**Python homework 1**1. Запросите у пользователя имя и месячную зарплату в долларах и выведите их годовую зарплату в **тысячах** долларов.  
*Например: «Мишель», «12345» → «Годовая зарплата Мишель составляет 148 тыс. долларов».*

2. Запросите целое число и выведите «True», если это четное число в диапазоне от 100 до 999, в противном случае - «False».

3. В качестве входящих данных возьмем целое число; **Предположим**, что это число от 101 до 999, а его последняя цифра не равна нулю. Выведите **обратное** число. Например: 256 → 652.

4. Запросите два целых числа и выведите:  
 a. Их сумму  
 b. Их разность  
 c. Произведение  
 d. Результат деления первого на второе  
 e. Остаток от деления первого на второе  
 f. «True», если первое число больше или равно второму, иначе «False».

**Python homework 2**1. В качестве входящих данных запросите целое число. Если оно делится на 3, выведите «foo»; если оно делится на 5, выведите «bar»; если оно делится на оба, выведите «ham» (а не «foo» или «bar»).

2. В качестве входящих данных запросите два числа и выведите какое из них меньше и какое большее.

3. В качестве входящих данных запросите три числа и выведите наименьшее, среднее и наибольшее. Предположим, все они разные.

**Дополнительные задания повышенной сложности**4. Прочтите это объяснение list slicing <https://www.learnbyexample.org/python-list-slicing/>

5. Сыграйте в игру Fizz-Buzz: выведите все числа от 1 до 100; если число делится на 3, вместо числа выведите «fizz». Если оно делится на 5, вместо числа выведите “buzz”. Если оно делится на оба, выведите «fizz buzz» вместо числа.

6. Сыграйте в игру 7-boom: выведите все числа от 1 до 100; если число делится на 7 или содержит цифру 7, выведите “BOOM” вместо числа.

**Python homework 3**1. В классе профессора Грубла только что был экзамен. Напишите программу, которая принимает в качестве входящих данных оценку каждого ученика и то, сдал ли он экзамен. Затем программе необходимо напечатать две вещи:  
 a. Был ли профессор Грубл последовательным в проставлении отметки “Passed” для студентов.  
 b. Если профессор Грубл был последовательным, выведите диапазон, в котором находится порог для сдачи экзамена.

Примеры:  
● Предположим, мы получаем в качестве входящих данных следующие значения:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Student1 | Student2 | Student3 | Student4 | Student5 | Student6 |
| 78 | 82 | 97 | 86 | 67 | 75 |
| Failed | Passed | Passed | Passed | Passed | Passed |

В данном случае профессор, к сожалению, непоследователен, потому что Student 6 имеет отметку “Passed” с оценкой 75, а Student 1 имеет отметку “Failed” при более высокой оценке - 78.

● Теперь предположим, что мы получили следующие значения:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Student1 | Student2 | Student3 | Student4 | Student5 |
| 84 | 78 | 65 | 90 | 72 |
| Passed | Passed | Failed | Passed | Failed |

В этом случае профессор последовательный, а порог сдачи экзамена находится в диапазоне **73 - 78** баллов.

**Python homework 4**

Обращаюсь к некоторым важным базовым знаниям :)

1.Прочтите официальную спецификацию формата и примеры <https://docs.python.org/3.4/library/string.html#formatspec>

2.Прочтите документацию по формату дата и время <https://docs.python.org/3/library/datetime.html#strftime-strptime-behavior>

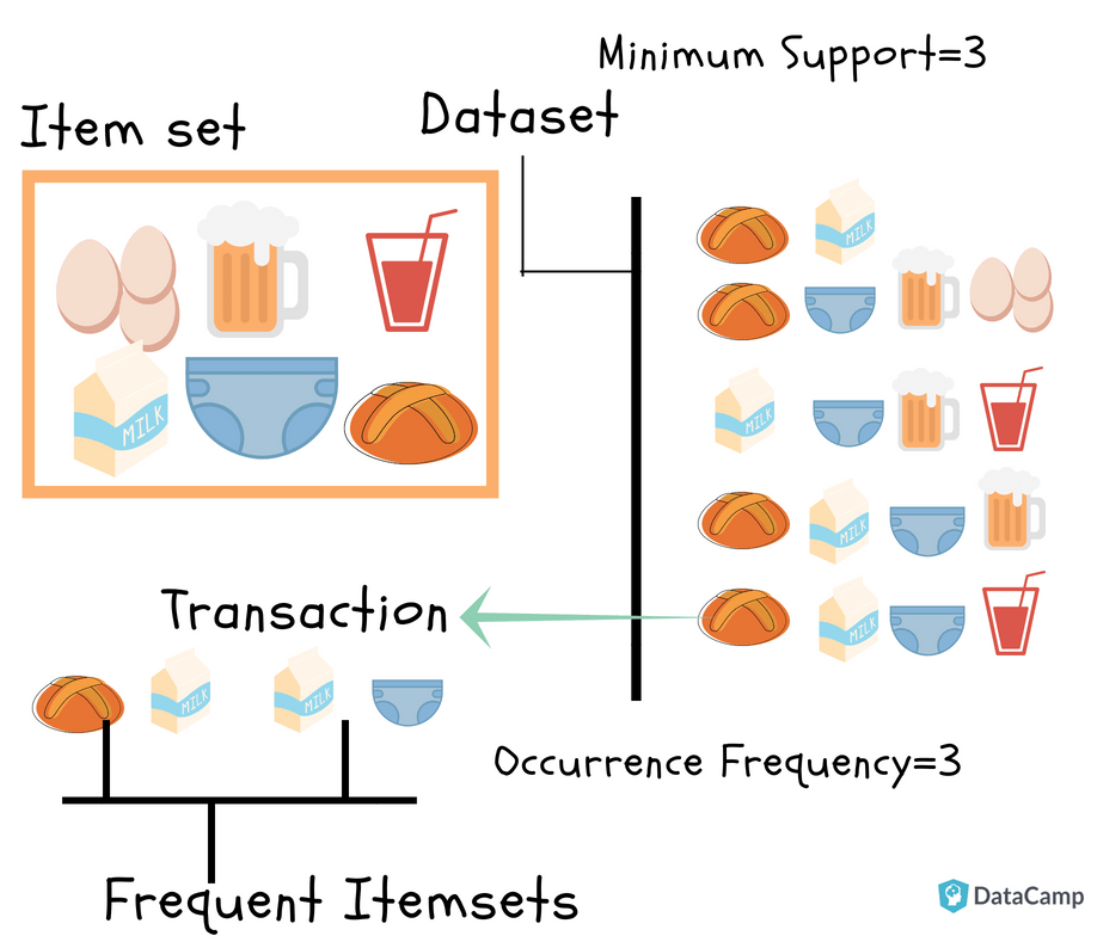
3.Прочтите этот раздел руководства по Python. <https://docs.python.org/3.7/tutorial/controlflow.html#defining-functions>

4.Используйте модуль времени, чтобы сравнить производительность «эффективного» метода поиска простых чисел с простой реализацией (без перерывов, тестирования по всем числам и т.д.). Проверьте несколько диапазонов поиска простых чисел (например, до 100, до 1000 и т. д.)

**Python homework 5**

Анализ рыночной корзины <https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B0%D0%BB%D0%B8%D0%B7_%D1%80%D1%8B%D0%BD%D0%BE%D1%87%D0%BD%D0%BE%D0%B9_%D0%BA%D0%BE%D1%80%D0%B7%D0%B8%D0%BD%D1%8B> (market basket analysis) <https://en.wikipedia.org/wiki/Affinity_analysis>

- это семейство методов в BI для извлечения ассоциаций между событиями из данных. Анализ рыночной корзины - это конкретное приложение анализа сродства, в котором заказы клиентов (например, рыночные корзины супермаркетов) анализируются с целью выявления общих моделей покупок, то есть продуктов, которые обычно покупаются вместе.



Эти шаблоны затем можно использовать для принятия бизнес решений. Например, если 70% людей, которые покупают макароны, также покупают томатный соус, может быть хорошей идеей разместить их рядом друг с другом, чтобы повысить вероятность того, что покупатель макаронных изделий также купит томатный соус. Или, если 99% людей, которые покупают утренние хлопья, также покупают молоко, они могут быть размещены дальше друг от друга, чтобы покупатели, которые хотят купить и то, и другое, прошли длинный путь через супермаркет и увидели множество других продуктов, которые они, возможно, решат купить. На практике интересны даже более «слабые» ассоциации (например, 2%) из-за высокой вариативности покупательских привычек.

В этом упражнении вы реализуете простую версию анализа рыночной корзины и получите информацию из заказов клиентов. В частности, мы будем искать покупательские закономерности между **парами товаров.**

*Данные*

Набор данных предоставляется в одном текстовом файле, где:

● Каждая строка представляет собой отдельный заказ клиента (для удобства чтения заказы разделены двумя строками).

● В каждом заказе содержится список продуктов, приобретенных в этом заказе, разделенных символом «@@@».

Например, вот строка из набора данных:

**Bag of Organic Bananas@@@Celery Sticks@@@Sparkling**

**Water@@@Organic Strawberries**

В этой строке описывается заказ со следующими четырьмя товарами:

● Bag of Organic Bananas

● Celery Sticks

● Sparkling Water

● Organic Strawberries

*Требование к решению*

В этом упражнении мы хотим выявить покупательские ассоциации между парами продуктов, например - Pears => Clementines.

Мы будем использовать два критерия для определения интересных ассоциаций. Для двух продуктов p1 и p2 определим:

1.Уверенность: процент заказов, содержащих оба продукта, от заказов, содержащих p1. Например, если 200 заказов содержали apples, а 30 из них содержали апельсины, то уверенность ассоциации apples => oranges составляет 15% (30/200).

2.Поддержка: Абсолютное количество заказов, содержащих оба продукта; в приведенном выше примере поддержка равна 30.

Почему мы используем два критерия?

●Редкие продукты могут иметь высокую степень достоверности, но на самом деле это не указывает на значительную связь. Например, если было только 2 заказа, содержащих «Aunt Esme’s Special Apple Sauce», и один из них содержал Tomatoes, достоверность составила бы 50% (что считается очень высоким).

●Очень распространенные продукты могут иметь высокую поддержку из-за своей популярности, но, опять же, это не обязательно указывает на значительную связь. Например, если есть 1 миллион заказов, содержащих «bottled water», и 2 миллиона заказов, содержащих «cucumbers», у нас может быть 10 000 заказов, содержащих и то, и другое, высокое абсолютное число, но в контексте это не показывает сильной заимосвязи.

Ваше задание -написать функцию, которая при минимальной уверенности и поддержке печатает все парные ассоциации, которые соответствуют критериям, т.е. имеют больше, чем минимальную уверенность и поддержку.

Инструкции и руководство

1.Сначала напишите код для чтения и анализа данных заказа -разделение заказов по строкам и продуктов по «@@@».

2.Затем напишите код для расчета поддержки и уверенности для каждой пары продуктов.

a.Создайте два словаря -один для подсчета каждого продукта и один для подсчета каждой пары продуктов.

b.Переберите все заказы и обновите два словаря для каждого заказа по мере необходимости.

c.Словарь подсчета пар содержит поддержку каждой пары элементов.

d.Используйте два словаря, чтобы рассчитать уверенность для каждой пары элементов.

e.Обратите внимание, что поддержка симметрична, а уверенность -нет. Например, если есть 100 заказов с “bananas”, 200 заказов с «peanut butter» и 30 заказов с обоими, тогда уверенность bananas => peanut butter составляет 30%, тогда как уверенность peanut butter => bananas -15 %.

f.Помните, что нельзя рассчитывать уверенность продукта в себе (она всегда будет 100%, но это не интересно).

g.Наконец, учитывая минимальные требования к уверенности и поддержке, переберите все пары продуктов и напечатайте пары, соответствующие критериям, в следующем формате:

Chunky Guacamole => Organic Tortilla Chips (5.37% confidence), 38 support

База покупок -вот здесь <https://drive.google.com/file/d/1IOPTVq2ooQfZRkF3rAjGkTjRtbotG7FF/view?usp=sharing>

Удачи!

**Python homework 6**1. Прочтите статью Array Broadcasting in Numpy. <https://numpy.org/doc/stable/user/theory.broadcasting.html#array-broadcasting-in-numpy>

Для следующих упражнений загрузите набор данных фильма по этой ссылке <https://gist.githubusercontent.com/tiangechen/b68782efa49a16edaf07dc2cdaa855ea/raw/0c794a9717f18b094eabab2cd6a6b9a226903577/movies.csv> (откройте ссылку и используйте «Сохранить как», чтобы загрузить файл).

2. Загрузите набор данных фильмов в pandas.

3. Перечислите все столбцы набора данных и изучите их типы. Изучите статистику по различным областям. Опишите, какие данные у нас есть.

4. Сколько всего фильмов в наборе данных?

5. Сколько фильмов содержится в наборе данных за каждый год?

6. Покажите подробную информацию о наименее и наиболее прибыльных фильмах в наборе данных.

7. Значение «Жанр» временами кажется непоследовательным; попробуйте найти эти несоответствия и исправить их.

8. Сохраните (в новый файл CSV) 10 лучших комедий по количеству зрителей; покажите только название фильма, год и студию.

9. Используйте pip для установки двух библиотек: **lxml, MySQL-connector-python**.

**Python homework 7**

1.Прочтите этот <https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/data/2019/2019-10-08/ipf_lifts.csv>

Набор данных по пауэрлифтингу в pandas DataFrame.

2.Найдите рекорды каждого пола и каждого подразделения в каждом из 3 упражнений (жим лежа, приседания и становая тяга).

3.Посчитайте количество побед каждого участника, принимая за победу 1-е место. Сохраните результаты во вторичном DataFrame.

4.Используйте DataFrame из упражнения 3, чтобы показать для каждой комбинации подразделения и пола, участника с наибольшим количеством побед среди участников, когда-либо участвовавших в этом дивизионе.

5.Подключитесь к базе данных employees и выполните с помощью pandas последний запрос из финального проекта по SQL. Убедитесь, что результаты загружены в pandas DataFrame.

**Python homework 8**

*Рекомендуемые упражнения (не оцениваются)*1. Изучите галерею matplotlib <https://matplotlib.org/3.1.1/gallery/index.html> и галерею seaborn <https://seaborn.pydata.org/examples/index.html>, чтобы увидеть больше типов визуализации и другие возможности построения графиков.

Для следующих упражнений загрузите этот набор данных рейтинга кофе <https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/data/2020/2020-07-07/coffee_ratings.csv> и создайте визуализации, чтобы ответить на вопросы. Вы можете использовать  
дополнительные типы графиков к тем, которые мы узнали на уроках, если вы думаете, что они помогут ответить на вопросы. Вы можете найти описание набора данных здесь. <https://github.com/rfordatascience/tidytuesday/blob/master/data/2020/2020-07-07/readme.md>

2. Какие страны являются крупными экспортерами кофе?

3. Каковы корреляции между различными показателями оценки кофе?

4. Как (если есть) влияние цвета зерен на общий сорт кофе?

5. Влияет ли страна происхождения на качество кофе?

6. Существенно ли влияет высота на качество кофе?

**Python homework ADD**

**Дополнительные задания для практики по Python**1. Запросите два вещественных числа в качестве следующих переменных:  
- Длина стороны квадрата  
- Радиус круга  
Выведите «True», если окружность круга больше квадрата, иначе «False».

2. Джонни строит забор. В качестве входящих данных запросите – название желаемой формы («круг» или «квадрат») и длину забора Джонни. Выведите площадь, которую покрывает забор Джонни.

3. Запустите игру «камень-ножницы-бумага» против компьютера. Выведите победителя.

4. Запустите серию игр «камень, ножницы, бумага», пока пользователь не захочет выйти. Выведите статистику после каждой игры (количество побед компьютера, количество побед пользователя, количество ничьих).  
Когда пользователь уйдет, выведите окончательного победителя.

5. Получите от пользователя вещественные числа, пока он не захочет выйти. Когда он закончит ввод, выведите наименьшее число в последовательности. **Не используйте списки для этого упражнения**.

6. Получите от пользователя ввод вещественных чисел, пока он не захочет выйти. Когда он закончит ввод, выведите самое большое число, затем второе по величине число, затем третье по величине число. **Не используйте списки для этого упражнения.**

7. Выведите все простые числа от 1 до 100. Постарайтесь сделать вашу программу максимально эффективной, используя только те инструменты, которые мы изучили до сих пор. Например, для каждого числа, которое мы проверяем на простоту, нам нужно только проверить, делится ли оно на меньшие простые числа  
(поэтому нам не нужно проверять деление на 4, 6, 8, 9, 10, ...).

**SP**

**Python**

База данных IKEA.

1.Скачайте этот набор данных IKEA <https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/data/2020/2020-11-03/ikea.csv>.

2. Выполните исследовательский анализ для набора данных, включая описательную статистику и визуализации. Опишите результаты.

3. Основываясь на EDA и вашем здравом смысле, выберите две гипотезы, которые вы хотите проверить / проанализировать. Для каждой гипотезы перечислите нулевую гипотезу и другие возможные альтернативные гипотезы, разработайте тесты, чтобы различать их, и выполните их. Опишите результаты.

4. Обучите модель предсказывать цену на мебель.

- Укажите, какие столбцы не следует включать в модель и почему.

- Создайте конвейер перекрестной проверки для обучения и оценки модели, включая (при необходимости) такие шаги как, вменение пропущенных значений и нормализация.

- Предложите методы повышения производительности модели. Опишите результаты.