DSL1_C5_S5_Challenge

Task 1

In [57]:

```
from scipy.stats import norm
X1 = 2000
zscore2k=(X1-1332)/725
print(zscore2k)
p_value = norm.cdf(zscore2k)
1-p_value
```

```
0.9213793103448276
```

Out[57]:

```
0.1784262133979203
```

Task 2

In [7]:

```
X2 = 0
zscore0=(X2-1332)/725
print(zscore0)
p_value1 = norm.cdf(zscore0)
p_value1
```

```
-1.8372413793103448
```

Out[7]:

```
0.03308713475892688
```

Task 3

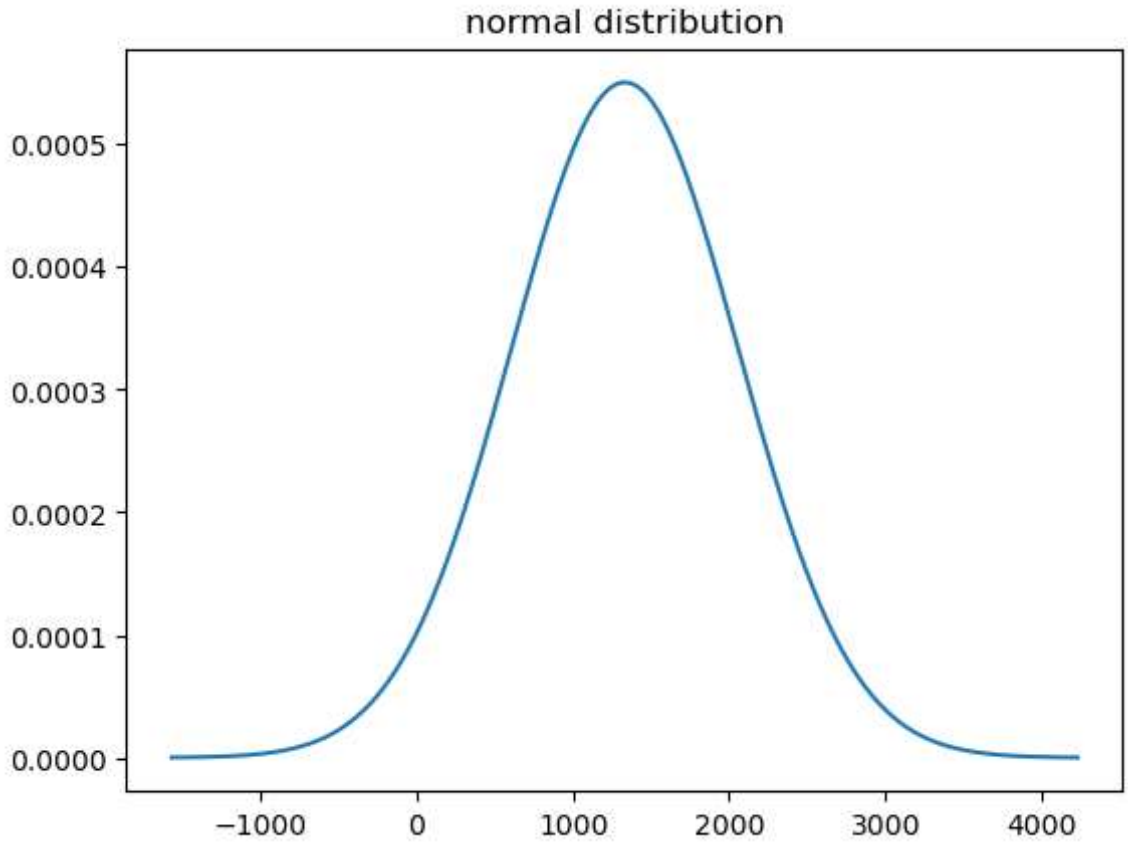
In [8]:

```
X3 = 100
X4 = 700
zscore100=(X1-1332)/725
zscore700=(X2-1332)/725

print(zscore100)
print(zscore700)

p_value100 = norm.cdf(zscore100)
p_value700 = norm.cdf(zscore700)

print('zscore80','-', 'zscore55 =',p_value700 - p_value100)
```

```
0.9213793103448276
-1.8372413793103448
zscore80 - zscore55 = -0.7884866518431528
```

In [56]:

```
mean = 1332
sd=725

lower1 = mean-4*sd
upper1 = mean+4*sd
norm1=np.arange(lower1,upper1)
plt.plot(norm1,norm.pdf(norm1,mean,sd))
plt.title("normal distribution")
plt.show()
```



High - end video Games

In [54]:

```
import pandas as pd
import warnings
warnings.filterwarnings('ignore')
import seaborn as sns
import numpy as np
```

In [16]:

```
game_df = pd.read_csv(r'E:\Aishwarya official\Aishwarya Data Scince\course 5\DS1_C5_S5_Comp
game_df
```

Out[16]:

	index	price	speed	hd	ram	screen	cd	multi	premium	ads	trend
0	1	1499	25	80	4	14	no	no	yes	94	1
1	2	1795	33	85	2	14	no	no	yes	94	1
2	3	1595	25	170	4	15	no	no	yes	94	1
3	4	1849	25	170	8	14	no	no	no	94	1
4	5	3295	33	340	16	14	no	no	yes	94	1
...	...	...	...	...	...	...	...	...	...	...	...
6254	6255	1690	100	528	8	15	no	no	yes	39	35
6255	6256	2223	66	850	16	15	yes	yes	yes	39	35
6256	6257	2654	100	1200	24	15	yes	no	yes	39	35
6257	6258	2195	100	850	16	15	yes	no	yes	39	35
6258	6259	2490	100	850	16	17	yes	no	yes	39	35

6259 rows × 11 columns

In [20]:

```
game = game_df[(game_df.price<4000)]
game
```

Out[20]:

	index	price	speed	hd	ram	screen	cd	multi	premium	ads	trend
0	1	1499	25	80	4	14	no	no	yes	94	1
1	2	1795	33	85	2	14	no	no	yes	94	1
2	3	1595	25	170	4	15	no	no	yes	94	1
3	4	1849	25	170	8	14	no	no	no	94	1
4	5	3295	33	340	16	14	no	no	yes	94	1
...	...	...	...	...	...	...	...	...	...	...	...
6254	6255	1690	100	528	8	15	no	no	yes	39	35
6255	6256	2223	66	850	16	15	yes	yes	yes	39	35
6256	6257	2654	100	1200	24	15	yes	no	yes	39	35
6257	6258	2195	100	850	16	15	yes	no	yes	39	35
6258	6259	2490	100	850	16	17	yes	no	yes	39	35

6229 rows × 11 columns

In [23]:

```
population= game['price']
```

In [24]:

```
samp1 = population.sample(50,replace=True , random_state=1)
```

Task 2 : central tendancy

In [26]:

```
import  scipy.stats as sts
import statistics as st
```

In [27]:

```
tables=[ population, samp1]
std=[]
mean=[]
mode=[]
median=[]
skew1=[]
kurt1=[]

for sample in tables:
    std.append(sample.std())
    mean.append(sample.mean())
    median.append(sample.median())
    mode.append(st.mode(sample))
    skew1.append(sample.skew())
    kurt1.append(sample.kurtosis())

pd.DataFrame([std,mean,median,mode,skew1,kurt1],
             columns=["Population","samp1"],index=["std","Mean","Median","Mode","Skewness",
```

Out[27]:

	Population	samp1
std	560.501267	532.160389
Mean	2208.855515	2093.120000
Median	2144.000000	2018.500000
Mode	1999.000000	1890.000000
Skewness	0.518217	0.560012
Kurtosis	-0.125533	0.795868

Task 3

In [28]:

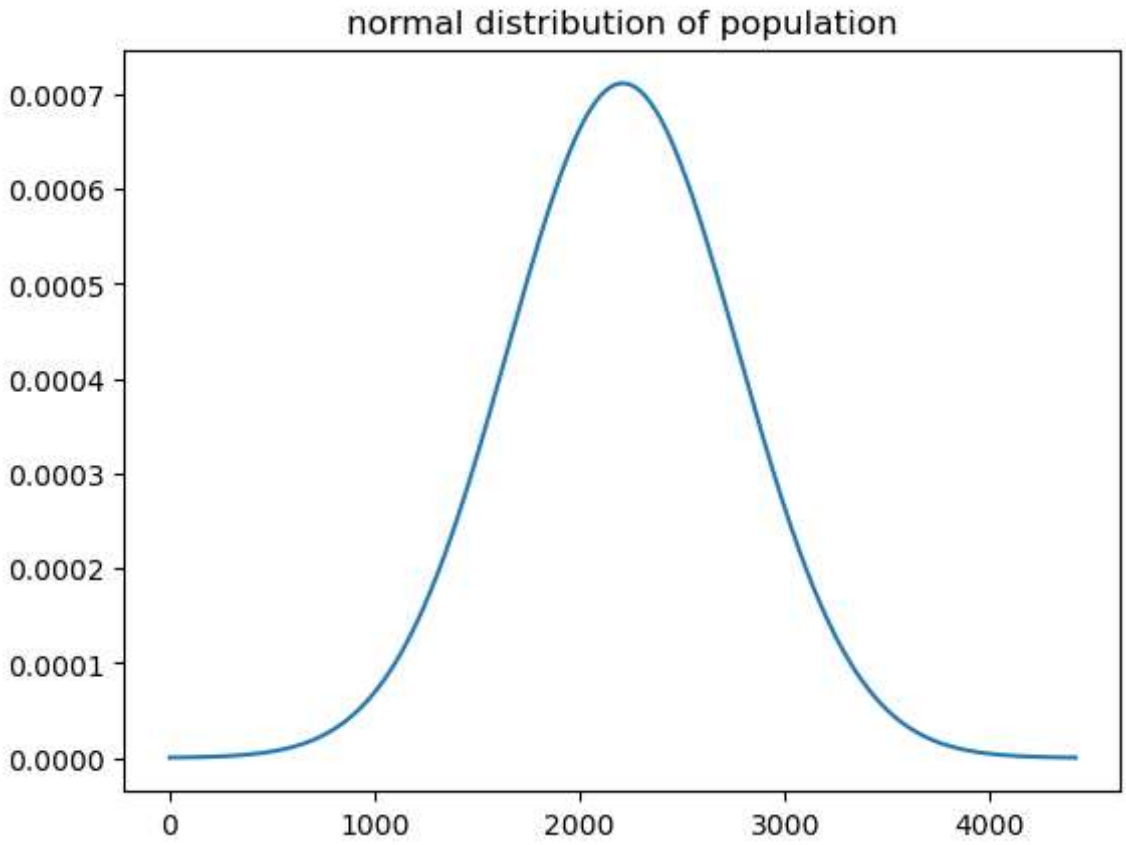
```
popmean=st.mean()
popstd=st.stdev(population)

samp1mean=st.mean(samp1)
samp1std=st.stdev(samp1)
popmean
samp1mean
```

Out[28]:

```
2093.12
```

In [34]:

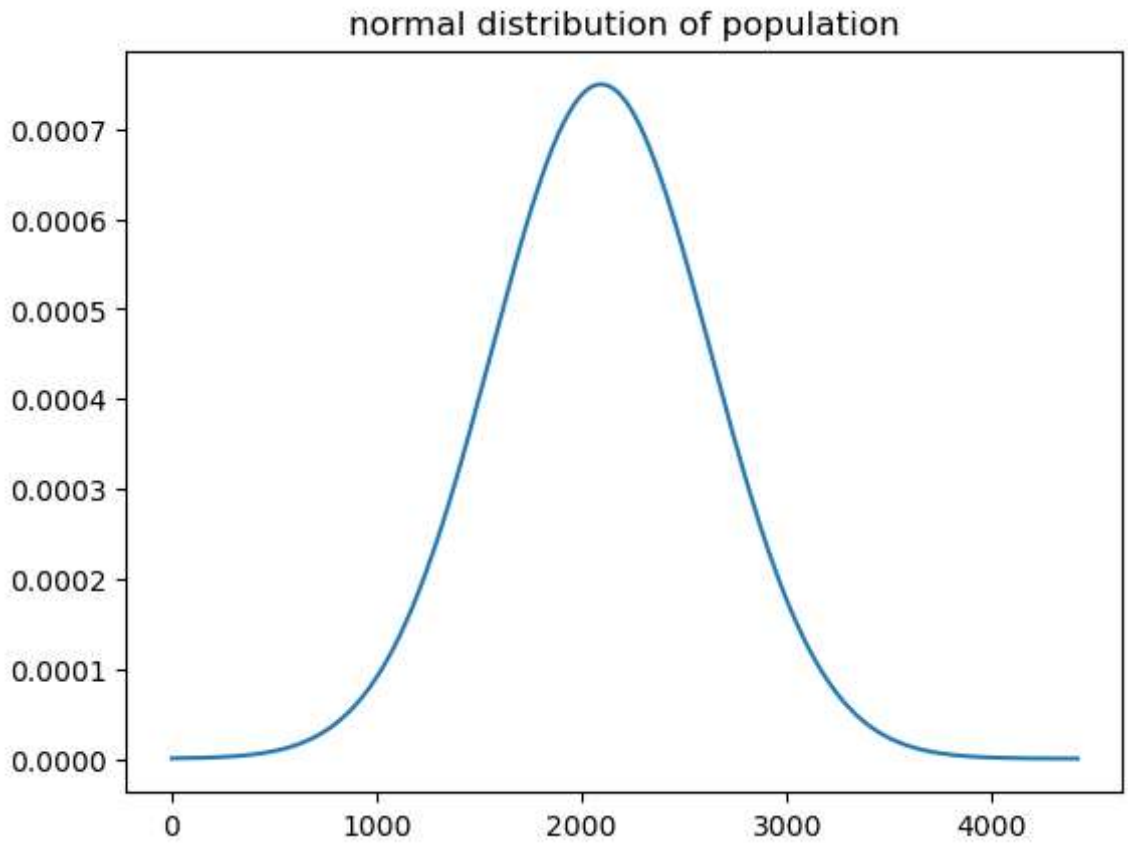
```
from scipy.stats import norm
import matplotlib.pyplot as plt
lower=popmean-1*popmean
upper=popmean+1*popmean

norm_p=np.arange(lower,upper)
plt.plot(norm_p,norm.pdf(norm_p,popmean,popstd))
plt.title("normal distribution of population")
plt.show()
```



Task 4

In [31]:

```
lower_s=samp1mean-1*samp1mean
upper_s=samp1mean+1*samp1mean

norm_p=np.arange(lower,upper)
plt.plot(norm_p,norm.pdf(norm_p,samp1mean,samp1std))
plt.title("normal distribution of population")
plt.show()
```



Task 5 : Calculate zscore

In [38]:

```
#scipy.stats.zscore(population)
X = population
zscore = (X- popmean)/popstd

print("z score :", zscore)
```

```
z score : 0      -1.266465
1       -0.738367
2       -1.095190
3       -0.642024
4        1.937809
           ...
6254    -0.925699
6255     0.025235
6256     0.794190
6257    -0.024720
6258     0.501595
Name: price, Length: 6229, dtype: float64
```

In [41]:

```
p_value = norm.cdf(zscore)
p_value
```

Out[41]:

```
array([0.10602109, 0.99930822, 0.35134587, 0.24632221, 0.65057594,
       0.83893187, 0.07961057, 0.6779741 , 0.41217128, 0.20198263,
       0.24454832, 0.64429521, 0.28766911, 0.32659385, 0.42906971,
       0.06514906, 0.35413784, 0.68133559, 0.96162956, 0.68867825,
       0.04646211, 0.09320804, 0.76581071, 0.80157583, 0.17221989,
       0.89111612, 0.68066447, 0.36397077, 0.45949793, 0.80676991,
       0.35134587, 0.46696261, 0.38965579, 0.63374051, 0.82477156,
       0.82670393, 0.68133559, 0.95476019, 0.06781591, 0.35134587,
       0.90950565, 0.4602438 , 0.04573569, 0.82910024, 0.55743443,
       0.19934697, 0.35134587, 0.17462795, 0.82477156, 0.30258262])
```

In [42]:

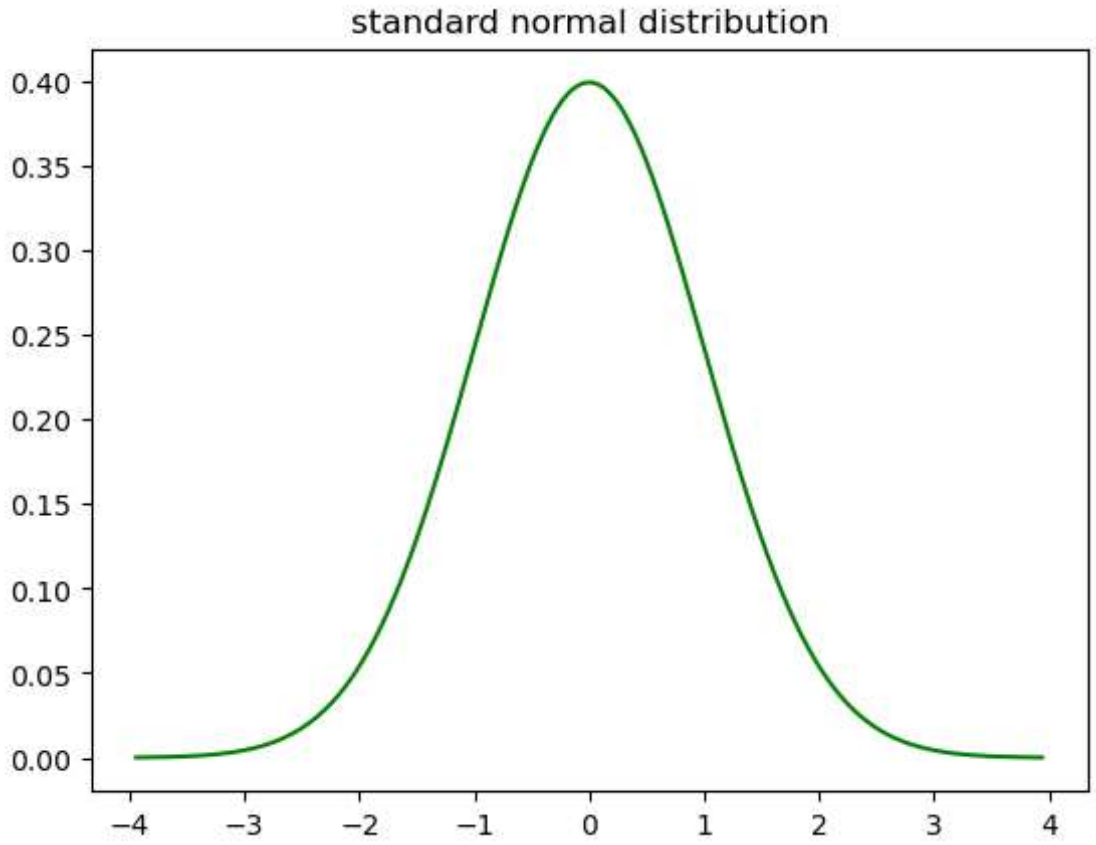
```
standard_data = []
norm_s = np.arange(lower,upper)

for x in norm_s:
    zscore1=(x-popmean)/popstd
    standard_data.append(zscore1)
standard_data
```

Out[42]:

```
[-3.9408573093102737,
 -3.9390731920226605,
 -3.9372890747350477,
 -3.9355049574474346,
 -3.9337208401598214,
 -3.9319367228722086,
 -3.9301526055845954,
 -3.9283684882969823,
 -3.926584371009369,
 -3.9248002537217563,
 -3.923016136434143,
 -3.92123201914653,
 -3.9194479018589172,
 -3.917663784571304,
 -3.915879667283691,
 -3.914095549996078,
 -3.912311432708465,
 -3.9105273154208517,
```

In [43]:

```
plt.plot(standard_data, norm.pdf(standard_data,st.mean(standard_data)), color='green')
plt.title("standard normal distribution")
plt.show()
```



Task 8

In [44]:

```
X3 = 2700
zscore = (X3- popmean)/popstd

print("z score :", zscore)
```

```
z score : 0.8762593672449527
```

In [46]:

```
p_value = norm.cdf(zscore)
prob_above = 1- p_value
prob_above
```

Out[46]:

```
0.19044452449275684
```

Task 9

In [50]:

```
X4=1301
zscore3=(X4-popmean)/popstd
p_value1=norm.cdf(zscore3)
print("probability of price greater thab 2700 :", p_value1)
```

```
probability of price greater thab 2700 : 0.052646141810628
```

Task 10

In [52]:

```
X5= 2000
X6=2900
zscore5=(X5-popmean)/popstd
zscore6=(X6-popmean)/popstd

p_value5=norm.cdf(zscore5)
p_value6=norm.cdf(zscore6)
print("probability of price between 2000 and 2900 :", p_value5-p_value6 )
```

```
probability of price between 2000 and 2900 : -0.5365129425849419
```