

**Липецкий государственный технический университет**

**Факультет автоматизации и информатики**

**Кафедра автоматизированных систем управления**

**ЛАБОРАТОРНАЯ РАБОТА №1**

**по дисциплине «Прикладные интеллектуальные системы и экспертные  
системы»**

**Экспертные системы. Оценка планирования продаж.**

Студент

Мастылина А.А.

Группа М-ИАП-22

Руководитель

Кургасов В.В.

Липецк 2022 г.

### Задание кафедры

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

## Ход работы

### Используемые библиотеки:

- numpy предназначена для поддержки многомерных массивов (включая матрицы), поддержки высокоуровневых математических функций;
- pandas предназначена для обработки и анализа данных.

#### Сгенерируем информацию о продажах 10-ти товаров за 12 месяцев

```
Ввод [2]: product = {
    'Мандарин': numpy.random.normal(20, 5, 12),
    'Апельсин': numpy.random.normal(30, 6, 12),
    'Сыр': numpy.random.normal(70, 5, 12),
    'Халва': numpy.random.normal(20, 5, 12),
    'Хлеб': numpy.random.normal(90, 6, 12),
    'Шоколад': numpy.random.normal(60, 20, 12),
    'Ментай': numpy.random.normal(40, 5, 12),
    'Карп': numpy.random.normal(80, 10, 12),
    'Макароны': numpy.random.normal(90, 15, 12),
    'Лаваш': numpy.random.normal(30, 10, 12),
}
```

Рисунок 1 – Генерация данных

#### Сформируем единую таблицу из сгенерированных данных.

```
Ввод [3]: dataframe = pandas.DataFrame(product)
dataframe
```

Out[3]:

	Мандарин	Апельсин	Сыр	Халва	Хлеб	Шоколад	Ментай	Карп	Макароны	Лаваш
0	9.811846	21.959409	61.848248	23.271025	94.628954	32.363633	45.700550	99.001333	95.531549	29.910203
1	15.271764	32.460863	69.743459	13.953545	89.611626	79.253148	37.015474	88.700956	91.075317	36.739077
2	9.654317	21.831715	61.505943	11.253538	88.915697	95.636437	42.688007	79.429050	108.024575	28.494973
3	23.314499	37.631374	61.422785	19.605064	85.109626	32.939636	51.648871	95.539333	100.778377	26.811033
4	17.503738	31.448129	60.189718	24.996658	90.866177	66.824334	36.167796	73.860914	110.853173	28.117465
5	23.899980	30.668903	74.014545	21.330899	86.946915	97.294648	35.477159	93.792970	105.215073	31.474088
6	18.381948	31.378741	71.035342	19.401342	91.294328	31.443281	29.340539	85.442236	109.502706	21.847648
7	21.375107	31.780447	74.295248	16.798385	94.363248	78.513904	37.749642	68.175179	94.520905	34.489691
8	12.770799	33.824142	62.037091	17.017827	93.591638	64.663140	39.621295	78.866330	93.396984	30.999823
9	16.271636	17.719043	69.703919	17.693947	98.930813	56.567183	32.948477	77.672839	112.197481	21.934413
10	24.894806	41.717621	81.649210	15.662535	86.147088	37.444455	38.709275	84.123567	102.704044	11.805920
11	22.785073	37.784263	73.674880	20.462059	83.585679	97.499618	44.298144	76.334848	72.442261	3.603467

Рисунок 2 – Сформированные данные

#### Рассчитаем среднее арифметическое величины $X_i$

```
Ввод [4]: p0 = dataframe.sum() / 12
p0
```

Out[4]:

Мандарин	17.994626
Апельсин	30.850387
Сыр	68.426699
Халва	18.453902
Хлеб	90.332649
Шоколад	64.203618
Ментай	39.280436
Карп	83.411630
Макароны	99.686870
Лаваш	25.518983
dtype:	float64

Рисунок 3 – Расчёт среднего арифметического ( $\hat{p}_0$ )

### Рассчитаем среднее квадратичное отклонение величины $X_i$ и сразу извлечем квадратный корень

```
Ввод [5]: sigma = (((dataFrame - p0) ** 2).sum() / (12 - 1)) ** (1 / 2)
sigma

Out[5]: Мандарин      5.391436
        Апельсин     7.121967
        Сыр          6.922394
        Халва        3.882545
        Хлеб         4.520400
        Шоколад      26.057354
        Ментай       6.044904
        Карп         9.406962
        Макароны     11.232540
        Лаваш        9.549363
        dtype: float64
```

Рисунок 4 – Расчет среднеквадратичного отклонения

### Значение планируемого показателя на 13-ый месяц

```
Ввод [6]: x_i = p0 + numpy.random.normal(0, sigma, len(p0))
columns = x_i.index
pandas.concat([dataFrame, pandas.DataFrame([x_i], columns=columns)]).reset_index(drop=True)

Out[6]:
```

	Мандарин	Апельсин	Сыр	Халва	Хлеб	Шоколад	Ментай	Карп	Макароны	Лаваш
0	9.811846	21.959409	61.848248	23.271025	94.628954	32.363633	45.700550	99.001333	95.531549	29.910203
1	15.271764	32.460863	69.743459	13.953545	89.611626	79.253148	37.015474	88.700956	91.075317	36.739077
2	9.654317	21.831715	61.505943	11.253538	88.915697	95.636437	42.688007	79.429050	108.024575	28.494973
3	23.314499	37.631374	61.422785	19.605064	85.109626	32.939636	51.648871	95.539333	100.778377	26.811033
4	17.503738	31.448129	60.189718	24.996658	90.866177	66.824334	36.167796	73.860914	110.853173	28.117465
5	23.899980	30.668903	74.014545	21.330899	86.946915	97.294648	35.477159	93.792970	105.215073	31.474088
6	18.381948	31.378741	71.035342	19.401342	91.294328	31.443281	29.340539	85.442236	109.502706	21.847648
7	21.375107	31.780447	74.295248	16.798385	94.363248	78.513904	37.749642	68.175179	94.520905	34.489691
8	12.770799	33.824142	62.037091	17.017827	93.591638	64.663140	39.621295	78.866330	93.396984	30.999823
9	16.271636	17.719043	69.703919	17.693947	98.930813	56.567183	32.948477	77.672839	112.197481	21.934413
10	24.894806	41.717621	81.649210	15.662535	86.147088	37.444455	38.709275	84.123567	102.704044	11.805920
11	22.785073	37.784263	73.674880	20.462059	83.585679	97.499618	44.298144	76.334848	72.442261	3.603467
12	21.542578	34.871664	66.086589	24.796169	86.380606	48.887855	30.589523	82.329157	105.036841	27.977388

Рисунок 5 – Расчёт планируемого показателя на 13-ый месяц

### Проверим три условия, характеризующие достоверность прогноза

```
Ввод [7]: authenticity = sigma / p0
authenticity

Out[7]: Мандарин      0.299614
        Апельсин     0.230855
        Сыр          0.101165
        Халва        0.210392
        Хлеб         0.050042
        Шоколад      0.405855
        Ментай       0.153891
        Карп         0.112778
        Макароны     0.112678
        Лаваш        0.374206
        dtype: float64
```

Рисунок 6 – Рассчитаем соотношение  $\frac{\hat{\sigma}_p}{\hat{p}_0}$

## Условие 1

```
Ввод [8]: condition1 = ((dataFrame - p0) < 2 * sigma).all()  
condition1
```

```
Out[8]: Мандарин      True  
        Апельсин     True  
        Сыр          True  
        Халва        True  
        Хлеб         True  
        Шоколад      True  
        Ментай       False  
        Карп         True  
        Макароны     True  
        Лаваш        True  
dtype: bool
```

Рисунок 7 – Проверка первого условия

## Условие 2

```
Ввод [9]: condition2 = p0 > 2 * sigma  
condition2
```

```
Out[9]: Мандарин      True  
        Апельсин     True  
        Сыр          True  
        Халва        True  
        Хлеб         True  
        Шоколад      True  
        Ментай       True  
        Карп         True  
        Макароны     True  
        Лаваш        True  
dtype: bool
```

Рисунок 8 – Проверка второго условия

### Условие 3

```
Ввод [10]: condition3 = (dataFrame > 0).all()  
condition3
```

```
Out[10]: Мандарин      True  
Апельсин      True  
Сыр           True  
Халва         True  
Хлеб          True  
Шоколад       True  
Ментай        True  
Карп          True  
Макароны      True  
Лаваш         True  
dtype: bool
```

Рисунок 9 – Проверка третьего условия

```
Out[11]:
```

	keyword	color
0	Мандарин	green
1	Апельсин	green
2	Сыр	green
3	Халва	green
4	Хлеб	green
5	Шоколад	green
6	Ментай	yellow
7	Карп	green
8	Макароны	green
9	Лаваш	green

Рисунок 10 – Сведем итоги в таблицу

Таким образом, автоматизированное планирование продаж достоверно у всех, кроме позиции «ментай», для нее нужна корректировка полученных данных.

Код программы

```
import numpy as numpy
import pandas as pandas
import seaborn as sns

## Сгенерируем информацию о продажах 10-ти товаров за 12 месяцев
product = {
    'Мандарин': numpy.random.normal(20, 5, 12),
    'Апельсин': numpy.random.normal(30, 6, 12),
    'Сыр': numpy.random.normal(70, 5, 12),
    'Халва': numpy.random.normal(20, 5, 12),
    'Хлеб': numpy.random.normal(90, 6, 12),
    'Шоколад': numpy.random.normal(60, 20, 12),
    'Ментай': numpy.random.normal(40, 5, 12),
    'Карп': numpy.random.normal(80, 10, 12),
    'Макароны': numpy.random.normal(90, 15, 12),
    'Лаваш': numpy.random.normal(30, 10, 12),
}

## Сформируем единую таблицу из сгенерированных данных.
dataFrame = pandas.DataFrame(product)
dataFrame

## Рассчитаем среднее арифметическое величины  $X_i$ 
p0 = dataFrame.sum() / 12
p0

## Рассчитаем среднее квадратичное отклонение величины  $X_i$  и сразу
извлечем квадратный корень
sigma = (((dataFrame - p0) ** 2).sum() / (12 - 1)) ** (1 / 2)
sigma

## Значение планируемого показателя на 13-ый месяц
x_i = p0 + numpy.random.normal(0, sigma, len(p0))
columns = x_i.index
```

```

pandas.concat([dataFrame, pandas.DataFrame([x_i],
columns=columns)]).reset_index(drop=True)

## Проверим три условия, характеризующие достоверность прогноза
authenticity = sigma / p0
authenticity

## Условие 1
condition1 = ((dataFrame - p0) < 2 * sigma).all()
condition1

## Условие 2
condition2 = p0 > 2 * sigma
condition2

## Условие 3
condition3 = (dataFrame > 0).all()
condition3

## Итоги
df_1 = pandas.DataFrame({'keyword':dataFrame.columns,
                        'color': 'green'})

i = 0

for name in dataFrame.columns:
    if (not condition3[name]):
        df_1.color[i] = 'red'
    elif (not condition1[name] and not condition2[name]):
        df_1.color[i] = 'orange'
    elif (not condition1[name] or not condition2[name]):
        df_1.color[i] = 'yellow'
    i += 1

new_dict = pandas.Series(df_1.color.values,index=df_1.keyword).to_dict()

```



```
df_1.style.applymap(lambda v: f"background-color: {new_dict.get(v,  
'None')}")
```

## Вывод

В ходе выполнения данной лабораторной работы мы получили базовые навыки работы с языком python и набором функций для анализа и обработки данных. Спрогнозировали количество продаж по десяти товарам на тринадцатый месяц и провели анализ достоверности планирования продаж.