In [14]:
```
import numpy as np
import scipy.stats as st
import statsmodels.datasets
import matplotlib.pyplot as plt
import pandas as pd
from math import log,sqrt
%matplotlib inline

data = [3.0,3.2,4.2,4.5,4.7,4.9,4.9,5.0,5.1,5.5,5.5,5.6,5.6,5.7,5.7,5.7,5.8,5.
9,6.0,6.0,6.1,6.2,6.2,6.3,6.3,6.4,6.5,6.5,6.5,6.6,6.8,6.8,7.0,7.0,7.1,7.1,7.2,
7.2,7.3,7.4,7.4,7.5,7.6,7.6,7.7,7.8,7.8,7.9,7.9,8.0,8.0,8.0,8.0,8.0,8.1,8.1,8.
3,8.3,8.3,8.4,8.6,8.7,8.8,9.0,9.0,9.0,9.0,9.1,9.1,9.2,9.2,9.3,9.4,9.4,9.5,9.7,
9.8,9.8,9.9,9.9,10.0,10.0,10.2,10.4,10.4,10.4,10.5,10.6,10.7,10.8,10.8,10.9,1
1.0,11.1,11.2,11.5,11.8,11.9,12.1,12.1]
len(data)
```

Out[14]: 100

In [15]:
```
df = pd.DataFrame(data, columns=['xi'])
df
```

Out[15]:

	xi
0	3.0
1	3.2
2	4.2
3	4.5
4	4.7
...	...
95	11.5
96	11.8
97	11.9
98	12.1
99	12.1

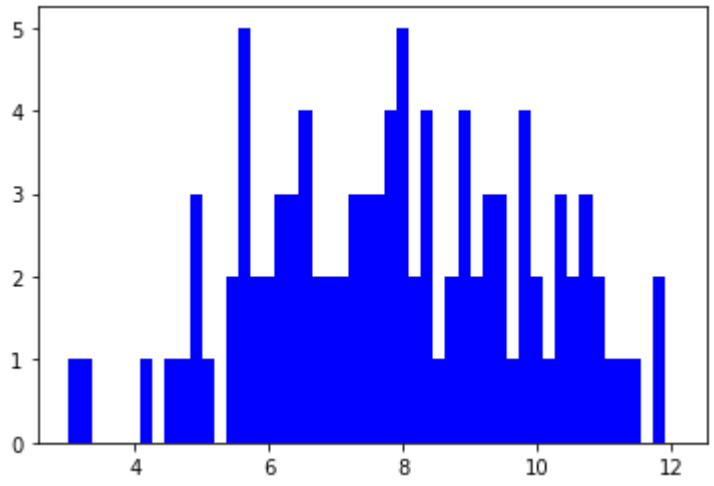
100 rows × 1 columns

In [16]:
```
limits_list = []
n = len(df)
k = round(sqrt(n))
nhigh =  df['xi'].max()
nlow =  df['xi'].min()
range1 = nhigh - nlow
w = range1 / k

#limites inferiores y superiores
xlimite = nlow
for x in range(0,k):
    if x == 0:
        linf = nlow
        lsup = nlow + w
    else:
        linf = xlimite
        lsup = xlimite + w

    xlimite = xlimite + w
    limits_list.append([linf,lsup])

limits_list = np.around(limits_list, decimals=4)
limits_df = pd.DataFrame(limits_list, columns=['linf','lsup'])

frecuencies_list = []
ii = 0
for index1, irow in limits_df.iterrows():
    i = 0
    for index2, jrow in df.iterrows():
        if index1 == 0:
            if jrow['xi'] >= irow['linf'] and jrow['xi'] <= irow['lsup'] :
                i = i+1
        else:
            if jrow['xi'] > irow['linf'] and jrow['xi'] <= irow['lsup'] :
                i = i+1
    #print(irow['linf'], irow['lsup'],jrow['xi'])
    ii = ii+i
    frecuencies_list.append([irow['linf'], irow['lsup'],i,ii/n,(i/n)*100,ii,(i
i/n)*100])

frecuencies_df = pd.DataFrame(frecuencies_list, columns=['linf','lsup','fabsol
uta','frelativa','fporcentual','Facumulada','Facmporcentual'])
frecuencies_df
```

Out[16]:

	linf	lsup	fabsoluta	frelativa	fporcentual	Facumulada	Facmporcentual
0	3.00	3.91	2	0.02	2.0	2	2.0
1	3.91	4.82	3	0.05	3.0	5	5.0
2	4.82	5.73	11	0.16	11.0	16	16.0
3	5.73	6.64	14	0.30	14.0	30	30.0
4	6.64	7.55	12	0.42	12.0	42	42.0
5	7.55	8.46	18	0.60	18.0	60	60.0
6	8.46	9.37	12	0.72	12.0	72	72.0
7	9.37	10.28	11	0.83	11.0	83	83.0
8	10.28	11.19	11	0.94	11.0	94	94.0
9	11.19	12.10	6	1.00	6.0	100	100.0

In [17]:
```
def plot_function(size = 100, bins = 50, loc=0, scale=1, color='blue'):
    binwidth = (max(df['xi']) - min(df['xi']))/ bins
    plt.hist(df['xi'], bins=np.arange(min(data), max(data) + binwidth, binwidth),color=color)
    plt.show()

plot_function()
```



In [ ]:
```
```