```
import gdown
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import norm
```

```
! gdown https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/293/original/walmart_data.csv
```

```
Downloading...
From: https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/293/original/walmart_data.csv
To: /content/walmart_data.csv
100% 23.0M/23.0M [00:00<00:00, 67.9MB/s]
```

```
df = pd.read_csv('walmart_data.csv')
df
```

| | User_ID | Product_ID | Gender | Age | Occupation | City_Category | Stay_In_Current_City_Years | Marital_Status | Product_Ca |
|---|---|---|---|---|---|---|---|---|---|
| **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 | |
| **3** | 1000001 | P00085442 | F | 0-17 | 10 | A | 2 | 0 | |
| **4** | 1000002 | P00285442 | M | 55+ | 16 | C | 4+ | 0 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | |
| **550063** | 1006033 | P00372445 | M | 51-55 | 13 | B | 1 | 1 | |
| **550064** | 1006035 | P00375436 | F | 26-35 | 1 | C | 3 | 0 | |

```
df.head()
```

| | User_ID | Product_ID | Gender | Age | Occupation | City_Category | Stay_In_Current_City_Years | Marital_Status | Product_Category |
|---|---|---|---|---|---|---|---|---|---|
| **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: |

```
#We have total 10 Columns
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
```

```
#Data is Notnull or we have 0 cell with null value
df.isnull().sum(axis=0)
```

```
User_ID                       0
Product_ID                    0
Gender                        0
Age                           0
Occupation                    0
City_Category                 0
Stay_In_Current_City_Years    0
Marital_Status                0
Product_Category              0
Purchase                      0
dtype: int64
```

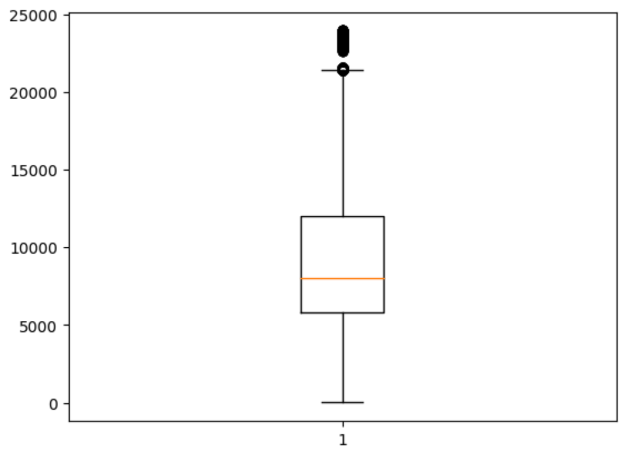
```
#we have apporx 0.55 Millon data point
df.shape
```

```
(550068, 10)
```

```
#As we observe we have only some outliers on Purchase (mean=9263,median=8047)
df.describe()
```

| | 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 |

```
# data["Purchase"]
plt.boxplot(data=df,x="Purchase")
plt.show()
```




```
#We have only 0.64 % of data of puchase as outliers with max value of (23961)
df[df["Purchase"]>21000].shape[0]/df.shape[0] * 100
```

```
0.643738592319495
```

Objective 1:Spending Habits Differ between Males and Female Customers

Function Required for Analysys

```
# Creating the Function for selecting random Samples
import random
def samples(datai,n):
  x=random.sample(range(0,len(datai)),n)
  return df.loc[x,"Purchase"]
```

```
#As per CLT mean of 50 million population and mean of mean sample distribution are same
#CI with X% of confidence is Upop+Z*sigmaPop/root(sample size)
def CI(mu,n,sigma,confidence):
  a=(1-confidence)/2
  return (mu+norm.ppf(a)*(sigma/n**0.5),mu+norm.ppf(a+confidence)*(sigma/n**0.5))
```

*Average Spending of male and female Customers on population data *

```
#Avg spending of male and female customer are significantly differ as per available data
#As we Observe on population data average spending of Male is grater then Female
female_avg_pop=df[df["Gender"]=="F"]["Purchase"].mean()
print("Female:-",female_avg_pop)
male_avg_pop=df[df["Gender"]=="M"]["Purchase"].mean()
print("Male:-",male_avg_pop)
```

```
Female:- 8734.565765155476
Male:- 9437.526040472265
```

```
#Standard deviation for male and female customer is also diff as we see male data have more spread than female
male_sigma_pop=df[df["Gender"]=="M"]["Purchase"].std()
female_sigma_pop=df[df["Gender"]=="F"]["Purchase"].std()
print("male_std:-",male_sigma_pop,"female_std:-",female_sigma_pop)
```

```
male_std:- 5092.18620977797 female_std:- 4767.233289291458
```

```
# sample of size 30 have different mean values
male_sample_30=samples(df[df["Gender"]=="M"],30).mean()
female_sample_30=samples(df[df["Gender"]=="F"],30).mean()
print("male_sample_30=",male_sample_30)
print("female_sample_30=",female_sample_30)
```

```
male_sample_30= 8267.166666666666
female_sample_30= 7995.166666666667
```

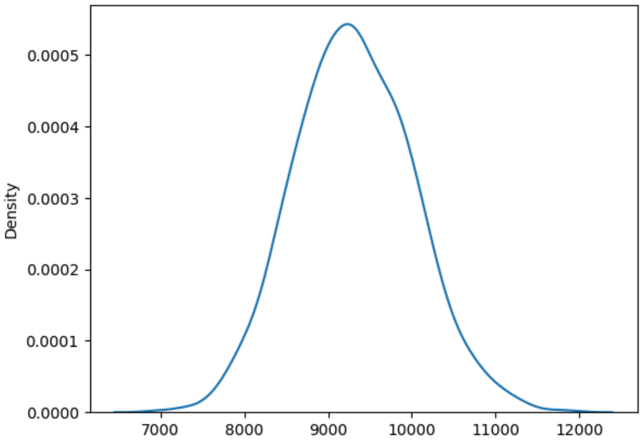
```
#Lets see interval of population mean with 95% of confidence
#Population mean for male purchase amount will lie in (8416.791640437727, 8749.475026228938) with 95% confidence
#Population mean for female purchase amount will lie in (8416.791640437727, 8749.475026228938) with 95% confidence
male_interval=(male_sample_30+(norm.ppf(0.025)*male_sigma_pop/30*0.5),(male_sample_30)+norm.ppf(0.975)*male_sigma_pop/30*0.5)
print("male interval=",male_interval)
female_interval=(female_sample_30+(norm.ppf(0.025)*female_sigma_pop/30*0.5),(female_sample_30)+norm.ppf(0.975)*female_sigma_pop/30*0.5)
print("female interval=",female_interval)
```

```
male interval= (8100.82497377106, 8433.508359562271)
female interval= (7839.439907451472, 8150.893425881862)
```

Sample mean distribution for good value population mean using bootstrap and CLT

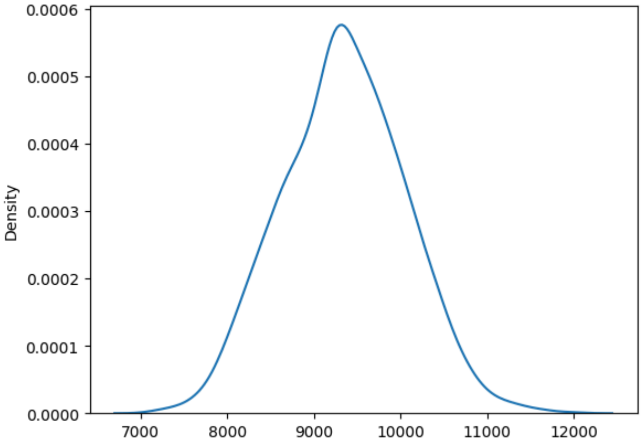
```
#As we see 2000 samples of avg female spending follow Normal distribution with mean=9287.05355
m=2000
n=50
mean_data=[]
inpt=df[df["Gender"]=="F"]
for i in range(m):
  mean_data.append(round(samples(inpt,n).mean(),2))
mean_data.sort()
CLT_POP_mean_female=np.mean(mean_data)
print("Avg_spending_female_pop=",np.mean(mean_data))
sns.kdeplot(mean_data)
plt.show()
```

```
Avg_spending_female_pop= 9311.4169
```




```
#As we see 2000 samples of avg male spending follow Normal distribution with mean=9287.05355
m=2000
n=50
mean_data=[]
inpt=df[df["Gender"]=="M"]
for i in range(m):
  mean_data.append(round(samples(inpt,n).mean(),2))
mean_data.sort()
CLT_POP_mean_male=np.mean(mean_data)
print("Avg_spending_female_pop=",np.mean(mean_data))
sns.kdeplot(mean_data)
plt.show()
```

```
Avg_spending_female_pop= 9345.597370000001
```



```
#As per CLT both mean never differ tomuch
print("Pop_female=",CLT_POP_mean_female,"Pop_male=",CLT_POP_mean_male)
```

```
Pop_female= 9311.4169 Pop_male= 9345.597370000001
```

Central Limit Thereom for finding confidence interval of mean of average spending of male and female over different sizes

Case1:Sample Size=30 and CI with (90%,95%,99%) of confidence

```
#we are Taking Population mean of 50M male and 50M female as per CLT value Pop_male and Pop_female respectively
#sample size=30
for i in (0.90,0.95,0.99):
  female_pop_mean_CI=CI(female_avg_pop,30,female_sigma_pop,i)
  male_pop_mean_CI=CI(male_avg_pop,30,male_sigma_pop,i)
  print("sample size",30,"Confidence",i*100,"%")
  print("female_pop_mean_CI:",female_pop_mean_CI)
  print("male_pop_mean_CI:",male_pop_mean_CI)
  print("")
```

```
sample size 30 Confidence 90.0 %
female_pop_mean_CI: (7302.928367906067, 10166.203162404883)
male_pop_mean_CI: (7908.302742747017, 10966.749338197513)

sample size 30 Confidence 95.0 %
female_pop_mean_CI: (7028.664588568725, 10440.466941742226)
male_pop_mean_CI: (7615.344091421863, 11259.707989522667)

sample size 30 Confidence 99.0 %
female_pop_mean_CI: (6492.63158969568, 10976.499940615271)
male_pop_mean_CI: (7042.773025730858, 11832.279055213672)
```

1)we can see of sample size 30 CI will overlap and we imporving the Confidence, range of CI is also improving

```
#we are Taking Population mean of 50M male and 50M female as per CLT value Pop_male and Pop_female respectively
#sample size=10
for i in (0.90,0.95,0.99):
  female_pop_mean_CI=CI(female_avg_pop,10,female_sigma_pop,i)
  male_pop_mean_CI=CI(male_avg_pop,10,male_sigma_pop,i)
  print("sample size",10,"Confidence",i*100,"%")
  print("female_pop_mean_CI:",female_pop_mean_CI)
  print("male_pop_mean_CI:",male_pop_mean_CI)
  print("")
```

```
sample size 10 Confidence 90.0 %
female_pop_mean_CI: (6254.897055103832, 11214.234475207119)
male_pop_mean_CI: (6788.833592694107, 12086.21848825042)

sample size 10 Confidence 95.0 %
female_pop_mean_CI: (5779.858254615695, 11689.273275695257)
male_pop_mean_CI: (6281.414324082085, 12593.637756862445)

sample size 10 Confidence 99.0 %
female_pop_mean_CI: (4851.421866034072, 12617.70966427688)
male_pop_mean_CI: (5289.692147361407, 13585.359933583124)
```

As size=10 is low we are observing more standard error in CI have more range

```
#we are Taking Population mean of 50M male and 50M female as per CLT value Pop_male and Pop_female respectively
#sample size=100
for i in (0.90,0.95,0.99):
  female_pop_mean_CI=CI(female_avg_pop,100,female_sigma_pop,i)
  male_pop_mean_CI=CI(male_avg_pop,100,male_sigma_pop,i)
  print("sample size",100,"Confidence",i*100,"%")
  print("female_pop_mean_CI:",female_pop_mean_CI)
  print("male_pop_mean_CI:",male_pop_mean_CI)
  print("")
```

```
sample size 100 Confidence 90.0 %
female_pop_mean_CI: (7950.42566851399, 9518.705861796961)
male_pop_mean_CI: (8599.935944845707, 10275.11613609882)

sample size 100 Confidence 95.0 %
female_pop_mean_CI: (7800.205209864308, 9668.926320446642)
male_pop_mean_CI: (8439.47588309863, 10435.5761978459)

sample size 100 Confidence 99.0 %
female_pop_mean_CI: (7506.607844814401, 9962.523685496551)
male_pop_mean_CI: (8125.865794644895, 10749.186286299635)
```

As sample size=100 CI of mean is still overlapping and the range of CI is decrease due to decrease in Std Error

Central Limit Thereom for finding confidence interval of mean of average spending of married and unmarried over different sizes

```
df.head()
```

| | User_ID | Product_ID | Gender | Age | Occupation | City_Category | Stay_In_Current_City_Years | Marital_Status | Product_Category |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 1000001 | P00069042 | F | 0-17 | 10 | A | 2 | 0 | [illegible] |
| 1 | 1000001 | P00248942 | F | 0-17 | 10 | A | 2 | 0 | [illegible] |
| 2 | 1000001 | P00087842 | F | 0-17 | 10 | A | 2 | 0 | 1[illegible] |

```
data_unmarried=df[df["Marital_Status"]==0]["Purchase"]
data_married=df[df["Marital_Status"]==1]["Purchase"]
```

```
data_unmarried.head()
```

```
0     8370
1    15200
2     1422
3     1057
4     7969
Name: Purchase, dtype: int64
```

```
data_married.head()
```

```
6     19215
7     15854
8     15686
9      7871
10     5254
Name: Purchase, dtype: int64
```

```
#population mean for Unmarried customer is 9265
pop_unmarried_avg=data_unmarried.mean()
pop_unmarried_sigma=data_unmarried.std()
pop_unmarried_avg
```

```
9265.907618921507
```

```
#population mean for married customer is 9261 as same like Unmarried Customer
pop_married_avg=data_married.mean()
pop_married_sigma=data_married.std()
pop_married_avg
```

```
9261.174574082374
```

Confidence interval for married and unmarried customer of population mean using CLT with sample size=30 and confidence =95%

```
CI(pop_unmarried_avg,n,pop_unmarried_sigma,0.95)
```

```
(7872.423494186891, 10659.391743656122)
```

```
CI(pop_married_avg,n,pop_married_sigma,0.95)
```

```
(7870.587121631721, 10651.762026533026)
```

As we can see both of CI are overlap or nearly same

Central Limit Thereom for finding confidence interval of mean of average spending of different age people

```
age_data=df[["Age","Purchase"]]
age_data=pd.DataFrame(age_data.groupby("Age").aggregate({"Purchase":["mean","std"]})).reset_index()
age_data
```

| | Age | Purchase | |
|---|---|---|---|
| | | mean | std |
| 0 | 0-17 | 8933.464640 | 5111.114046 |
| 1 | 18-25 | 9169.663606 | 5034.321997 |
| 2 | 26-35 | 9252.690633 | 5010.527303 |
| 3 | 36-45 | 9331.350695 | 5022.923879 |
| 4 | 46-50 | 9208.625697 | 4967.216367 |
| 5 | 51-55 | 9534.808031 | 5087.368080 |
| 6 | 55+ | 9336.280459 | 5011.493996 |

Next steps: Generate code with age_data View recommended plots

Confidence interval of population mean for different age group of customer using CLT with sample size=30 and confidence =95%

```
for i in range(len(age_data)):
  print("Confidense Interval of ",age_data.iloc[i][0],"==",CI(age_data.iloc[i][1],30,age_data.iloc[i][2],0.95))
```

```
Confidense Interval of  0-17 == (7104.509576649243, 10762.419704240705)
Confidense Interval of  18-25 == (7368.187719274321, 10971.139493248256)
Confidense Interval of  26-35 == (7459.729411376321, 11045.651854363454)
Confidense Interval of  36-45 == (7533.953497115872, 11128.747892719875)
Confidense Interval of  46-50 == (7431.162810534684, 10986.08858440197)
Confidense Interval of  51-55 == (7714.350195968683, 11355.26586595179)
Confidense Interval of  55+ == (7542.973317807921, 11129.587601090887)
```

Confidence Interval for different age group customer are overlap

Conclusion and Recommendation

```
#As we see walmart have very balanced type of customer on black friday sale as per there purchasing behaviour
#We can say very small difference in female purchase avg as per male so its king of same(Pop_female= 9318.83732 Pop_male= 9301.6628099999
#Most of CI are overlapped to each other with good percentage so we can say they not differ alot
#Customer with age group after 30s is have more purchasing behaviour
```

```
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
```

```
File "<ipython-input-75-58b461a8d053>", line 1
  .
  ^
SyntaxError: invalid syntax
```

Next steps: Fix error