

**МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
им М.В. Ломоносова**

Факультет вычислительной математики и кибернетики

**Компьютерный практикум по учебному курсу
«Исследование операций»
ЗАДАНИЕ №4**

ОТЧЕТ

о выполненном задании

студентов 311 - 312 учебных групп факультета ВМК МГУ
Волкова А., Камкия Н., Кожух П., Мирзоева С., Сюаньлина В.

(фамилия имя студентов)

гор.Москва
2021

Оглавление

1	Постановка задачи	3
2	Решение	4
2.1	Ход решения	4
2.2	Ход решения	4
3	Используемые средства	6
4	Литература	7

Глава 1

Постановка задачи

После оглушительного успеха в освобождении Астапора, Миэрина и Юнкая от власти работорговцев Дейенерис Бурерожденная открыла себе доступ к Летнему морю, а следовательно – путь в Вестерос.

Для ведения войны с Семью Королевствами нужно оружие, а для оружия нужна сталь. Нет никаких сомнений в кузнечном искусстве Безупречных, однако поставщики стали не столь надежны.

Два основных поставщика стали - это Westeros Inc. и Hargry and Co. На протяжении нескольких месяцев мы покупаем сталь у обеих компаний, и каждая из них предлагает ощутимую скидку при заключении эксклюзивного договора на поставку.

Советник королевы Тирион Ланнистер знает о твоём умении принимать взвешенные рациональные решения и просит помощи в объективном решении вопроса о том, с какой из компаний следует заключить эксклюзивный договор на поставку стали.

У Тириона есть записи о производстве мечей каждым из кузнецов-безупречных, а также данные о количестве сломанных мечей в каждый из месяцев ведения боевых действий.

Данные имеют вид:

unsullens.id	production.date	report.date	produced	defects	supplier
int	int	int	int	int	string

Цель задания 4 в проведении разведывательного анализа данных с целью ответа на вопрос: "С каким из поставщиков стали следует заключить договор?".

Формально, задача заключается в следующем:

1. Считать данные из входной таблицы;
2. Построить по этим данным некие таблицы;
3. Ответить на поставленный вопрос;

Глава 2

Решение

2.1 Ход решения

Получим обзорную информацию, построив различные графики:

1. Общее количество произведенного и сломавшегося оружия для каждой из компаний за все месяцы сотрудничества;
2. Количество произведенного и сломавшегося оружия обеих компаний для каждого месяца производства;
3. Среднее число поломанной продукции по сроку службы;
4. Качество продукции за каждый месяц (процент сломавшейся продукции, которая была произведена в i -й месяц);
5. Зависимость количества поломок от кузнеца.

2.2 Ход решения

all – time(book). Посмотрим для каждого поставщика, сколько всего мечей было произведено и какое количество из них сломалось. Используя *drop*, исключим колонки *report.date*, *production.date*, *unsullen.id* за ненадобностью. С помощью *groupby* сгруппируем данные по поставщику и применим *sum*, таким образом, для каждой компании получим суммарное количество произведённой продукции и число сломавшихся мечей. Сгенерируем объекты *fig* и *axes*. Используя *subplots*, будем строить диаграммы для каждого поставщика. Перебираем компании, вызывая функцию *book.supplier.unique()*, и формируем с помощью *loc* подтаблицы, где хранится информация только про одного поставщика. Теперь с помощью *plot* отобразим на диаграммах данные для каждого производителя с номером *number*, где по оси ординат будут отложены *produced* и *defects*, а две колонки будут отвечать за количество произведённых и сломавшихся мечей. Положение диаграмм зададим используя *axes* и *number*.

crack – and – make(book). Используя *drop*, исключим колонки *report.date*, *unsullen.id* за ненадобностью. С помощью *groupby* сгруппируем данные по поставщику и месяцу изготовления, а затем применим *sum*. Таким образом получим суммарное количество произведённой продукции в каждый месяц и число сломавшихся мечей. Создадим новые наборы данных *prod_book* и *crack_book*. Для этого с помощью *pivot* преобразуем таблицу, используя значения столбцов или индексов. Индексом будет служить *production.date*, колонками – *supplier*. Для набора *prod_book* ячейки будут заполнены значениями из колонки *produced*, для *crack_book* – *defects*. Сгенерируем объекты *fig* и *axes*, используя *subplots*, и построим две диаграммы для произведённых и сломавшихся мечей, с помощью *plot* и *ax* задав их положение.

month – cracked – prod(book). Определим для каждого *production.date* среднюю долю поломанных к данному месяцу мечей. Используя *drop*, исключим колонки *report.date*, *unsullen.id* за ненадобностью. С помощью *groupby* сгруппируем данные по поставщику и месяцу изготовления, далее применим *sum*, таким образом получим суммарное количество произведённой продукции в каждый месяц и число сломавшихся мечей. Поделим число поломок на число произведённых изделий, запишем полученное частное в колонку *defects* и разделим её на число месяцев ($7 - book['production.date']$), прошедших со дня изготовления продукции. Построим график.

Перебирая поставщиков (*book.supplier.unique()*), формируем с помощью *loc* подтаблицы, где хранится информация только про одного поставщика. Теперь отобразим данные на графике с помощью *fig.plot*, где по оси ординат отложены *defects*, а по оси абсцисс – месяц изготовления.

mean – after – month(book). Определим среднее число сломавшихся мечей после каждого месяца эксплуатации, то есть среднее количество поломок в месяц после одного месяца использования, после двух и т.д. Вычтем из *report.date* колонку *production.date* и переименуем её с помощью *rename* в *time_to_death*, получая срок службы. Используя *drop*, исключим колонки *unsullen.id*, *produced* за ненадобностью. С помощью *groupby* сгруппируем данные по месяцу изготовления, поставщику и сроку службы, следом применим *sum*. После, исключив столбец *production.date*, перегруппируем по поставщику и сроку службы. Вслед за этим, используя *mean*, возьмём среднее и построим график. Перебираем поставщиков, вызывая функцию *book.supplier.unique()* и формируем с помощью *loc* подтаблицы, где хранится информация только про одного поставщика. Теперь отобразим данные на графике, где по оси ординат отложены *defects*.

unsullens(book). Проверим не портит ли какой-либо кузнец общий результат. Используя *drop*, исключим колонки *report.date*, *production.date* за ненадобностью. С помощью *groupby* сгруппируем данные по поставщику и кузнецу, а затем применим *sum*. Таким образом, получаем для каждого кузнеца суммарное количество произведённой им продукции в каждый месяц и число сломавшихся мечей. Построим график, далее сгенерируем объекты *fig* и *axes*, используя *subplots*. Также введём переменную *number*, отвечающую за номер поставщика. Перебираем поставщиков, вызывая функцию *book.supplier.unique()*, и формируем с помощью *loc* подтаблицы, где хранится информация только про одного производителя. Индексной делаем колонку, содержащую идентификационный номер кузнеца. Используя *drop*, исключим за ненадобностью колонки *supplier*, *unsullen.id*. Теперь, с помощью *plot*, отобразим на диаграммах данные для каждого производителя с номером *number*, где по оси ординат будут отложены *produced* и *defects*, а по оси абсцисс – номер кузнеца.

crack – production(book). Используя *drop*, исключим колонки *report.date*, *unsullen.id* за ненадобностью. С помощью *groupby* сгруппируем данные по поставщику и месяцу изготовления и применим *sum*. Таким образом, получаем суммарное количество произведённой продукции в каждый месяц. Подсчитаем сколько процентов из всей продукции, произведённой в *i* – й месяц, сломалось к текущему моменту, то есть найдём отношение *book['defects']* к *book['produced']* и сохраним его в колонку *percent*. Построим график. Перебираем поставщиков, вызывая функцию *book.supplier.unique()*, и формируем с помощью *loc* подтаблицы, где хранится информация только про одного поставщика. Теперь, отобразим данные на графике, где по оси ординат отложены *percent*.

Глава 3

Используемые средства

Модули:

1. *csv*;
2. *copy*;
3. *matplotlib*;
4. *pandas*;

Функции:

1. *read_csv(filepath_or_buffer, sep = ',')* – считывает *csv* файл в *dataframe*;
2. *sep* – разделитель;
3. *copy.deepcopy(x)* – возвращает полную копию *x*;
4. *drop* – удаляет указанную строку или столбец;
5. *groupby()* – группирует элементы таблицы по колонке;
6. *sum()* – суммирует элементы группы;
7. *rename(columns)* – переименование колонок;
8. *plt.figure(figsize = None).add_subplot(numrows, numcols, fignum)* – *figsize* задаёт размер, *add – subplot* добавляет объект для рисования графика по указанным координатам;
9. *unique()* – возвращает уникальные значения объекта *Series*;
10. *loc* – доступ к группе строк и столбцов по меткам или логическому выражению;
11. *fig.plot(x, y, color, label)* – рисует график, где точки имеют координаты (x, y) , *label* используется для генерации легенды;
12. *plt.show()* – показ изображения (графика);
13. *plt.subplots(nrows, ncols)* – возвращает объекты *Figure* и *Axes*;
14. *pivot(index, columns, values)*
 - (a) *index* и *columns* – столбцы, используемые для создания индекса и столбца нового датафрейма;
 - (b) *values* – столбцы, используемые для заполнения значений нового фрейма;

Иной инструментарий:

1. *Jupyter Notebook*;

Глава 4

Литература

1. *wikipedia* - всякие базовые методы статистики;
2. [https : //docs.python.org/3/library/csv.html](https://docs.python.org/3/library/csv.html) - библиотека *csv*;