In [ ]:

```
#outliers
#Extreme values
# values that can distort the mean

# lower and Upper Whisker/cap/limit - we'll have to decide
# Any value < LC and > UC : outlier

#Symmentry(skewness ~ 0 : then curve is Symmetrical)
# If the distribution is symmertic we have 2 options
#           1. IQR Method   (symmertic)
#           2.Mean +/- Standard deviation(Symmetric and normal)
# If the distribution is skewed : 95th %ile
#                                 99th %ile
```

In [1]:

```
#IQR Rule
# quartiles  :   0%        25%       50%        75%        100%
# deciles
# Q3 : 75%
# Q1 : 25%
# IQR  : Inter Quartile Range = Q3 - Q1
# LC = Q1 - 1.5 * IQR
# UC = Q3 + 1.5 * IQR
```

In [8]:

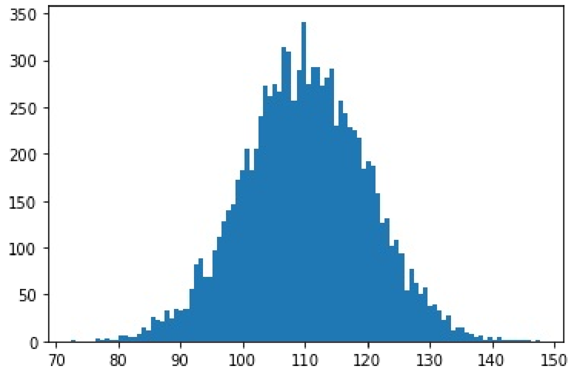
```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
```

In [3]:

```
mydata = np.random.normal(size=10000,loc=110,scale=10)
```

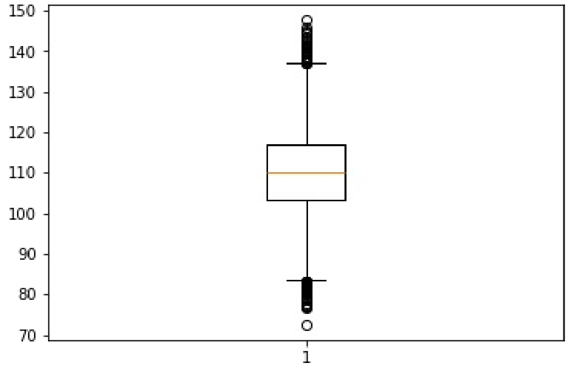
In [12]:

```
mydata = pd.Series(mydata)
mydata.describe()
plt.hist(mydata,bins=100)
plt.show()
```



In [14]:

```
plt.boxplot(mydata)
plt.show()
#Remove mejority of black dots ~ 80%
```



In [15]:

```
UC = mydata.mean()+3*mydata.std()  # Normally distributed curves
LC = mydata.mean()-3*mydata.std()
```

In [6]:

```
len(mydata.loc[(mydata<LC)|(mydata>UC)])
```

Out[6]:

```
28
```

In [24]:

```
# IQR
# calculate the quantiles
mydata3 = mydata.copy(deep=True)  #set deep =True for different copies
quant=mydata.quantile([0,0.25,0.5,0.75,1])
# the quantile takes [] as input
# the list can have values min 0 max 1
```

```
quant
Q1=quant.iloc[1]
Q3=quant.loc[0.75]
IQR = Q3-Q1
IQR
```

Out[24]:

```
np.float64(13.460909412216907)
```

In [25]:

```
LC=Q1-1.5*IQR
UC=Q3+1.5*IQR
```

In [26]:

```
len(mydata.loc[(mydata<LC) |(mydata>UC)])
```

Out[26]:

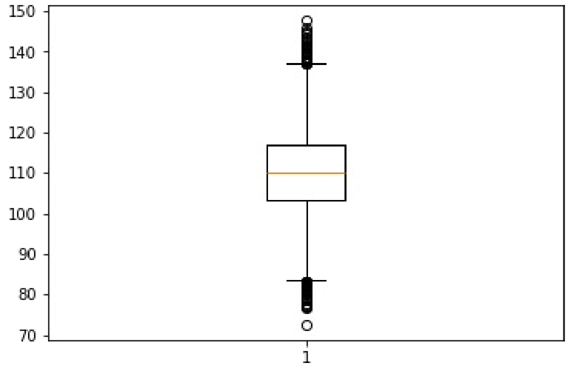
```
67
```

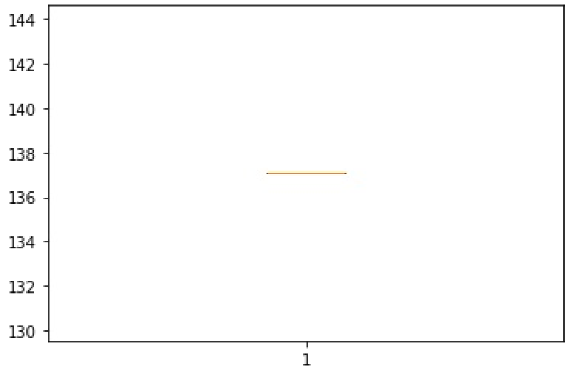
In [31]:

```
mydata3.iloc[mydata3<LC]=LC
mydata3.loc[mydata3>UC]=UC
```

In [32]:

```
plt.boxplot(mydata)
plt.show()
plt.boxplot(mydata3)
plt.show()
```





In [33]:

```
# Asymmetric curves
#p95 or P99 rule
```

#P95 rule : UC can be set to the 95% value and Lower cap set to the 5% #if p99 p95 value are far from each other p99 and max are bit close to each other # Use p95 as upper Cap and P5 as lower cap mydata.loc[mydatapct[0.95]] # P99 rule :UC can be set to the 99% value and Lower cap set to the 1% # if p99 and p95 are close to from each other #P99 and max are far from each other then consider p99 as upper cap and p1 as lower cap mydata.loc[mydatapct[0.99]]

In [35]:

```
pct=mydata3.quantile([0,0.01,0.05,0.1,0.2,0.3,.4,0.5,0.6,0.75,0.8,0.9,0.95,0.99,1])
pct
```

Out[35]:

```
0.00    137.071035
0.01    137.071035
0.05    137.071035
0.10    137.071035
0.20    137.071035
0.30    137.071035
0.40    137.071035
0.50    137.071035
0.60    137.071035
0.75    137.071035
0.80    137.071035
0.90    137.071035
0.95    137.071035
0.99    137.071035
1.00    137.071035
dtype: float64
```

In [39]:

```
def AddNum(a,b):
    return(a+b)
```

In [46]:

```
AddNum.  doc   = "Add a,b"
```

In [47]:

```
?AddNum
```

In [50]:

```
def AddNum3(a,b,c):
    '''Add Three number
    a,b,c'''
    return(a+b+c)
```

In [49]:

```
?AddNum3
```

????.pdf# continous categorical --------- ----------- Nominal Ordinal Age -------- ---------- Operating cost location Survey score income Gender Rank profits Dept Grades empcount Designation TaxSlab distance Season Offers Payscale weight OSType Empperformance Car Types DressSize Aqc cost Payment Methods Band Score product types Salary Region/Area Discount expenseswe derived a categorical var from continous var This is called binning Age ------> AgeCategories Salary-----> Salary bracket Tax/Income -----> Tax Salbs Encoding All categories in a cat var , are given a number and a column is created where we substitute the categorical with these numbersnp.where(condition,if true,if false) #in numpy where works same as if condition

In [52]:

```
import pandas as pd
from numpy import where as IF
?IF
```

In [55]:

```
student=pd.read_csv(r"C:\Users\sival\Downloads\DataSet Student Expenses.csv",sep=",")
```

In [56]:

```
student
```

Out[56]:

	schoolid	schoolname	CalendarYear	SchoolType	TermType	FTResTuition	PTResTuition	FTNonResTuition	PTNonResTuition	FTResTe
0	100300	Faulkner University	2024	PRI	SEM	40400	NaN	40400	NaN	
1	103600	Samford University	2024	PRI	SEM	46326	NaN	46326	NaN	
2	105100	University of Alabama	2024	PUB	SEM	25317	NaN	47537	NaN	
3	108100	Arizona State University	2024	PUB	SEM	28299	NaN	50317	NaN	
4	108300	University of Arizona	2024	PUB	SEM	25353	NaN	29988	NaN	
...	...	...	...	...	...	...	...	...	...	
190	3191300	City University of New York	2024	PUB	SEM	16013	10482.0	26203	17502.0	
191	3559300	Appalachian School of Law	2024	PRI	SEM	41000	NaN	41000	NaN	
192	3691400	Ave Maria School of Law	2024	PRI	SEM	50750	NaN	50750	NaN	
193	4096300	Charleston School of Law	2024	PRI	SEM	48234	38834.0	48234	38834.0	
194	4242101	UNT Dallas College Of Law	2024	PUB	SEM	23833	20439.0	40150	34180.0	

195 rows × 17 columns

In [68]:

```
fee=IF((student.FTResTuition<20000),"low",
  IF((student.FTResTuition<25000)&(student.FTResTuition>20000),"med",
    IF((student.FTResTuition<30000)&(student.FTResTuition>25000),"AVg","veryHigh")))
```

In [70]:

```
feelevel=IF(fee=="low",1,
           IF(fee=="med",2,
             IF(fee=="Avg",3,4)))
feelevel
```

Out[70]:

```
array([4, 4, 4, 4, 4, 1, 1, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
       4, 4, 4, 1, 4, 4, 4, 4, 4, 1, 2, 4, 4, 4, 2, 4, 4, 1, 4, 1, 2, 4,
       4, 4, 4, 2, 4, 2, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 2, 4, 2, 4,
       4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 1, 4, 4, 2, 4,
       4, 4, 1, 4, 4, 4, 4, 2, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
       4, 4, 1, 4, 4, 1, 4, 4, 4, 4, 4, 4, 2, 4, 4, 4, 2, 4, 4, 4, 4, 4,
       4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 2, 1, 4, 4, 2, 2, 4, 4, 4, 4, 4, 1,
       4, 4, 4, 1, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 1, 1, 1, 4, 4, 1, 4,
       4, 2, 2, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 1, 4, 4, 4, 2])
```

In [79]:

```
student[feelevel] = feelevel
```

In [75]:

```
student
```

Out[75]:

	schoolid	schoolname	CalendarYear	SchoolType	TermType	FTResTuition	PTResTuition	FTNonResTuition	PTNonResTuition	FTResTe
0	100300	Faulkner University	2024	PRI	SEM	40400	NaN	40400	NaN	
1	103600	Samford University	2024	PRI	SEM	46326	NaN	46326	NaN	
2	105100	University of Alabama	2024	PUB	SEM	25317	NaN	47537	NaN	
3	108100	Arizona State University	2024	PUB	SEM	28299	NaN	50317	NaN	
4	108300	University of Arizona	2024	PUB	SEM	25353	NaN	29988	NaN	
...	...	...	...	...	...	...	...	...	...	
190	3191300	City University of New York	2024	PUB	SEM	16013	10482.0	26203	17502.0	
191	3559300	Appalachian School of Law	2024	PRI	SEM	41000	NaN	41000	NaN	
192	3691400	Ave Maria School of Law	2024	PRI	SEM	50750	NaN	50750	NaN	
193	4096300	Charleston School of Law	2024	PRI	SEM	48234	38834.0	48234	38834.0	
194	4242101	UNT Dallas College Of Law	2024	PUB	SEM	23833	20439.0	40150	34180.0	

195 rows × 20 columns

In [ ]:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js