Для изучения статистики загружаем библиотеки: math, numpy, pandas, statistics, scipy.stats. Посмотрим, каким образом можно рассчитать центральные метрики, средневзвешенное, гармоническое среднее, среднее геометрическое, медиану, моду, дисперсию, среднеквадратичное отклонение, смещение, процентили, диапазон.

Программный код для расчёта данных показателей:

**import math**

**import statistics**

**import numpy as np**

**import scipy.stats**

**import pandas as pd**

**print("Исходные данные")**

**x = [8.0, 1, 2.5, 4, 28.0]**

**x\_with\_nan = [8.0, 1, 2.5, math.nan, 4, 28.0]**

**y, y\_with\_nan = np.array(x), np.array(x\_with\_nan)**

**z, z\_with\_nan = pd.Series(x), pd.Series(x\_with\_nan)**

**print(y)**

**print(y\_with\_nan)**

**print(z)**

**print(z\_with\_nan)**

**# Среднее значение**

**print("Среднее значение")**

**mean\_=sum(x)/len(x)**

**print(mean\_)**

**mean\_=statistics.mean(x)**

**print(mean\_)**

**m=np.nanmean(y\_with\_nan)**

**print(m)**

**# Средневзвешенное значение**

**print("Средневзвешенное значение")**

**x = [8.0, 1, 2.5, 4, 28.0]**

**w = [0.1, 0.2, 0.3, 0.25, 0.15]**

**wmean = sum(w[i] \* x[i] for i in range(len(x))) / sum(w)**

**print(wmean)**

**wmean = sum(x\_ \* w\_ for (x\_, w\_) in zip(x, w)) / sum(w)**

**print(wmean)**

**# Средневзвешенное значение, использование массивов Numpy и Pandas**

**x = [8.0, 1, 2.5, 4, 28.0]**

**y, z, w = np.array(x), pd.Series(x), np.array(w)**

**wmean = np.average(y, weights=w)**

**print(wmean)**

**wmean = np.average(z, weights=w)**

**print(wmean)**

**# Гармоническое среднее**

**print("Гармоническое среднее")**

**hmean = len(x) / sum(1 / item for item in x)**

**print(hmean)**

**hmean==scipy.stats.hmean(y)**

**print(hmean)**

**# Среднее геометрическое**

**print("Среднее геометрическое")**

**gmean = 1**

**for item in x:**

**gmean \*= item**

**gmean \*\*= 1 / len(x)**

**print(gmean)**

**# Медиана**

**print("Медиана")**

**n = len(x)**

**if n % 2:**

**median\_ = sorted(x)[round(0.5\*(n-1))]**

**else:**

**x\_ord, index = sorted(x), round(0.5 \* n)**

**median\_ = 0.5 \* (x\_ord[index-1] + x\_ord[index])**

**print(median\_)**

**print(z.median())**

**print(z\_with\_nan.median())**

**# Медиана**

**u = [2, 3, 2, 8, 12]**

**mode\_ = max((u.count(item), item) for item in set(u))[1]**

**print(mode\_)**

**# Дисперсия**

**print("Дисперсия")**

**n = len(x)**

**mean\_ = sum(x) / n**

**var\_ = sum((item - mean\_)\*\*2 for item in x) / (n - 1)**

**print(var\_)**

**# Среднеквадратическое отклонение**

**print("Среднеквадратическое отклонение")**

**std\_ = var\_ \*\* 0.5**

**print(std\_)**

**std\_=np.std(y, ddof=1)**

**print(std\_)**

**# Смещение**

**print("Смещение")**

**y, y\_with\_nan = np.array(x), np.array(x\_with\_nan)**

**print(scipy.stats.skew(y, bias=False))**

**print(scipy.stats.skew(y\_with\_nan, bias=False))**

**# Процентили**

**print("Процентили")**

**y = np.array(x)**

**print(np.percentile(y, 5))**

**print(np.percentile(y, 95))**

**# Диапазон**

**print("Диапазон")**

**print(np.amax(y) - np.amin(y))**

**print(np.nanmax(y\_with\_nan) - np.nanmin(y\_with\_nan))**

**print(y.max() - y.min())**

**print(z.max() - z.min())**

**print(z\_with\_nan.max() - z\_with\_nan.min())**

Результат работы программы:

