

## Савельев Михаил, 3.8, вариант 6

In [1]:

```
import numpy as np
```

In [2]:

```
data = np.array([2.49, 1.46, 3.95, 1.7, 3.27, 4.17, 3.82, 1.33, 2.44, 1.69, 3.33, 2.9, 2.31, 3.98, 2.21, 2.55, 2.14, 2.86, 5.49, 3.48, 3.51, 3.42])
```

а) найти выборочное среднее, выборочную дисперсию, исправленную выборочную дисперсию

Выборочное среднее

$$\bar{x} = \frac{1}{n} \cdot \sum_{i=1}^n x_i$$

In [3]:

```
mean = sum(data)/len(data)
print(f'Выборочное среднее = {mean}')
print(f'Выборочное среднее из библиотеки numpy = {data.mean()}')
```

Выборочное среднее = 2.931818181818181

Выборочное среднее из библиотеки numpy = 2.9318181818181817

Выборочная дисперсия

$$s^2 = \frac{1}{n} \cdot \sum_{i=1}^n (x_i - \bar{x})^2$$

In [4]:

```
dis = sum((xi - mean) ** 2 for xi in data) / len(data)
print(f'Выборочная дисперсия = {dis}')
print(f'Выборочная дисперсия из библиотеки numpy = {data.var()}')
```

Выборочная дисперсия = 1.0028603305785124

Выборочная дисперсия из библиотеки numpy = 1.0028603305785122

Исправленная выборочная дисперсия

$$s^2 = \frac{1}{n-1} \cdot \sum_{i=1}^n (x_i - \bar{x})^2$$

In [5]:

```
c_dis = sum((xi - mean) ** 2 for xi in data) / (len(data)-1)
print(f'Исправленная выборочная дисперсия = {c_dis}')
print(f'Исправленная выборочная дисперсия из библиотеки numpy = {data.var(ddof=1)}')
```

Исправленная выборочная дисперсия = 1.0506155844155844

Исправленная выборочная дисперсия из библиотеки numpy = 1.0506155844155842

**b) приближенно вычислить начальные и центральные моменты вплоть до 4-го порядка включительно**

In [6]:

```
for i in range(1,5):
    print(f'Начальный момент {i}-го порядка = {(1 / len(data)) * sum(data ** i)}.mean()')
```

Начальный момент 1-го порядка = 2.931818181818181  
Начальный момент 2-го порядка = 9.598418181818182  
Начальный момент 3-го порядка = 34.460635909090904  
Начальный момент 4-го порядка = 133.7242930309091

In [7]:

```
m = []
for i in range(1,5):
    m.append(((data - [mean] * len(data)) ** i).mean())
    print(f'Центральный момент {i}-го порядка = {m[i-1]}')
```

Центральный момент 1-го порядка = 5.853903220750825e-16  
Центральный момент 2-го порядка = 1.0028603305785124  
Центральный момент 3-го порядка = 0.43941066491360076  
Центральный момент 4-го порядка = 2.966725620981492

In [8]:

```
from scipy.stats import moment
for i in range(1,5):
    print(f'Центральный момент {i}-го порядка из библиотеки scipy = {moment(data, moment=i)}')
```

Центральный момент 1-го порядка из библиотеки scipy = 0.0  
Центральный момент 2-го порядка из библиотеки scipy = 1.0028603305785122  
Центральный момент 3-го порядка из библиотеки scipy = 0.43941066491359937  
Центральный момент 4-го порядка из библиотеки scipy = 2.966725620981492

**c) приближенно найти эксцесс  $\gamma$  распределения**

In [9]:

```
ex = m[4-1] / (dis ** 2) - 3
print(f'Эксцесс  $\gamma$  распределения = {ex}')
```

Эксцесс  $\gamma$  распределения = -0.050173471041932594

In [10]:

```
from scipy.stats import kurtosis
print(f'Эксцесс  $\gamma$  распределения из библиотеки scipy = {kurtosis(data)}')
```

Эксцесс  $\gamma$  распределения из библиотеки scipy = -0.05017347104193126

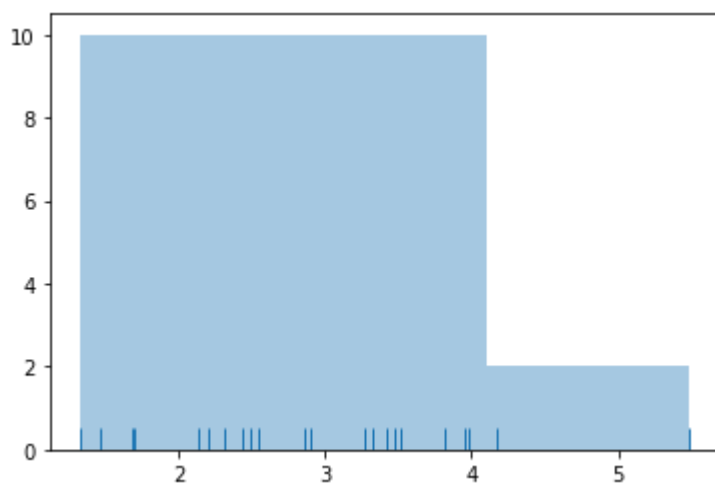
**d) построить гистограмму ( число разрядов гистограммы задавать самостоятельно)**

In [11]:

```
import seaborn as sns
sns.distplot(data, kde = False, rug = True, bins = 3)
```

Out[11]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x11eca9d30>

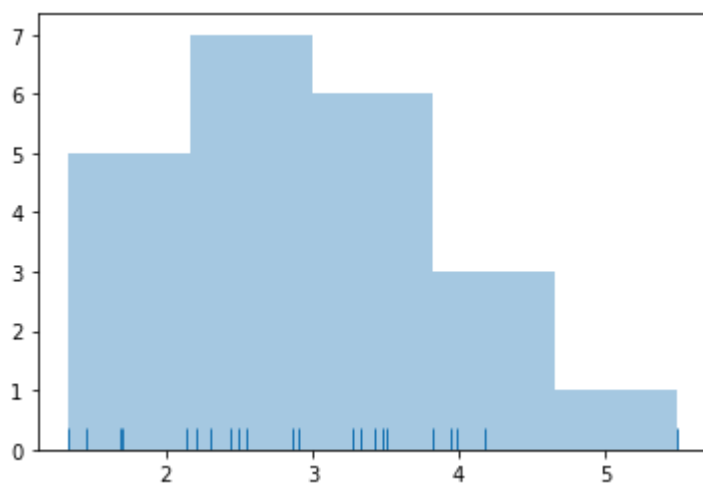


In [12]:

```
sns.distplot(data, kde = False, rug = True, bins = 5)
```

Out[12]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x120d4ecc0>

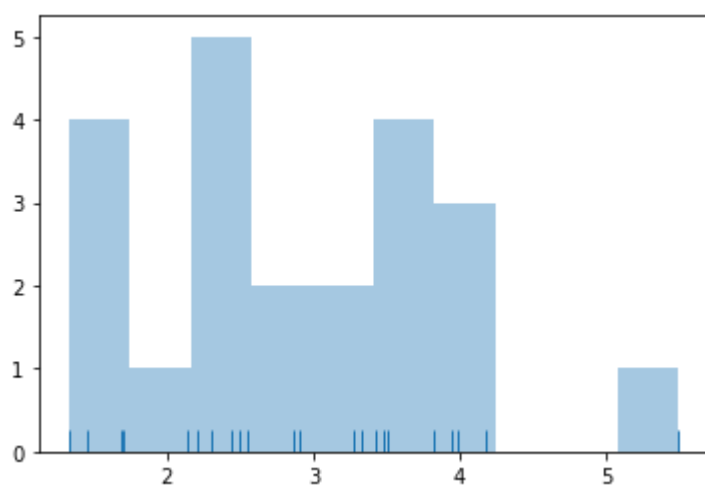


In [13]:

```
sns.distplot(data, kde = False, rug = True, bins = 10)
```

Out[13]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x120ff3f60>

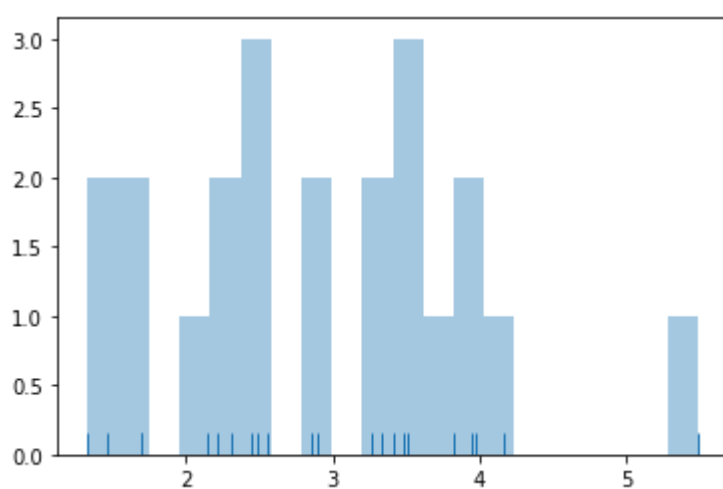


In [14]:

```
sns.distplot(data, kde = False, rug = True, bins = 20)
```

Out[14]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x1210cae48>



In [15]:

```
sns.distplot(data, kde = False, rug = True, bins = 25)
```

Out[15]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x121199978>

