Задание 1

Борисов Д.А.

Вариант **1**

In [6]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [7]:

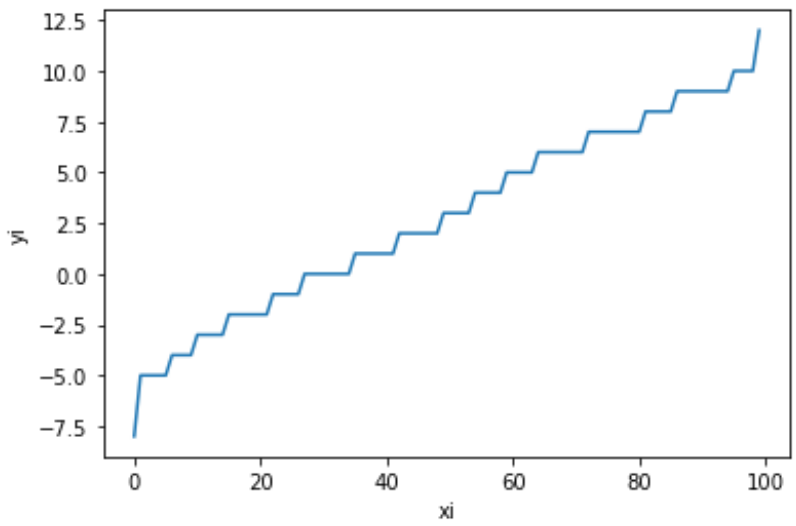
```
df = pd.read_excel('Data.xlsx')
```

In [9]:

```
# Изобразим исходные данные в виже графика
Y = df.sort_values('xi')['xi'].tolist()
X = df.index.tolist()
plt.plot(X,Y)
plt.xlabel('xi')
plt.ylabel('yi')
```

Out[9]:

```
Text(0, 0.5, 'yi')
```



Вывод: очищать данные не нужно, выбросов нет. На графике просто линейная зависимость.

In [ ]:

```

```

In [ ]:

```

```

In [10]:

```
df.head(5)
```

Out[10]:

	xi
0	-8
1	-5
2	-5
3	-5
4	-5

Проверим на наличае выбросов - код для очистки от выбросов не понадобился

In [11]:

```
# найдем среднее значение
# mean_x = np.mean(abs(df))
```

In [12]:

```
# max_xi = float(1.8*mean_x)
# min_xi = float(0.8*mean_x)
```

In [13]:

```
# выберем только значения в этом интервале
# df_clean = df[(df['xi'] <= max_xi) & (df['xi'] >= min_xi)].copy()
# df_clean.head(100)
# df_clean.reset_index(drop=True)
```

In [14]:

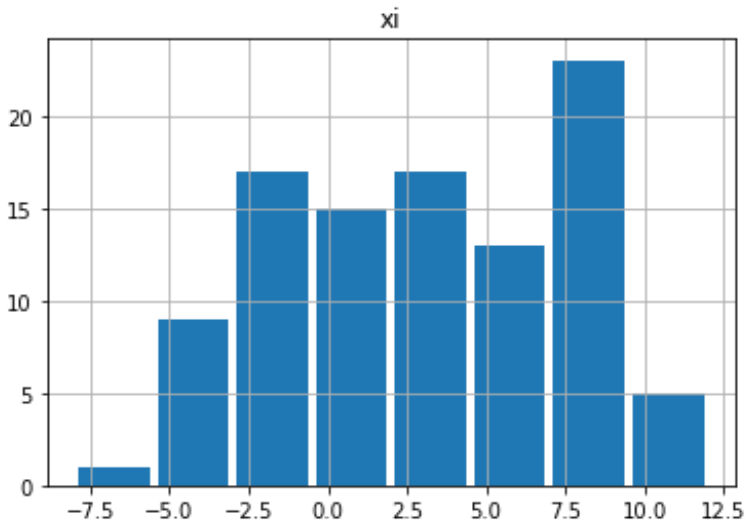
```
df_clean = df.copy()
```

In [15]:

```
# Построим гистаграмму
df_clean.hist(bins=8, rwidth=0.9)
```

Out[15]:

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x00000227A1DDAE80>]],
      dtype=object)
```

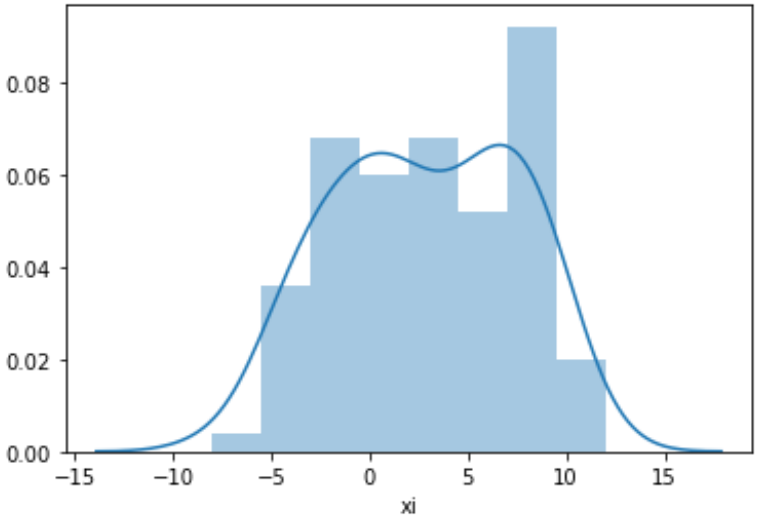


In [16]:

```
# Эмпирическая функция распределения
sns.distplot(df_clean.xi, bins = 8)
```

Out[16]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x227a217b100>
```



In [106]:

```
# посчитаем частоту встречаемости каждого значения

df_list = df_clean.xi.tolist()

df_count = {}

for xi in df_list:
    count = df_count.get(xi,0)
    df_count[xi] = count + 1

df2 = pd.DataFrame()
df2['xi'] = df_count.keys()
df2['ni'] = df_count.values()
df2.sort_values('xi', ascending=True, inplace = True)
# запись в файл
# df2.to_excel("Task_1.xlsx")
df2.head()
```

Out[106]:

	xi	ni
0	-8	1
1	-5	5
2	-4	4
3	-3	5
4	-2	7

In [14]:

```
# посчитаем наращиваемые частоты
wi = []
count = 0
for i in df_count.values() :
    count = count + i
    wi.append(count)
```

```
df2['n+'] = wi
df2.head()
```

Out[14]:

	xi	ni	n+
0	**-8**	**1**	**1**
1	**-5**	**5**	**6**
2	**-4**	**4**	**10**
3	**-3**	**5**	**15**
4	**-2**	**7**	**22**

In [15]:

```
# посчитаем относительные частота

df2['wi'] = df2['ni']/sum(df2['ni'].tolist())
df2.head()
```

Out[15]:

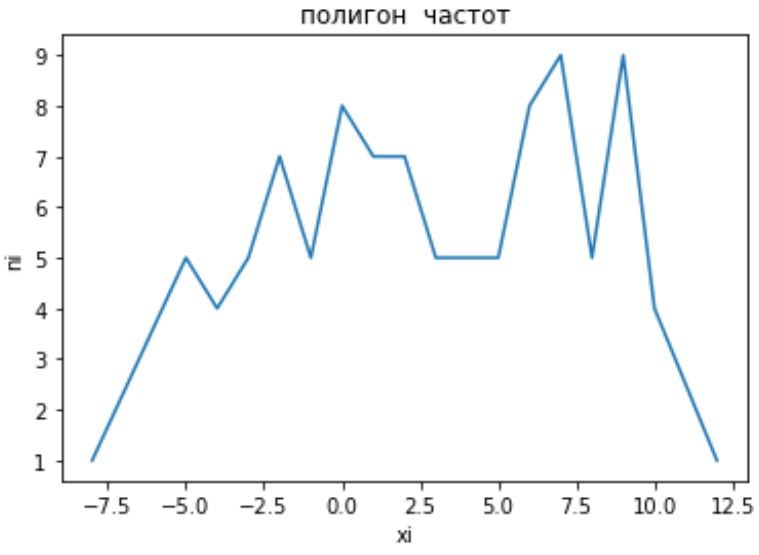
	xi	ni	n+	wi
0	**-8**	**1**	**1**	**0.01**
1	**-5**	**5**	**6**	**0.05**
2	**-4**	**4**	**10**	**0.04**
3	**-3**	**5**	**15**	**0.05**
4	**-2**	**7**	**22**	**0.07**

In [16]:

```
# построим полигон
X = df2['xi'].tolist()
Y = df2['ni'].tolist()
plt.plot(X, Y)
plt.xlabel('xi')
plt.ylabel('ni')
plt.title('полигон  частот')
```

Out[16]:

```
Text(0.5, 1.0, 'полигон  частот')
```



In [17]:

```
# построим полигон относительных частот
Y = df2['wi'].tolist()
plt.plot(X, Y)
plt.xlabel('xi')
plt.ylabel('wi')
plt.title('полигон относительных частот')
```

Out[17]:

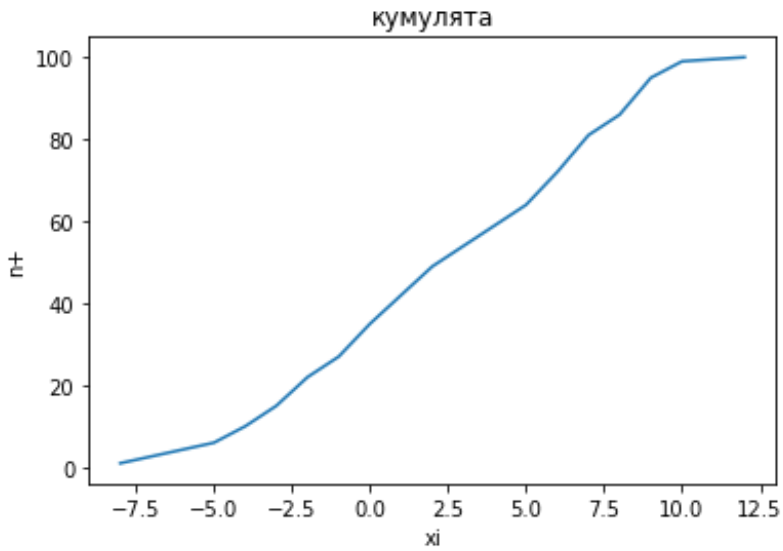
```
Text(0.5, 1.0, 'полигон относительных частот')
```

In [18]:

```
# построим кумулятивную кривую
Y = df2['n+'].tolist()
plt.plot(X, Y)
plt.xlabel('xi')
plt.ylabel('n+')
plt.title('кумулята')
```

Out[18]:

```
Text(0.5, 1.0, 'кумулята')
```



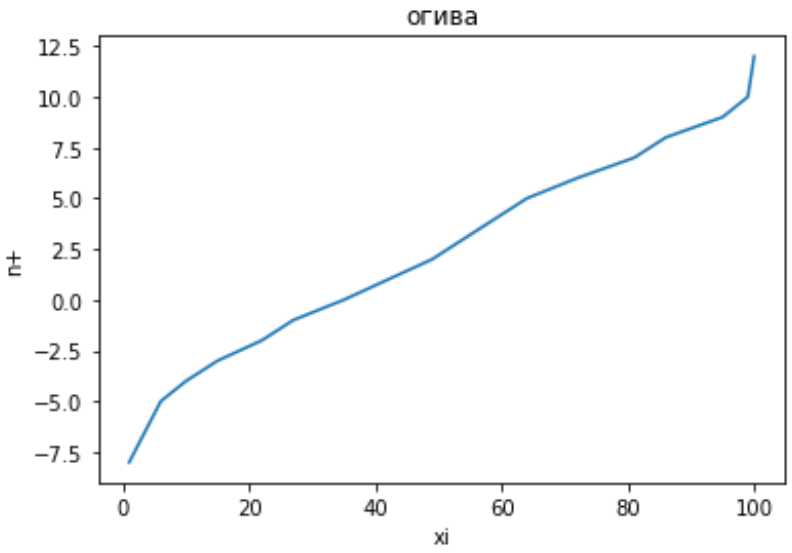
In [19]:

```
# построим огиву
Y = df2['n+'].tolist()
```

```
plt.plot(Y, X)
plt.xlabel('xi')
plt.ylabel('n+')
plt.title('огива')
```

Out[19]:

```
Text(0.5, 1.0, 'огива')
```



In [41]:

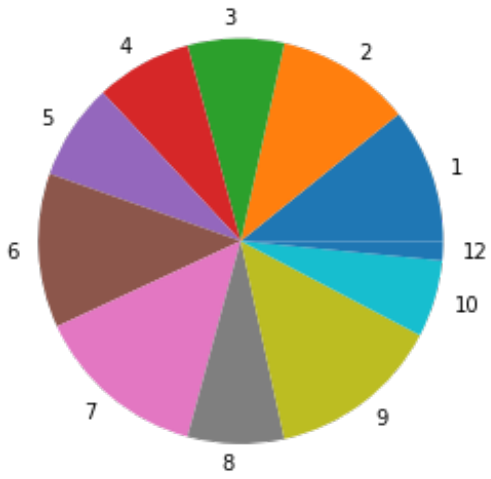
```
# круговая диаграмма - только для положительных значений

temp = df2[df2['xi'] > 0].copy()
vals = temp['ni'].tolist()
labels = temp['xi'].tolist()

fig, ax = plt.subplots()
ax.pie(vals, labels=labels, radius = 200)
ax.axis("equal")
```

Out[41]:

```
(-221.06647946375435,
 220.0507853092253,
 -220.36450723243485,
 220.3645088184055)
```



In [21]:

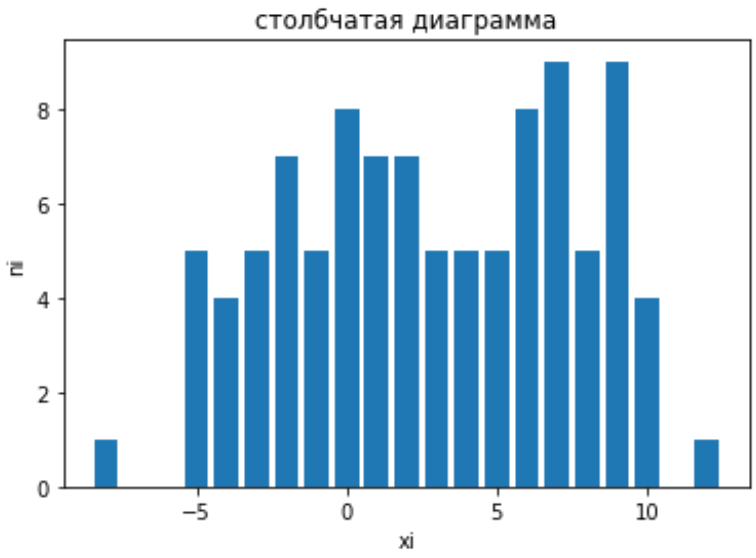
```
# столбчатая диаграмма

X = df2['xi'].tolist()
```

```
Y = df2['ni'].tolist()
plt.bar(X, Y)
plt.xlabel('xi')
plt.ylabel('ni')
plt.title('столбчатая диаграмма')
```

Out[21]:

```
Text(0.5, 1.0, 'столбчатая диаграмма')
```



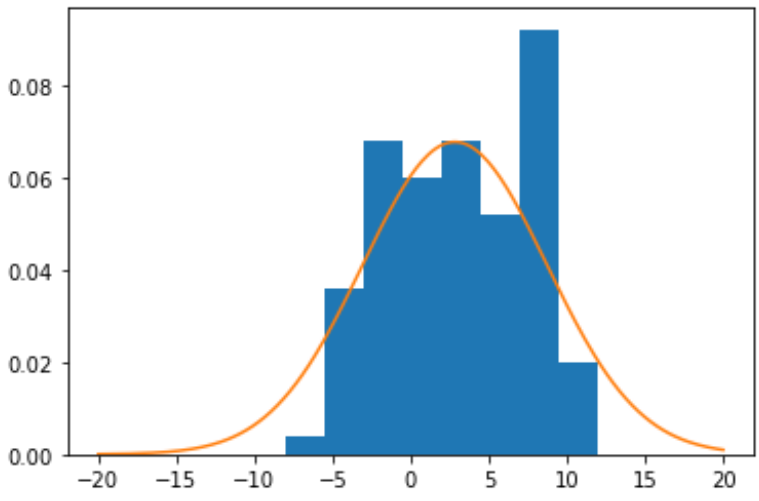
In [102]:

```
# империческая фнукция распредления
import scipy
# sns.distplot(df_clean.xi, bins = 8)

mean = df.mean()
std = np.sqrt(df.describe().std())
X = np.linspace(-20,20,200)
Y = scipy.stats.norm.pdf(X, mean, std)
plt.hist(df['xi'], density=1, bins = 8)
plt.plot(X,Y)

print("Среднее =",round(float(mean),2),'Дисперсия =',round(float(std),2))
```

```
Среднее = 2.82 Дисперсия = 5.9
```



In [95]:

```
# Отдельно империческая функция распредления
plt.plot(X,Y)
plt.xlabel('xi')
```

```
plt.ylabel('ni')
```

Out[95]:

```
Text(0, 0.5, 'ni')
```