

**Федеральное государственное образовательное
бюджетное учреждение
высшего образования**

**«ФИНАНСОВЫЙ УНИВЕРСИТЕТ
ПРИ ПРАВИТЕЛЬСТВЕ РОССИЙСКОЙ
ФЕДЕРАЦИИ»**

(Финансовый университет)

**Факультет
информационных технологий и анализа больших данных
Кафедра «Бизнес-информатика»**

Домашнее задание № 4

«Игры с природой в условиях риска и неопределенности.»

Студенты группы БИ20-8:

Луканина Полина

Аверкин Никита

Филимонова Арина

Совин Владимир

Горшков Георгий

Киселева Евгения

Руководитель:

Аксенов Дмитрий Андреевич

Москва 2022

Оглавление

Оглавление	2
1. 5	
2. 5	
2.1. Решение задачи о принятии оптимального решения в условиях риска	6
2.2. Решение задачи о принятии оптимального решения в условиях неопределенности	8
3. 10	
3.1. 10	
3.1.1. 10	
3.1.2. 12	
3.1.3. 14	
3.2. Решение задачи о нахождении выигрышной стратегии в условиях неопределенности.	15
3.2.1. Описание входных данных	15
3.2.2. Описание алгоритма решения	16
3.2.3. Описание входных данных	16
4. 19	
4.1.1. ВИ 1	18
4.1.2. ВИ 2	20
4.1.3. ВИ 3	21
4.2. Варианты использования в условиях неопределенности	22
4.2.1. ВИ 1	22
4.2.2. ВИ 2	24
4.2.3. ВИ 3	26

5.	28	
5.1	28	
5.1.1.	Функции считывания информации	27
5.1.2.	Функции обработки информации	32
5.1.3	Функции вывода информации	40
5.2	43	
5.2.1.	Функции считывания информации	42
5.2.2.	Функции обработки информации	45
5.2.3.	Функции вывода информации	53
6	55	
6.1.	Тестирование задачи на нахождение выигрышной стратегии в условиях риска	54
6.1.1.	Проверка №1 матрица 3x3:	54
6.1.2.	Проверка №2 матрица 3x3:	57
6.1.3.	Проверка №3 матрица 3x3:	60
6.1.4.	Проверка №4 матрица 4x4:	63
6.1.5.	Проверка №5 матрица 4x4:	66
6.2.	Тестирование задачи на нахождение выигрышной стратегии в условиях неопределенности	70
6.2.1.	Проверка №1 матрица 3x3	70
6.2.2.	Проверка №2 матрица 3x3:	72
6.2.3.	Проверка №3 матрица 3x3:	74
6.2.4.	Проверка №4 матрица 4x4:	76
6.2.5.	Проверка №5 матрица 4x4:	78
7	83	

7.1. Заключение по задачи на нахождение выигрышной стратегии в условиях риска	81
7.2. Заключение по задачи на нахождение выигрышной стратегии в условиях неопределенности	83

1. Постановка задачи (физическая модель)

Одна Российская IT-компания производит игры. у нее есть 4 основных продукта (игры), которые приносят большую часть прибыли. Величина прибыли определяется состоянием спроса на рынке, который может находиться в одном из 4 рабочих состояний: повышенный спрос, умеренный спрос, пассивная покупательская способность и депрессивный рынок.

Для наглядности зависимости компания предоставила нам таблицу зависимости величины прибыли от вида игры и состояния рынка:

Таблица 1. Зависимость величины прибыли от вида игры

Виды выпускаемых игр	Возможные состояния рынка спроса			
	Повышенный	Стагнирующий	Сокращающийся	Депрессивный
стратегия	50	30	35	45
шутер	40	40	55	40
аркада	35	40	45	55
гонки	10	20	25	35

Необходимо определить оптимальные пропорции вкладываемых средств и сил компании в виды игр, продажа которых обеспечила бы ему максимально возможную выручку вне зависимости от того, какое состояние спроса будет реализовано.

2. Математическая модель

Таблица 2. Исходные данные

A_1	50	30	35	45
A_2	40	40	55	40
A_3	35	40	45	55
A_4	10	20	25	35
p_j	0.15	0.25	0.5	0.1

2.1. Решение задачи о принятии оптимального решения в условиях риска

1) Критерий Байеса:

По критерию Байеса за оптимальные принимается та стратегия (чистая) A_i , при которой максимизируется средний выигрыш a или минимизируется средний риск r .

Считаем значения $\sum(a_{ij}p_j)$:

$$\sum(a_{1,j}p_j) = 50*0.15 + 30*0.25 + 35*0.5 + 45*0.1 = 37$$

$$\sum(a_{2,j}p_j) = 40*0.15 + 40*0.25 + 55*0.5 + 40*0.1 = 47.5$$

$$\sum(a_{3,j}p_j) = 35*0.15 + 40*0.25 + 45*0.5 + 55*0.1 = 43.25$$

$$\sum(a_{4,j}p_j) = 10*0.15 + 20*0.25 + 25*0.5 + 35*0.1 = 22.5$$

Таблица 3. Значения $\sum(a_{ij}p_j)$

A_i	Π_1	Π_2	Π_3	Π_4	$\sum(a_{ij}p_j)$
A_1	7.5	7.5	17.5	4.5	37
A_2	6	10	27.5	4	47.5
A_3	5.25	10	22.5	5.5	43.25
A_4	1.5	5	12.5	3.5	22.5
p_j	0.15	0.25	0.5	0.1	

Выбираем из (37; 47.5; 43.25; 22.5) максимальный элемент $\max=47.5$

Вывод: выбираем стратегию $N=2$.

2) Критерий Лапласа:

Если вероятности состояний природы правдоподобны, для их оценки используют принцип недостаточного основания Лапласа, согласно которого все состояния природы полагаются равновероятными, т.е.:

$$q_1 = q_2 = \dots = q_n = 1/n.$$

$$q_i = 1/4$$

Таблица 4. Критерий Лапласа

A_i	Π_1	Π_2	Π_3	Π_4	$\sum(a_{ij})$
A_1	12.5	7.5	8.75	11.25	40
A_2	10	10	13.75	10	43.75
A_3	8.75	10	11.25	13.75	43.75
A_4	2.5	5	6.25	8.75	22.5
p_j	0.25	0.25	0.25	0.25	

Выбираем из (40; 43.75; 43.75; 22.5) максимальный элемент $\max=43.75$

Вывод: выбираем стратегию $N=2$.

3) Критерий Гермейера:

Преобразуем матрицу в соответствии с методом Гермейера:

$$e_{ij}=a_{ij} \cdot q_j, \text{ если } a_{ij} < 0$$

$$e_{ij}=a_{ij}/q_j, \text{ если } a_{ij} > 0$$

Далее к этой матрице применяется принцип максимина.

Таким образом, новую матрицу необходимо дополнить справа еще одним столбцом, в который нужно внести наименьшие значения элементов каждой строки. Затем из элементов добавленного столбца нужно выбрать наибольший. Строка, в которой он стоит и будет оптимальной стратегией.

A_i	Π_1	Π_2	Π_3	Π_4	$\min(e_i)$
A_1	333.3333333333	120	70	45	70
A_2	266.6666666666	160	11	40	110
A_3	233.3333333333	160	90	55	90
A_4	66.6666666666	80	50	35	50

4	7			0	
p_j	0.15	0.2	0.5	0.1	
		5			

Выбираем из (70; 110; 90; 50) максимальный элемент $\max=110$

Вывод: выбираем стратегию $N=2$.

Таким образом, в результате решения статистической игры по различным критериям чаще других рекомендовалась стратегия A_2 .

2.2. Решение задачи о принятии оптимального решения в условиях неопределенности

1) Критерий «пессимизма»

Таблица 5. Min - min

30
40
35
10

2) Критерий «оптимизма»

Таблица 6. Max – max

50
55
55
35

3) Критерий Вальда

Min - max

Правило Вальда (правило крайнего пессимизма).

Рассматривая i -е решение будем полагать, что на самом деле ситуация складывается самая плохая, т.е. приносящая самый малый доход a_i , но теперь уж выберем решение i_0 с наибольшим a_{i_0} .

Итак, правило Вальда рекомендует принять решение i_0 , такое что $a_{i_0} = \max (a_i) = \max (\min q_{ij})$

Так, в нижеуказанной матрице имеем $a_1 = 30$, $a_2 = 40$, $a_3 = 35$ $a_4 = 10$. Из этих чисел максимальным является число 40. Значит, правило Вальда рекомендует принять 2-е решение:

Таблица 7. Рекомендованное число 40

30
40
35
10

4) Правило Гурвица (взвешивающее пессимистический и оптимистический подходы к ситуации).:

Принимается решение i , на котором достигается максимум,

$$\text{где } 0 \leq \lambda \leq 1.$$

Значение λ выбирается из субъективных соображений. Если λ приближается к 1, то правило Гурвица приближается к правилу Вальда, при приближении λ к 0, правило Гурвица приближается к правилу "розового оптимизма":

Таблица 8. В нашем случае правило Гурвица рекомендует 3-е решение.

Alpha	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
A1	30	32	34	36	38	40	42	44	46	48	50
A2	40	41,5	43	44,5	46	47,5	49	50,5	52	53,5	55

A3	35	37	39	41	43	45	47	49	51	53	55
A4	10	12,5	15	17,5	20	22,5	25	27,5	30	32,5	35

5) Правило Сэвиджа (правило минимального риска).

При применении этого правила анализируется матрица рисков $R = (r_{ij})$. Рассматривая i -е решение будем полагать, что на самом деле складывается ситуация максимального риска $b_i = \max [r_{ij}]$

Но теперь уж выберем решение i_0 с наименьшим b_{i_0} .

Итак, правило Сэвиджа рекомендует принять решение i_0 , такое что $b_{i_0} = \min (b_i) = \min (\max r_{ij})$

В рассматриваемом примере получим:

Таблица 9. выбираем решение i_0 с наименьшим b_{i_0} .

0	10	20	10
10	0	0	15
15	0	10	0
40	20	30	20

Минимальным из этих чисел является число 15. Т.е. правило Сэвиджа рекомендует принять 2-е или 3-е решение.

3. Алгоритмы решения задачи

Алгоритмы решения реализованы с помощью программного кода в Python.

3.1. Решение задачи о нахождении выигрышной стратегии в условиях риска

3.1.1. Описание входных данных.

Вид входных данных зависит от способа, которым будут вводиться данными. Данный алгоритм позволяет самим ввести данные, или ввести с помощью CSV файла.

Для ручного ввода входными данными являются:

- Количество стратегий для компании А, название стратегий для компании А, количество состояний природы, название состояния природы, значения весовой матрицы, значения вектора вероятности.

Способ ввода данных: 1

Введите количество стратегий для компании А: 2

Введите название 1 стратегии компании А: Продукты

Введите название 2 стратегии компании А: Косметика

Введите количество состояний природы П: 2

Введите название 1 состояний природы П: Оживление

Введите название 2 состояний природы П: Стагнация

Введите матрицу весовых коэффициентов поэлементно:

Введите элемент строки 1 столбца 1: 40

Введите элемент строки 1 столбца 2: 55

Введите элемент строки 2 столбца 1: 35

Введите элемент строки 2 столбца 2: 55

Введите вектор вероятности поэлементно:

Введите элемент строки 1 столбца 1: 0.15

Введите элемент строки 1 столбца 2: 0.25

Рисунок 1. Терминал в питоне

- Количество стратегий игрока А, количество состояний природы, значения вектора вероятности.

Способ ввода данных: 2

Введите количество стратегий для компании А: 2

Введите количество состояний природы П: 2

Введите вектор вероятности поэлементно:

Введите элемент строки 1 столбца 1: 0.15

Введите элемент строки 1 столбца 2: 0.25

Рисунок 2. 2 способ ввода данных

Импорт входных данных из csv файла:

- Весовая матрица с названиями столбцов и строк.

Введите путь к файлу: \Users\Полина\Downloads\12.csv

Рисунок 3. Способ ввода данных 3

3.1.2. Методы вывода информации

После того как данные введены, программе необходимо преобразовать данные для дальнейшего использования.

Шаг 1: необходимо рассчитать оценку по критерию Лапласа, для этого нужно рассчитать вероятности: единицу разделить на количество стратегий компании А. Затем с помощью цикла перебираются значения, находится оптимальное.

Шаг 2: необходимо рассчитать оценку Байеса. Для этого необходимо перемножить вероятности, и значения по строке платежной матрицы, а после сложить. Затем с помощью цикла перебрать значения и найти оптимальное.

Шаг 3: необходимо рассчитать оценку по критерию Гермейера, а именно находятся минимальное значение по строке и округляется до трех знаков после запятой.

Шаг 4: берем весовую матрицу, находим минимум каждой строки, а далее - максимум — это получилась цена игры для игрока А.

Шаг 5: находим максимум для каждого столбца весовой матрицы, и затем находим минимальное из них — это цена игры для игрока В.

Шаг 6: Переход к канонической форме задачи линейного

программирования путем введения неотрицательных дополнительных балансовых (базисных) переменных. Запись задачи в симплекс-таблицу.

Между системой ограничений задачи и симплекс-таблицей взаимно-однозначное соответствие. Строчек в таблице столько, сколько равенств в

системе ограничений, а столбцов - столько, сколько свободных переменных.

Базисные переменные заполняют первый столбец, свободные - верхнюю

строку таблицы. Нижняя строка называется индексной, в ней записываются

коэффициенты при переменных в целевой функции.

Шаг 7: Проверка опорного плана на оптимальность. Для этого необходимо анализировать строку целевой функции F. Если найдется хотя

бы один коэффициент индексной строки меньше нуля, то план не оптимальный, и его необходимо улучшить.

Шаг 8: Улучшение опорного плана. Из отрицательных коэффициентов индексной строки выбирается наибольший по абсолютной

величине. Затем элементы столбца свободных членов симплексной таблицы

делит на элементы того же знака ведущего столбца. Далее идет построение

нового опорного плана.

Шаг 9: Выписывание оптимального решения.

3.1.3. Описание входных данных

В конце программа рассчитает весовую матрицу, состояние природы, оптимальную стратегию игрока А по критерию Лапласа, цену игры оптимальной стратегией по критерию Лапласа, оптимальную стратегию игрока А по критерию Байеса, цену игры оптимальной стратегией по критерию Байеса, оптимальную чистую стратегию игрока А по критерию Гермейера, цену игры оптимальной чистой стратегией по критерию Гермейера, цену игры для игрока А при выборе смешанной оптимальной стратегии по критерию Гермейера и таблицу оптимальных смешанных стратегий для игрока А по критерию Гермейера.

Весовая матрица стратегий игрока А:

	1	2
1	765	462
2	820	257

Состояния природы П:

	1	2
0	0.15	0.25

Оптимальная стратегия игрока А по критерию Лапласа: А1-1

Цена игры оптимальной стратегией по критерию Лапласа: 613.5

Оптимальная стратегия игрока А по критерию Байеса: А1-1

Цена игры оптимальной стратегией по критерию Байеса: 230.25

Оптимальная чистая стратегия игрока А по критерию Гермейера: А1-1

Цена игры оптимальной чистой стратегией по критерию Гермейера: 114.75/765.0

Цена игры для игрока А при выборе смешанной оптимальной стратегии по критерию Гермейера: 114.

Таблица оптимальных смешанных стратегий для игрока А по критерию Гермейера:

+-----+		
	1	2
+-----+		
	99%	1%
+-----+		

Рисунок 4. Предоставление программой оптимальной стратегии

3.2. Решение задачи о нахождении выигрышной стратегии в условиях неопределенности.

3.2.1. Описание входных данных

Вид входных данных зависит от способа, которым будут вводиться данными. Данный алгоритм позволяет самим ввести данные, или ввести с помощью CSV файла.

Для ручного ввода входными данными являются:

- Количество стратегий для компании А, название стратегий для компании А, количество состояний природы, название состояния природы, значения весовой матрицы.

```
Выберите способ ввода данных: 1
Введите количество стратегий для компании А: 2
Введите количество состояний природы П: 2
Введите название 1 стратегии компании А: Косметика
Введите название 2 стратегии компании А: Продукты
Введите название 1 состояния природы П: Ожидание
Введите название 2 состояния природы П: Стагнация
Введите матрицу весовых коэффициентов поэлементно:
Введите элемент строки 1 столбца 1: 44
Введите элемент строки 1 столбца 2: 50
Введите элемент строки 2 столбца 1: 55
Введите элемент строки 2 столбца 2: 25
```

Рисунок 5. Терминал питона при 1 способе ввода

- Количество стратегий игрока А, количество состояний природы, минимальное и максимальное значение.

```
Выберите способ ввода данных: 2
Введите количество стратегий для компании А: 2
Введите количество состояний природы П: 2
Введите диапазон значений для матрицы весовых коэффициентов :
Введите минимальное значение: 40
Введите максимальное значение: 90
```

Рисунок 6. Терминал питона при 2 варианте ввода

Импорт входных данных из csv файла:

- Файл с весовая матрица с названиями столбцов и строк.

```
Введите путь к файлу: \Users\Полина\Downloads\12.csv
```

Рисунок 7. Терминал питона при 3 способе ввода

3.2.2. Описание алгоритма решения

После того как данные введены, программе необходимо преобразовать данные для дальнейшего использования.

Шаг 1: необходимо транспонировать матрицу.

Шаг 2: необходимо рассчитать стратегию по критерию оптимизма и пессимизма, а также Вальда.

Шаг 3: необходимо рассчитать линейную свертку склонности к риску по критерию Гурвица, а также разработать гистограмму максимумов.

Шаг 4: также создать некую защиту от неверного ввода данных.

3.2.3. Описание входных данных

В конце программа выдаст весовую матрицу; оптимальную стратегию игрока А по критерию пессимизма, а также цена игры оптимальной стратегии этому критерию; оптимальную стратегию игрока А по критерию оптимизма, а также цена игры оптимальной стратегии этому критерию; оптимальную стратегию игрока А по критерию Вальда, а также цена игры оптимальной стратегии этому критерию; оптимальная стратегия игрока А по критерию Сэвиджа, а также величину минимальной недополученной прибыли по этому

критерию.

Весовая матрица:

	П1	П2
A1	72	56
A2	46	46

Оптимальная стратегия игрока А по критерию пессимизма: A2
 Цена игры оптимальной стратегии по критерию пессимизма: 46

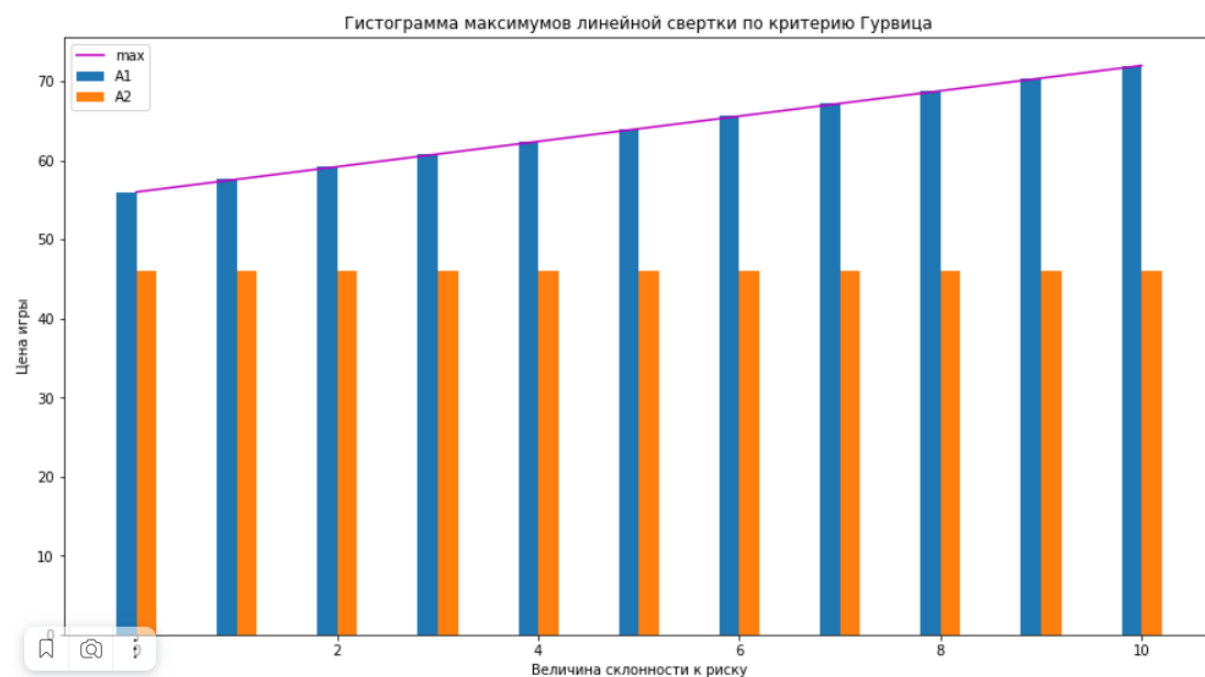
Оптимальная стратегия игрока А по критерию оптимизма: A1
 Цена игры оптимальной стратегии по критерию оптимизма: 72

Оптимальная стратегия игрока А по критерию Вальда: A1
 Цена игры оптимальной стратегии по критерию Вальда: 56

Линейная свертка склонности к риску по критерию Гурвица:

	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
A1	56.0	57.6	59.2	60.8	62.4	64.0	65.6	67.2	68.8	70.4	72.0
A2	46.0	46.0	46.0	46.0	46.0	46.0	46.0	46.0	46.0	46.0	46.0
max	56.0	57.6	59.2	60.8	62.4	64.0	65.6	67.2	68.8	70.4	72.0

Рисунок 8. Вывод оптимальной стратегии и линейная свертка



Оптимальная стратегия игрока А по критерию Сэвиджа: A1
 Величина минимальной недополученной прибыли по критерию Сэвиджа: 0

Рисунок 9. Гистограмма максимумов

4. Варианты использования системы

В нашей системе есть три варианта использования.

4.1.1. ВИ 1

Данный вариант использования включает в себя ручной ввод данных с клавиатуры. Для того, чтобы его активировать в графу «Каким способом вы хотите ввести значения?» надо ввести цифру «1».

```
Способ ввода данных: 1
Введите количество стратегий для компании А: 2
Введите название 1 стратегии компании А: Косметика
Введите название 2 стратегии компании А: Продукты

Введите количество состояний природы П: 2
Введите название 1 состояний природы П: Стагнация
Введите название 2 состояний природы П: Оживление

Введите матрицу весовых коэффициентов поэлементно:
Введите элемент строки 1 столбца 1: 44
Введите элемент строки 1 столбца 2: 55
Введите элемент строки 2 столбца 1: 40
Введите элемент строки 2 столбца 2: 25

Введите вектор вероятности поэлементно:
Введите элемент строки 1 столбца 1: 0.15
Введите элемент строки 1 столбца 2: 0.25

Весовая матрица стратегий игрока А:
      Косметика  Продукты
Косметика      44.0      55.0
Продукты       40.0      25.0

Состояния природы П:
      Стагнация  Оживление
0      0.15      0.25

Оптимальная стратегия игрока А по критерию Лапласа: А1-Косметика
Цена игры оптимальной стратегией по критерию Лапласа: 49.5

Оптимальная стратегия игрока А по критерию Байеса: А1-Косметика
Цена игры оптимальной стратегией по критерию Байеса: 20.35

Оптимальная чистая стратегия игрока А по критерию Гермейера: А1-Косметика
Цена игры оптимальной чистой стратегией по критерию Гермейера: 6.6/44.0
Цена игры для игрока А при выборе смешанной оптимальной стратегии по критерию Гермейера: 6.6/20.35
```

Рисунок 10. Терминал кода при ВИ 1

Таблица оптимальных смешанных стратегий для игрока А по критерию Гермейера:

-----+	-----+
Косметика	Продукты
-----+	-----+
100%	0%
-----+	-----+

Рисунок 11. Оптимальный путь при ВИ 1

После этого необходимо ввести количество стратегий для компании А цифрой. (Например, 2).

Затем нужно ввести названия стратегий, их количество зависит от того, сколько стратегий вы указали ранее. (Например, Косметика).

Далее переходим к заполнению данных для состояния природы. Мы также вводим количество состояний, а также их названия.

После этого вы должны ввести данные для заполнения весовой матрицы. Количество значений зависит от того, сколько стратегий вы указали ранее. (Например, 45).

Затем вводятся данные вектора вероятности поэлементно. (Например: 0.15). Важно вводить не целые значения через точку.

После этого от пользователя требуется лишь нажатие клавиши «Enter» и на экране выведется весовая матрица, состояние природы, оптимальная стратегия игрока А по критерию Лапласа, цену игры оптимальной стратегией по критерию Лапласа, оптимальную стратегию игрока А по критерию Байеса, цену игры оптимальной стратегией по критерию Байеса, оптимальную чистую стратегию игрока А по критерию Гермейера, цену игры оптимальной чистой стратегией по критерию Гермейера, цену игры для игрока А при выборе смешанной оптимальной стратегии по критерию Гермейера и таблицу оптимальных смешанных стратегий для игрока А по критерию Гермейера.

4.1.2. ВИ 2

Данный вариант позволяет сгенерировать весовую матрицу, для его выбора в первой строке необходимо ввести «2».

Способ ввода данных: 2

Введите количество стратегий для компании А: 2

Введите количество состояний природы П: 2

Введите вектор вероятности поэлементно:

Введите элемент строки 1 столбца 1: 0.15

Введите элемент строки 1 столбца 2: 0.25

Весовая матрица стратегий игрока А:

	1	2
1	765	462
2	820	257

Состояния природы П:

	1	2
0	0.15	0.25

Оптимальная стратегия игрока А по критерию Лапласа: А1-1

Цена игры оптимальной стратегией по критерию Лапласа: 613.5

Оптимальная стратегия игрока А по критерию Байеса: А1-1

Цена игры оптимальной стратегией по критерию Байеса: 230.25

Оптимальная чистая стратегия игрока А по критерию Гермейера: А1-1

Цена игры оптимальной чистой стратегией по критерию Гермейера: 114.75/765.0

Цена игры для игрока А при выборе смешанной оптимальной стратегии по критерию Гермейера: 114.854/229.708

Таблица оптимальных смешанных стратегий для игрока А по критерию Гермейера:

+	-----	+	-----	+
	1		2	
+	-----	+	-----	+
	99%		1%	
+	-----	+	-----	+

Рисунок 12. Терминал и ответ при 2 ВИ

Далее Вам необходимо ввести количество стратегий игрока А, а также количество состояний природы.

Затем нужно ввести вектор вероятности поэлементно.

После этого от пользователя требуется лишь нажатие клавиши «Enter» и на экране выведется весовая матрица, состояние природы, оптимальная стратегия игрока А по критерию Лапласа, цену игры оптимальной стратегией по критерию Лапласа, оптимальную стратегию игрока А по критерию Байеса, цену игры оптимальной стратегией по критерию Байеса, оптимальную чистую стратегию игрока А по критерию Гермейера, цену игры оптимальной чистой стратегией по критерию Гермейера, цену игры для игрока А при выборе

смешанной оптимальной стратегии по критерию Гермейера и таблицу оптимальных смешанных стратегий для игрока А по критерию Гермейера .

4.1.3. ВИ 3

Данный вариант использования включает в себя ввод данных с помощью файла csv. Для того, чтобы его активировать в графу «Каким способом вы хотите ввести значения?» надо ввести цифру «3».

После этого появляется окно, в котором вводим путь к csv файлу. Например:

Введите путь к файлу: \Users\Полина\Downloads\12.csv

Рисунок 13. Терминал при 3 ВИ

4.2. Варианты использования в условиях неопределенности

4.2.1. ВИ 1

Данный вариант использования включает в себя ручной ввод данных с клавиатуры. Для того, чтобы его активировать в графу «Каким способом вы хотите ввести значения?» надо ввести цифру «1».

```
Выберите способ ввода данных: 1
Введите количество стратегий для компании А: 2
Введите количество состояний природы П: 2
Введите название 1 стратегии компании А: Косметика
Введите название 2 стратегии компании А: Продукты
Введите название 1 состояния природы П: Стагнация
Введите название 2 состояния природы П: Ожидание
Введите матрицу весовых коэффициентов поэлементно:
Введите элемент строки 1 столбца 1: 44
Введите элемент строки 1 столбца 2: 50
Введите элемент строки 2 столбца 1: 25
Введите элемент строки 2 столбца 2: 5
```

```
Оптимальная стратегия игрока А по критерию пессимизма: А2 - Продукты
Цена игры оптимальной стратегии по критерию пессимизма: 5
```

```
Оптимальная стратегия игрока А по критерию оптимизма: А1 - Косметика
Цена игры оптимальной стратегии по критерию оптимизма: 50
```

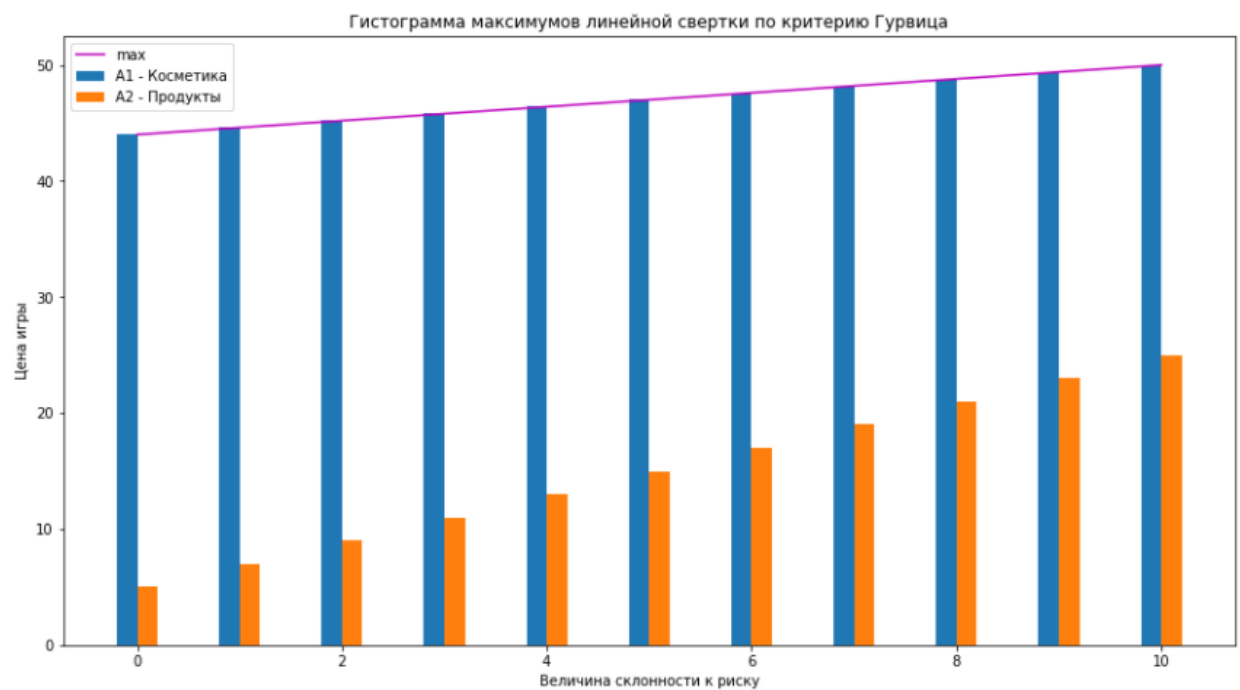
```
Оптимальная стратегия игрока А по критерию Вальда: А1 - Косметика
Цена игры оптимальной стратегии по критерию Вальда: 44
```

```
Линейная свертка склонности к риску по критерию Гурвица:
```

	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	\
A1 - Косметика	44.0	44.6	45.2	45.8	46.4	47.0	47.6	48.2	48.8	49.4	
A2 - Продукты	5.0	7.0	9.0	11.0	13.0	15.0	17.0	19.0	21.0	23.0	
max	44.0	44.6	45.2	45.8	46.4	47.0	47.6	48.2	48.8	49.4	

	1.0
A1 - Косметика	50.0
A2 - Продукты	25.0
max	50.0

Рисунок 14. Терминал питона с решением



Оптимальная стратегия игрока А по критерию Сэвиджа: A1 - Косметика
 Величина минимальной недополученной прибыли по критерию Сэвиджа: 0

Рисунок 15. Гистограмма максимумов

После этого необходимо ввести количество стратегий для компании А цифрой. (Например, 2).

Затем нужно ввести названия стратегий, их количество зависит от того, сколько стратегий вы указали ранее. (Например, Косметика).

Далее переходим к заполнению данных для состояния природы. Мы также вводим количество состояний, а также их названия.

После этого вы должны ввести данные для заполнения весовой матрицы. Количество значений зависит от того, сколько стратегий вы указали ранее. (Например, 45).

После этого от пользователя требуется лишь нажатие клавиши «Enter» и на экране выведется весовая матрица; оптимальная стратегия игрока А по критерию пессимизма, а также цена игры оптимальной стратегии этому критерию; оптимальную стратегию игрока А по критерию оптимизма, а также

цена игры оптимальной стратегии этому критерию; оптимальную стратегию игрока А по критерию Вальда, а также цена игры оптимальной стратегии этому критерию; оптимальная стратегия игрока А по критерию Сэвиджа, а также величину минимальной недополученной прибыли по этому критерию.

4.2.2. ВИ 2

Данный вариант позволяет сгенерировать весовую матрицу, для его выбора в первой строке необходимо ввести «2».

Выберите способ ввода данных: 2
 Введите количество стратегий для компании А: 2
 Введите количество состояний природы П: 2
 Введите диапазон значений для матрицы весовых коэффициентов :
 Введите минимальное значение: 40
 Введите максимальное значение: 90

Весовая матрица:

	П1	П2
A1	72	56
A2	46	46

Оптимальная стратегия игрока А по критерию пессимизма: A2
 Цена игры оптимальной стратегии по критерию пессимизма: 46

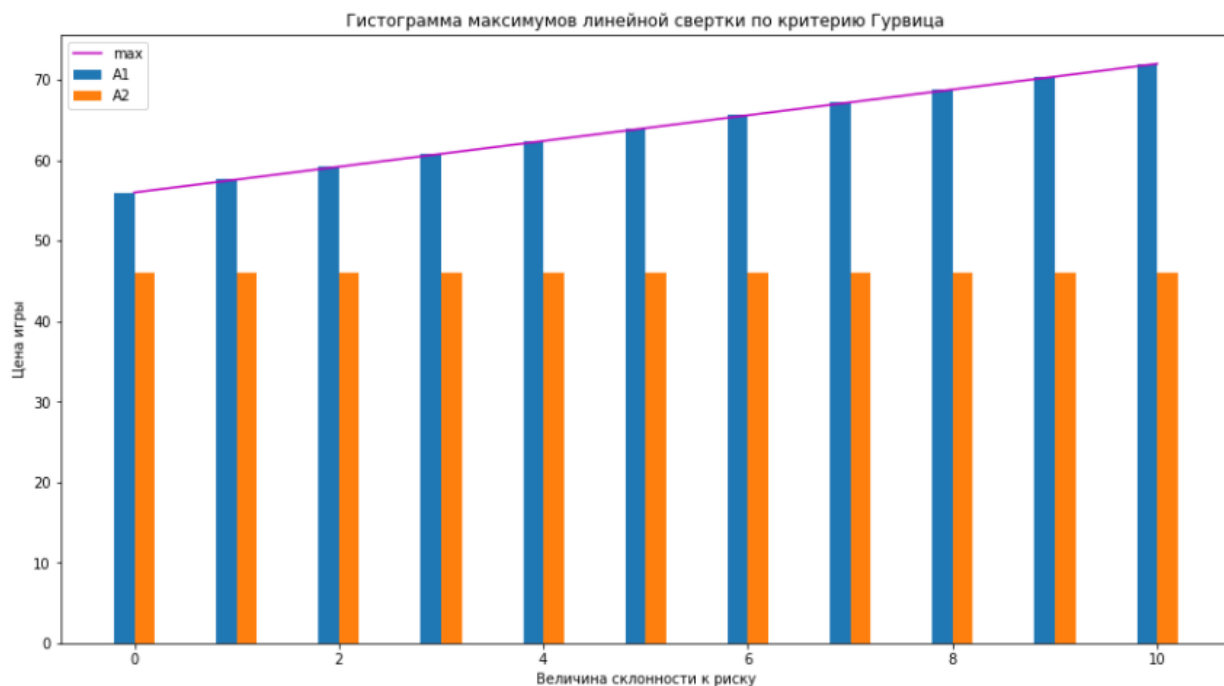
Оптимальная стратегия игрока А по критерию оптимизма: A1
 Цена игры оптимальной стратегии по критерию оптимизма: 72

Оптимальная стратегия игрока А по критерию Вальда: A1
 Цена игры оптимальной стратегии по критерию Вальда: 56

Линейная свертка склонности к риску по критерию Гурвица:

	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
A1	56.0	57.6	59.2	60.8	62.4	64.0	65.6	67.2	68.8	70.4	72.0
A2	46.0	46.0	46.0	46.0	46.0	46.0	46.0	46.0	46.0	46.0	46.0
max	56.0	57.6	59.2	60.8	62.4	64.0	65.6	67.2	68.8	70.4	72.0

Рисунок 16. ВИ с генерацией весовой матрицы



Оптимальная стратегия игрока А по критерию Сэвиджа: A1
 Величина минимальной недополученной прибыли по критерию Сэвиджа: 0

Рисунок 17. Гистограмма при данном способе ввода информации

Далее Вам необходимо ввести количество стратегий игрока А, а также количество состояний природы.

Затем нужно ввести минимальное и максимальное значение.

После этого от пользователя требуется лишь нажатие клавиши «Enter» и на экране выведется весовая матрица; оптимальная стратегия игрока А по критерию пессимизма, а также цена игры оптимальной стратегии этому критерию; оптимальную стратегию игрока А по критерию оптимизма, а также цена игры оптимальной стратегии этому критерию; оптимальную стратегию игрока А по критерию Вальда, а также цена игры оптимальной стратегии этому критерию; оптимальная стратегия игрока А по критерию Сэвиджа, а также величину минимальной недополученной прибыли по этому критерию.

4.2.3. ВИ 3

Данный вариант использования включает в себя ввод данных с помощью файла csv. Для того, чтобы его активировать в графу «Каким способом вы хотите ввести значения?» надо ввести цифру «3».

После этого появляется окно, в котором вводим путь к csv файлу. Например:

Введите путь к файлу: \Users\Полина\Downloads\12.csv

Рисунок 18. Способ ввода информации 3

5. Архитектура решения

Для решения задачи использовались методы (функции), которые можно разделить на 3 принципиальных кода.

5.1 Алгоритм 1

5.1.1. Функции считывания информации

После запуска программы необходимо определиться каким способом будет происходить ввод данных: 1 – ручной ввод; 2 – случайные числа; 3 – файл csv.

1) Если введено «1»:

Входные данные:

- Нет входных данных

Выходные данные:

- name_rows – список названий стратегий компании А (тип данных: list);
- name_columns – список названий стратегий компании В (тип данных: list);
- matrix_A – список значений весовой платёжной матрицы (тип данных: list);
- matrix_P – список значений весовой матрицы вероятностей (тип данных: list);

Переменные, затрагиваемые в ходе работы:

- sum_strategy_A – количество стратегий для компании А (тип данных: int);

- `sum_strategy_B` – количество стратегий для компании В (тип данных: `int`);
- `name_rows` – список названий стратегий компании А (тип данных: `list`);
- `name_columns` – список названий стратегий компании В (тип данных: `list`);
- `matrix_A` – список значений весовой платёжной матрицы (тип данных: `list`);
- `matrix_P` – список значений весовой `name` – название стратегий (тип данных: `str`);
- `value` – элементы весовой матрицы (тип данных: `int`);

```
if way == 1:
    sum_strategy_A = int(input('Введите количество стратегий для компании А: '))
    name_rows, name_columns = [], []
    for num in range(sum_strategy_A):
        name = input('Введите название {} стратегии компании А: '.format(num + 1))
        name_rows.append(name)

    sum_strategy_B = int(input('\nВведите количество состояний природы П: '))
    for num in range(sum_strategy_B):
        name = input('Введите название {} состояний природы П: '.format(num + 1))
        name_columns.append(name)

    print('\nВведите матрицу весовых коэффициентов поэлементно:')
    matrix_A = []
    rows = []
    for num_row, row in enumerate(name_rows, start=1):
        for num_column, column in enumerate(name_columns, start=1):
            value = float(input('Введите элемент строки {} столбца {}: '.format(num_row, num_column)))
            rows.append(value)
        matrix_A.append(rows)
        rows = []

    print('\nВведите вектор вероятности поэлементно:')
    matrix_P = []
    for num_column, column in enumerate(name_columns, start=1):
        value = float(input('Введите элемент строки 1 столбца {}: '.format(num_column)))
        matrix_P.append(value)
```

Рисунок 19. Часть кода, в которой упоминаются эти функции

2) Если введено «2»:

Входные данные:

- Нет входных данных

Выходные данные:

- name_rows – список названий стратегий компании А (тип данных: list);
- name_columns – список названий стратегий компании В (тип данных: list);
- matrix_A – список значений весовой платёжной матрицы (тип данных: list);
- matrix_P – список значений весовой матрицы вероятностей (тип данных: list);

Переменные, затрагиваемые в ходе работы:

- name_rows – список названий стратегий компании А (тип данных: list);
- name_columns – список названий стратегий компании В (тип данных: list);
- matrix_A – список значений весовой платёжной матрицы (тип данных: list);
- matrix_P – список значений весовой матрицы вероятностей (тип данных: list);
- size_rows – количество стратегий игрока А (тип данных: int);

```

elif way == 2:
    name_rows, name_columns = [], []

    # Заполняем матрицу для игрока A
    size_rows = int(input("Введите количество стратегий для компании A: "))
    size_cols = size_rows

    # Последовательно генерируем случайные числа типа int в диапазоне (0;1000) и добавляем их в матрицу
    matrix_A = []
    for i in range(size_rows):
        rows = []
        for j in range(size_cols):
            x = random.randint(0,1000)
            rows.append(x)
        matrix_A.append(rows)

    # список для имен строк и столбцов матрицы:
    name_rows = []
    for i in range(size_rows):
        name_rows.append(i+1)

    # -----
    # Заполняем список состояний природы (вероятности)
    sum_strategy_B = int(input('Введите количество состояний природы П: '))
    name = 0
    for num in range(sum_strategy_B):
        name += 1
        name_columns.append(name)

    print('\nВведите вектор вероятности поэлементно:')
    matrix_P = []
    for num_column, column in enumerate(name_columns, start=1):
        value = float(input('Введите элемент строки 1 столбца {}: '.format(num_column)))
        matrix_P.append(value)

```

Рисунок 20. Часть кода, в которой были затронуты эти переменные

3) Если ведено «3»:

Входные данные:

- Нет входных данных

Выходные данные:

- matrix_A – список значений весовой платёжной матрицы (тип данных: list);
- matrix_P – список значений весовой матрицы вероятностей (тип данных: list);
- name_rows – список названий стратегий компании A (тип данных: list);

- name_columns – список названий стратегий компании В (тип данных: list);

Переменные, затрагиваемые в ходе работы:

- name_rows – список названий стратегий компании А (тип данных: list);
- name_columns – список названий стратегий компании В (тип данных: list);
- file_way – путь к файлу (тип данных: str).
- matrix_A – список значений весовой платёжной матрицы (тип данных: list);
- matrix_P – список значений весовой матрицы вероятностей (тип данных: list);

```
elif way == 3:
    matrix_input = []
    matrix_A = []
    file_way = input('Введите путь к файлу: ')
    with open(file_way, 'r', newline='') as csvfile:
        spamreader = csv.reader(csvfile, delimiter=';')
        for row in spamreader:
            matrix_input.append(row)

    name_columns = matrix_input[0][1:] # Образование списка с названиями колонок матрицы
    name_rows = []
    for row in range(1, len(matrix_input)):
        if row != len(matrix_input)-1:
            name_rows.append(matrix_input[row][0])
            matrix_A.append(matrix_input[row][1:])
        else:
            matrix_P = matrix_input[row][1:]

    matrix_A = [[int(value) for value in row] for row in matrix_A]
    matrix_P = [float(value.replace(',', '.')) for value in matrix_P]
```

Рисунок 21. Часть кода, в которой были использованы данные переменные

5.1.2. Функции обработки информации

После того, как вы введете все необходимые данные, программа их получит и начнет первичную обработку.

При выборе ручного ввода все необходимые данные заносятся в словари, с помощью метода `append ()`.

В случае случайной генерации, нужные данные генерируются с помощью `random.randint()`, при этом программа получает случайное целое число в заданном диапазоне. После этого данные также заносятся в пустые словари.

Если ввод данных осуществляется с помощью файла `csv`, то для начала файл необходимо открыть и прочитать программе, а затем разделить данные и также занести их в пустой словарь. Более того, программа образует список с названиями колонок матрицы, а также список с названиями строк. Не мало важен и перевод коэффициентов из типа данных `str` в тип `int`.

После всех необходимых операций с данными реализуются важные функции, рассчитывающие оптимальную чистую стратегию и оптимальную смешанную стратегию.

В функции `pure_strategy` реализуется оптимальная чистая стратегия. Она представляет собой нахождение минимума в каждой строке, а также нахождения максимума в каждом столбце. Это находится с помощью функции `min ()`, `max ()`.

В функции `nash_equilibrium` реализуется оптимальная смешанная стратегия. С помощью функции `linprog ()` находится оптимальное решение переменных с помощью, которых мы находим процентное соотношение по стратегиям. Более того, здесь также составляются таблицы смешанных стратегий.

Также в коде присутствует вычисление по критерию Лапласа, Байеса и Греймейра.

Критерий Лапласа реализуется с помощью вычисления вероятностей, а также с помощью цикла перебираются значения, находится оптимальное и записывается в переменную `strategy_Laplace` строкового типа.

Критерий Байеса реализуется с помощью цикл, который перебираются значения, находится оптимальное значение и записываются в переменную `strategy_Bayes` строкового типа.

Оценка по критерию Греймейра

Далее следует подробное описание входных, выходных и переменных, затрагивается в ходе работы.

Критерий Лапласа:

Входные данные:

- `value` – элементы весовой матрицы (тип данных: `int`);

Выходные данные:

- `strategy_Laplace` – оптимальная стратегия (тип данных: `str`);
- `average_Laplace` – список средневзвешенных (тип данных: `list`);

Переменные, затрагиваемые в ходе работы:

- `strategy_Laplace` – оптимальная стратегия (тип данных: `str`);
- `average_Laplace` – список средневзвешенных (тип данных: `str`);
- `value` – элементы весовой матрицы (тип данных: `int`);
- `average` – средневзвешенное значение (тип данных: `int`);

- `probability_Laplace` – вероятность состояния природы (тип данных: float).

```

# Оценка по критерию Лапласа

probability_Laplace = 1/len(name_rows)

average_Laplace = []
average = 0
for row in matrix_A_DF:
    for value in row:
        average += value * probability_Laplace
    average_Laplace.append(average)
    average = 0

strategy_Laplace = ''
for num, avrg in enumerate(average_Laplace, start=1):
    if avrg == max(average_Laplace):
        strategy_Laplace += 'A{}-{}; '.format(num, name_rows[num-1])
print('Оптимальная стратегия игрока А по критерию Лапласа: ', strategy_Laplace[:-2])
print('Цена игры оптимальной стратегией по критерию Лапласа: ', round(max(average_Laplace), 4))

```

Рисунок 22. Оценка по критерию Лапласа

Критерий Байеса:

Входные данные:

- `value` – элементы весовой матрицы (тип данных: int);
- `strategy_Bayes` – матрица Байеса (тип данных: str);
- `average` – средневзвешенное значение (тип данных: int);
- `average_Bayes` – список средневзвешенных значений (тип данных: str);
- `matrix_A_DF` - список значений весовой платёжной матрицы (тип данных: list);

Выходные данные:

- `strategy_Bayes` – матрица Байеса (тип данных: str);
- `average_Bayes`– список средневзвешенных значений (тип данных: str);

Переменные, затрагиваемые в ходе работы:

- value – элементы весовой матрицы (тип данных: int);
- strategy_Bayes – оптимальная стратегия
- average – средневзвешенное значение (тип данных: int);
- average_Bayes – список средневзвешенных значений (тип данных: str);
- probability_Bayes – значения вероятностей природы (тип данных: float).

```
# Оценка по критерию Байеса

probability_Bayes = matrix_P_DF

average_Bayes = []
average = 0
for row in matrix_A_DF:
    for num_value, value in enumerate(row):
        average += value * probability_Bayes[num_value]
    average_Bayes.append(average)
    average = 0

strategy_Bayes = ''
for num, avrg in enumerate(average_Bayes, start=1):
    if avrg == max(average_Bayes):
        strategy_Bayes += 'A{}-{}; '.format(num, name_rows[num-1])

print('\nОптимальная стратегия игрока А по критерию Байеса: ', strategy_Bayes[:-2])
print('Цена игры оптимальной стратегией по критерию Байеса: ', round(max(average_Bayes), 4))
```

Рисунок 23. Оценка по критерию Байеса

Критерий Гермейера:

Входные данные:

- matrix_P_DF – список значений весовой name – название стратегий (тип данных: str);
- value – элементы весовой матрицы (тип данных: int);
- matrix_A_DF - список значений весовой платёжной матрицы (тип данных: list);

Выходные данные:

- matrix_Germeier - матрица Гермейера (тип данных: list);

Переменные, затрагиваемые в ходе работы:

- probability_Germeier – значения вероятностей природы (тип данных: float);
- matrix_Germeier - матрица Гермейера (тип данных: list);
- value – элементы весовой матрицы (тип данных: int);
- matrix_P_DF – список значений весовой name – название стратегий (тип данных: str);
- matrix_A_DF - список значений весовой платёжной матрицы (тип данных: list);

```
# Оценка по критерию Гермейера

probability_Germeier = matrix_P_DF

matrix_Germeier = []
rows = []
num_value = 0
for num_row, row in enumerate(matrix_A_DF):
    for value in row:
        rows.append(round(value * probability_Germeier[num_value], 3))
        num_value += 1
    matrix_Germeier.append(rows)
    rows = []
    num_value = 0
```

Функция pure_strategy:

Входные данные:

- name_rows – список названий стратегий компании А (тип данных: list);
- name_columns – список названий стратегий компании В (тип данных: list);

- probability_Germeier – значения вероятностей природы (тип данных: float);
- matrix_Germeier - матрица Гермейера (тип данных: list);

Выходные данные:

- probability_Germeier – значения вероятностей природы (тип данных: float);
- strategy_Germeier – оптимальное стратегия (тип данных: str);
- average_Germeier - список с минимальные значениями по стратегии (тип данных: str);
- max_strategy - значение исходной матрицы (тип данных: int);

Переменные, затрагиваемые в ходе работы:

- name_rows – список названий стратегий компании А (тип данных: list);
- name_columns – список названий стратегий компании В (тип данных: list);
- probability_Germeier – значения вероятностей природы (тип данных: float);
- matrix_Germeier - матрица Гермейера (тип данных: list);
- strategy_Germeier – оптимальное стратегия (тип данных: str);
- average – минимальные значения по стратегии (тип данных: int);
- average_Germeier - список с минимальные значениями по стратегии (тип данных: str);
- max_strategy - значение исходной матрицы (тип данных: int);
- probability_Germeier - значения вероятностей природы (тип данных: float);
- matrix_Germeier - матрица Гермейера (тип данных: list);

```

# Оптимальная чистая стратегия
def pure_strategy(matrix_Germeier, name_columns, name_rows, probability_Germeier):
    average_Germeier = []

    for row in matrix_Germeier:
        average = min(row)
        average_Germeier.append(average)

    strategy_Germeier = ''
    for num, avrg in enumerate(average_Germeier, start=1):
        if avrg == max(average_Germeier):
            strategy_Germeier += 'A{}-{}; '.format(num, name_rows[num-1])

    max_strategy = round(max(average_Germeier) / probability_Germeier[matrix_Germeier[average_Germeier.index(max(average_Germeier))].index(max(average_Germeier))], 4)
    print('\nОптимальная чистая стратегия игрока А по критерию Гермейера: ', strategy_Germeier[:-2])
    print('Цена игры оптимальной чистой стратегией по критерию Гермейера: {} / {}'.format(max(average_Germeier), round(max_strategy, 4)))

pure_strategy(matrix_Germeier, name_columns, name_rows, probability_Germeier)

```

Функция nash_equilibrium:

Входные данные:

- name_rows – список названий стратегий компании А (тип данных: list);
- name_columns – список названий стратегий компании В (тип данных: list);
- probability_Germeier - значения вероятностей природы (тип данных: float);
- matrix_A_DF - список значений весовой платёжной матрицы (тип данных: list);

Выходные данные:

- a – матрица Гермейера в формате numpy.array
- name_rows – список названий стратегий компании А (тип данных: list);
- name_columns – список названий стратегий компании В (тип данных: list);

- `probability_Germeier` - значения вероятностей природы (тип данных: float);
- `matrix_A_DF` - список значений весовой платёжной матрицы (тип данных: list);
- `average_Germier` – список с мин значениями по стратегии матрица Гермейера

Переменные, затрагиваемые в ходе работы:

- `name_rows` – список названий стратегий компании А (тип данных: list);
- `name_columns` – список названий стратегий компании В (тип данных: list);
- `probability_Germeier` - значения вероятностей природы (тип данных: float);
- `matrix_A_DF` - список значений весовой платёжной матрицы (тип данных: list);
- `res` — это массив с вероятностями стратегий игроков (тип данных: `scipy.optimize.optimize.OptimizeResult`);
- `mytable` - – таблицы, которые выводят стратегии и проценты (тип данных: `prettytable.prettytable.PrettyTable`);
- `mix_strategy_A` – таблица смешанной стратегии для игрока А (тип данных: list);
- `a_T` – транспонированная матрица Гермейера (тип данных: int);
- `matrix_Germeier` - матрица Гермейера (тип данных: list);
- `a` – матрица Гермейера в формате `numpy.array`
- `sum_procent` – суммарная вероятность (тип данных: list).


```

# Оптимальная смешанная стратегия
def nash_equilibrium(a, name_columns, name_rows, probability_Germeier, matrix_A_DF):
    c = np.array([1 for x in range(len(a[0]))]) # список с единицами, количество которых равно количеству стратегий игрока A
    b = np.array([1 for x in range(len(a))]) # список с единицами, количество которых равно количеству стратегий игрока B

    # bounds - определяет границы для переменных xi. В нашей программе мы определяем неотрицательные границы (0, None) для каждого xi в связи с условиями задачи.
    bounds = [(0, None)]
    for x in range(len(a[0]) - 1):
        bounds += [(0, None)]

    a_T = -np.transpose(a)
    res = linprog(c, A_ub=a_T, b_ub=-b, bounds=bounds)
    p = [round(res.x[i]/(res.fun), 6) for i in range(len(res.x))]

    procent = []
    pr = 0
    for num_cols in range(len(name_columns)):
        for num_row, row in enumerate(matrix_A_DF):
            pr += row[num_cols] * p[num_row]
        procent.append(round(pr, 4))
        pr = 0

    sum_procent = []
    for num, value in enumerate(procent):
        sum_procent.append(value * probability_Germeier[num])

    print('Цена игры для игрока A при выборе смешанной оптимальной стратегии по критерию Гермейера: {}/{}'.format(round(1/res.fun, 4), round(sum(sum_procent), 4)))

    # Таблица смешанной стратегии для игрока A
    mix_strategy_A = []
    for i in range(len(p)):
        mix_strategy_A.append(str(round(p[i]*100)) + '%')

    a = np.array(matrix_Germeier)
    nash_equilibrium = nash_equilibrium(a, name_columns, name_rows, probability_Germeier, matrix_A_DF)

```

Рисунок 26. Часть кода, в которой используются данные переменные

```

# Таблица смешанной стратегии для игрока A
mix_strategy_A = []
for i in range(len(p)):
    mix_strategy_A.append(str(round(p[i]*100)) + '%')

# Для вывода таблицы PrettyTable
mytable = PrettyTable()
mytable.field_names = name_rows # имена полей таблицы
mytable.add_row(mix_strategy_A) # добавление данных по одной строке за раз
print('Таблица оптимальных смешанных стратегий для игрока A по критерию Гермейера:\n' + str(mytable))

a = np.array(matrix_Germeier)
nash_equilibrium = nash_equilibrium(a, name_columns, name_rows, probability_Germeier, matrix_A_DF)

```

Рисунок 27. Вывод таблицы PrettyTable

5.1.3 Функции вывода информации

Метод вывода информации (он заключен внутри каждой функции)

Что делает: осуществляет вывод необходимой информации

Вывод информации осуществляется с помощью функции print ()

Затрагиваемые переменные:

- `matrix_P` – список значений весовой `name` – название стратегий (тип данных: `str`);
- `strategy_Laplace` – оптимальная стратегия (тип данных: `str`);
- `average_Laplace` – список средневзвешенных (тип данных: `list`);
- `strategy_Bayes` – матрица Байеса (тип данных: `str`);
- `average_Bayes` – список средневзвешенных значений (тип данных: `str`);
- `strategy_Germeier` – оптимальная стратегия (тип данных: `str`);
- `average_Germeier` – список с минимальными значениями по стратегии (тип данных: `str`);
- `max_strategy` – значение исходной матрицы (тип данных: `int`);
- `sum_procent`
- `res` — это массив с вероятностями стратегий игроков (тип данных: `scipy.optimize.optimize.OptimizeResult`);
- `mytable` – таблицы, которые выводят стратегии и проценты (тип данных: `prettytable.prettytable.PrettyTable`);

```
print('Оптимальная стратегия игрока А по критерию Лапласа: ', strategy_Laplace[:-2])
print('Цена игры оптимальной стратегией по критерию Лапласа: ', round(max(average_Laplace), 4))
```

Рисунок 28. Вывод оптимальной стратегии по критерию Лапласа

```
print('\nОптимальная стратегия игрока А по критерию Байеса: ', strategy_Bayes[:-2])
print('Цена игры оптимальной стратегией по критерию Байеса: ', round(max(average_Bayes), 4))
```

Рисунок 29. Вывод оптимальной стратегии по критерию Байеса

```
print('\nОптимальная чистая стратегия игрока А по критерию Гермейера: ', strategy_Germeier[:-2])
print('Цена игры оптимальной чистой стратегией по критерию Гермейера: {}'.format(max(average_Germeier), round(max_strategy, 4)))
```

Рисунок 30. Вывод оптимальной стратегии по критерию Гермейера

```
print('Цена игры для игрока А при выборе смешанной оптимальной стратегии по критерию Гермейера: {}'.format(round(1/res.fun, 4), round(sum(sum_procent), 4)))
```

Рисунок 31. Вывод оптимальной стратегии по критерию Гермейера

```
print('Таблица оптимальных смешанных стратегий для игрока А по критерию Гермейера:\n' + str(mytable))
```

Рисунок 32. Таблица оптимальных смешанных стратегий для игрока Ф по критерию Гермейера

5.2 Алгоритм 2

5.2.1. Функции считывания информации

После запуска программы необходимо определиться каким способом будет происходить ввод данных: 1 – ручной ввод; 2 – случайные числа; 3 – файл csv.

1) Если введено «1»:

Входные данные:

- Нет входных данных

Выходные данные:

- num - список количества стратегий компании А и количество состояния природы (тип данных: list);

Переменные, затрагиваемые в ходе работы:

- sum_strategy_company – количество стратегий для компании А (тип данных: int);
- name_strategy_company – список названий стратегий компании А (тип данных: list);
- name_strategy_nature - список названий природы (тип данных: list);
- sum_strategy_nature – количество состояния природы (тип данных: int);
- num – список количества стратегий компании А и количество состояния природы (тип данных: list);
- name – название состояния природы (тип данных: str);

```

if way == 1:
    sum_strategy = sum_strategy()
    sum_strategy_company, sum_strategy_nature = sum_strategy[0], sum_strategy[1]

    name_strategy_company = []
    for i in range(sum_strategy_company):
        name = input('Введите название {} стратегии компании А: '.format(i+1))
        name_strategy_company.append('A{} - {}'.format(i+1, name))

    name_strategy_nature = []
    for i in range(sum_strategy_nature):
        name = input('Введите название {} состояния природы П: '.format(i+1))
        name_strategy_nature.append('A{} - {}'.format(i+1, name))

    num = input_matrix(sum_strategy_company, sum_strategy_nature)

    PesOpVal(num, name_strategy_company)
    Gurvits(num, name_strategy_company)
    Sevice(num, name_strategy_company)

```

Рисунок 33. Часть кода с данными переменными

2) Если введено «2»:

Входные данные:

- Нет входных данных

Выходные данные:

- num – список количества стратегий компании А и количество состояния природы (тип данных: list);

Переменные, затрагиваемые в ходе работы:

- min_value – минимальное значение (тип данных: int);
- max_value – максимальное значение (тип данных: int);
- sum_strategy_company – количество стратегий для компании А (тип данных: int);
- sum_strategy_nature – количество состояния природы (тип данных: int);
- num – список количества стратегий компании А и количество состояния природы (тип данных: list);

```

elif way == 2:
    sum_strategy = sum_strategy()
    sum_strategy_company, sum_strategy_nature = sum_strategy[0], sum_strategy[1]

    name_strategy_company, name_strategy_nature = [], []
    for i in range(sum_strategy_company):
        name_strategy_company.append('A' + str(i+1))
        name_strategy_nature.append('П' + str(i+1))

    print('Введите диапазон значений для матрицы весовых коэффициентов :')
    min_value = int(input('Введите минимальное значение: '))
    max_value = int(input('Введите максимальное значение: '))

    num = [[randint(min_value, max_value) for i in range(sum_strategy_nature)] for j in range(sum_strategy_company)]
    print('\nВесовая матрица:\n', pd.DataFrame(num, index = name_strategy_company, columns = name_strategy_nature), '\n')

    PesOpVal(num, name_strategy_company)
    Gurvits(num, name_strategy_company)
    Sevice(num, name_strategy_company)

```

Рисунок 34. Часть кода с данными переменными

3) Если ведено «3»:

Входные данные:

- Нет входных данных

Выходные данные:

- num - список количества стратегий компании А и количество состояния природы (тип данных: list);

Переменные, затрагиваемые в ходе работы:

- file_way – путь к файлу (тип данных: str).
- sum_strategy_company – количество стратегий для компании А (тип данных: int);
- name_strategy_company – список названий стратегий компании А (тип данных: list);
- name_strategy_nature - список названий природы (тип данных: list);
- sum_strategy_nature – количество состояния природы (тип данных: int);
- num – список количества стратегий компании А и количество состояния природы (тип данных: list);
- name – название состояния природы (тип данных: str);

```

elif way == 3:
    num = []
    try:
        file_way = input('Введите путь к файлу: ')
        with open(file_way, 'r', newline='') as csvfile:
            spamreader = csv.reader(csvfile, delimiter=';')
            for row in spamreader:
                num.append(row)

        # Для образования списка с названиями колонок матрицы
        name_strategy_nature = num[0]
        name_strategy_nature = name_strategy_nature[1:]
        del num[0]

        # Для образования списка с названиями строк
        name_strategy_company = []
        for num_value, value in enumerate(num):
            name_strategy_company.append(value[0])
            del num[num_value][0]

        # Перевод коэффициентов из str в тип int
        for num_row, row in enumerate(num):
            for num_value, value in enumerate(row):
                num[num_row][num_value] = int(value)

        print('\nВесовая матрица:\n', pd.DataFrame(num, index = name_strategy_company, columns = name_strategy_nature), '\n')

        PesOpVal(num, name_strategy_company)
        Gurvits(num, name_strategy_company)
        Sevice(num, name_strategy_company)

```

Рисунок 35. Часть кода с данными переменными

5.2.2. Функции обработки информации

После того, как вы введете все необходимые данные, программа их получит и начнет первичную обработку.

При выборе ручного ввода все необходимые данные заносятся в словари, с помощью метода `append()`.

В случае случайной генерации, нужные данные генерируются с помощью `random.randint()`, при этом программа получает случайное целое число в заданном диапазоне. После этого данные также заносятся в пустые словари.

Если ввод данных осуществляется с помощью файла csv, то для начала файл необходимо открыть и прочитать программе, а затем разделить данные и также занести их в пустой словарь. Более того, программа образует список с

названиями колонок матрицы, а также список с названиями строк. Не мало важен и перевод коэффициентов из типа данных `str` в тип `int`.

После всех необходимых операций с данными реализуются важные функции, рассчитывающие оптимальную чистую стратегию и оптимальную смешанную стратегию.

В функции `transpose_mtrx` транспонируется матрица с помощью метода `transpose`.

В функции `input_matrix` осуществляется создание матрицы вручную. Это обеспечивается за счёт ввода элементов с помощью `int (input ())`.

В функции `PesOpVal` просчитывается стратегия по критерию пессимизма, оптимизма, Вальда. Здесь необходимые значения подсчитываются с помощью цикла `for i in range`, подсчитываются количество стратегий компании А.

В функции `Gurvits` осуществляется линейная свертка склонности к риску по критерию Гурвица, а также построение гистограммы. Здесь необходимые значения подсчитываются с помощью цикла `for i in range`, подсчитываются количество стратегий компании А. График же формируется с помощью метода `plt`.

В функции `Sevige` подсчитывается оптимальная стратегия по критерию Сэвиджа. Здесь необходимые значения подсчитываются с помощью цикла `for i in range`, подсчитываются количество стратегий компании А.

В функции `sum_strategy` реализуется защита от неверного ввода данных с помощью конструкции `try` и `except`, а также с помощью конструкции `if`.

Функция `transpose_mtrx`:

Входные данные:

- a – матрица (тип данных: int);

Выходные данные:

- trns - транспонированная матрица (тип данных: int);

Переменные, затрагиваемые в ходе работы:

- trns - транспонированная матрица (тип данных: int);
- a – матрица (тип данных: int);

```
def transpose_mtx(a): # Транспонирование матрицы
    a = np.array(a)
    trns = a.transpose()
    return trns
```

Рисунок 36. Транспонирование матрицы

Функция input_matrix:**Входные данные:**

- x – количество стратегий для компании А
- y – количество стратегий состояния природы

Выходные данные:

- num – элемент весовой матрицы (тип данных: int);

Переменные, затрагиваемые в ходе работы:

- num – элемент весовой матрицы (тип данных: int);


```

def input_matrix(x,y): # Функция создания матрицы вручную
    def input_values():
        try:
            num[i][j] = int(input('Введите элемент строки {} столбца {}: '.format(i+1, j+1)))
            return num[i][j]
        except ValueError:
            print('Ошибка ввода!\nВведите элемент заново:\n')
            return input_values()

    try:
        num = [[0 for n in range(y)] for nn in range(x)]
        print('Введите матрицу весовых коэффициентов поэлементно:')
        for i in range(x):
            for j in range(y):
                input_values()
        return num
    except ValueError:
        print('\nОшибка!\nВведите заново количество строк/столбцов:\n')
        return input_matrix()

```

Рисунок 37. Функция создания матрицы вручную

Функция PesOpVal:

Входные данные:

- num – элемент весовой матрицы (тип данных: int);

Выходные данные:

- answer – список с оптимальными значениями по критериям (тип данных: list);
- pes – список значений по стратегии пессимизма (тип данных: list);
- opt - список значений по стратегии оптимизма (тип данных: list);

Переменные, затрагиваемые в ходе работы:

- min_and_max – список минимальных и максимальных значений весовой матрицы (тип данных: list);
- row – строки матрицы (тип данных: list);
- Vald – список минимальных значений по критерию Вальда (тип данных: int)

- transpose_min_and_max – транспонированная матрица (тип данных: int)
- answr – цена игры оптимальной стратегии по критериям (тип данных: list);
- num – элемент весовой матрицы (тип данных: int);
- pes – список значений по стратегии пессимизма (тип данных: list);
- opt - список значений по стратегии оптимизма (тип данных: list);
- strategy_Vald - список значений по стратегии оптимизма (тип данных: list);

```
def PesOpVal(num, c): # Функция просчитывания стратегии по критериям оптимизма/пессимизма/Вальда
    min_and_max = []
    for i in range(len(num)):
        row = []
        row.append(min(num[i]))
        row.append(max(num[i]))
        min_and_max.append(row)

    Vald = []
    for i in range(len(num)):
        Vald.append(min(num[i]))
    for i in range(len(min_and_max)):
        min_and_max[i].append(Vald[i])

    answr = []
    transpose_min_and_max = transpose_mtrx(min_and_max)
    answr.append(min(transpose_min_and_max[0]))
    answr.append(max(transpose_min_and_max[1]))
    answr.append(max(transpose_min_and_max[2]))
```

Рисунок 38. Функция просчитывания стратегии PesOpVal

```
# Пессимизм
pes = []
for i in range(len(num)):
    if min(num[i]) == answr[0]:
        pes.append(c[i])
print('\nОптимальная стратегия игрока А по критерию пессимизма: {}'.format(pes[0]))
print('Цена игры оптимальной стратегии по критерию пессимизма: {}'.format(answr[0]))
```

Рисунок 39 SEQ Рисунок *ARABIC 38. Пессимизм

Функция Gurvits:

Входные данные:

- num – элемент весовой матрицы (тип данных: int);

Выходные данные:

- DF_matrix_Gurvits - список матрицы Гурвица (тип данных: list);

Переменные, затрагиваемые в ходе работы:

- DF_matrix_Gurvits – список матрицы Гурвица (тип данных: list);
- num – элемент весовой матрицы (тип данных: int);
- min_and_max – список минимальных значений весовой матрицы (тип данных: list);

```
# Вальд
strategy_Vald = []
for i in range(len(num)):
    if Vald[i] == answr[2]:
        strategy_Vald.append(c[i])
print('Оптимальная стратегия игрока А по критерию Вальда: {}'.format(strategy_Vald[0]))
print('Цена игры оптимальной стратегии по критерию Вальда: {}'.format(answr[2]))
return answr
```

- matrix_Gurvits – матрица Гурвица (тип данных: int);
- row – строки матрицы (тип данных: list);
- transpose_matrix_Gurvits – транспонированная матрица Гурвица (тип данных: int);
- answr - – цена игры оптимальной стратегии по критериям (тип данных: list);

```

# Оптимальная чистая стратегия
def pure_strategy(matrix_Germeier, name_columns, name_rows, probability_Germeier):
    average_Germeier = []

    for row in matrix_Germeier:
        average = min(row)
        average_Germeier.append(average)

    strategy_Germeier = ''
    for num, avrg in enumerate(average_Germeier, start=1):
        if avrg == max(average_Germeier):
            strategy_Germeier += 'A{}-{}; '.format(num, name_rows[num-1])

    max_strategy = round(max(average_Germeier) / probability_Germeier[matrix_Germeier[average_Germeier.index(max(average_Germeier))].index(max(average_Germeier))], 4)
    print('\nОптимальная чистая стратегия игрока А по критерию Гермейера: ', strategy_Germeier[:-2])
    print('Цена игры оптимальной чистой стратегией по критерию Гермейера: {} / {}'.format(max(average_Germeier), round(max_strategy, 4)))

pure_strategy(matrix_Germeier, name_columns, name_rows, probability_Germeier)

```

Функция Sevice:

Входные данные:

- num – элемент весовой матрицы (тип данных: int);

Выходные данные:

- answr – цена игры оптимальной стратегии по критериям (тип данных: list);
- strategy_Sevice – список оптимальных значений по стратегии Сэвиджа (тип данных: list);

Переменные, затрагиваемые в ходе работы:

- transpose_num – транспонированная весовая матрица (тип данных: int);
- max_value – максимальные значения весовой матрицы (тип данных: int);
- matrix Sevice – матрица Сэвиджа (тип данных: int);
- row – строки матрицы (тип данных: list);
- answr – цена игры оптимальной стратегии по критериям (тип данных: list);

- strategy_Sevige - список оптимальных значений по стратегии Сэвиджа (тип данных: list);

```
def Sevige(num, c): # Оптимальная стратегия по критерию Сэвиджа
    transpose_num = transpose_mtrx(num)

    max_values = [max(transpose_num[i]) for i in range(len(transpose_num))]

    matrix_Sevige = []
    for i in range(len(num)):
        row = []
        for j in range(len(max_values)):
            row.append(max_values[j] - num[i][j])
        matrix_Sevige.append(row)

    answr = min([max(matrix_Sevige[i]) for i in range(len(matrix_Sevige))])

    strategy_Sevige = []
    for i in range(len(matrix_Sevige)):
        if max(matrix_Sevige[i]) == answr:
            strategy_Sevige.append(c[i])

    print('Оптимальная стратегия игрока А по критерию Сэвиджа: {}'.format(strategy_Sevige[0]))
    print('Величина минимальной недополученной прибыли по критерию Сэвиджа: {}'.format(answr))
```

Рисунок 40. Оптимальная стратегия по критерию Сэвиджа

Функция sum_strategy:

Входные данные:

- нет входных данных

Выходные данные:

- x – количество стратегий для компании А
- y – количество стратегий состояния природы

Переменные, затрагиваемые в ходе работы:

- x – количество стратегий для компании А
- y – количество стратегий состояния природы

```
def sum_strategy(): # Защита от неверного ввода данных
    try:
        x = int(input('Введите количество стратегий для компании А: '))
        if x <= 0:
            print('Количество стратегий должно быть больше 0!')
            return sum_strategy()
        y = int(input('Введите количество состояний природы П: '))
        if y <= 0:
            print('Количество стратегий должно быть больше 0!')
            return sum_strategy()
        return [x,y]
    except ValueError:
        print('Количество стратегий должно быть числом!')
        return sum_strategy()
```

Рисунок 41. Защита от неверного ввода данных

5.2.3. Функции вывода информации

Метод вывода информации (он заключен внутри каждой функции)

Что делает: осуществляет вывод необходимой информации

Вывод информации осуществляется с помощью функции print ()

Затрагиваемые переменные:

- answer – цена игры оптимальной стратегии по критериям (тип данных: list);
- pes – список значений по стратегии пессимизма (тип данных: list);
- opt - список значений по стратегии оптимизма (тип данных: list);
- strategy_Vald -
- DF_matrix_Gurvits - список матрицы Гурвица (тип данных: list);
- strategy_Sevidge – список оптимальных значений по стратегии Сэвиджа (тип данных: list);

```
print('\nОптимальная стратегия игрока А по критерию пессимизма: {}'.format(pes[0]))
print('Цена игры оптимальной стратегии по критерию пессимизма: {}'.format(answr[0]))
```

Рисунок 42. Оптимальная стратегия по критерию пессимизма

```
print('Оптимальная стратегия игрока А по критерию оптимизма: {}'.format(opt[0]))
print('Цена игры оптимальной стратегии по критерию оптимизма: {}'.format(answr[1]))
```

Рисунок 43. Оптимальная стратегия по критерию оптимизма

```
print('Оптимальная стратегия игрока А по критерию Вальда: {}'.format(strategy_Vald[0]))
print('Цена игры оптимальной стратегии по критерию Вальда: {}'.format(answr[2]))
```

Рисунок 44.. Оптимальная стратегия по критерию Вальда

```
print('Линейная свертка склонности к риску по критерию Гурвица:\n', pd.DataFrame(Df_matrix_Gurvits, index = Df_index, columns=[(i/10) for i in range(11)]))
```

Рисунок 45. Оптимальная стратегия по критерию Гурвица

```
print('Оптимальная стратегия игрока А по критерию Сэвиджа: {}'.format(strategy_Sevige[0]))
print('Величина минимальной недополученной прибыли по критерию Сэвиджа: {}'.format(answr))
```

Рисунок 46. Оптимальная стратегия по критерию Сэвиджа

6 Тестирование

Проведём тестирование нашей программы и сравним полученные показатели, чтобы сделать вывод о предпочтительном варианте использования нашей программы или excel-заготовки под условия заказчика.

6.1. Тестирование задачи на нахождение выигрышной стратегии в условиях риска

6.1.1. Проверка №1 матрица 3x3:

6.1.1.1. Тест №1 кодом Python:

Выбираем размер матрицы, задаём названия столбцов и строк, вводим вероятности и данные матрицы и получаем оптимальную стратегию по критерию Байеса, Лапласа и Гермейера.

```

Способ ввода данных: 1
Введите количество стратегий для компании A: 3
Введите название 1 стратегии компании A: A1
Введите название 2 стратегии компании A: A2
Введите название 3 стратегии компании A: A3

Введите количество состояний природы П: 3
Введите название 1 состояний природы П: П1
Введите название 2 состояний природы П: П2
Введите название 3 состояний природы П: П3

Введите матрицу весовых коэффициентов поэлементно:
Введите элемент строки 1 столбца 1: 754
Введите элемент строки 1 столбца 2: 465
Введите элемент строки 1 столбца 3: 630
Введите элемент строки 2 столбца 1: 43
Введите элемент строки 2 столбца 2: 645
Введите элемент строки 2 столбца 3: 252
Введите элемент строки 3 столбца 1: 98
Введите элемент строки 3 столбца 2: 74
Введите элемент строки 3 столбца 3: 123

Введите вектор вероятности поэлементно:
Введите элемент строки 1 столбца 1: 0.15
Введите элемент строки 1 столбца 2: 0.25
Введите элемент строки 1 столбца 3: 0.6

Весовая матрица стратегий игрока A:
      П1      П2      П3
A1  754.0  465.0  630.0
A2   43.0  645.0  252.0
A3   98.0   74.0  123.0

Состояния природы П:
      П1      П2      П3
0  0.15  0.25  0.6

Оптимальная стратегия игрока A по критерию Лапласа: A1-A1
Цена игры оптимальной стратегией по критерию Лапласа: 616.3333

Оптимальная стратегия игрока A по критерию Байеса: A1-A1
Цена игры оптимальной стратегией по критерию Байеса: 607.35

Оптимальная чистая стратегия игрока A по критерию Гермейера: A1-A1
Цена игры оптимальной чистой стратегией по критерию Гермейера: 113.1/754.0
Цена игры для игрока A при выборе смешанной оптимальной стратегии по критерию Гермейера: 113.1/607.35
Таблица оптимальных смешанных стратегий для игрока A по критерию Гермейера:
+-----+-----+-----+
|  A1  |  A2  |  A3  |
+-----+-----+-----+
| 100% |  0%  |  0%  |
+-----+-----+-----+

```

Рисунок 47. Решение в питон

6.1.1.2. Тест №1 Excel:

Вводим необходимые данные в матрицу, проверяем корректность введенных данных и получаем искомые данные по критерию Байеса, Лапласа и Гермейера.

Платежная матрица					
	A	П1	П2	П3	П4
A1	754	465	630		
A2	43	645	252		
A3	98	74	123		
A4					
Вероятности	0,15	0,25	0,6		

Рисунок 48. Платежная матрица

Оценка по критерию Байеса						
	A	П1	П2	П3	П4	средневзвешенное
	A1	754	465	630	0	607,35
	A2	43	645	252	0	318,9
	A3	98	74	123	0	107
	A4	0	0	0	0	0
	Вероятности	0,15	0,25	0,6	0	

Рисунок 49. Оценка по критерию Байеса

Оценка по критерию Лапласа						
	A	П1	П2	П3	П4	средневзвешенное
	A1	754	465	630	0	616,333
	A2	43	645	252	0	313,333
	A3	98	74	123	0	98,3333
	A4	0	0	0	0	0
Вероятности		0,33333	0,33333	0,33333		
расчет вероятности:						
pi=1/n	0,33333					

Рисунок 50. Оценка по критерию Лапласа

Оценка по критерию Гермейера													
Матрица Гермейера													
G	П1	П2	П3	П4	min-max		Ограничения:			Переменные		Стратегия	
A1	113,1	116,25	378		113,1	754				x1	0,00884	P1	100%
A2	6,45	161,25	151,2		6,45		1	1,02785	3,34218	x2	0	P2	0%
A3	14,7	18,5	73,8		14,7		>=	>=	>=	x3	0	P3	0%
A4							1	1	1				
Целевая функция		Цена игры								754	465	630	
L=	0,00884	v=1/L(min =				113,1	607,35			113,1	116,25	378	

Рисунок 51. Матрица Гермейера

6.1.2. Проверка №2 матрица 3x3:

6.1.2.1. Тест №2 кодом Python:

Выбираем размер матрицы, задаём названия столбцов и строк, вводим вероятности и данные матрицы и получаем оптимальную стратегию по критерию Байеса, Лапласа и Гермейера.

```

Способ ввода данных: 1
Введите количество стратегий для компании А: 3
Введите название 1 стратегии компании А: А1
Введите название 2 стратегии компании А: А2
Введите название 3 стратегии компании А: А3

Введите количество состояний природы П: 3
Введите название 1 состояний природы П: П1
Введите название 2 состояний природы П: П2
Введите название 3 состояний природы П: П3

Введите матрицу весовых коэффициентов поэлементно:
Введите элемент строки 1 столбца 1: 63
Введите элемент строки 1 столбца 2: 45
Введите элемент строки 1 столбца 3: 213
Введите элемент строки 2 столбца 1: 423
Введите элемент строки 2 столбца 2: 74
Введите элемент строки 2 столбца 3: 386
Введите элемент строки 3 столбца 1: 234
Введите элемент строки 3 столбца 2: 864
Введите элемент строки 3 столбца 3: 943

Введите вектор вероятности поэлементно:
Введите элемент строки 1 столбца 1: 0.15
Введите элемент строки 1 столбца 2: 0.25
Введите элемент строки 1 столбца 3: 0.6

Весовая матрица стратегий игрока А:
      П1      П2      П3
А1  63.0  45.0  213.0
А2  423.0  74.0  386.0
А3  234.0  864.0  943.0

Состояния природы П:
      П1      П2      П3
0  0.15  0.25  0.6

Оптимальная стратегия игрока А по критерию Лапласа:  А3-А3
Цена игры оптимальной стратегией по критерию Лапласа:  680.3333

Оптимальная стратегия игрока А по критерию Байеса:  А3-А3
Цена игры оптимальной стратегией по критерию Байеса:  816.9

Оптимальная чистая стратегия игрока А по критерию Гермейера:  А3-А3
Цена игры оптимальной чистой стратегией по критерию Гермейера:  35.1/234.0
Цена игры для игрока А при выборе смешанной оптимальной стратегии по критерию Гермейера:  57.8076/413.7297
Таблица оптимальных смешанных стратегий для игрока А по критерию Гермейера:
+----+-----+-----+
| А1 |  А2 |  А3 |
+----+-----+-----+
| 0% | 80% | 20% |
+----+-----+-----+

```

Рисунок 52. Решение в питоне 2

6.1.2.2. Тест №2 Excel:

Вводим необходимые данные в матрицу, проверяем корректность введенных данных и получаем искомые данные по критерию Байеса, Лапласа и Гермейера.

Платежная матрица					
A	П1	П2	П3	П4	
A1	63	45	213		
A2	423	74	386		
A3	234	864	943		
A4					
Вероятности	0,15	0,25	0,6		

Рисунок 53. Платежная матрица 2

Оценка по критерию Байеса						
	A	П1	П2	П3	П4	средневзвешенное
	A1	63	45	213		148,5
	A2	423	74	386		313,55
	A3	234	864	943		816,9
	A4					
	Вероятности	0,15	0,25	0,6		

Рисунок 54. Оценка по критерию Байеса 2

Оценка по критерию Лапласа						
	A	П1	П2	П3	П4	средневзвешенное
	A1	63	45	213		107
	A2	423	74	386		294,333
	A3	234	864	943		680,333
	A4					
	Вероятности	0,33333	0,33333	0,33333		
расчет вероятности:						
$p_i = 1/n$	0,33333					

Рисунок 55. Оценка по критерию Лапласа

Оценка по критерию Гермейера												
Матрица Гермейера					Ограничения:					Переменные		
G	П1	П2	П3	П4	min-max							Стратегия
A1	9,45	11,25	127,8		9,45					x1	0	P1 0%
A2	63,45	18,5	231,6		18,5		1	1	5,15701	x2	0,01386	P2 80%
A3	35,1	216	565,8		35,1	234	>=	>=	>=	x3	0,00344	P3 20%
							1	1	1			
Целевая функция					Цена игры					385,384	231,23	496,857
L=	0,0173				v=1/L{mir =	57,8076				57,8076	57,8076	298,114

Рисунок 56. Матрица Гермейера 2

6.1.3. Проверка №3 матрица 3x3:

6.1.3.1. Тест №3 кодом Python:

Выбираем размер матрицы, задаём названия столбцов и строк, вводим вероятности и данные матрицы и получаем оптимальную стратегию по критерию Байеса, Лапласа и Гермейера.

```

Способ ввода данных: 1
Введите количество стратегий для компании А: 3
Введите название 1 стратегии компании А: А1
Введите название 2 стратегии компании А: А2
Введите название 3 стратегии компании А: А3

Введите количество состояний природы П: 3
Введите название 1 состояний природы П: П1
Введите название 2 состояний природы П: П2
Введите название 3 состояний природы П: П3

Введите матрицу весовых коэффициентов поэлементно:
Введите элемент строки 1 столбца 1: 10
Введите элемент строки 1 столбца 2: 75
Введите элемент строки 1 столбца 3: 3
Введите элемент строки 2 столбца 1: 42
Введите элемент строки 2 столбца 2: 4
Введите элемент строки 2 столбца 3: 721
Введите элемент строки 3 столбца 1: 789
Введите элемент строки 3 столбца 2: 5
Введите элемент строки 3 столбца 3: 497

Введите вектор вероятности поэлементно:
Введите элемент строки 1 столбца 1: 0.15
Введите элемент строки 1 столбца 2: 0.25
Введите элемент строки 1 столбца 3: 0.6

Весовая матрица стратегий игрока А:
      П1      П2      П3
А1  10.0  75.0   3.0
А2  42.0   4.0 721.0
А3 789.0   5.0 497.0

Состояния природы П:
      П1      П2      П3
0  0.15  0.25  0.6

Оптимальная стратегия игрока А по критерию Лапласа:  А3-А3
Цена игры оптимальной стратегией по критерию Лапласа:  430.3333

Оптимальная стратегия игрока А по критерию Байеса:  А2-А2
Цена игры оптимальной стратегией по критерию Байеса:  439.9

Оптимальная чистая стратегия игрока А по критерию Гермейера:  А1-А1
Цена игры оптимальной чистой стратегией по критерию Гермейера:  1.5/10.0
Цена игры для игрока А при выборе смешанной оптимальной стратегии по критерию Гермейера:  16.5031/72.8
627
Таблица оптимальных смешанных стратегий для игрока А по критерию Гермейера:
+-----+-----+-----+
|  А1  |  А2  |  А3  |
+-----+-----+-----+
| 87% |  0% | 13% |
+-----+-----+-----+

```

Рисунок 57. Решение в питоне 3

6.1.3.2. Тест №3 Excel:

Вводим необходимые данные в матрицу, проверяем корректность введенных данных и получаем искомые данные по критерию Байеса, Лапласа и Гермейера.

Платежная матрица					
	A	П1	П2	П3	П4
A1		10	75	3	
A2		42	4	721	
A3		789	5	497	
A4					
Вероятности		0,15	0,25	0,6	

Рисунок 58. Платежная матрица 3

Оценка по критерию Байеса						
	A	П1	П2	П3	П4	средневзвешенное
A1		10	75	3		22,05
A2		42	4	721		439,9
A3		789	5	497		417,8
A4						
Вероятности		0,15	0,25	0,6		

Рисунок 59. Оценка по критерию Байеса 3

Оценка по критерию Лапласа						
	A	П1	П2	П3	П4	средневзвешенное
A1		10	75	3		29,3333
A2		42	4	721		255,667
A3		789	5	497		430,333
Вероятности		0,33333	0,33333	0,33333		
расчет вероятности:						
pi=1/n	0,33333					

Рисунок 60. Оценка по критерию Лапласа 3

Оценка по критерию Гермейера												
Матрица Гермейера					Ограничения:					Переменные		
G	П1	П2	П3		min-max					x1		Стратегия
A1	1,5	18,75	1,8		1,5					0,05281		P1 87%
A2	6,3	1	432,6		1		1	2,4151		0		P2 0%
A3	118,35	1,25	298,2		1,25	10	>=	>=	>=	0,00778		P3 13%
							1	1	1			
Целевая функция		Цена игры								110,02	66,0123	66,4276
L=	0,06059	v=1/L{mir =		16,5031		72,8627				16,5031	16,5031	39,8566

Рисунок 61. Матрица Гермейера 3

6.1.4. Проверка №4 матрица 4x4:

6.1.4.1. Тест №4 кодом Python:

Выбираем размер матрицы, задаём названия столбцов и строк, вводим вероятности и данные матрицы и получаем оптимальную стратегию по критерию Байеса, Лапласа и Гермейера.


```

Введите количество стратегий для компании A: 4
Введите название 1 стратегии компании A: A1
Введите название 2 стратегии компании A: A2
Введите название 3 стратегии компании A: A3
Введите название 4 стратегии компании A: A4

Введите количество состояний природы P: 4
Введите название 1 состояний природы P: P1
Введите название 2 состояний природы P: P2
Введите название 3 состояний природы P: P3
Введите название 4 состояний природы P: P4

Введите матрицу весовых коэффициентов поэлементно:
Введите элемент строки 1 столбца 1: 766
Введите элемент строки 1 столбца 2: 75
Введите элемент строки 1 столбца 3: 630
Введите элемент строки 1 столбца 4: 978
Введите элемент строки 2 столбца 1: 905
Введите элемент строки 2 столбца 2: 850
Введите элемент строки 2 столбца 3: 281
Введите элемент строки 2 столбца 4: 358
Введите элемент строки 3 столбца 1: 723
Введите элемент строки 3 столбца 2: 7
Введите элемент строки 3 столбца 3: 609
Введите элемент строки 3 столбца 4: 970
Введите элемент строки 4 столбца 1: 341
Введите элемент строки 4 столбца 2: 423
Введите элемент строки 4 столбца 3: 535
Введите элемент строки 4 столбца 4: 904

Введите вектор вероятности поэлементно:
Введите элемент строки 1 столбца 1: 0.15
Введите элемент строки 1 столбца 2: 0.25
Введите элемент строки 1 столбца 3: 0.5
Введите элемент строки 1 столбца 4: 0.1

Весовая матрица стратегий игрока A:
      P1      P2      P3      P4
A1  766.0   75.0  630.0  978.0
A2  905.0  850.0  281.0  358.0
A3  723.0    7.0  609.0  970.0
A4  341.0  423.0  535.0  904.0

Состояния природы P:
      P1      P2      P3      P4
0  0.15  0.25  0.5   0.1

Оптимальная стратегия игрока A по критерию Лапласа: A1-A1
Цена игры оптимальной стратегией по критерию Лапласа: 612.25

Оптимальная стратегия игрока A по критерию Байеса: A1-A1
Цена игры оптимальной стратегией по критерию Байеса: 546.45

Оптимальная чистая стратегия игрока A по критерию Гермейера: A4-A4
Цена игры оптимальной чистой стратегией по критерию Гермейера: 51.15/341.0
Цена игры для игрока A при выборе смешанной оптимальной стратегии по критерию Гермейера: 86.2439/528.1029
Таблица оптимальных смешанных стратегий для игрока A по критерию Гермейера:
+-----+-----+-----+-----+
|  A1  |  A2  |  A3  |  A4  |
+-----+-----+-----+-----+
| 38%  | 13%  | 0%   | 49%  |
+-----+-----+-----+-----+

```

Рисунок 62. Решение в питоне 4

6.1.4.2. Тест №4 Excel:

Вводим необходимые данные в матрицу, проверяем корректность введённых данных и получаем искомые данные по критерию Байеса, Лапласа и Гермейера.

Платежная матрица					
	A	П1	П2	П3	П4
A1		766	75	630	978
A2		905	850	281	358
A3		723	7	609	970
A4		341	423	535	904
Вероятности		0,15	0,25	0,5	0,1

Рисунок 63. Платежная матрица 4

Оценка по критерию Байеса						
	A	П1	П2	П3	П4	средневзвешенное
A1		766	75	630	978	546,45
A2		905	850	281	358	524,55
A3		723	7	609	970	511,7
A4		341	423	535	904	514,8
Вероятности		0,15	0,25	0,5	0,1	

Рисунок 64. Оценка по критерию Байеса 4

Оценка по критерию Лапласа						
	A	П1	П2	П3	П4	средневзвешенное
	A1	766	75	630	978	612,25
	A2	905	850	281	358	598,5
	A3	723	7	609	970	577,25
	A4	341	423	535	904	550,75
Вероятности		0,25	0,25	0,25	0,25	
расчет вероятности:						
pi=1/n		0,25				

Рисунок 65. Оценка по критерию Лапласа 4

Оценка по критерию Гермейера													
Матрица Гермейера						Ограничения:				Переменные		Стратегия	
G	П1	П2	П3	П4	min-max					x1		P1	
A1	114,9	18,75	315	97,8	18,75	0,99999999	1	3,12336537	0,99999999	0,00441724		38%	
A2	135,75	212,5	140,5	35,8	35,8	>=	>=	>=	>=	0,00148127		P2	13%
A3	108,45	1,75	304,5	97	1,75	1	1	1	1	0		P3	0%
A4	51,15	105,75	267,5	90,4	51,15					0,00569651		P4	49%
Целевая функция						Цена игры				574,9594	344,975625	538,7425	862,4391
L= 0,01159502						v=1/L(min = 86,243906				86,24391	86,2439063	269,3712	86,24391

Рисунок 66. Матрица Гермейера 4

6.1.5. Проверка №5 матрица 4x4:

6.1.5.1. Тест №5 кодом Python:

Выбираем размер матрицы, задаём названия столбцов и строк, вводим вероятности и данные матрицы и получаем оптимальную стратегию по критерию Байеса, Лапласа и Гермейера.

```

Введите количество стратегий для компании A: 4
Введите название 1 стратегии компании A: A1
Введите название 2 стратегии компании A: A2
Введите название 3 стратегии компании A: A3
Введите название 4 стратегии компании A: A4

Введите количество состояний природы P: 4
Введите название 1 состояний природы P: P1
Введите название 2 состояний природы P: P2
Введите название 3 состояний природы P: P3
Введите название 4 состояний природы P: P4

Введите матрицу весовых коэффициентов поэлементно:
Введите элемент строки 1 столбца 1: 99
Введите элемент строки 1 столбца 2: 75
Введите элемент строки 1 столбца 3: 383
Введите элемент строки 1 столбца 4: 12
Введите элемент строки 2 столбца 1: 75
Введите элемент строки 2 столбца 2: 77
Введите элемент строки 2 столбца 3: 466
Введите элемент строки 2 столбца 4: 784
Введите элемент строки 3 столбца 1: 723
Введите элемент строки 3 столбца 2: 7
Введите элемент строки 3 столбца 3: 66
Введите элемент строки 3 столбца 4: 970
Введите элемент строки 4 столбца 1: 231
Введите элемент строки 4 столбца 2: 83
Введите элемент строки 4 столбца 3: 535
Введите элемент строки 4 столбца 4: 33

Введите вектор вероятности поэлементно:
Введите элемент строки 1 столбца 1: 0.15
Введите элемент строки 1 столбца 2: 0.25
Введите элемент строки 1 столбца 3: 0.5
Введите элемент строки 1 столбца 4: 0.1

Весовая матрица стратегий игрока A:
      P1    P2    P3    P4
A1  99.0  75.0  383.0  12.0
A2  75.0  77.0  466.0  784.0
A3  723.0  7.0   66.0  970.0
A4  231.0  83.0  535.0  33.0

Состояния природы P:
      P1    P2    P3    P4
0  0.15  0.25  0.5   0.1

Оптимальная стратегия игрока A по критерию Лапласа:  A3-A3
Цена игры оптимальной стратегией по критерию Лапласа:  441.5

Оптимальная стратегия игрока A по критерию Байеса:  A2-A2
Цена игры оптимальной стратегией по критерию Байеса:  341.9

Оптимальная чистая стратегия игрока A по критерию Гермейера:  A2-A2
Цена игры оптимальной чистой стратегией по критерию Гермейера:  11.25/75.0
Цена игры для игрока A при выборе смешанной оптимальной стратегии по критерию Гермейера:  20.4083/329.7766
Таблица оптимальных смешанных стратегий для игрока A по критерию Гермейера:
+-----+-----+
| A1 |  A2 |  A3 |  A4 |
+-----+-----+
| 0% | 23% | 0% | 77% |
+-----+-----+

```

Рисунок 67. Результат тестирования в питоне 5

6.1.5.2. Тест №5 Excel:

Вводим необходимые данные в матрицу, проверяем корректность введенных данных и получаем искомые данные по критерию Байеса, Лапласа и Гермейера.

Платежная матрица					
A	П1	П2	П3	П4	
A1	99	75	383	12	
A2	75	77	466	784	
A3	723	7	66	970	
A4	231	83	535	33	
Вероятности	0,15	0,25	0,5	0,1	

Рисунок 68. Платежная матрица 5

Оценка по критерию Байеса						
	A	П1	П2	П3	П4	средневзвешенное
	A1	99	75	383	12	226,3
	A2	75	77	466	784	341,9
	A3	723	7	66	970	240,2
	A4	231	83	535	33	326,2
	Вероятности	0,15	0,25	0,5	0,1	

Рисунок 69. Оценка по критерию Байеса 5

Оценка по критерию Лапласа						
	A	П1	П2	П3	П4	средневзвешенное
	A1	99	75	383	12	142,25
	A2	75	77	466	784	350,5
	A3	723	7	66	970	441,5
	A4	231	83	535	33	220,5
Вероятности		0,25	0,25	0,25	0,25	
расчет вероятности:						
pi=1/n		0,25				

Рисунок 70. Оценка по критерию Лапласа 5

Оценка по критерию Гермейера													
Матрица Гермейера							Ограничения:				Переменные		Стратегия
G	П1	П2	П3	П4	min-max						x1		P1
A1	14,85	18,75	191,5	1,2	1,2						0		0%
A2	11,25	19,25	233	78,4	11,25	75	1,43664	1	12,7223	1	0,01116		P2
A3	108,45	1,75	33	97	1,75		>=	>=	>=	>=	0		P3
A4	34,65	20,75	267,5	3,3	3,3		1	1	1	1	0,03784		P4
Целевая функция							Цена игры				195,462	81,6332	519,281
L= 0,049							v=1/L(min = 20,4083				29,3193	20,4083	259,641
													204,083
													20,4083

Рисунок 71. Матрица Гермейера 5

Для подсчёта времени на выполнение алгоритмов будет использовано время, затраченное на ввод необходимых данных в предназначенные поля и расчёт данных самой программой с дальнейшим выводом их пользователю на экран.

Таблица 10. Время, затраченное на выполнение программы

Входные данные	Python	Excel
Тестирование №1	~25 секунд	~50 секунд
Тестирование №2	~25 секунд	~51 секунд
Тестирование №3	~23 секунд	~49 секунд
Тестирование №4	~22 секунд	~50 секунд
Тестирование №5	~22 секунд	~48 секунд

6.2. Тестирование задачи на нахождение выигрышной стратегии в условиях неопределенности

6.2.1. Проверка №1 матрица 3x3

6.2.1.1. Тест №1 кодом Python:

Выбираем размер матрицы, для неё вводим диапазон допустимых значений и получаем оптимальную стратегию по критерию Вальда, гистограмму максимумов линейной свёртки по критерию Гурвица и оптимальную стратегию по критерию Сэвиджа.

Выберите способ ввода данных: 2
 Введите количество стратегий для компании A: 3
 Введите количество состояний природы P: 3
 Введите диапазон значений для матрицы весовых коэффициентов :
 Введите минимальное значение: 1
 Введите максимальное значение: 20

Весовая матрица:

	п1	п2	п3
A1	9	16	20
A2	6	16	5
A3	5	2	6

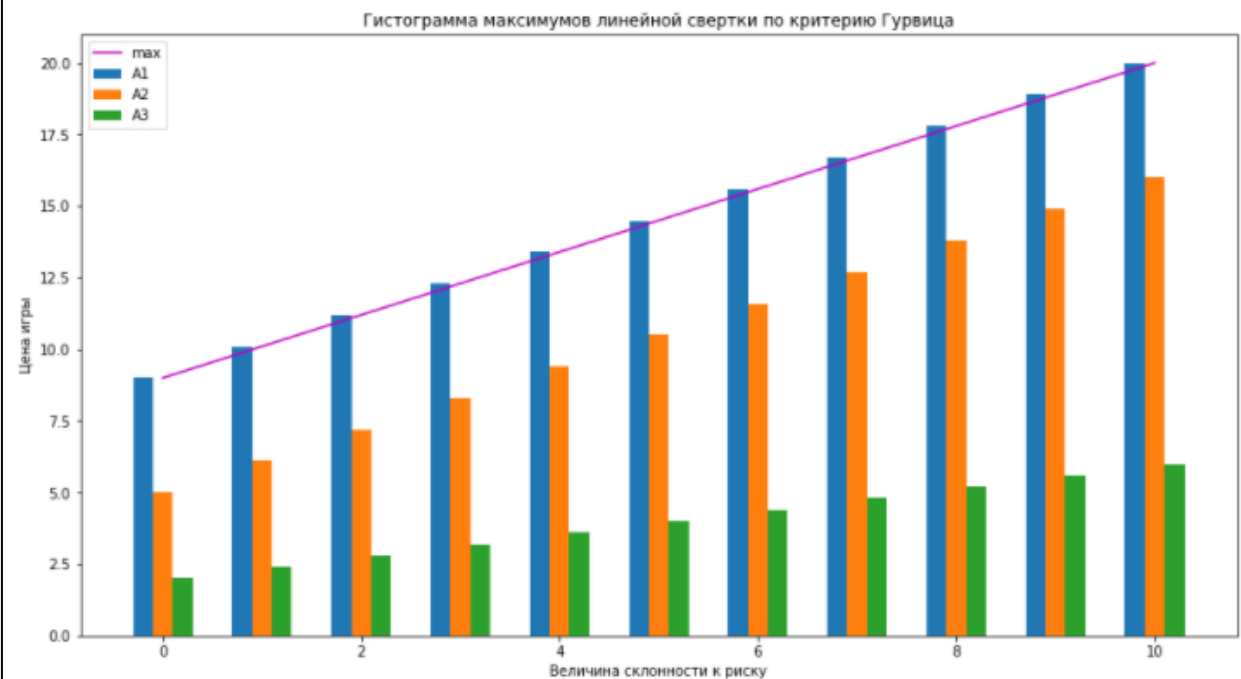
Оптимальная стратегия игрока A по критерию пессимизма: A3
 Цена игры оптимальной стратегии по критерию пессимизма: 2

Оптимальная стратегия игрока A по критерию оптимизма: A1
 Цена игры оптимальной стратегии по критерию оптимизма: 20

Оптимальная стратегия игрока A по критерию Вальда: A1
 Цена игры оптимальной стратегии по критерию Вальда: 9

Линейная свертка склонности к риску по критерию Гурвица:

	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
A1	9.0	10.1	11.2	12.3	13.4	14.5	15.6	16.7	17.8	18.9	20.0
A2	5.0	6.1	7.2	8.3	9.4	10.5	11.6	12.7	13.8	14.9	16.0
A3	2.0	2.4	2.8	3.2	3.6	4.0	4.4	4.8	5.2	5.6	6.0
max	9.0	10.1	11.2	12.3	13.4	14.5	15.6	16.7	17.8	18.9	20.0



Оптимальная стратегия игрока A по критерию Сэвиджа: A1
 Величина минимальной недополученной прибыли по критерию Сэвиджа: 0

Рисунок 72. Решение в питоне 1

6.2.1.2. Тест №1 онлайн-калькулятором:

Вводим необходимые данные в матрицу онлайн-калькулятора¹, выбираем критерий «Максимально ожидаемый выигрыш», критерий Вальда, критерий Сэвиджа и обобщенный критерий Гурвица и запускаем программу.

Выбираем из (9; 5; 2) максимальный элемент $\max=9$
Вывод: выбираем стратегию $N=1$.

Рисунок 73. Критерий Вальда 1

Выбираем из (0; 15; 14) минимальный элемент $\min=0$
Вывод: выбираем стратегию $N=1$.

Рисунок 74. Критерий Сэвиджа 1

Таким образом, в результате решения статистической игры по различным критериям чаще других рекомендовалась стратегия A_1 .

Рисунок 75. Критерий Гурвица 1

6.2.2. Проверка №2 матрица 3x3:

6.2.2.1. Тест №2 кодом Python:

Выбираем размер матрицы, для неё вводим диапазон допустимых значений и получаем оптимальную стратегию по критерию Вальда, гистограмму максимумов линейной свёртки по критерию Гурвица и оптимальную стратегию по критерию Сэвиджа.

¹ <https://math.semestr.ru/games/stat.php>

Выберите способ ввода данных: 2
 Введите количество стратегий для компании A: 3
 Введите количество состояний природы P: 3
 Введите диапазон значений для матрицы весовых коэффициентов :
 Введите минимальное значение: 5
 Введите максимальное значение: 25

Весовая матрица:

	p1	p2	p3
A1	8	24	13
A2	5	19	12
A3	7	8	10

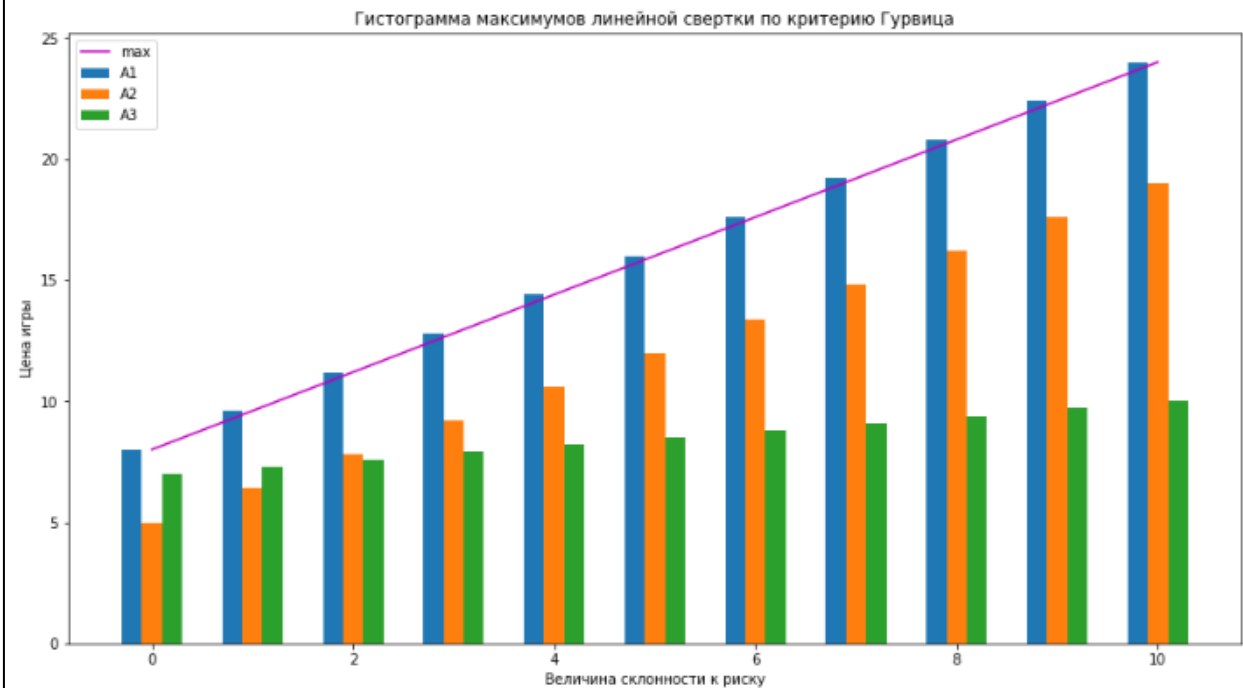
Оптимальная стратегия игрока A по критерию пессимизма: A2
 Цена игры оптимальной стратегии по критерию пессимизма: 5

Оптимальная стратегия игрока A по критерию оптимизма: A1
 Цена игры оптимальной стратегии по критерию оптимизма: 24

Оптимальная стратегия игрока A по критерию Вальда: A1
 Цена игры оптимальной стратегии по критерию Вальда: 8

Линейная свертка склонности к риску по критерию Гурвица:

	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
A1	8.0	9.6	11.2	12.8	14.4	16.0	17.6	19.2	20.8	22.4	24.0
A2	5.0	6.4	7.8	9.2	10.6	12.0	13.4	14.8	16.2	17.6	19.0
A3	7.0	7.3	7.6	7.9	8.2	8.5	8.8	9.1	9.4	9.7	10.0
max	8.0	9.6	11.2	12.8	14.4	16.0	17.6	19.2	20.8	22.4	24.0



Оптимальная стратегия игрока A по критерию Сэвиджа: A1
 Величина минимальной недополученной прибыли по критерию Сэвиджа: 0

Рисунок 76. Решение в питоне 2

6.2.2.2. Тест №2 онлайн-калькулятором:

Вводим необходимые данные в матрицу онлайн-калькулятора², выбираем критерий «Максимально ожидаемый выигрыш», критерий Вальда, критерий Сэвиджа и обобщённый критерий Гурвица и запускаем программу.

Выбираем из (8; 5; 7) максимальный элемент $\max=8$
Вывод: выбираем стратегию $N=1$.

Рисунок 77. Критерий Вальда 2

Выбираем из (0; 5; 16) минимальный элемент $\min=0$
Вывод: выбираем стратегию $N=1$.

Рисунок 78. Критерий Сэвиджа 2

Таким образом, в результате решения статистической игры по различным критериям чаще других рекомендовалась стратегия A_1 .

Рисунок 79. Критерий Гурвица 2

6.2.3. Проверка №3 матрица 3x3:

6.2.3.1. Тест №3 кодом Python:

Выбираем размер матрицы, для неё вводим диапазон допустимых значений и получаем оптимальную стратегию по критерию Вальда, гистограмму максимумов линейной свёртки по критерию Гурвица и оптимальную стратегию по критерию Сэвиджа.

² <https://math.semestr.ru/games/stat.php>

Выберите способ ввода данных: 2
 Введите количество стратегий для компании А: 3
 Введите количество состояний природы P: 3
 Введите диапазон значений для матрицы весовых коэффициентов :
 Введите минимальное значение: 7
 Введите максимальное значение: 17

Весовая матрица:

	p1	p2	p3
A1	7	10	11
A2	11	16	14
A3	17	15	11

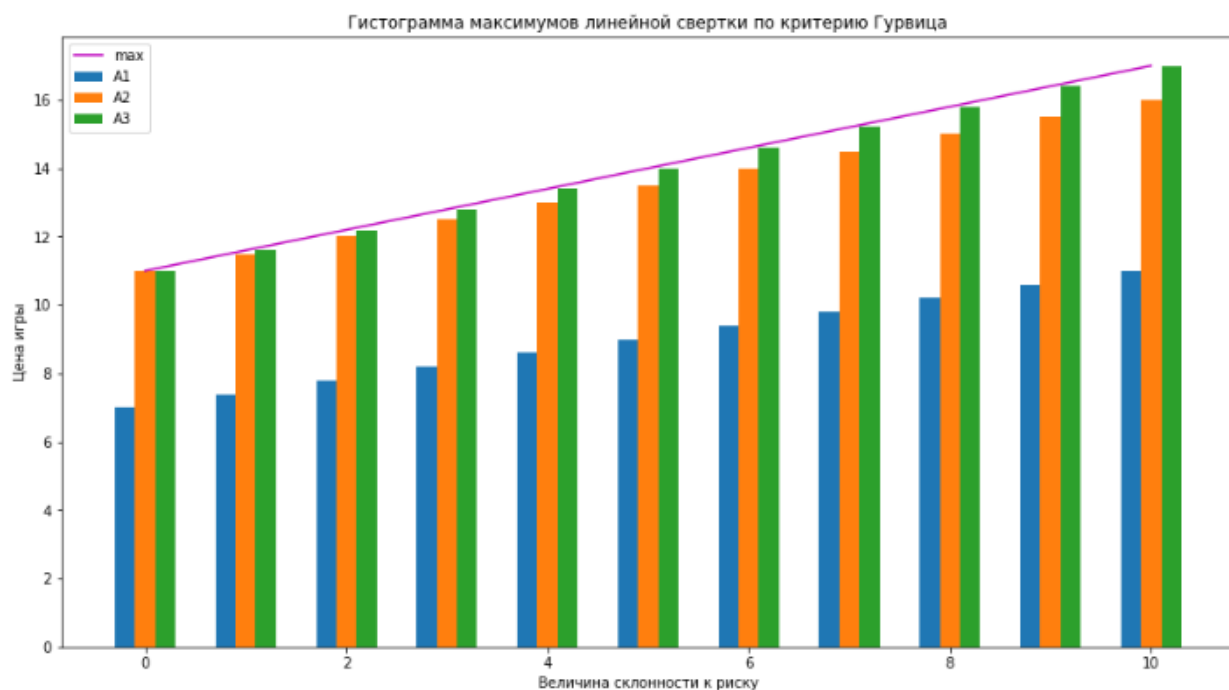
Оптимальная стратегия игрока А по критерию пессимизма: A1
 Цена игры оптимальной стратегии по критерию пессимизма: 7

Оптимальная стратегия игрока А по критерию оптимизма: A3
 Цена игры оптимальной стратегии по критерию оптимизма: 17

Оптимальная стратегия игрока А по критерию Вальда: A2
 Цена игры оптимальной стратегии по критерию Вальда: 11

Линейная свертка склонности к риску по критерию Гурвица:

	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
A1	7.0	7.4	7.8	8.2	8.6	9.0	9.4	9.8	10.2	10.6	11.0
A2	11.0	11.5	12.0	12.5	13.0	13.5	14.0	14.5	15.0	15.5	16.0
A3	11.0	11.6	12.2	12.8	13.4	14.0	14.6	15.2	15.8	16.4	17.0
max	11.0	11.6	12.2	12.8	13.4	14.0	14.6	15.2	15.8	16.4	17.0



Оптимальная стратегия игрока А по критерию Сэвиджа: A3
 Величина минимальной недополученной прибыли по критерию Сэвиджа: 3

Рисунок 80. Решение в питоне 3

6.2.3.2. Тест №3 онлайн-калькулятором:

Вводим необходимые данные в матрицу онлайн-калькулятора³, выбираем критерий «Максимально ожидаемый выигрыш», критерий Вальда, критерий Сэвиджа и обобщённый критерий Гурвица и запускаем программу.

Выбираем из (7; 11; 11) максимальный элемент $\max=11$
Вывод: выбираем стратегию $N=2$.

Рисунок 81. Критерий Вальда 3

Выбираем из (10; 6; 3) минимальный элемент $\min=3$
Вывод: выбираем стратегию $N=3$.

Рисунок 82. Критерий Сэвиджа 3

Таким образом, в результате решения статистической игры по различным критериям чаще других рекомендовалась стратегия A_3 .

Рисунок 83. Критерий Гурвица 3

6.2.4. Проверка №4 матрица 4x4:

6.2.4.1. Тест №4 кодом Python:

Выбираем размер матрицы, для неё вводим диапазон допустимых значений и получаем оптимальную стратегию по критерию Вальда, гистограмму максимумов линейной свёртки по критерию Гурвица и оптимальную стратегию по критерию Сэвиджа.

³ <https://math.semestr.ru/games/stat.php>

Введите количество стратегий для компании A: 4
 Введите количество состояний природы P: 4
 Введите диапазон значений для матрицы весовых коэффициентов :
 Введите минимальное значение: 3
 Введите максимальное значение: 23

Весовая матрица:

	p1	p2	p3	p4
A1	21	8	3	9
A2	11	4	20	14
A3	22	15	18	4
A4	7	18	20	19

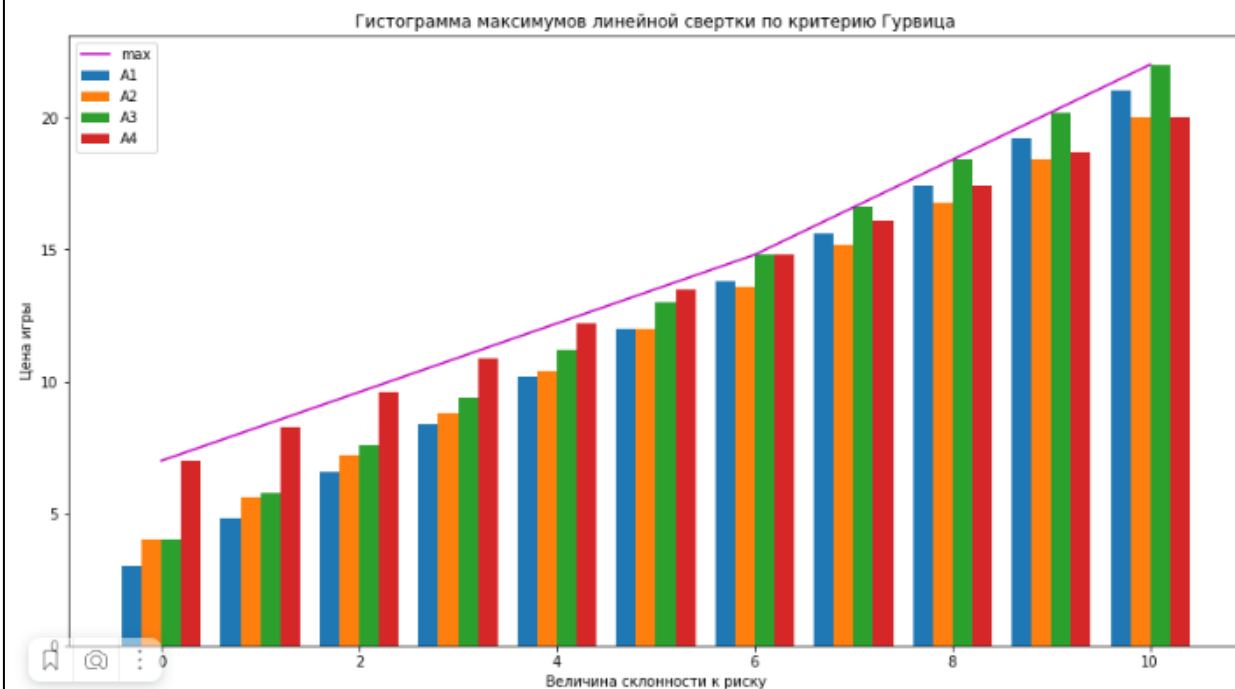
Оптимальная стратегия игрока A по критерию пессимизма: A1
 Цена игры оптимальной стратегии по критерию пессимизма: 3

Оптимальная стратегия игрока A по критерию оптимизма: A3
 Цена игры оптимальной стратегии по критерию оптимизма: 22

Оптимальная стратегия игрока A по критерию Вальда: A4
 Цена игры оптимальной стратегии по критерию Вальда: 7

Линейная свертка склонности к риску по критерию Гурвица:

	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
A1	3.0	4.8	6.6	8.4	10.2	12.0	13.8	15.6	17.4	19.2	21.0
A2	4.0	5.6	7.2	8.8	10.4	12.0	13.6	15.2	16.8	18.4	20.0
A3	4.0	5.8	7.6	9.4	11.2	13.0	14.8	16.6	18.4	20.2	22.0
A4	7.0	8.3	9.6	10.9	12.2	13.5	14.8	16.1	17.4	18.7	20.0
max	7.0	8.3	9.6	10.9	12.2	13.5	14.8	16.6	18.4	20.2	22.0



Оптимальная стратегия игрока A по критерию Сэвиджа: A2
 Величина минимальной недополученной прибыли по критерию Сэвиджа: 14

Рисунок 84. Решение в питоне 4

6.2.4.2. Тест №4 онлайн-калькулятором:

Вводим необходимые данные в матрицу онлайн-калькулятора⁴, выбираем критерий «Максимально ожидаемый выигрыш», критерий Вальда, критерий Сэвиджа и обобщённый критерий Гурвица и запускаем программу.

Выбираем из (3; 4; 4; 7) максимальный элемент $\max=7$
Вывод: выбираем стратегию $N=4$.

Рисунок 85. Критерий Вальда 4

Выбираем из (17.792; 17.149; 18.792; 17.683) максимальный элемент $\max=18.79$
Вывод: выбираем стратегию $N=3$.

Рисунок 86. Критерий Сэвиджа 4

Таким образом, в результате решения статистической игры по различным критериям чаще других рекомендовалась стратегия A_2 .

Рисунок 87. Критерий Гурвица 4

6.2.5. Проверка №5 матрица 4x4:

6.2.5.1. Тест №5 кодом Python:

Выбираем размер матрицы, для неё вводим диапазон допустимых значений и получаем оптимальную стратегию по критерию Вальда, гистограмму максимумов линейной свёртки по критерию Гурвица и оптимальную стратегию по критерию Сэвиджа.

⁴ <https://math.semestr.ru/games/stat.php>

Введите количество стратегий для компании А: 4
 Введите количество состояний природы P: 4
 Введите диапазон значений для матрицы весовых коэффициентов :
 Введите минимальное значение: 6
 Введите максимальное значение: 32

Весовая матрица:

	П1	П2	П3	П4
A1	28	29	28	30
A2	31	10	20	6
A3	23	25	29	25
A4	11	21	9	24

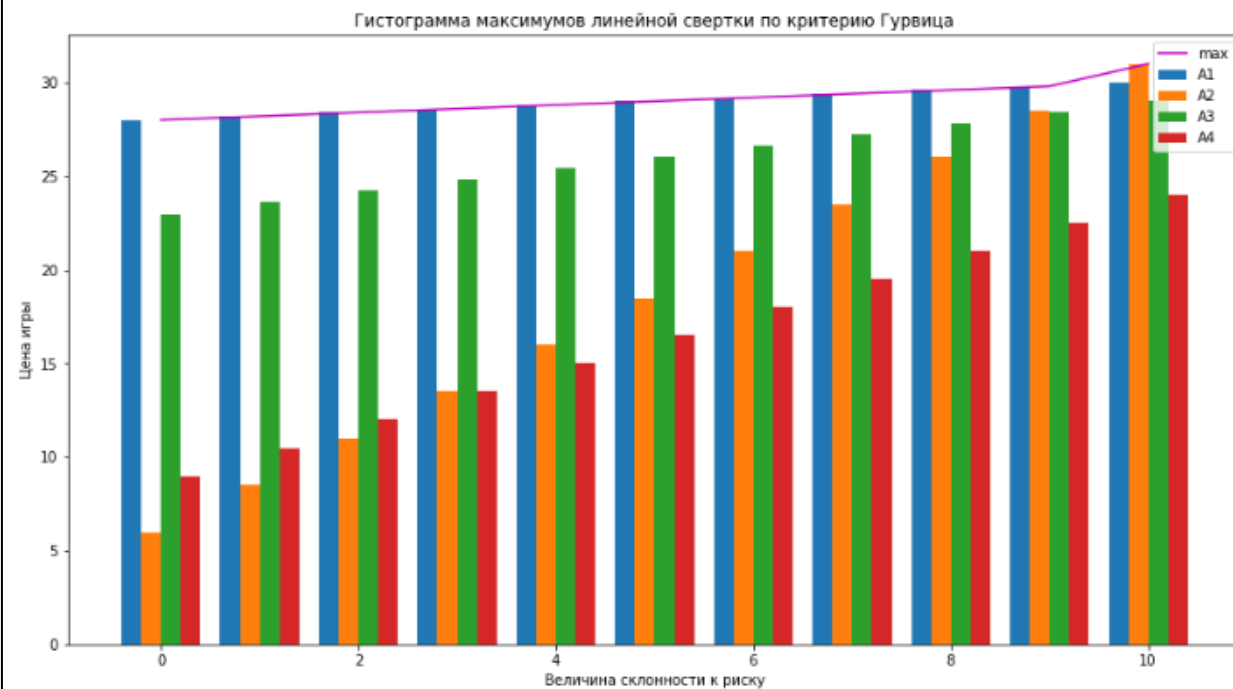
Оптимальная стратегия игрока А по критерию пессимизма: A2
 Цена игры оптимальной стратегии по критерию пессимизма: 6

Оптимальная стратегия игрока А по критерию оптимизма: A2
 Цена игры оптимальной стратегии по критерию оптимизма: 31

Оптимальная стратегия игрока А по критерию Вальда: A1
 Цена игры оптимальной стратегии по критерию Вальда: 28

линейная свертка склонности к риску по критерию Гурвица:

	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
A1	28.0	28.2	28.4	28.6	28.8	29.0	29.2	29.4	29.6	29.8	30.0
A2	6.0	8.5	11.0	13.5	16.0	18.5	21.0	23.5	26.0	28.5	31.0
A3	23.0	23.6	24.2	24.8	25.4	26.0	26.6	27.2	27.8	28.4	29.0
A4	9.0	10.5	12.0	13.5	15.0	16.5	18.0	19.5	21.0	22.5	24.0
max	28.0	28.2	28.4	28.6	28.8	29.0	29.2	29.4	29.6	29.8	31.0



Оптимальная стратегия игрока А по критерию Сэвиджа: A1
 Величина минимальной недополученной прибыли по критерию Сэвиджа: 3

Рисунок 88. Решение в питоне 5

6.2.5.2. Тест №5 онлайн-калькулятором:

Вводим необходимые данные в матрицу онлайн-калькулятора⁵, выбираем критерий «Максимально ожидаемый выигрыш», критерий Вальда, критерий Сэвиджа и обобщённый критерий Гурвица и запускаем программу.

Выбираем из (28; 6; 23; 9) максимальный элемент $\max=28$
Вывод: выбираем стратегию $N=1$.

Рисунок 89. Критерий Вальда 5

Выбираем из (3; 24; 8; 20) минимальный элемент $\min=3$
Вывод: выбираем стратегию $N=1$.

Рисунок 90. Критерий Сэвиджа 5

Таким образом, в результате решения статистической игры по различным критериям чаще других рекомендовалась стратегия A_1 .

Рисунок 91. Критерий Гурвица 5

Для подсчёта времени на выполнение алгоритмов будет использовано время, затраченное на ввод необходимых данных в предназначенные поля и расчёт данных самой программой с дальнейшим выводом их пользователю на экран.

Таблица 11. Сравнение решения в питоне и эксель

Входные данные	Python	Онлайн-калькулятор
Тестирование №1	~15 секунд	~18 секунд
Тестирование №2	~14 секунд	~18 секунд
Тестирование №3	~14 секунд	~17 секунд
Тестирование №4	~13 секунд	~17 секунд
Тестирование №5	~13 секунд	~17 секунд

⁵ <https://math.semestr.ru/games/stat.php>

7 Заключение

7.1. Заключение по задачи на нахождение выигрышной стратегии в условиях риска

Наш представленный код решает поставленную задачу. На основании тестирования данного алгоритма можно сделать вывод о том, что Python выводит самое оптимальное решение достаточно быстро. Ниже представлено решение поставленной задачи через Python:

```

Введите количество стратегий для компании А: 4
Введите название 1 стратегии компании А: Стратегия
Введите название 2 стратегии компании А: Шутер
Введите название 3 стратегии компании А: Аркада
Введите название 4 стратегии компании А: Гонки

Введите количество состояний природы П: 4
Введите название 1 состояний природы П: Повышенный
Введите название 2 состояний природы П: Стагнирующий
Введите название 3 состояний природы П: Сокращающийся
Введите название 4 состояний природы П: Депрессивный

Введите матрицу весовых коэффициентов поэлементно:
Введите элемент строки 1 столбца 1: 50
Введите элемент строки 1 столбца 2: 30
Введите элемент строки 1 столбца 3: 35
Введите элемент строки 1 столбца 4: 45
Введите элемент строки 2 столбца 1: 40
Введите элемент строки 2 столбца 2: 40
Введите элемент строки 2 столбца 3: 55
Введите элемент строки 2 столбца 4: 40
Введите элемент строки 3 столбца 1: 35
Введите элемент строки 3 столбца 2: 40
Введите элемент строки 3 столбца 3: 45
Введите элемент строки 3 столбца 4: 55
Введите элемент строки 4 столбца 1: 10
Введите элемент строки 4 столбца 2: 20
Введите элемент строки 4 столбца 3: 25
Введите элемент строки 4 столбца 4: 35

Введите вектор вероятности поэлементно:
Введите элемент строки 1 столбца 1: 0.25
Введите элемент строки 1 столбца 2: 0.25
Введите элемент строки 1 столбца 3: 0.25
Введите элемент строки 1 столбца 4: 0.25

Весовая матрица стратегий игрока А:
      Повышенный  Стагнирующий  Сокращающийся  Депрессивный
Стратегия      50.0         30.0         35.0         45.0
Шутер           40.0         40.0         55.0         40.0
Аркада          35.0         40.0         45.0         55.0
Гонки           10.0         20.0         25.0         35.0

Состояния природы П:
      Повышенный  Стагнирующий  Сокращающийся  Депрессивный
0      0.25         0.25         0.25         0.25

Оптимальная стратегия игрока А по критерию Лапласа: А2-Шутер; А3-Аркада
Цена игры оптимальной стратегией по критерию Лапласа: 43.75

Оптимальная стратегия игрока А по критерию Байеса: А2-Шутер; А3-Аркада
Цена игры оптимальной стратегией по критерию Байеса: 43.75

Оптимальная чистая стратегия игрока А по критерию Гермейера: А2-Шутер
Цена игры оптимальной чистой стратегией по критерию Гермейера: 10.0/40.0
Цена игры для игрока А при выборе смешанной оптимальной стратегии по критерию Гермейера: 10.0/43.75
Таблица оптимальных смешанных стратегий для игрока А по критерию Гермейера:
+-----+-----+-----+-----+
| Стратегия | Шутер | Аркада | Гонки |
+-----+-----+-----+-----+
| 0%        | 100%  | 0%      | 0%      |
+-----+-----+-----+-----+

```

Рисунок 92. Решение поставленной задачи через питон

Теперь сравним два алгоритма по критериям: эффективности, скорости использования алгоритма, простоты использования, надёжности в разрезе человеческого фактора и точности предоставляемого решения.

Таблица 12. Сравнение Питона и Эксель

Критерий	Python	Excel
Эффективность	Высокая	Высокая
Скорость использования алгоритма	Высокая	Низкая
Простота использования	Высокая	Низкая
Надёжность (человеческий фактор)	Высокая	Средняя
Точность	Высокая	Высокая

Мы считаем, что представленный рукописный код на языке Python лучше, потому что он удобнее, быстрее и проще, а ещё имеет импорт .csv файлов и случайный ввод данных. Улучшением кода, к примеру, может послужить время выполнения запроса.

7.2. Заключение по задаче на нахождение выигрышной стратегии в условиях неопределенности

Наш представленный код решает поставленную задачу. На основании тестирования данного алгоритма можно сделать вывод о том, что Python выводит самое оптимальное решение достаточно быстро. Ниже представлено решение поставленной задачи через Python и проверка в онлайн-калькуляторе:

Выберите способ ввода данных: 1
 Введите количество стратегий для компании A: 4
 Введите количество состояний природы P: 4
 Введите название 1 стратегии компании A: Повышенный
 Введите название 2 стратегии компании A: Стагнирующий
 Введите название 3 стратегии компании A: Сокращающийся
 Введите название 4 стратегии компании A: Депрессивный
 Введите название 1 состояния природы P: Стратегия
 Введите название 2 состояния природы P: Шутер
 Введите название 3 состояния природы P: Аркада
 Введите название 4 состояния природы P: Гонки
 Введите матрицу весовых коэффициентов поэлементно:
 Введите элемент строки 1 столбца 1: 50
 Введите элемент строки 1 столбца 2: 30
 Введите элемент строки 1 столбца 3: 35
 Введите элемент строки 1 столбца 4: 45
 Введите элемент строки 2 столбца 1: 40
 Введите элемент строки 2 столбца 2: 40
 Введите элемент строки 2 столбца 3: 55
 Введите элемент строки 2 столбца 4: 40
 Введите элемент строки 3 столбца 1: 35
 Введите элемент строки 3 столбца 2: 40
 Введите элемент строки 3 столбца 3: 45
 Введите элемент строки 3 столбца 4: 55
 Введите элемент строки 4 столбца 1: 10
 Введите элемент строки 4 столбца 2: 20
 Введите элемент строки 4 столбца 3: 25
 Введите элемент строки 4 столбца 4: 35

 Оптимальная стратегия игрока A по критерию пессимизма: A4 - Депрессивный
 Цена игры оптимальной стратегии по критерию пессимизма: 10

 Оптимальная стратегия игрока A по критерию оптимизма: A2 - Стагнирующий
 Цена игры оптимальной стратегии по критерию оптимизма: 55

 Оптимальная стратегия игрока A по критерию Вальда: A2 - Стагнирующий
 Цена игры оптимальной стратегии по критерию Вальда: 40

 Линейная свертка склонности к риску по критерию Гурвица:

	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	\
A1 - Повышенный	30.0	32.0	34.0	36.0	38.0	40.0	42.0	44.0	46.0	
A2 - Стагнирующий	40.0	41.5	43.0	44.5	46.0	47.5	49.0	50.5	52.0	
A3 - Сокращающийся	35.0	37.0	39.0	41.0	43.0	45.0	47.0	49.0	51.0	
A4 - Депрессивный	10.0	12.5	15.0	17.5	20.0	22.5	25.0	27.5	30.0	
max	40.0	41.5	43.0	44.5	46.0	47.5	49.0	50.5	52.0	

	0.9	1.0
A1 - Повышенный	48.0	50.0
A2 - Стагнирующий	53.5	55.0
A3 - Сокращающийся	53.0	55.0
A4 - Депрессивный	32.5	35.0
max	53.5	55.0

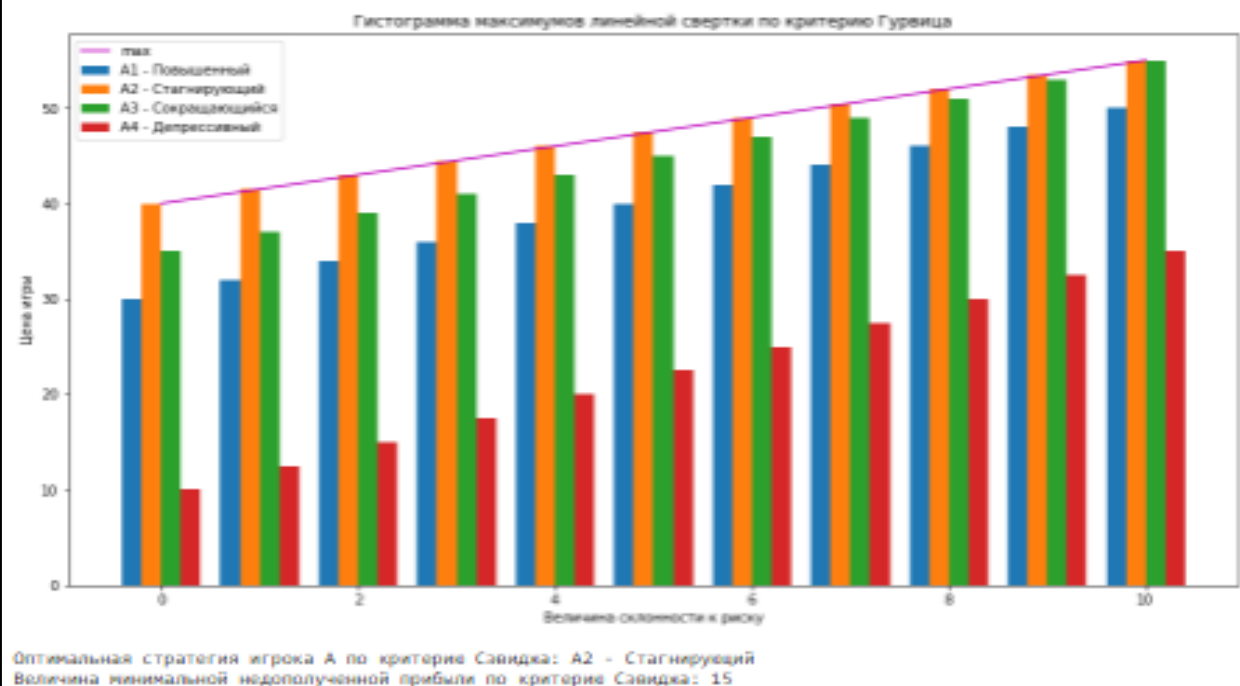


Рисунок 93. Решение поставленной задачи через питон

Выбираем из (30; 40; 35; 10) максимальный элемент $\max=40$
Вывод: выбираем стратегию $N=2$.

Рисунок 94. Метод Вальда

Выбираем из (20; 15; 15; 40) минимальный элемент $\min=15$
Вывод: выбираем стратегию $N=2$.

Рисунок 95. Критерий Сэвиджа

Выбираем из (42.581; 49.435; 47.581; 25.726) максимальный элемент $\max=49.44$
Вывод: выбираем стратегию $N=2$.

Рисунок 96. Критерий Гурвица

Теперь сравним два алгоритма по критериям: эффективности, скорости использования алгоритма, простоты использования, надёжности в разрезе человеческого фактора и точности предоставляемого решения.

Таблица 13. Сравнение питона и онлайн-калькулятора

Критерий	Python	Онлайн-калькулятор
Эффективность	Высокая	Высокая
Скорость использования алгоритма	Высокая	Средняя
Простота использования	Высокая	Высокая
Надёжность (человеческий фактор)	Высокая	Средняя
Точность	Высокая	Высокая

Мы считаем, что представленный рукописный код на языке Python лучше, потому что он удобнее, быстрее и проще, а ещё имеет импорт .csv файлов и случайный ввод данных. Улучшением кода, к примеру, может послужить добавление времени выполнения запроса.