In [105]:
```
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


uberdata_df = pd.read_csv('uberdata/rider/trips_data.csv')
uberdata_df['Distance (miles)'] = uberdata_df['Distance (miles)'].apply(lambda
x : x*1.61)

uberdata_df['Distance (miles)'].head(100)
```

Out[105]:
```
0      3.6869
1      3.1556
2      3.6869
3      3.4776
4      3.6386
       ...
95    12.2038
96     6.0697
97    12.1716
98    12.9766
99     4.5563
Name: Distance (miles), Length: 100, dtype: float64
```

In [97]:
```
uberdata_df['Product Type'] = uberdata_df['Product Type'].replace('uberX','Ube
rX')
uberdata_df['Product Type'] = uberdata_df['Product Type'].str.replace('Uber_X'
,'UberX')
uberdata_df = uberdata_df.drop(uberdata_df.loc[uberdata_df['Product Type'] ==
'UberEATS Marketplace'].index)
uberdata_df['Fare Amount'].sum()
uberdata_df['Fare Amount'].max()
# print(pd.value_counts(uberdata_df['Product Type'].values, sort=True))
```

Out[97]:
```
174.8
```

In [99]:
```
# uberdata_df.loc[uberdata_df['Trip or Order Status'] == 'COMPLETED'].tail()
uberdata_df = uberdata_df.loc[uberdata_df['Trip or Order Status'] == 'COMPLETE
D']
# uberdata_df = uberdata_df.sort_values(by=['Distance (miles)'])
# random_df
# uberdata_df['Distance (miles)'].tail()

limits_list = []
n = len(uberdata_df['Distance (miles)'])
k = round(sqrt(n))
nhigh =  uberdata_df['Distance (miles)'].max()
nlow =  uberdata_df['Distance (miles)'].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 uberdata_df.iterrows():
        if index1 == 0:
            if jrow['Distance (miles)'] >= irow['linf'] and jrow['Distance (mi
les)'] <= irow['lsup'] :
                i = i+1
        else:
            if jrow['Distance (miles)'] > irow['linf'] and jrow['Distance (mil
es)'] <= 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
```

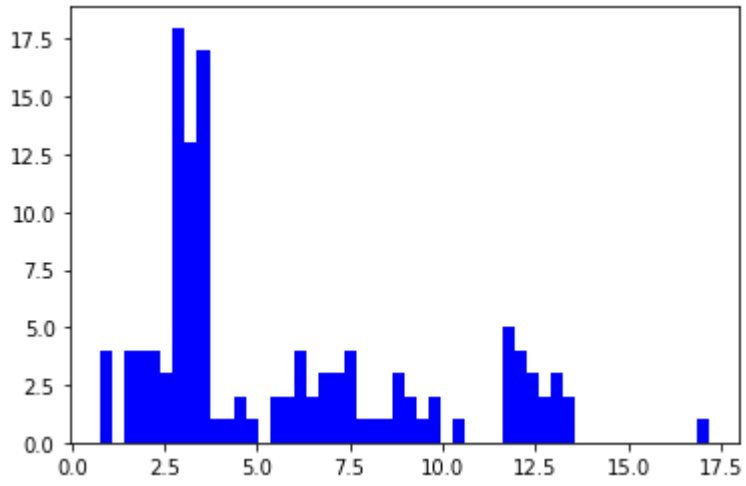
Out[99]:

	linf	lsup	fabsoluta	frelativa	fporcentual	Facumulada	Facmporcentual
0	0.7567	2.2511	14	0.112903	11.290323	14	11.290323
1	2.2511	3.7454	53	0.540323	42.741935	67	54.032258
2	3.7454	5.2398	5	0.580645	4.032258	72	58.064516
3	5.2398	6.7342	10	0.661290	8.064516	82	66.129032
4	6.7342	8.2286	12	0.758065	9.677419	94	75.806452
5	8.2286	9.7229	8	0.822581	6.451613	102	82.258065
6	9.7229	11.2173	2	0.838710	1.612903	104	83.870968
7	11.2173	12.7117	14	0.951613	11.290323	118	95.161290
8	12.7117	14.2061	5	0.991935	4.032258	123	99.193548
9	14.2061	15.7004	0	0.991935	0.000000	123	99.193548
10	15.7004	17.1948	1	1.000000	0.806452	124	100.000000

In [101]:
```
data = uberdata_df['Distance (miles)']

def plot_function(size = 100, bins = 50, loc=0, scale=1, color='blue'):
    binwidth = (max(data) - min(data))/ bins
    plt.hist(data, bins=np.arange(min(data), max(data) + binwidth, binwidth),c
olor=color)
    plt.show()

plot_function()
```



In [ ]:
```

```