = ",np.std(sample_mean_marital))
print()

plt.title("Sample Mean distribution of total spend by a marital customer on Black Friday
")

plt.subplot(1,2,2)
sns.histplot(sample_mean_single, kde = True, bins = 100, fill = True, element = 'step')
ll95 = np.percentile(sample_mean_single, 2.5)
ul95 = np.percentile(sample_mean_single, 97.5)
ll90 = np.percentile(sample_mean_single, 5)
ul90 = np.percentile(sample_mean_single, 95)
ll99 = np.percentile(sample_mean_single, 0.5)
ul99 = np.percentile(sample_mean_single, 99.5)

plt.axvline(ul90, label = f'll_90 : {round(ll90, 2)}', linestyle = '--', color = 'b')
plt.axvline(ll90, label = f'ul_90 : {round(ul90, 2)}', linestyle = '--', color = 'b')
plt.axvline(ul95, label = f'll_95 : {round(ll95, 2)}', linestyle = '--', color = 'y')
plt.axvline(ll95, label = f'ul_95 : {round(ul95, 2)}', linestyle = '--', color = 'y')
plt.axvline(ul99, label = f'll_99 : {round(ll99, 2)}', linestyle = '--', color = 'g')
plt.axvline(ll99, label = f'ul_99 : {round(ul99, 2)}', linestyle = '--', color = 'g')
print()
print("90 CI for Average Spending by a single customer on black friday is ({0},{1}) ".for
mat(ll90,ul90))
print("95 CI for Average Spending by a single customer on black friday is ({0},{1}) ".for
mat(ll95,ul95))
print("99 CI for Average Spending by a single customer on black friday is ({0},{1}) ".for
mat(ll99,ul99))
print("std devaiation of sample means is = ",np.std(sample_mean_single))
plt.title("Sample Mean distribution of total spend by a single customer on Black Friday "
)
plt.legend()
```

```
plt.tight_layout()
```

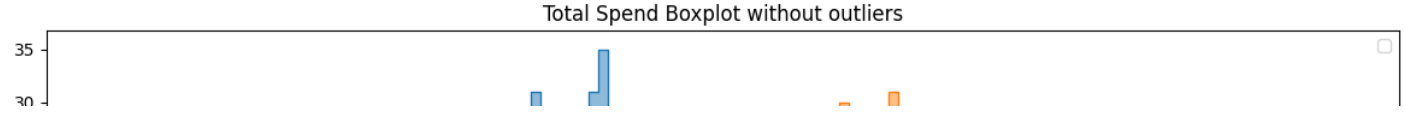
```
Actual population mean for total spend for a married customer is   843526.7966855295
Approximate population mean for total spend for a married customer is   843860.821709
Actual population mean for total spend for a single customer is   880575.7819724905
Approximate population mean for total spend for a single customer is   879924.780052

90 CI for Average Spending by a marital customer on black friday is (807438.34375,880796.
4443)
95 CI for Average Spending by a marital customer on black friday is (800084.6087999999,88
9695.2020249999)
99 CI for Average Spending by a marital customer on black friday is (789321.143355,900241
.23829)
std devaiation of sample means is =  21961.648038685948


90 CI for Average Spending by a single customer on black friday is (839186.66965,919513.8
8185)
95 CI for Average Spending by a single customer on black friday is (833226.59905,929482.7
17875)
99 CI for Average Spending by a single customer on black friday is (818597.660425,946468.
8749)
std devaiation of sample means is =  25134.09491024556
```

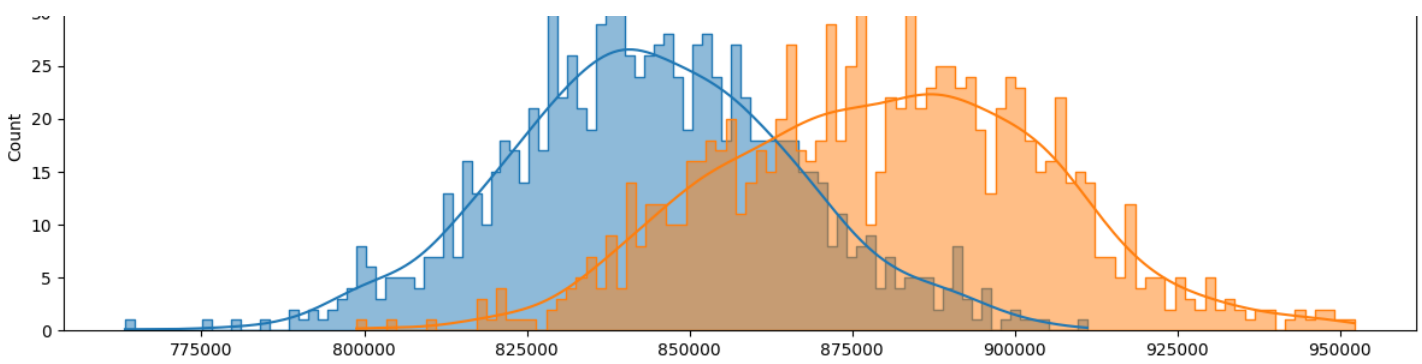


In [77]:

```
fig = plt.figure(figsize=(12, 4))

sns.histplot(sample_mean_marital, kde = True, bins = 100, fill = True, element = 'step')

ll95 = np.percentile(sample_mean_marital, 2.5)
ul95 = np.percentile(sample_mean_marital, 97.5)
ll90 = np.percentile(sample_mean_marital, 5)
ul90 = np.percentile(sample_mean_marital, 95)
ll99 = np.percentile(sample_mean_marital, 0.5)
ul99 = np.percentile(sample_mean_marital, 99.5)


sns.histplot(sample_mean_single, kde = True, bins = 100, fill = True, element = 'step')
ll95 = np.percentile(sample_mean_single, 2.5)
ul95 = np.percentile(sample_mean_single, 97.5)
ll90 = np.percentile(sample_mean_single, 5)
ul90 = np.percentile(sample_mean_single, 95)
ll99 = np.percentile(sample_mean_single, 0.5)
ul99 = np.percentile(sample_mean_single, 99.5)

plt.title("Total Spend Boxplot without outliers")

plt.legend()
plt.tight_layout()
```

```
WARNING:matplotlib.legend:No artists with labels found to put in legend.  Note that artis
ts whose label start with an underscore are ignored when legend() is called with no argum
ent.
```





1. **Sample mean's mean for total spend for a married and single customer is 843860 and 879924.**

1. **95 CI for Average Spending by a marital customer and single customer on black friday is (800084 ,889695) and (833226,929482) respectively.**
2. **Single customers have higher average mean spend then married customers. 95 Confidence interval is also in a higher spend range for single customers then married customers.**
3. **There is a lot of overlap b/w avg spend distribution for married and single customers with single customere lying on thr right tail of distribution i.e**
4. **There is no significant difference in spending habits of married and single customers based on amount spend.**

Q Results when the same activity is performed for Age

In [78]:

```
df_cust['Age'].value_counts()
```

Out[78]:

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

In [81]:

```
df_26_35=df_cust.loc[df_cust['Age']=='26-35','total_spend']
df_36_45=df_cust.loc[df_cust['Age']=='36-45','total_spend']
df_18_25=df_cust.loc[df_cust['Age']=='18-25','total_spend']
df_46_50=df_cust.loc[df_cust['Age']=='46-50','total_spend']
df_51_55=df_cust.loc[df_cust['Age']=='51-55','total_spend']
df_55=df_cust.loc[df_cust['Age']=='55+','total_spend']
df_17=df_cust.loc[df_cust['Age']=='0-17','total_spend']

print(df_26_35.shape)
print(df_36_45.shape)
print(df_18_25.shape)
print(df_46_50.shape)
print(df_51_55.shape)
print(df_55.shape)
print(df_17.shape)
```

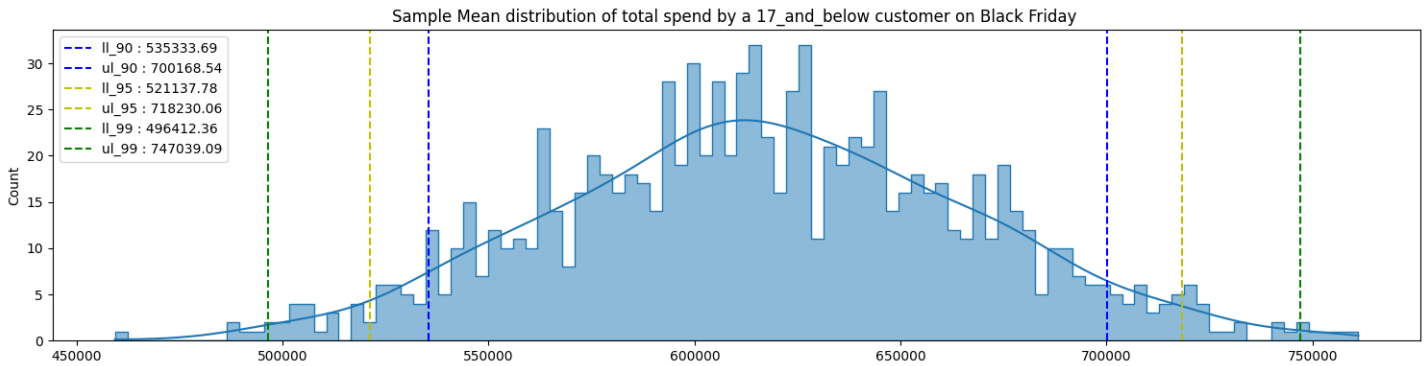
```
(2053,)
(1167,)
(1069,)
(531,)
(481,)
(372,)
(218,)
```

In [82]:

```
sample_mean_17_and_below=[]
```

```
sample_mean_17_and_below=[]
sample_size=100
for i in range(1000):
  sample_mean_17_and_below.append(df_17.sample(sample_size).mean())


print("Actual population mean for total spend for a 17_and_below customer is  ",df_17.mea
n())
print("Approximate population mean for total spend for a 17_and_below customer is  ",np.m
ean(sample_mean_17_and_below))
print()


fig = plt.figure(figsize=(15, 4))



sns.histplot(sample_mean_17_and_below, kde = True, bins = 100, fill = True, element = 'st
ep')
ll95 = np.percentile(sample_mean_17_and_below, 2.5)
ul95 = np.percentile(sample_mean_17_and_below, 97.5)
ll90 = np.percentile(sample_mean_17_and_below, 5)
ul90 = np.percentile(sample_mean_17_and_below, 95)
ll99 = np.percentile(sample_mean_17_and_below, 0.5)
ul99 = np.percentile(sample_mean_17_and_below, 99.5)

plt.axvline(ul90, label = f'll_90 : {round(ll90, 2)}', linestyle = '--', color = 'b')
plt.axvline(ll90, label = f'ul_90 : {round(ul90, 2)}', linestyle = '--', color = 'b')
plt.axvline(ul95, label = f'll_95 : {round(ll95, 2)}', linestyle = '--', color = 'y')
plt.axvline(ll95, label = f'ul_95 : {round(ul95, 2)}', linestyle = '--', color = 'y')
plt.axvline(ul99, label = f'll_99 : {round(ll99, 2)}', linestyle = '--', color = 'g')
plt.axvline(ll99, label = f'ul_99 : {round(ul99, 2)}', linestyle = '--', color = 'g')
print()
print("90 CI for Average Spending by a 17_and_below customer on black friday is ({0},{1})
".format(ll90,ul90))
print("95 CI for Average Spending by a 17_and_below customer on black friday is ({0},{1})
".format(ll95,ul95))
print("99 CI for Average Spending by a 17_and_below customer on black friday is ({0},{1})
".format(ll99,ul99))
print("std devaiation of sample means is = ",np.std(sample_mean_17_and_below))
plt.title("Sample Mean distribution of total spend by a 17_and_below customer on Black Fr
iday ")
plt.legend()

plt.tight_layout()
```

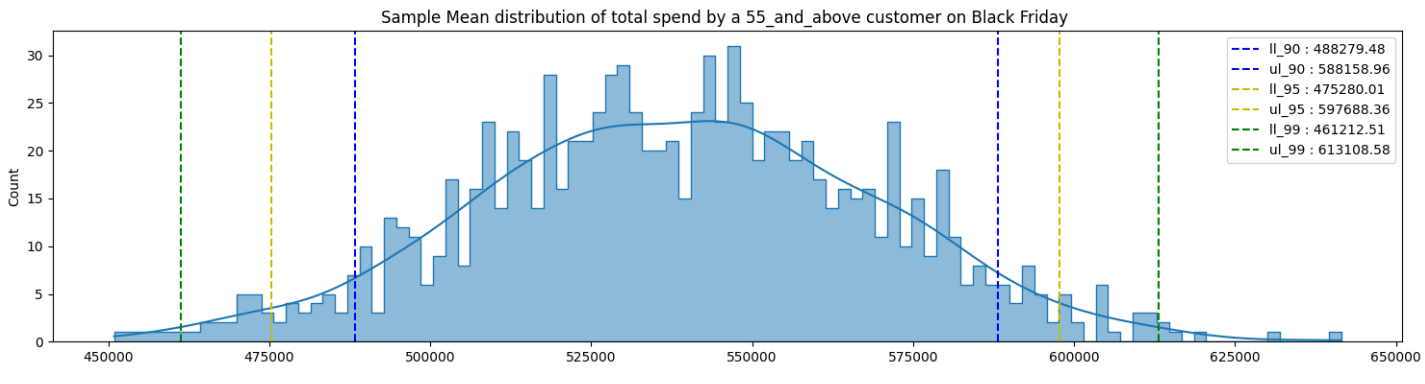
```
Actual population mean for total spend for a 17_and_below customer is   618867.8119266055
Approximate population mean for total spend for a 17_and_below customer is   616613.44645


90 CI for Average Spending by a 17_and_below customer on black friday is (535333.695,7001
68.5375)
95 CI for Average Spending by a 17_and_below customer on black friday is (521137.78225000
005,718230.0615)
99 CI for Average Spending by a 17_and_below customer on black friday is (496412.3576,747
039.08895)
std devaiation of sample means is =  50367.46869615543
```



In [83]:

```
sample_mean_55_and_above=[]
sample_size=200
for i in range(1000):
  sample_mean_55_and_above.append(df_55.sample(sample_size).mean())


print("Actual population mean for total spend for a 55_and_above customer is  ",df_55.mea
n())
print("Approximate population mean for total spend for a 55_and_above customer is  ",np.m
ean(sample_mean_55_and_above))
print()


fig = plt.figure(figsize=(15, 4))



sns.histplot(sample_mean_55_and_above, kde = True, bins = 100, fill = True, element = 'st
ep')
ll95 = np.percentile(sample_mean_55_and_above, 2.5)
ul95 = np.percentile(sample_mean_55_and_above, 97.5)
ll90 = np.percentile(sample_mean_55_and_above, 5)
ul90 = np.percentile(sample_mean_55_and_above, 95)
ll99 = np.percentile(sample_mean_55_and_above, 0.5)
ul99 = np.percentile(sample_mean_55_and_above, 99.5)

plt.axvline(ul90, label = f'll_90 : {round(ll90, 2)}', linestyle = '--', color = 'b')
plt.axvline(ll90, label = f'ul_90 : {round(ul90, 2)}', linestyle = '--', color = 'b')
plt.axvline(ul95, label = f'll_95 : {round(ll95, 2)}', linestyle = '--', color = 'y')
plt.axvline(ll95, label = f'ul_95 : {round(ul95, 2)}', linestyle = '--', color = 'y')
plt.axvline(ul99, label = f'll_99 : {round(ll99, 2)}', linestyle = '--', color = 'g')
plt.axvline(ll99, label = f'ul_99 : {round(ul99, 2)}', linestyle = '--', color = 'g')
print()
print("90 CI for Average Spending by a 55_and_above customer on black friday is ({0},{1})
".format(ll90,ul90))
print("95 CI for Average Spending by a 55_and_above customer on black friday is ({0},{1})
".format(ll95,ul95))
print("99 CI for Average Spending by a 55_and_above customer on black friday is ({0},{1})
".format(ll99,ul99))
print("std devaiation of sample means is = ",np.std(sample_mean_55_and_above))
plt.title("Sample Mean distribution of total spend by a 55_and_above customer on Black Fr
iday ")
plt.legend()

plt.tight_layout()
```

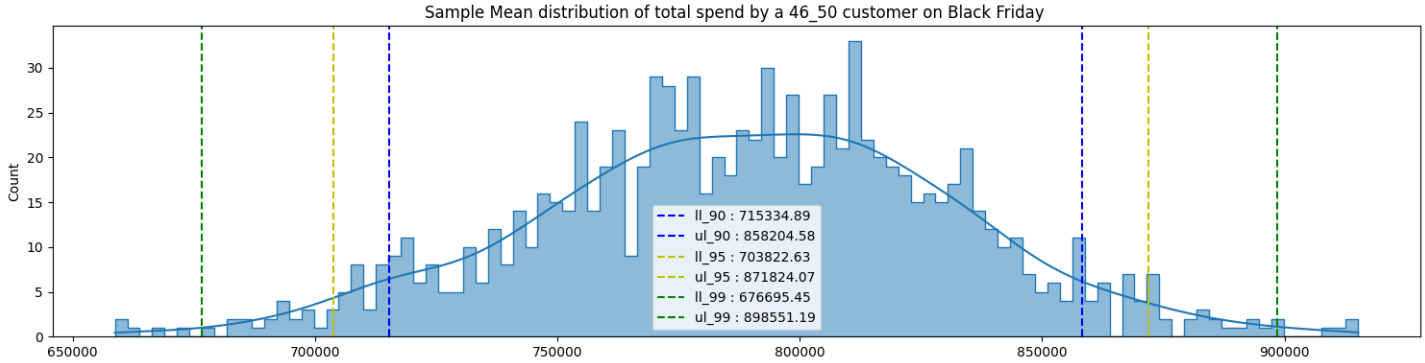
```
Actual population mean for total spend for a 55_and_above customer is   539697.2446236559
Approximate population mean for total spend for a 55_and_above customer is   538247.39767
49999


90 CI for Average Spending by a 55_and_above customer on black friday is (488279.47525,58
8158.96025)
95 CI for Average Spending by a 55_and_above customer on black friday is (475280.006625,5
97688.363125)
99 CI for Average Spending by a 55_and_above customer on black friday is (461212.509775,6
13108.580825)
std devaiation of sample means is =  30828.627106180164
```



In [84]:

```
sample_mean_46_50=[]
sample_size=250
for i in range(1000):
  sample_mean_46_50.append(df_46_50.sample(sample_size).mean())


print("Actual population mean for total spend for a 46_50 customer is  ",df_46_50.mean())
print("Approximate population mean for total spend for a 46_50 customer is  ",np.mean(sam
ple_mean_46_50))
print()


fig = plt.figure(figsize=(15, 4))



sns.histplot(sample_mean_46_50, kde = True, bins = 100, fill = True, element = 'step')
ll95 = np.percentile(sample_mean_46_50, 2.5)
ul95 = np.percentile(sample_mean_46_50, 97.5)
ll90 = np.percentile(sample_mean_46_50, 5)
ul90 = np.percentile(sample_mean_46_50, 95)
ll99 = np.percentile(sample_mean_46_50, 0.5)
ul99 = np.percentile(sample_mean_46_50, 99.5)

plt.axvline(ul90, label = f'll_90 : {round(ll90, 2)}', linestyle = '--', color = 'b')
plt.axvline(ll90, label = f'ul_90 : {round(ul90, 2)}', linestyle = '--', color = 'b')
plt.axvline(ul95, label = f'll_95 : {round(ll95, 2)}', linestyle = '--', color = 'y')
plt.axvline(ll95, label = f'ul_95 : {round(ul95, 2)}', linestyle = '--', color = 'y')
plt.axvline(ul99, label = f'll_99 : {round(ll99, 2)}', linestyle = '--', color = 'g')
plt.axvline(ll99, label = f'ul_99 : {round(ul99, 2)}', linestyle = '--', color = 'g')
print()
print("90 CI for Average Spending by a 46_50 customer on black friday is ({0},{1}) ".form
at(ll90,ul90))
print("95 CI for Average Spending by a 46_50 customer on black friday is ({0},{1}) ".form
at(ll95,ul95))
print("99 CI for Average Spending by a 46_50 customer on black friday is ({0},{1}) ".form
at(ll99,ul99))
print("std devaiation of sample means is = ",np.std(sample_mean_46_50))
plt.title("Sample Mean distribution of total spend by a 46_50 customer on Black Friday ")
plt.legend()

plt.tight_layout()
```

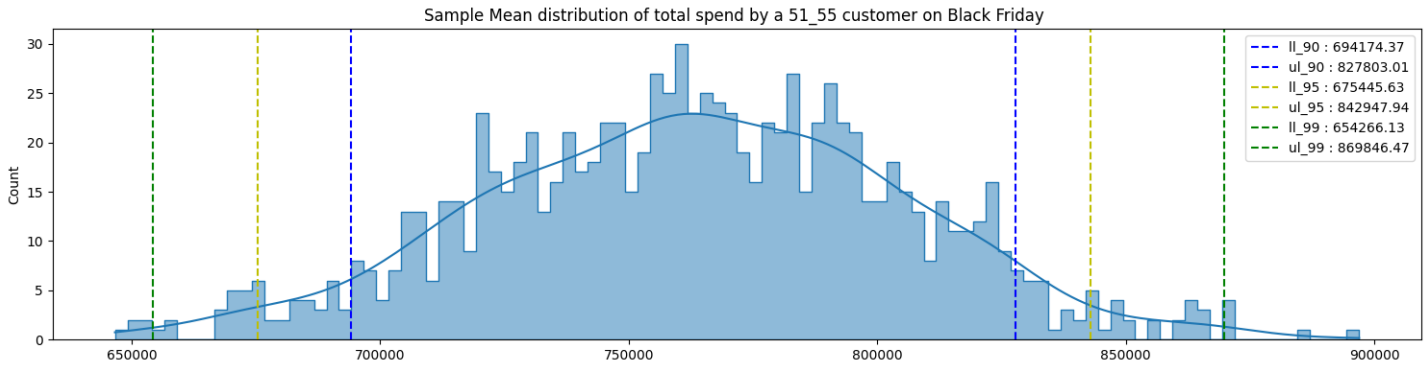
```
Actual population mean for total spend for a 46_50 customer is   792548.7815442561
Approximate population mean for total spend for a 46_50 customer is   788888.125856


90 CI for Average Spending by a 46_50 customer on black friday is (715334.8928,858204.575
)
95 CI for Average Spending by a 46_50 customer on black friday is (703822.6263,871824.073
4999999)
99 CI for Average Spending by a 46_50 customer on black friday is (676695.44814,898551.18
784)
std devaiation of sample means is =  43061.771647289774
```



In [85]:

```
sample_mean_51_55=[]
sample_size=200
for i in range(1000):
  sample_mean_51_55.append(df_51_55.sample(sample_size).mean())


print("Actual population mean for total spend for a 51_55 customer is  ",df_51_55.mean())
print("Approximate population mean for total spend for a 51_55 customer is  ",np.mean(sam
ple_mean_51_55))
print()


fig = plt.figure(figsize=(15, 4))



sns.histplot(sample_mean_51_55, kde = True, bins = 100, fill = True, element = 'step')
ll95 = np.percentile(sample_mean_51_55, 2.5)
ul95 = np.percentile(sample_mean_51_55, 97.5)
ll90 = np.percentile(sample_mean_51_55, 5)
ul90 = np.percentile(sample_mean_51_55, 95)
ll99 = np.percentile(sample_mean_51_55, 0.5)
ul99 = np.percentile(sample_mean_51_55, 99.5)

plt.axvline(ul90, label = f'll_90 : {round(ll90, 2)}', linestyle = '--', color = 'b')
plt.axvline(ll90, label = f'ul_90 : {round(ul90, 2)}', linestyle = '--', color = 'b')
plt.axvline(ul95, label = f'll_95 : {round(ll95, 2)}', linestyle = '--', color = 'y')
plt.axvline(ll95, label = f'ul_95 : {round(ul95, 2)}', linestyle = '--', color = 'y')
plt.axvline(ul99, label = f'll_99 : {round(ll99, 2)}', linestyle = '--', color = 'g')
plt.axvline(ll99, label = f'ul_99 : {round(ul99, 2)}', linestyle = '--', color = 'g')
print()
print("90 CI for Average Spending by a 51_55 customer on black friday is ({0},{1}) ".form
at(ll90,ul90))
print("95 CI for Average Spending by a 51_55 customer on black friday is ({0},{1}) ".form
at(ll95,ul95))
print("99 CI for Average Spending by a 51_55 customer on black friday is ({0},{1}) ".form
at(ll99,ul99))
print("std devaiation of sample means is = ",np.std(sample_mean_51_55))
plt.title("Sample Mean distribution of total spend by a 51_55 customer on Black Friday ")
plt.legend()

plt.tight_layout()
```

```
Actual population mean for total spend for a 51_55 customer is   763200.9230769231
Approximate population mean for total spend for a 51_55 customer is   763122.02053


90 CI for Average Spending by a 51_55 customer on black friday is (694174.3655,827803.008
75)
95 CI for Average Spending by a 51_55 customer on black friday is (675445.62975,842947.93
525)
99 CI for Average Spending by a 51_55 customer on black friday is (654266.1269,869846.469
7)
std devaiation of sample means is =  42008.77265427249
```

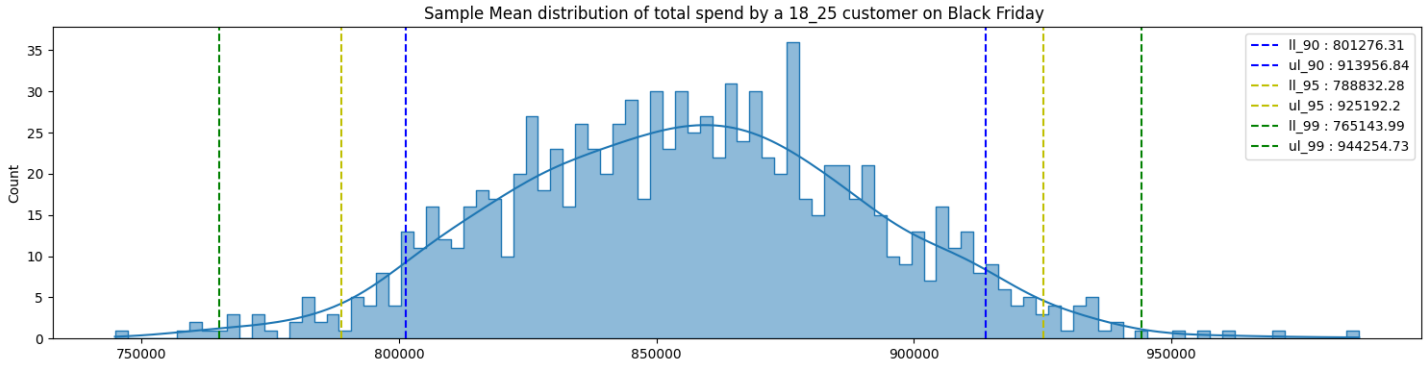


In [86]:

```
sample_mean_18_25=[]
sample_size=400
```

```
for i in range(1000):
  sample_mean_18_25.append(df_18_25.sample(sample_size).mean())


print("Actual population mean for total spend for a 18_25 customer is  ",df_18_25.mean())
print("Approximate population mean for total spend for a 18_25 customer is  ",np.mean(sam
ple_mean_18_25))
print()


fig = plt.figure(figsize=(15, 4))



sns.histplot(sample_mean_18_25, kde = True, bins = 100, fill = True, element = 'step')
ll95 = np.percentile(sample_mean_18_25, 2.5)
ul95 = np.percentile(sample_mean_18_25, 97.5)
ll90 = np.percentile(sample_mean_18_25, 5)
ul90 = np.percentile(sample_mean_18_25, 95)
ll99 = np.percentile(sample_mean_18_25, 0.5)
ul99 = np.percentile(sample_mean_18_25, 99.5)

plt.axvline(ul90, label = f'll_90 : {round(ll90, 2)}', linestyle = '--', color = 'b')
plt.axvline(ll90, label = f'ul_90 : {round(ul90, 2)}', linestyle = '--', color = 'b')
plt.axvline(ul95, label = f'll_95 : {round(ll95, 2)}', linestyle = '--', color = 'y')
plt.axvline(ll95, label = f'ul_95 : {round(ul95, 2)}', linestyle = '--', color = 'y')
plt.axvline(ul99, label = f'll_99 : {round(ll99, 2)}', linestyle = '--', color = 'g')
plt.axvline(ll99, label = f'ul_99 : {round(ul99, 2)}', linestyle = '--', color = 'g')
print()
print("90 CI for Average Spending by a 18_25 customer on black friday is ({0},{1}) ".form
at(ll90,ul90))
print("95 CI for Average Spending by a 18_25 customer on black friday is ({0},{1}) ".form
at(ll95,ul95))
print("99 CI for Average Spending by a 18_25 customer on black friday is ({0},{1}) ".form
at(ll99,ul99))
print("std devaiation of sample means is = ",np.std(sample_mean_18_25))
plt.title("Sample Mean distribution of total spend by a 18_25 customer on Black Friday ")
plt.legend()

plt.tight_layout()
```

```
Actual population mean for total spend for a 18_25 customer is   854863.119738073
Approximate population mean for total spend for a 18_25 customer is   856124.327935


90 CI for Average Spending by a 18_25 customer on black friday is (801276.309,913956.8396
25)
95 CI for Average Spending by a 18_25 customer on black friday is (788832.2761875,925192.
20125)
99 CI for Average Spending by a 18_25 customer on black friday is (765143.9888,944254.732
1)
std devaiation of sample means is =  35466.14826476161
```

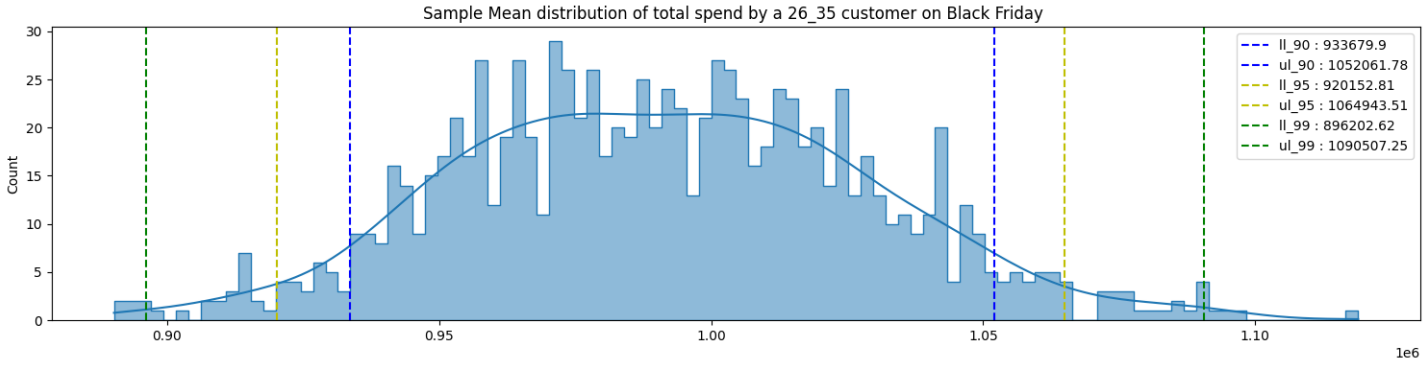


In [92]:

```
sample_mean_26_35=[]
sample_size=500
for i in range(1000):
  sample_mean_26_35.append(df_26_35.sample(sample_size).mean())
```

```
print("Actual population mean for total spend for a 26_35 customer is  ",df_26_35.mean())
print("Approximate population mean for total spend for a 26_35 customer is  ",np.mean(sam
ple_mean_26_35))
print()


fig = plt.figure(figsize=(15, 4))



sns.histplot(sample_mean_26_35, kde = True, bins = 100, fill = True, element = 'step')
ll95 = np.percentile(sample_mean_26_35, 2.5)
ul95 = np.percentile(sample_mean_26_35, 97.5)
ll90 = np.percentile(sample_mean_26_35, 5)
ul90 = np.percentile(sample_mean_26_35, 95)
ll99 = np.percentile(sample_mean_26_35, 0.5)
ul99 = np.percentile(sample_mean_26_35, 99.5)

plt.axvline(ul90, label = f'll_90 : {round(ll90, 2)}', linestyle = '--', color = 'b')
plt.axvline(ll90, label = f'ul_90 : {round(ul90, 2)}', linestyle = '--', color = 'b')
plt.axvline(ul95, label = f'll_95 : {round(ll95, 2)}', linestyle = '--', color = 'y')
plt.axvline(ll95, label = f'ul_95 : {round(ul95, 2)}', linestyle = '--', color = 'y')
plt.axvline(ul99, label = f'll_99 : {round(ll99, 2)}', linestyle = '--', color = 'g')
plt.axvline(ll99, label = f'ul_99 : {round(ul99, 2)}', linestyle = '--', color = 'g')
print()
print("90 CI for Average Spending by a 26_35 customer on black friday is ({0},{1}) ".form
at(ll90,ul90))
print("95 CI for Average Spending by a 26_35 customer on black friday is ({0},{1}) ".form
at(ll95,ul95))
print("99 CI for Average Spending by a 26_35 customer on black friday is ({0},{1}) ".form
at(ll99,ul99))
print("std devaiation of sample means is = ",np.std(sample_mean_26_35))
plt.title("Sample Mean distribution of total spend by a 26_35 customer on Black Friday ")
plt.legend()

plt.tight_layout()
```

```
Actual population mean for total spend for a 26_35 customer is   989659.3170969313
Approximate population mean for total spend for a 26_35 customer is   991312.0958199999


90 CI for Average Spending by a 26_35 customer on black friday is (933679.9046,1052061.77
88)
95 CI for Average Spending by a 26_35 customer on black friday is (920152.8148,1064943.51
465)
99 CI for Average Spending by a 26_35 customer on black friday is (896202.6237,1090507.25
299)
std devaiation of sample means is =  37645.81093707861
```

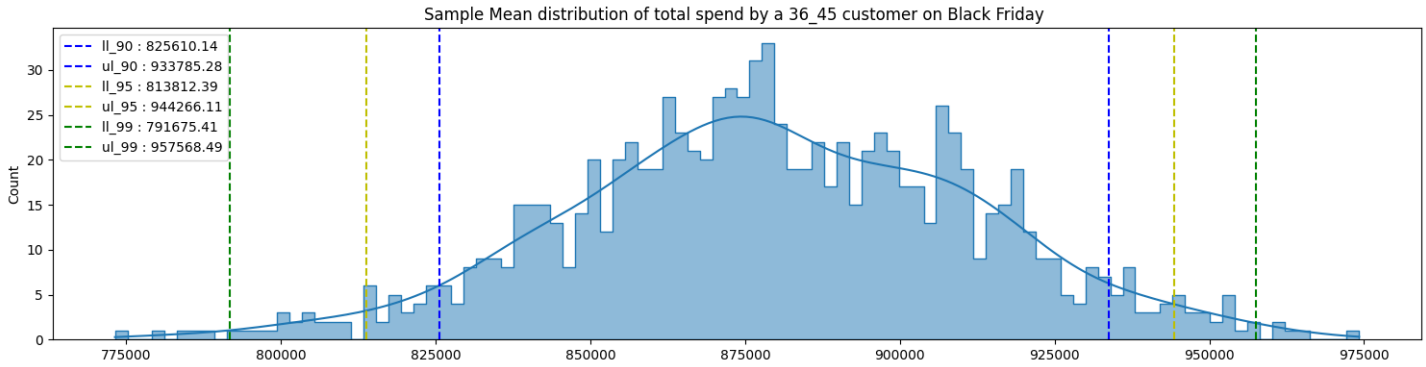


In [93]:

```
sample_mean_36_45=[]
sample_size=500
for i in range(1000):
  sample_mean_36_45.append(df_36_45.sample(sample_size).mean())
```

```
print("Actual population mean for total spend for a 36_45 customer is  ",df_36_45.mean())
print("Approximate population mean for total spend for a 36_45 customer is  ",np.mean(sam
ple_mean_36_45))
print()


fig = plt.figure(figsize=(15, 4))



sns.histplot(sample_mean_36_45, kde = True, bins = 100, fill = True, element = 'step')
ll95 = np.percentile(sample_mean_36_45, 2.5)
ul95 = np.percentile(sample_mean_36_45, 97.5)
ll90 = np.percentile(sample_mean_36_45, 5)
ul90 = np.percentile(sample_mean_36_45, 95)
ll99 = np.percentile(sample_mean_36_45, 0.5)
ul99 = np.percentile(sample_mean_36_45, 99.5)

plt.axvline(ul90, label = f'll_90 : {round(ll90, 2)}', linestyle = '--', color = 'b')
plt.axvline(ll90, label = f'ul_90 : {round(ul90, 2)}', linestyle = '--', color = 'b')
plt.axvline(ul95, label = f'll_95 : {round(ll95, 2)}', linestyle = '--', color = 'y')
plt.axvline(ll95, label = f'ul_95 : {round(ul95, 2)}', linestyle = '--', color = 'y')
plt.axvline(ul99, label = f'll_99 : {round(ll99, 2)}', linestyle = '--', color = 'g')
plt.axvline(ll99, label = f'ul_99 : {round(ul99, 2)}', linestyle = '--', color = 'g')
print()
print("90 CI for Average Spending by a 36_45 customer on black friday is ({0},{1}) ".form
at(ll90,ul90))
print("95 CI for Average Spending by a 36_45 customer on black friday is ({0},{1}) ".form
at(ll95,ul95))
print("99 CI for Average Spending by a 36_45 customer on black friday is ({0},{1}) ".form
at(ll99,ul99))
print("std devaiation of sample means is = ",np.std(sample_mean_36_45))
plt.title("Sample Mean distribution of total spend by a 36_45 customer on Black Friday ")
plt.legend()

plt.tight_layout()
```

```
Actual population mean for total spend for a 36_45 customer is   879665.7103684661
Approximate population mean for total spend for a 36_45 customer is   879474.940502


90 CI for Average Spending by a 36_45 customer on black friday is (825610.1415,933785.277
2)
95 CI for Average Spending by a 36_45 customer on black friday is (813812.3916,944266.110
75)
99 CI for Average Spending by a 36_45 customer on black friday is (791675.40818,957568.48
605)
std devaiation of sample means is =  32723.214689248947
```



In [97]:

```
data = [['17_and_below',616613,(521137,718230)],['55_and_above',538247,(475280,597688)],
['46_50',788888,(703822,871824)],['51_55',763122,(675445,842947)],
        ['18_25',856124,(788832,925192)],['26_35',991312,(920152,1064943)],['36_45',8794
74,( 813812,944266)]]
df_age = pd.DataFrame(data,columns=['Age_Group','Mean','95thPercentConfidenceRange'])
df_age
```

Out[97]:

	Age_Group	Mean	95thPercentConfidenceRange
0	17_and_below	616613	(521137, 718230)
1	55_and_above	538247	(475280, 597688)
2	46_50	788888	(703822, 871824)
3	51_55	763122	(675445, 842947)
4	18_25	856124	(788832, 925192)
5	26_35	991312	(920152, 1064943)
6	36_45	879474	(813812, 944266)

In [98]:

```
df_age.sort_values(by=['Mean'],inplace=True)
df_age
```

Out[98]:

	Age_Group	Mean	95thPercentConfidenceRange
1	55_and_above	538247	(475280, 597688)
0	17_and_below	616613	(521137, 718230)
3	51_55	763122	(675445, 842947)
2	46_50	788888	(703822, 871824)
4	18_25	856124	(788832, 925192)
6	36_45	879474	(813812, 944266)
5	26_35	991312	(920152, 1064943)

1. Customers in 26_25 have highest avg total_Spend value per customer.
2. Customers in age group 55_and_above have lowest mean value for total spend per customer.
3. There is very less overlap between mean spending of age group 26-35 and 36-45 for 95 % CI range.

Answers

1.

Q Defining Problem Statement and Analyzing basic metrics (10 Points) Ans . If spending habits differ between male and female customers.

Q. Observations on shape of data, Ans. (550068 ,10) Total 5,50,068 products bought on Black Friday.

Q. Data types of all the attributes, Ans
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

Q.Conversion of categorical attributes to 'category' (If required), statistical summary : Ans. Coverted Occupation, Product Category and Marital Status int64 to Object.

Q. Value counts and unique attributes Ans 0 User_ID 5891 unique users. 1 Product_ID 3631 unique products. 2 Gender 4423 Males, 1666 Females 3 Age Total 7 age bins, 26-35(40%),36-45(20%),18-25(18%) 4 Occupation 20 unique product categories 5 City_Category A(27%),B(42%),C(31%) 6 Stay_In_Current_City_Years 0(14%) 1(35%) ,2(19%),3 (17%) or 4+(15%) 7 Marital_Status 0(59%) or 1(41%) 8 Product_Category 20 unique product categories 9 Purchase

2

2.

Q Missing Value & Outlier Detection Ans There are no missing value in dataset. There are outlier present for the varaible purchase and total spend per user when purchase aggregated per user.

3.

Q Business Insights based on Non- Graphical and Visual Analysis (10 Points) Final Insights

1. **Total 5891 unique customers and 3631 unique products.**
2. **Most sold product is P00265242, sold 1880 times. Increase the inventory of these products.**
3. **There are no null values in the dataset.**
4. **Cheapest product cost 12 , mean cost of a product is 9263.96, and max cost of a product is 23961.**
5. **Total 5,50,068 items sold.**
6. **Total 20 unique product categories and 20 differnent types of occupations. 7.Product Category 5,1,8 are sold the most. They should be kept in high visibility area of Walmart stores.**
7. **Mean,Median purchase amount for males is = 9438 and 8098 respectively.**
8. **Mean,Median purchase amount for females is = 8735 and 7914 respectively.**
9. **Clearly men on average are spending per transaction more then women. This could be due to number of reasons.**
10. **Men prefer expensive products.**
11. **Maybe, men are paid more than women hence spend more.**
12. **Target men with campaigns and advertisement of products with high price.**
13. **Women make more economical choices while buying products then men. #Value count Insight**
14. **75% transactions on Black friday are done by males.**
15. **4225 (72%) people are Male while 1666 (28%) customers are Female.**
16. **3417 (58%) customers are single while 2474 (42%) are married.**
17. **Most no (42%) transactions are done in City B.**
18. **59% transactions are done by single people.**
19. **Most no (40%) transactions are done BY people in age group 26-35.**
20. **35% transactions are done by people who have lived for one year in the city.**
21. **People with occupation 4(13%),0(13%),7(13%) transact the most on Black friday.**
22. **Product category 5 (27%),1 (26%),8 (21%) are sold the most during Black Friday. # Product insight**
23. **No of unique products sold are == 3631 25.No of unique products bought by Males are == 3588 26.No of unique products bought by Females are == 3367**

Product Category Insight

1. **Product Category 1 is most bought category by Male while 5 is the most bought category by females. #Insight on total_spend and Total products per user on Black friday**
2. **Total spend follows a log normal distribution.**
3. **There are outliers present in the dataset for the total spend by each customer during black friday sales.**
4. **Median value for total spend on black friday per user for male and female is 5,23,983 and 5,19,347 respectively.**
5. **Median value for total products bought per user for male and female is 10 and 9 respectively.**

Insights on Categorical Variable per user statistics

1. **4225 (72%) people are Male while 1666 (28%) customers are Female.**
2. **3417 (58%) customers are unmarried while 2474 (42%) are married.**
3. **Majoity of customers fall in 26-25 age group followed by 36-45 group. Walmart Can include more products for this age group. Target more advertisement towards this age group. Least no of 35. customers in age group 0-17 and 46 and above. Can take meausres to improve their purchase behaviour by introducing new products for this age group.**
4. **Majority customers belong to C city category and least to city A. City C customers should be of prime focus as they generate most of the revenue. Send discount offers to the people who have been staying less than a year in the city to increase their sales.**

4.

Q1. Are women spending more money per transaction than men? Why or Why not? Ans1. Mean,Median purchase amount for males is = 9438 and 8098 respectively. Mean,Median purchase amount for females is = 8735 and 7914 respectively.

Clearly men on average are spending per transaction more then women. This could be due to number of reasons.

1. **Men prefer expensive products.**
2. **Maybe, men are paid more than women hence spend more.**
3. **Men are targetted with campaigns and advertisement of products with high price.**
4. **Women make more sensible/conservative choices while buying products.**

Q2. Confidence intervals and distribution of the mean of the expenses by female and male customers (10 Points) Ans2.
Approximate population mean for total spend per Male is 925479 Approximate population mean for total spend per Female is 711445

95 CI for Average Spending by a male customer on black friday is (876429,978355) 95 CI for Average Spending by a female customer on black friday is (681338,743533)

Q3. Are confidence intervals of average male and female spending overlapping? How can Walmart leverage this conclusion to make changes or improvements?

1. **Confidence interval of average male and female customer spendings are not overlapping.**
2. **We can conclude that on average males customer spend more than a female customer.**
3. **Male customers are more likely to buy expensive products in comparison to female customers.**
4. **Walmart should target ads with expensive products to male customers. Female customers should be targetted with economical products.**

Q4. Results when the same activity is performed for Married vs Unmarried (10 Points)

1. **Sample mean's mean for total spend for a married and single customer is 843860 and 879924.**
2. **95 CI for Average Spending by a marital customer and single customer on black friday is (800084 ,889695) and (833226,929482) respectively.**
3. **Single customers have higher average mean spend then married customers. 95 Confidence interval is also in a higher spend range for single customers then married customers.**
4. **There is a lot of overlap b/w avg spend distribution for married and single customers with single customere lying on thr right tail of distribution i.e**
5. **There is no significant difference in spending habits of married and single customers based on amount spend.**

Q5. Results when the same activity is performed for Age

1. **Customers in 26_25 have highest avg total_Spend value per customer.**
2. **Customers in age group 55_and_above have lowest mean value for total spend per customer.**

   Age_Group Mean 95thPercentConfidenceRange 55_and_above 538247 (475280, 597688) 17_and_below 616613 (521137, 718230) 51_55 763122 (675445, 842947) 46_50 788888 (703822, 871824) 18_25 856124 (788832, 925192) 36_45 879474 (813812, 944266) 26_35 991312 (920152, 1064943)

5.

Q5 Final Insights (10 Points) - Illustrate the insights based on exploration and CLT

1. **Married and single people on average exhibit not too different purchase behaviour while shopping on black friday.**
2. **Male customers spend more on average then female customers while shopping on black friday.**
3. **People in age group 55 and above and 17 and below on average spend the lowest while shopping on black friday.**
4. **People in age group 26-35 and 36-45 on average spend the most while shopping on black friday. 5 P00265242 is the most sold product.**
5. **Product Category 5,1,8 are sold the most**
6. **75% transactions on Black friday are done by males.**

7. **Most no (42%) transactions are done in City B.**
8. **People with occupation 4(13%),0(13%),7(13%) transact the most on Black friday.**
9. **Product Category 1 is most bought category by Male while 5 is the most bought category by females.**
10. **As sample size increases there is lot less fluctuation between mean value of sample means.**

6.

Q6 Recommendations -->

1. **Target Men with campaigns and advertisement of products with high price.**
2. **Target Women with campaigns and advertisement of products with low price.**
3. **To increase the sales, keep more products liked by men as they make 75% of total balck friday sales.**
4. **Keep more stocks in CITY B of products due to most sales coming from CITY B.**
5. **Target people with Occupation 4,0 and 7 with more personalized advertisements and offers to increase the sales.**
6. **Product category 5,1,8 must be kept in high visibility area of Walmart stores to increase the sales.**
7. **Target male customer with product 1 while female with product 5.**
8. **P00265242 stocks should be available in excess quantity as the product has highest demand.**
9. **Send discount offers to the people who have been staying less than a year in the city to increase their sales.**
10. **Least no of customers in age group 0-17 and 46 and above. Can take meausres to improve their purchase behaviour by introducing new products for this age group.**