

Задание

Задать значения количества продаж по 10 товарам в течении 12 месяцев (помесячно). Для каждого из товаров спрогнозировать количество продаж на следующий, 13 месяц и провести анализ достоверности планирования продаж

```
In [1]: import random
# Для удобной работы с массивами
import numpy as np
# Для представления данных в виде датафреймов
import pandas as pd
# Для рисования графиков
import matplotlib.pyplot as plt
# Для отключения предупреждений
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)

class Product:
    def __init__(self, name, count=None):
        """
        Конструктор класса
        """
        self.name = name
        if count is not None:
            self.count = count
        else:
            min_rand = random.randint(100, 150)
            delt_rand = 20
            self.count = np.random.normal(min_rand, delt_rand, 12)

    def __str__(self):
        """
        Метод для конвертации в строку
        """
        return f'{self.name}: {[i for i in self.count]}'

    def __repr__(self):
        """
        Метод для представления в виде строки из дебаг-консоли
        """
        return f'{self.name}: {[i for i in self.count]}'

    def to_dict(self):
        """
        Возвращает класс в представлении словаря
        """
        return {
            self.name: self.count
        }
```

```

def sum(self):
    """
    Возвращает общую стоимость за все месяца
    """
    return sum(self.count)

def avg(self):
    """
    Возвращает среднюю цену
    """
    return round(sum(self.count) / len(self.count), 4)

def msd(self):
    """
    Возвращает СКО для продукции
    """
    avg_value = self.avg()
    upper_value = sum([(v - avg_value) ** 2 for v in self.count])
    msd_square = upper_value / (len(self.count) - 1)
    return msd_square ** 0.5

```

```

In [2]: products = [
    Product(name='Смартфон'),
    Product(name='Часы'),
    Product(name='Монитор'),
    Product(name='Наушники'),
    Product(name='Колонка'),
    Product(name='Камера'),
    Product(name='Принтер'),
    Product(name='Игровая приставка'),
    Product(name='Клавиатура'),
    Product(name='Видеокарта')
]

products

```

```
Out[2]: [Смартфон: [107.7975376328739, 114.30069256238663, 88.96966303124007, 97.521681
30786336, 108.58370962630686, 92.7905537385482, 154.22249435845356, 136.3318360
386703, 148.84977355000387, 109.69046709167563, 106.18119353936999, 100.6144361
7897096],
Часы: [108.82839269497705, 147.89736681458513, 100.03905111017365, 113.3243436
4146103, 79.65591201431525, 113.31147928433005, 116.87142246801939, 98.23313218
37287, 127.53303180260508, 109.33324282619236, 94.76475721376409, 106.277570097
77689],
Монитор: [111.51745390231288, 112.33620787433145, 130.17868977263635, 101.4426
7817506653, 111.33916241803682, 154.1457054375556, 121.53863967783379, 110.6194
6596843075, 88.89104479637376, 112.98586666666228, 123.76190174876533, 89.52149
058444175],
Наушники: [91.75309091207805, 81.21709046202244, 132.6103981232436, 116.030000
45735021, 141.40974506233283, 116.6021600153522, 99.9875997836876, 127.92130511
605312, 132.21361373550477, 125.30969225019588, 103.64651920898568, 107.2657382
3634851],
Колонка: [105.88844240700897, 80.01415192072898, 91.67479951331961, 89.7263879
6380369, 109.50113305252057, 116.29571169523615, 117.00309236699692, 113.309304
71409029, 109.11859201805594, 111.72119467522977, 105.53493351966544, 120.23113
941978286],
Камера: [141.28605874048975, 118.41990664812928, 91.54369258796093, 67.7543444
2571535, 84.09255444972607, 94.60515925575972, 117.63787356528624, 83.859145796
37172, 147.54386421653356, 78.97549091797276, 110.28704249552196, 83.1360501115
5819],
Принтер: [160.86310171319434, 151.66649351304713, 151.67242851775222, 110.7033
636904919, 101.50665306381734, 173.31583961213107, 118.28209765699708, 150.0013
4807784153, 159.2891626963951, 161.32817263255876, 166.62936006662258, 145.8690
876115951],
Игровая приставка: [100.65364143223353, 135.58192521638549, 142.0036293938955
5, 78.40749426054713, 125.01236751005285, 109.30472631281877, 136.8701251868019
7, 97.44531586186021, 90.8601068728075, 113.77652351089381, 128.17878957902738,
77.06940644493227],
Клавиатура: [109.34807886927872, 86.51433669527172, 82.20764255866777, 72.5944
3727737818, 109.29895933023538, 108.5718357299993, 109.41763338378034, 111.6358
2149411015, 57.024326518999544, 101.3729776993001, 115.97897766257793, 94.96827
960263623],
Видеокарта: [113.83206034640614, 120.53221983864466, 152.23193458395517, 135.9
3661023092525, 138.39119747354817, 100.52192379274966, 115.39456142208333, 119.
07576956394436, 81.33977022719398, 112.272966567159, 101.99938519493601, 93.955
73111762631]]
```

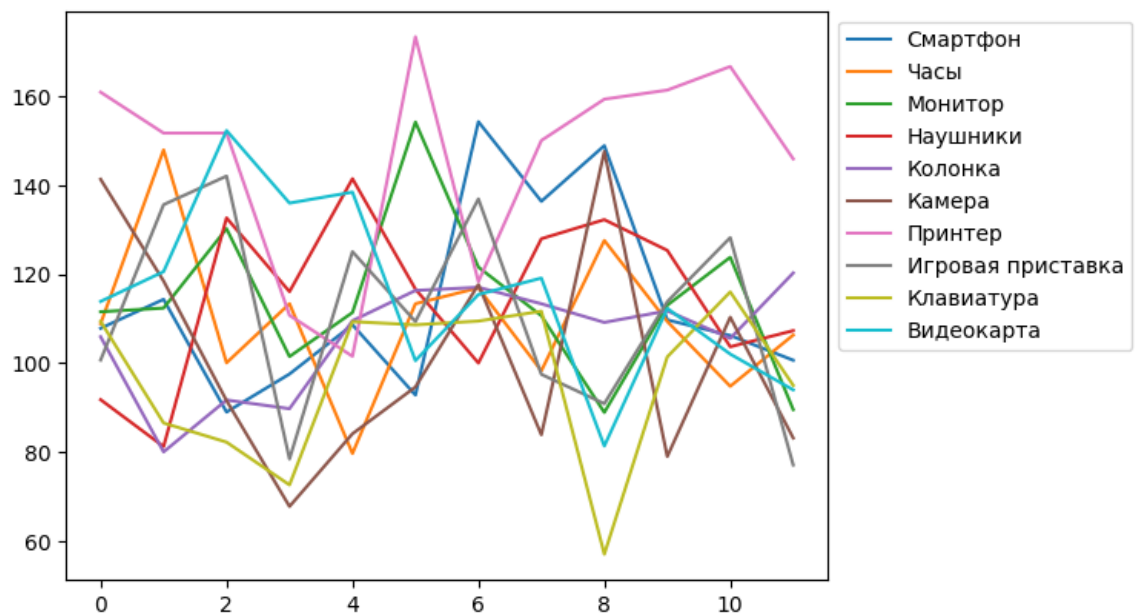
```
In [3]: def convert_list_products_to_dict(p_list: list):
        """
        Конвертирует лист продуктов в словарь для визуализации в датафрейме
        """
        result = {}
        for p in p_list:
            result[p.name] = p.count
        return result
```

```
In [4]: df = pd.DataFrame(convert_list_products_to_dict(products))
df
```

Out[4]:

	Смартфон	Часы	Монитор	Наушники	Колонка	Камера	Принтер	Игр прист
0	107.797538	108.828393	111.517454	91.753091	105.888442	141.286059	160.863102	100.65
1	114.300693	147.897367	112.336208	81.217090	80.014152	118.419907	151.666494	135.58
2	88.969663	100.039051	130.178690	132.610398	91.674800	91.543693	151.672429	142.00
3	97.521681	113.324344	101.442678	116.030000	89.726388	67.754344	110.703364	78.40
4	108.583710	79.655912	111.339162	141.409745	109.501133	84.092554	101.506653	125.01
5	92.790554	113.311479	154.145705	116.602160	116.295712	94.605159	173.315840	109.30
6	154.222494	116.871422	121.538640	99.987600	117.003092	117.637874	118.282098	136.87
7	136.331836	98.233132	110.619466	127.921305	113.309305	83.859146	150.001348	97.44
8	148.849774	127.533032	88.891045	132.213614	109.118592	147.543864	159.289163	90.86
9	109.690467	109.333243	112.985867	125.309692	111.721195	78.975491	161.328173	113.77
10	106.181194	94.764757	123.761902	103.646519	105.534934	110.287042	166.629360	128.17
11	100.614436	106.277570	89.521491	107.265738	120.231139	83.136050	145.869088	77.06

```
In [5]: for product in products:
plt.plot([i for i in range(12)], product.count, label=product.name)
plt.legend(bbox_to_anchor=(1, 1))
plt.show()
```



Вычисляем оценку тренда (среднее арифметическое величины)

```
In [6]: p0 = [product.avg() for product in products]
[f'{p.name}: {p_avg}' for p_avg, p in zip(p0, products)]
```

```
Out[6]: ['Смартфон: 113.8212',  
        'Часы: 109.6725',  
        'Монитор: 114.0232',  
        'Наушники: 114.6639',  
        'Колонка: 105.8349',  
        'Камера: 101.5951',  
        'Принтер: 145.9273',  
        'Игровая приставка: 111.2637',  
        'Клавиатура: 96.5778',  
        'Видеокарта: 115.457']
```

Вычисляем оценку среднеквадратичного отклонения

```
In [7]: msd_products = [product.msd() for product in products]  
        [f'{product.name}: {msd_value}' for msd_value, product in zip(msd_products, products)]
```

```
Out[7]: ['Смартфон: 21.33148027442298',  
        'Часы: 17.103411782511806',  
        'Монитор: 17.687587291184247',  
        'Наушники: 18.27872023301185',  
        'Колонка: 12.369112499450388',  
        'Камера: 25.353911148873852',  
        'Принтер: 23.116768205045965',  
        'Игровая приставка: 22.65535053030387',  
        'Клавиатура: 18.364525568905574',  
        'Видеокарта: 19.95421403892538']
```

Прогнозируем количество продаж на следующий, 13 месяц

```
In [8]: predict_values = p0 + np.random.normal(0, msd_products, len(msd_products))  
        predict_values
```

```
Out[8]: array([140.95869555, 101.18535081, 104.16311803, 122.12752784,  
              117.03441841,  75.34356682, 146.41389122, 123.039648 ,  
              76.90969946,  82.02308379])
```

```
In [9]: for product, predict_value in zip(products, predict_values):  
        product.count = np.append(product.count, predict_value)  
        products
```

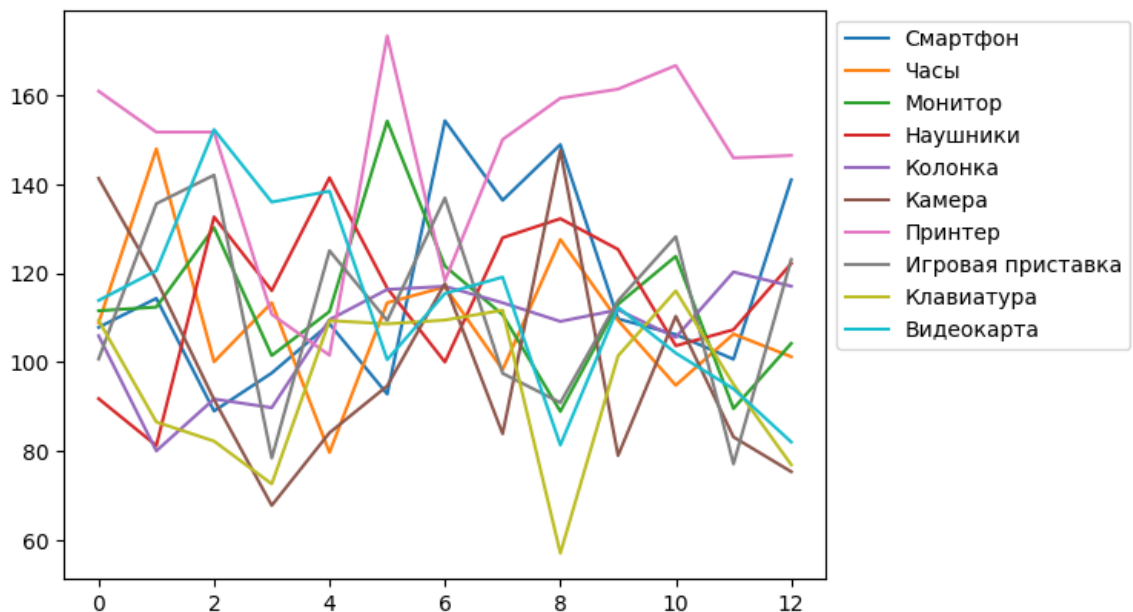
```
Out[9]: [Смартфон: [107.7975376328739, 114.30069256238663, 88.96966303124007, 97.521681
30786336, 108.58370962630686, 92.7905537385482, 154.22249435845356, 136.3318360
386703, 148.84977355000387, 109.69046709167563, 106.18119353936999, 100.6144361
7897096, 140.95869555281433],
Часы: [108.82839269497705, 147.89736681458513, 100.03905111017365, 113.3243436
4146103, 79.65591201431525, 113.31147928433005, 116.87142246801939, 98.23313218
37287, 127.53303180260508, 109.33324282619236, 94.76475721376409, 106.277570097
77689, 101.18535081156556],
Монитор: [111.51745390231288, 112.33620787433145, 130.17868977263635, 101.4426
7817506653, 111.33916241803682, 154.1457054375556, 121.53863967783379, 110.6194
6596843075, 88.89104479637376, 112.98586666666228, 123.76190174876533, 89.52149
058444175, 104.16311802934753],
Наушники: [91.75309091207805, 81.21709046202244, 132.6103981232436, 116.030000
45735021, 141.40974506233283, 116.6021600153522, 99.9875997836876, 127.92130511
605312, 132.21361373550477, 125.30969225019588, 103.64651920898568, 107.2657382
3634851, 122.12752783761158],
Колонка: [105.88844240700897, 80.01415192072898, 91.67479951331961, 89.7263879
6380369, 109.50113305252057, 116.29571169523615, 117.00309236699692, 113.309304
71409029, 109.11859201805594, 111.72119467522977, 105.53493351966544, 120.23113
941978286, 117.0344184127294],
Камера: [141.28605874048975, 118.41990664812928, 91.54369258796093, 67.7543444
2571535, 84.09255444972607, 94.60515925575972, 117.63787356528624, 83.859145796
37172, 147.54386421653356, 78.97549091797276, 110.28704249552196, 83.1360501115
5819, 75.34356681645926],
Принтер: [160.86310171319434, 151.66649351304713, 151.67242851775222, 110.7033
636904919, 101.50665306381734, 173.31583961213107, 118.28209765699708, 150.0013
4807784153, 159.2891626963951, 161.32817263255876, 166.62936006662258, 145.8690
876115951, 146.41389121945795],
Игровая приставка: [100.65364143223353, 135.58192521638549, 142.0036293938955
5, 78.40749426054713, 125.01236751005285, 109.30472631281877, 136.8701251868019
7, 97.44531586186021, 90.8601068728075, 113.77652351089381, 128.17878957902738,
77.06940644493227, 123.03964799978502],
Клавиатура: [109.34807886927872, 86.51433669527172, 82.20764255866777, 72.5944
3727737818, 109.29895933023538, 108.5718357299993, 109.41763338378034, 111.6358
2149411015, 57.024326518999544, 101.3729776993001, 115.97897766257793, 94.96827
960263623, 76.90969946403888],
Видеокарта: [113.83206034640614, 120.53221983864466, 152.23193458395517, 135.9
3661023092525, 138.39119747354817, 100.52192379274966, 115.39456142208333, 119.
07576956394436, 81.33977022719398, 112.272966567159, 101.99938519493601, 93.955
73111762631, 82.02308378624997]]
```

```
In [10]: df = pd.DataFrame(convert_list_products_to_dict(products))
df
```

Out[10]:

	Смартфон	Часы	Монитор	Наушники	Колонка	Камера	Принтер	Игр прист
0	107.797538	108.828393	111.517454	91.753091	105.888442	141.286059	160.863102	100.65
1	114.300693	147.897367	112.336208	81.217090	80.014152	118.419907	151.666494	135.58
2	88.969663	100.039051	130.178690	132.610398	91.674800	91.543693	151.672429	142.00
3	97.521681	113.324344	101.442678	116.030000	89.726388	67.754344	110.703364	78.40
4	108.583710	79.655912	111.339162	141.409745	109.501133	84.092554	101.506653	125.01
5	92.790554	113.311479	154.145705	116.602160	116.295712	94.605159	173.315840	109.30
6	154.222494	116.871422	121.538640	99.987600	117.003092	117.637874	118.282098	136.87
7	136.331836	98.233132	110.619466	127.921305	113.309305	83.859146	150.001348	97.44
8	148.849774	127.533032	88.891045	132.213614	109.118592	147.543864	159.289163	90.86
9	109.690467	109.333243	112.985867	125.309692	111.721195	78.975491	161.328173	113.77
10	106.181194	94.764757	123.761902	103.646519	105.534934	110.287042	166.629360	128.17
11	100.614436	106.277570	89.521491	107.265738	120.231139	83.136050	145.869088	77.06
12	140.958696	101.185351	104.163118	122.127528	117.034418	75.343567	146.413891	123.03

```
In [11]: for product in products:
plt.plot([i for i in range(13)], product.count, label=product.name)
plt.legend(bbox_to_anchor=(1, 1))
plt.show()
```



Анализ достоверности планирования продаж

Расчитаем достоверность прогноза

```
In [12]: reliability = [msd_value / p0_value for msd_value, p0_value in zip(msd_products,
[f'{product.name}: {reliability_value}' for product, reliability_value in zip(pr
```

```
Out[12]: ['Смартфон: 0.18741218924438485',
'Часы: 0.15594986694487503',
'Монитор: 0.15512270565274652',
'Наушники: 0.15941129015332506',
'Колонка: 0.11687177386146146',
'Камера: 0.24955840536476515',
'Принтер: 0.15841290975058103',
'Игровая приставка: 0.20361852545173198',
'Клавиатура: 0.1901526600202694',
'Видеокарта: 0.1728281008420917']
```

Первое условие

Величины X_i лежат близко к своему среднему значению. Это условие означает, что продажи стабильные

$$X_i - \hat{p}_0 < 2\hat{\sigma} \text{ для всех } i$$

```
In [13]: p0_np = np.array(p0)
msd_np = np.array(msd_products)
products_df = pd.DataFrame(convert_list_products_to_dict(products))

condition_one = products_df - p0_np < (2 * msd_np)
condition_one
```

```
Out[13]:
```

	Смартфон	Часы	Монитор	Наушники	Колонка	Камера	Принтер	Игровая приставка	Клави
0	True	True	True	True	True	True	True	True	
1	True	False	True	True	True	True	True	True	
2	True	True	True	True	True	True	True	True	
3	True	True	True	True	True	True	True	True	
4	True	True	True	True	True	True	True	True	
5	True	True	False	True	True	True	True	True	
6	True	True	True	True	True	True	True	True	
7	True	True	True	True	True	True	True	True	
8	True	True	True	True	True	True	True	True	
9	True	True	True	True	True	True	True	True	
10	True	True	True	True	True	True	True	True	
11	True	True	True	True	True	True	True	True	
12	True	True	True	True	True	True	True	True	

```
In [14]: condition_one.all()
```



```
Out[14]: Смартфон          True
         Часы             False
         Монитор          False
         Наушники         True
         Колонка          True
         Камера           True
         Принтер          True
         Игровая приставка True
         Клавиатура       True
         Видеокарта       True
         dtype: bool
```

Второе условие

Экстраполируемые значения X_i с большей степенью достоверности не равны 0. Это условие означает, что период для прогноза не слишком велик

$$\hat{p}_0 > 2\hat{\sigma}$$

```
In [15]: condition_two = p0_np > (2 * msd_np)
         condition_two
```

```
Out[15]: array([ True,  True,  True,  True,  True,  True,  True,  True,  True,
                True])
```

```
In [16]: [f'{product.name}: {condition_value}' for product, condition_value in zip(products, condition_two)]
```

```
Out[16]: ['Смартфон: True',
         'Часы: True',
         'Монитор: True',
         'Наушники: True',
         'Колонка: True',
         'Камера: True',
         'Принтер: True',
         'Игровая приставка: True',
         'Клавиатура: True',
         'Видеокарта: True']
```

Третье условие

Среди X_i не встречаются нулевые значения. Это условие означает, что периодичность анализа данных выбрана правильно

$$X_i > 0 \text{ для всех } i$$

```
In [17]: condition_three = products_df > 0
         condition_three
```

Out[17]:

	Смартфон	Часы	Монитор	Наушники	Колонка	Камера	Принтер	Игровая приставка	Клави
0	True	True	True	True	True	True	True	True	
1	True	True	True	True	True	True	True	True	
2	True	True	True	True	True	True	True	True	
3	True	True	True	True	True	True	True	True	
4	True	True	True	True	True	True	True	True	
5	True	True	True	True	True	True	True	True	
6	True	True	True	True	True	True	True	True	
7	True	True	True	True	True	True	True	True	
8	True	True	True	True	True	True	True	True	
9	True	True	True	True	True	True	True	True	
10	True	True	True	True	True	True	True	True	
11	True	True	True	True	True	True	True	True	
12	True	True	True	True	True	True	True	True	

In [18]: `condition_three.all()`

Out[18]: Смартфон True
Часы True
Монитор True
Наушники True
Колонка True
Камера True
Принтер True
Игровая приставка True
Клавиатура True
Видеокарта True
dtype: bool

Составим таблицу для визуализации достоверности планирования

```
In [19]: def get_color_by_condition(c1, c2, c3):
    """
    Получение уровня достоверности по условиям
    """
    if c1 and c2 and c3:
        return "Зеленый"
    elif (not c1 and c2 and c3) or (c1 and not c2 and c3):
        return "Желтый"
    elif not c1 and not c2 and c3:
        return "Оранжевый"
    else:
        return "Красный"
```

```
In [20]: table_header = ["Товар", "Уровень достоверности", "Условие 1", "Условие 2", "Усл
table = pd.DataFrame(columns=table_header)
```

```

condition_one_result = condition_one.all()
condition_two_result = condition_two
condition_three_result = condition_three.all()

for i in range(len(products)):
    table = table.append({
        "Товар": products[i].name,
        "Уровень достоверности": get_color_by_condition(condition_one_result[i],
        "Условие 1": condition_one_result[i],
        "Условие 2": condition_two_result[i],
        "Условие 3": condition_three_result[i]
    }, ignore_index=True)

table

```

Out[20]:

	Товар	Уровень достоверности	Условие 1	Условие 2	Условие 3
0	Смартфон	Зеленый	True	True	True
1	Часы	Желтый	False	True	True
2	Монитор	Желтый	False	True	True
3	Наушники	Зеленый	True	True	True
4	Колонка	Зеленый	True	True	True
5	Камера	Зеленый	True	True	True
6	Принтер	Зеленый	True	True	True
7	Игровая приставка	Зеленый	True	True	True
8	Клавиатура	Зеленый	True	True	True
9	Видеокарта	Зеленый	True	True	True