Оглавление

Глава 1

Язык

1.1 Описание языка

Предполагается, что свой скрипт пользователь сможет писать в программе Microsoft Excel. Скрипт состоит из набора лексем, каждая из которых записывается в отдельной ячейке. Предусмотрены лексемы следующих видов: переменная, присваивание, текстовая последовательность, название директивы, переменная цикла, границы цикла, элемент множества для цикла по множеству, конец цикла, слова, обозначающие начало и конец скрипта. Комментарии могут начинаться в любом месте. Пользователь должен сохранять скрипт в формате CSV (разделитель – точка с запятой). Этот формат делает достаточно простым написание скрипта как в Excel, так и в любом текстовом редакторе, воспринимающем формат CSV.

1.2 Директивы

В языке JACOBI 4 элементами сценария обработки информации являются директивы. Проще всего воспринимать директиву как команду, которая производит операции над файлами и записывает результат в новые файлы. Директиве могут требоваться дополнительные параметры.

Проверка правильности пользовательского скрипта может, в ряде случаев, показать заранее, что вызов директивы некорректен. Например, если директиве требуется три аргумента, а передано только два, или если нарушен жестко задекларированный порядок аргументов. В этом случае скрипт не будет запущен. Для его запуска потребуется сначала устранить все ошибки.

При использовании директивы в скрипте важно помнить следующие правила:

- имя директивы и все параметры должны располагаться в разных ячейках на одной строке;
- имя директивы должно быть расположено в самом начале;
- параметры директивы должны располагаться на одной строке с именем директивы;
- не допускается разбиение вызовов на несколько строк;
- порядок, в котором перечислены параметры, должен строго соответствовать документации;

- скрипт должен быть сохранён в формате .csv с разделителем ";".

Примеры использования директив:

transpose	matrixA.csv	out.csv

Продвинутому пользователю, вероятно, будет интересно узнать, что директивы представляют собой исполняемые файлы, которые могут быть запущены из командной строки. Исполняемые файлы можно найти в папке bin.

Основная идея написания сценария в том, чтобы исполнять последовательно несколько директив, передавая на вход последующей директиве результат работы предыдущей.

1.3 Комментарии

Комментарии обозначаются комбинацией "//". Все символы, начиная с символа комментария и до конца строки считаются комментарием и не рассматриваются как часть скрипта, т.е. не анализируются и не проверяются на синтаксическую корректность.

Пример строки с комментарием:

bootstrap	file1.csv	file2.csv//бутстреп

Подстрока "//бутстреп" и все последующие ячейки не являются частью скрипта и будут откинуты.

1.4 Начало/Конец

Программный комплекс JACOBI 4 создан для однотипной потоковой обработки данных. На практике это означает, что пользователь может обработать множество файлов с данными одним сценарием. Сценарий создается и отлаживается один раз, после чего переиспользуется. Блоки в языке JACOBI 4 являются удобным способом отладки. Они призваны ограничить "видимую" область скрипта, то есть по сути не обрабатывать команды вне блока.

Рассмотрим для начала простые примеры[1].

Пусть есть необходимость в предварительной обработке сырых данных. Допустим, есть несколько файлов с экспериментальными данными, сохраненными в табличном виде в формате .csv. Необходимо почистить их от нечисловых ячеек, транспонировать, нормализовать и прологарифмировать.

Для начала необходимо "отладить" сценарий обработки на каком-то одном из файлов. Пусть он называется raw.csv. Самый простой скрипт, который подойдёт для нашей цели, выглядит так:

[1]Примеры, приведенные в данном разделе, не несут никакой смысловой нагрузки с точки зрения обработки реальных данных и получения полезных результатов. Все примеры приведены для демонстрации языковых конструкций.

deleteAllRowsWithNotNumberValues	raw.csv	nums.csv	non.csv
transpose	nums.csv	transp.csv	
normalizeLength	transp.csv	norm.csv	
log	norm.csv	log.csv	2

Во время работы скрипта может обнаружиться, что что-то пошло не так, например, на этапе нормализации. Чтобы не исполнять действия, предшествующие нормализации, при отладке можно добавить блок “начало-конец”. Конечно, в данном случае, можно выполнить две первые директивы, но в реальных скриптах обычно довольно много шагов. Чтобы сэкономить время, достаточно начать с того шага, который не получается. Это будет выглядеть так:

deleteAllRowsWithNotNumberValues	raw.csv	nums.csv	non.csv
transpose	nums.csv	transp.csv	
НАЧАЛО			
normalizeLength	transp.csv	norm.csv	
log	norm.csv	log.csv	2
КОНЕЦ			

При исполнении скрипта все директивы, стоящие до ключевого слова НАЧАЛО и после КОНЕЦ, будут пропущены.

Блоков НАЧАЛО-КОНЕЦ может быть несколько. Главное, чтобы они не были вложенными.

Примеры использования блоков:

правильное использование

директива 1
директива 2
НАЧАЛО
директива 3
КОНЕЦ
директива 4
директива 5
директива 6
НАЧАЛО
директива 7
директива 8
КОНЕЦ
директива 9

неправильное использование

директива 1
КОНЕЦ
директива 2
НАЧАЛО
директива 3
НАЧАЛО
директива 4
НАЧАЛО
КОНЕЦ
директива 5
директива 6
НАЧАЛО
директива 7
КОНЕЦ
директива 8
КОНЕЦ
НАЧАЛО
директива 9

Вложенность блоков не предусмотрена ввиду отсутствия необходимости. Блок нужен для того, чтобы выделить, какая часть скрипта будет выполнена. Следует обратить внимание, что блоки не влияют на область видимости, поскольку всё, что

не входит в блоки, принимается за комментарий и отбрасывается на первом же этапе анализа скрипта.

По этой причине, если вне блоков НАЧАЛО-КОНЕЦ присутствуют ошибки, анализатор о них ничего не узнает и не выдаст ошибки. Это необходимо учитывать при отладке.

Вместо НАЧАЛО-КОНЕЦ можно также использовать BEGIN-END. Не будет ошибкой использование BEGIN-КОНЕЦ или НАЧАЛО-END, также значение не имеет регистр, поскольку ключевые слова проверяются после приведения к нижнему регистру. Рекомендуется использование одного языка и верхнего регистра, поскольку такие конструкции лучше видны в скриптах. Также рекомендуется оставлять пустые строки перед НАЧАЛО и после КОНЕЦ.

Ключевые слова НАЧАЛО и КОНЕЦ должны всегда стоять в начале строки и быть единственными функциональными словами в строке.

1.5 Переменные

В языке JACOBI 4 для удобства написания скриптов введены переменные. Поскольку в подавляющем большинстве случаев директивы требуют указания файлов в качестве входных и выходных параметров, длинные пути и имена файлов могут значительно снизить читаемость скрипта. Кроме того, копирование путей к файлам может привести к ошибкам. Стоит отметить, что для файлов, которые находятся непосредственно в рабочей директории, пути указывать не обязательно. При написании сценариев также существует проблема дублирования кода. Обычная ситуация, которая может произойти при дублировании - необходимые изменения внесены в одной части скрипта и не внесены в другой.

Эти проблемы призваны решить переменные. Поскольку для среднестатистического пользователя пакета определение типа переменной может показаться неочевидной задачей, все переменные было решено сделать строковыми. То есть по умолчанию считается, что любая переменная, которая определена пользователем, является строкой. Более того, даже аргументы директив, имеющие вид числа на первом этапе являются строкой. По этой причине пользователю не нужно беспокоиться заранее о типе переменной, однако необходимо помнить, что в процессе исполнения скрипта всё же могут быть выявлены проблемы.

Рассмотрим простой пример:

path	равно	data/in		
file	:=	work.csv		
c	equals	2		
transpose	≪path≫.csv	≪file≫		//data/in.csv
normalizeLength	≪file≫	n_≪file≫		//n_work.csv
log	n_≪file≫	l_≪file≫	≪c≫	//l_work.csv

При определении переменных никаких дополнительных символов не требуется. При использовании переменной необходимо заключить имя переменной в угловые скобки ≪≫.

Директива log требует число в качестве третьего параметра. Тип переменной в данном случае определяется во время исполнения, поэтому скрипт завершится с ошибкой, если переменная ≪c≫ не сможет быть преобразована к целому числу.

Если переменная используется в какой-то другой строке, то происходит конкатенация значения переменной со строкой (например, n_≪file≫ преобразуется в n_work.csv).

Для присвоения переменной можно использовать ":=", "равно", "assign", "equals".

Чтобы записать в переменную значение из файла нужно указать "from file" (или "из файла") и имя файла:

seed	:=	from file	next_seed.csv

При этом в пременную будет записано значение левого верхнего угла матрицы.

1.6 Циклы

В языке JACOBI 4 существует 3 вида циклов:

- n-цикл (положительное количество повторов);
- цикл по целочисленной переменной;
- цикл по множеству строчных значений.

Общие правила использования циклов:

- Циклы могут быть вложенными; поддерживается любой уровень вложенности.
- Переменные цикла разрешаются во время исполнения скрипта, а не во время проверки его корректности. Если значение переменной некорректно, скрипт будет остановлен с ошибкой, когда исполнение сценария дойдёт до этого шага.
- Несмотря на то, что области видимости переменных реализованы по аналогии с большинством процедурных языков, рекомендуется давать разные имена переменным цикла, поскольку одинаковые имена двух разных по смыслу переменных вносят путаницу и приводят к трудно отлавливаемым ошибкам. Общее правило: новый цикл - новая переменная цикла.
- Переменные (в частности переменные цикла) "видны" в том блоке, в котором они определены, после момента определения и во всех вложенных блоках, которые расположены после определения переменной.
- Циклы не могут пересекаться с блоками BEGIN-END. Нельзя включать ключевые слова BEGIN и END в тело цикла.

1.6.1. n-цикл (положительное количество повторов)

Самый простой цикл, который можно использовать в языке JACOBI 4 - это цикл, который повторяется n раз.

Рассмотрим житейский пример, когда необходимо к заданному столбцу присоединить сто его бутстреп-копий. Допустим, файл in.csv содержит исходный столбец

значений.

copy	in.csv	result.csv	
Loop	100	times	
bootstrap	in.csv	b_copy.csv	
appendRight	result.csv	b_copy.csv	result.csv
end loop			

В результате исполнения такого скрипта получим файл result.csv, содержаший исходный столбец и сто его бутстреп-копий.

1.6.2. Цикл по целочисленной переменной

Цикл по целочисленной переменной подразумевает возможность воспользоваться переменной цикла.

Рассмотрим пример:

Loop	for	i	from	1	to	4	step	1
transpose	in≪i≫.csv	in≪i≫_t.csv						
end loop								

В этом примере переменная i будет принимать значения 1, 2, 3, 4. Отсчёт значений начнется с 1 (from | 1) и продолжится до 4 включительно (to | 4), на каждом шаге увеличивая значение на 1 (step | 1).

Приведенный цикл эквивалентен следующему скрипту:

transpose	in1.csv	in1_t.csv
transpose	in2.csv	in2_t.csv
transpose	in3.csv	in3_t.csv
transpose	in4.csv	in4_t.csv

Для переменных цикла справедливы те же правила использования, что и для обычных переменных. Область видимости такой переменной ограничена блоком цикла. То есть вне цикла она не видна.

1.6.3. Цикл по множеству строчных значений

Цикл по множеству - это такой цикл, где переменная пробегает значения из множества значений, указанных после ключевого слова from. Использование переменных цикла аналогично использованию обычных переменных.

Пример такого цикла:

Loop	for	i	from	in1	in2	matrA
transpose	≪i≫.csv	≪i≫_t.csv				
end loop						

В результате работы этого скрипта будут созданы файлы in1_t.csv, in2_t.csv, matrA_t.csv.

1.7 Файл описания директив

При каждом запуске скрипта на исполнение, диспетчер загружает информацию о доступных директивах. Эти данные используются для проверки корректности параметров при вызове модулей комплекса, а также для сопоставления имени вызываемой директивы исполняемому файлу, который будет запущен.

Поиск файлов описаний производится сначала в папке bin, затем в папке, содержащей bin[2]. Имя файла описания модулей комплекса должено иметь префикс “list” и расширение “csv” или “txt”. Примеры подходящих имен: list.csv, LIST.ru.TXT, list_user.CSV.

Файл описания директив содержит таблицу в формате CSV, в которой каждая не пустая строка является описанием модуля комплекса. Пример описания модулей:

internal	module	impl	string	integer:positive
internal	module2	impl2	string	string:optional
internal	echo	echo	any	
external	my_module	path/to/exe	count	double
subroutine	my_script	path/to/script.csv	integer	string
alias	модуль	module		

Значение первого столбца является типом записи. Допустимы следующие значения типов записей:

1. internal - внутренний модуль; путь до исполняемого файла должен быть указан относительно текущего файла описания.

2. external - внешний модуль; рекомендуется указывать абсолютный путь до исполняемого файла, потому что относительный путь будет разрешён по отношению к текущей рабочей директории, которую задает пользователь.

3. subroutine - скрипт на языке JACOBI 4; рекомендуется указывать путь относительно текущего файла описания.

4. alias - специальный тип, позволяющий указать альтернативное имя для ранее описанного модуля.

Второй столбец содержит имя модуля, которое можно использовать в скрипте. Каждый модуль должен иметь уникальное имя без учёта регистра. При возникновении повторов будет использовано первое описание.

Для типа alias третий столбец содержит имя ранее описанного модуля, для которого добавляется альтернативное имя. Разрешение ссылок на модули осуществляется после загрузки всех описаний из текущей директории.

Для типов internal, external и subroutine третий столбец содержит путь до исполняемого файла (или скрипта); четвертый и все последующие содержат список типов аргументов, которые принимает модуль.

Поддерживаемые типы аргументов: string, integer, double, count, any. Типы string, integer, double указывают на строку, целое и вещественное значение соответственно.

[2]Расположение указано относительно директории, содержащей комплекс. Например, если комплекс расположен в D:\tools\JACOBI4, тогда первой папкой поиска будет D:\tools\JACOBI4\bin, а второй - D:\tools\JACOBI4

count используется для передачи вектора значений следующего за count типа. any указывает на произвольный список аргументов.

Для каждого типа можно указать дополнительные атрибуты после символа ":". Для всех типов можно указать атрибут optional, обозначающий необязательный параметр. Для типов integer и double можно указать атрибут nonnegative, обозначающий, что значение параметра должно быть больше или равно нулю.

Глава 2

Модули

2.1 Многомерный анализ

2.1.1. PCA (Метод главных компонент)

Описание

Метод главных компонент используется для понижения размерности входных данных с наименьшими потерями информации.

В основе алгоритма лежит сингулярное разложение входной матрицы:

$$X = USV^T,$$

где S – неотрицательно определенная диагональная матрица сингулярных чисел; U, V – ортонормированные матрицы.

Связь между PCA и SVD определяется следующими соотношениями:

$$T = US, \quad P = V,$$

где T – матрица счетов – дает проекции исходных образцов на подпространство главных компонент, таким образом, строки матрицы T – это координаты образцов в новой системе координат, а столбцы – ортогональны и представляют проекции всех образцов на одну новую ось; P – матрица нагрузок – матрица перехода из исходного пространства переменных в пространство главных компонент.

Параметры директивы

→ Имя входного файла, содержащего матрицу X.

← Имя выходного файла для матрицы счетов T.

← Имя выходного файла для матрицы нагрузок P.

← Имя выходного файла для диагональной матрицы сингулярных чисел из разложения матрицы X.

← (опционально) Имя выходного файла для вектора собственных чисел из разложения матрицы X.

Выход

Файлы, содержащие таблицу счетов, нагрузок, диагональную матрицу сингулярных чисел.

Вызов из командной строки

Вызов директивы из командной строки для файла in.csv:
pca in.csv t.csv p.csv s.csv
pca in.csv t.csv p.csv s.csv vs.csv

Вызов из пользовательского скрипта

НАЧАЛО					
pca	in.csv	t.csv	p.csv	s.csv	
pca	in.csv	t.csv	p.csv	s.csv	vs.csv
КОНЕЦ					

2.1.2. PCO (Метод главных координат)

Описание

Метод главных координат используется для получения матрицы главных компонент по матрице евклидовых расстояний между объектами.

Реализован классический алгоритм Гауэра (Gower, 1966):

1. Расстояния возводятся в квадрат, применяется двойное центрирование и умножение на $-\frac{1}{2}$.
2. Методом SVD вычисляются матрица собственных векторов и диагональная матрица сингулярных чисел.
3. Отрицательные сингулярные числа обнуляются. Из сингулярных чисел извлекаются квадратные корни.
4. Матрица главных компонент получается умножением матрицы собственных векторов на диагональную матрицу корней из сингулярных чисел (стандартных отклонений главных компонент). Сами сингулярные числа являются дисперсиями главных компонент.

Параметры директивы

→ Имя входного файла, содержащего таблицу евклидовых расстояний между объектами.

← Имя выходного файла для матрицы главных компонент.

← Имя выходного файла для диагональная матрица дисперсий главных компонент.

← (опционально) Имя выходного файла для вектора стандартных отклонений главных компонент.

Выход

Файл, содержащий таблицу координат объектов.

Вызов из командной строки

Для входного файла in.csv:
pco in.csv out.csv s.csv
pco in.csv out.csv s.csv vs.csv

Вызов из пользовательского скрипта

НАЧАЛО				
pco	in.csv	out.csv	s.csv	
pco	in.csv	out.csv	s.csv	vs.csv
КОНЕЦ				

2.1.3. SVD (Сингулярное разложение матрицы)

Описание

Директива производит сингулярное разложение матрицы:

$$X = USV^T,$$

где S – положительно определенная диагональная матрица сингулярных чисел; U, V – ортонормированные матрицы.

Параметры директивы

→ Имя входного файла.

← Имя выходного файла для матрицы U.

← Имя выходного файла для диагональной матрицы сингулярных чисел S.

← Имя выходного файла для матрицы V.

← (опционально) Имя выходного файла для вектора собственных чисел $\Lambda = S^2$.

Выход

Файлы, содержащие сингулярное разложение.

Вызов из командной строки

svd in.csv u.csv s.csv v.csv
svd in.csv u.csv s.csv v.csv vs.csv

Вызов из пользовательского скрипта

НАЧАЛО					
svd	in.csv	u.csv	s.csv	v.csv	
svd	in.csv	u.csv	s.csv	v.csv	vs.csv
КОНЕЦ					

2.1.4. LDA (Линейный дискриминантный анализ)

Описание

Линейный дискриминантный анализ используется для определения подпространства, в котором достигается максимум различия выборок по критерию Фишера.

Параметры директивы

→ Файл, содержащий столбцы-признаки.

→ Файл, содержащий столбец с группой для каждого объекта.

→ Файл, содержащий столбцы с тестовыми объектами.

← Имя выходного файла.

Выход

В выходном файле каждому тестовому объекту соответствует столбец, содержащий идентификатор группы, к которому этот объект должен принадлежать, и значение дискриминантной функции для каждой группы.

Вызов из командной строки

Для входного файла in.csv, данными о группе из файла group.csv и тестовым набором объектов из файла test.csv:

lda in.csv group.csv test.csv out.csv

Вызов из пользовательского скрипта

НАЧАЛО				
lda	in.csv	group.csv	test.csv	out.csv
КОНЕЦ				

2.1.5. MLR (Множественная линейная регрессия)

Описание

Используется для анализа связи между несколькими независимыми и одной зависимой переменной, при этом предполагается, что связь между вектором откликов y и матрицей предикторов X линейная:

$$y = Xa + \varepsilon.$$

Цель регрессионного анализа – найти оценки неизвестных коэффициентов a, которые минимизируют сумму квадратов остатков, что достигается при

$$a = (X^T X)^{-1} X^T y.$$

Замечание: в случае необратимости матрицы $(X^T X)^{-1}$ следует применить метод главных компонент ко входной матрице и удалить столбцы с нулевой и достаточно малой дисперсией.

Параметры директивы

→ Файл, содержащий таблицу предикторов X.

→ Файл, содержащий зависимую переменную.

← Имя выходного файла.

Выход

Значение вектора a.

Вызов из командной строки

Для входных файлов xs.csv и ys.csv:
regression xs.csv ys.csv out.csv

Вызов из пользовательского скрипта

НАЧАЛО			
regression	xs.csv	ys.csv	out.csv
КОНЕЦ			

2.1.6. NMDS (Неметрическое многомерное шкалирование)

Описание

Позволяет по матрице рангов мер сходства-различия между объектами сопоставить координаты этим объектам в евклидовом пространстве заданной размерности.

Для реализации используется алгоритм Тагучи—Оно[1].

Критерием остановки является выполнение одного из требований:

1. Достигнут заданный пользователем коэффициент корреляции Спирмена.
2. Совершено заданное количество итераций без улучшения решения.

Начальное значение шага равно $max(0.1/n^3, 10^{-7})$, где n - размерность матрицы расстояний; коэффициент изменения шага, если не произошло улучшения решения, равен 0.9.

Параметры директивы

→ Имя входного файла, содержащего таблицу расстояний.

← Имя выходного файла.

→ Размерность решения.

→ Требуемый коэффициент корреляции Спирмена.

→ Количество запусков алгоритма.

→ (опционально) Максимальное количество итераций без улучшения результата, по умолчанию равно 90.

Выход

Файл, содержащий таблицу координат объектов.

Вызов из командной строки

Для входного файла in.csv:
nmds in.csv out.csv 4 0.99 10 90

Вызов из пользовательского скрипта

НАЧАЛО						
nmds	in.csv	out.csv	4	0.99	10	
nmds	in.csv	out.csv	4	0.99	10	90
КОНЕЦ						

[1] Y.-h. Taguchi, Y. Oono; Relational patterns of gene expression via non-metric multidimensional scaling analysis, Bioinformatics, Volume 21, Issue 6, 15 March 2005, Pages 730–740, https://doi.org/10.1093/bioinformatics/bti067

2.1.7. 2B-PLS

Описание

Цель 2B-PLS[2] состоит в нахождении пар осей для каждого блока, выражающего наибольший шаблон ковариации между осями.

Параметры директивы

→ Имя первого входного файла.

→ Имя второго входного файла.

← Имя выходного файла для таблицы нагрузок первой матрицы (F_1).

← Имя выходного файла для диагональной матрицы сингулярных чисел (D).

← Имя выходного файла для таблицы нагрузок второй матрицы (F_2).

← Имя выходного файла для таблицы счетов первой матрицы (Z_1).

← Имя выходного файла для таблицы счетов второй матрицы (Z_2).

← (опционально) Имя выходного файла для вектора собственных чисел.

Выход

Файлы F_1, F_2, D, Z_1, Z_2.

Вызов из командной строки

Для входных файлов a.csv и b.csv:
2B-PLS a.csv b.csv f1.csv d.csv f2.csv z1.csv z2.csv

Вызов из пользовательского скрипта

НАЧАЛО							
2B-PLS	a.csv	b.csv	f1.csv	d.csv	f2.csv	z1.csv	z2.csv
КОНЕЦ							

[2]Rohlf, F. J., Corti, M. (2000). The use of two-block partial least-squares to study covariation in shape. Systematic Biology, 49(4), 740–753. doi: 10.1080/106351500750049806.

2.1.8. PLS регрессия

Описание

Цель PLS регрессии состоит в нахождении связей между матрицами X и Y. PLS модель:

$$X = TP^T + E$$
$$Y = UQ^T + F,$$

где X - матрица предикторов, Y - матрица откликов, T и U - матрицы счетов для X и Y, P и Q - матрицы нагрузок для X и Y, E и F - матрицы ошибок. Разложение X и Y производятся так, чтобы максимизировать ковариацию между T и U.

Параметры директивы

→ Имя входного файла с матрицой предикторов X.

→ Имя входного файла с матрицей откликов Y.

→ Число факторов n (для автоматического использования максимального числа факторов следует взять $n = 0$).

← Имя выходного файла для таблицы нагрузок матрицы X (P).

← Имя выходного файла для таблицы нагрузок матрицы Y (Q).

← Имя выходного файла для таблицы счетов матрицы X (T).

← Имя выходного файла для таблицы счетов матрицы Y (U).

Выход

Файлы с таблицами P, Q, T, U.

Вызов из командной строки

Для входных файлов a.csv и b.csv:
pls_regression x.csv y.csv 0 p.csv q.csv t.csv u.csv

Вызов из пользовательского скрипта

НАЧАЛО							
pls_regression	x.csv	y.csv	0	p.csv	q.csv	t.csv	u.csv
КОНЕЦ							

2.1.9. NNBP (нейронные сети: алгоритм обратного распространения ошибки)

Описание

Директива реализует алгоритм обратного распространения ошибки. Для этого используется библиотека FANN 2.2.0 (Fast Artificial Neural Network Library), в которой реализован алгоритм iRPROP- [Igel and Husken, 2000]. Обучение нейронной сети производится на основе данных входного файла (in.csv) и данных о выходных признаках (props.csv). Обученная нейронная сеть определяет признаки объектов, записанных в файле с тестовыми данными (test.csv), на основе проведенного обучения. Результат работы записывается в выходной файл (out.csv).

Для оценки достоверности результата следующая процедура выполняется B раз:

1. Формируется обучающая (train) и тестовая (test) выборки объектов на основе входного файла, для которого известны выходные признаки объектов. Формируется обучающая выборка train случайным выбором объектов. Её объем составляет 80% от исходной. В выборку test попадают все объекты, которые не были использованы в выборке train.

2. Производится обучение нейронной сети и формирование признаков тестовых объектов.

3. В файл статистики (stat.csv) записывается сумма квадратов отклонений полученного результата от правильного.

Параметры нейронной сети

Количество нейронов во входном слое: N (количество признаков у объектов равно числу строк входной матрицы)

Количество нейронов в выходном слое: M (количество объектов равно числу столбцов входной матрицы)

Функция активации нейронов: сигмоид

Критерий остановки тренировки: достижение порога среднеквадратичной ошибки (MSE)

Порог среднеквадратичной ошибки: 0.01

Максимальное количество эпох: 3000

Параметры директивы

→ Имя тренировочного файла (in.csv, объекты - строки).

→ Имя файла с информацией о выходных признаках объектов[3] (groups.csv)

→ Имя файла с тестовыми данными (test.csv, объекты - строки).

← Имя выходного файла (out.csv, объекты - строки).

← Имя файла для записи статистики (stat.csv).

[3]Здесь объекты - строки. Для создания файла groups.csv из вектора пар объект-признак можно использовать директиву “Преобразовать вектор пар объект-признак в матрицу признаков”.

→ Количество скрытых слоев R.

→ Количество нейронов на первом скрытом слое.

→ ...

→ Количество нейронов на R-том скрытом слое.

→ Число итераций B оценки достоверности результата.

Выход

Файл, содержащий результат классификации и файл со статистикой классификации объектов на основе тренировочных данных.

Вызов из командной строки

nnbp in.csv groups.csv test.csv out.csv stat.csv 3 15 10 5 10

Вызов из пользовательского скрипта

НАЧАЛО										
nnbp	in.csv	grp.csv	t.csv	o.csv	s.csv	3	15	10	5	10
КОНЕЦ										

Пример входных файлов

in.csv

XOR	in1	in2
ex1	0	0
ex2	0	1
ex3	1	0
ex4	1	1

groups.csv

obj/grp	-1	1
ex1	1	0
ex2	0	1
ex3	0	1
ex4	1	0
ex5	0	1
ex6	1	0

test.csv

XOR	in1	in2
ex1	0	0
ex2	0	1
ex3	1	0
ex4	1	1
ex5	0	1
ex6	0	0

Пример выходных файлов

out.csv

prediction: obj/grp	-1	1
ex1	0.998533	0.00096333
ex2	-0.000666678	0.995292
ex3	-0.000658721	0.995292
ex4	0.995145	0.935217
ex5	-0.000666678	0.995292
ex6	0.998533	0.00096333

stat.csv

bootstrep number	error
1	0.97826
2	6.23753
3	5.16965
4	7.11149
5	0.874654
6	1.98636
7	1.52321
8	1.99934
9	1.97571
10	4.84229

2.1.10. Active

Описание

Модуль производит оценку полезности от -1 до 1 с порогом 0 для детального регрессионного анализа двух наборов данных равных объемов.

Если указано два входных файла, то вычисление будет произведено для каждого столба первого входного файла с каждым столбом второго входного файла.

Параметры директивы

→ Имя первого входного файла.

→ (опционально) Имя второго входного файла.

← Имя выходного файла.

→ Число повторений.

Вызов из командной строки

Для входного файла in.csv:
act in.csv out.csv 20

Вызов из пользовательского скрипта

НАЧАЛО				
act	in.csv	out.csv	20	
act	in1.csv	int2.csv	out.csv	20
КОНЕЦ				

2.1.11. Join

Описание

Модуль производит разбиение на классы множества входных объектов исходя из расстояний между ними.

Параметры директивы

→ Имя входного файла.

→ Ключевое слово similarity\dissimilarity.

← Имя выходного файла для файла с булевой матрицей.

← Имя выходного файла для файла с матрицей объект-класс.

← Имя выходного файла для файла с квадратной матрицей сходства\различия.

Вызов из командной строки

Для входного файла in.csv:
join inFile.csv dissimilarity ouBool.csv outObj.csv outClass.csv

Вызов из пользовательского скрипта

НАЧАЛО					
join	in.csv	dissimilarity	bool.csv	obj.csv	class.csv
КОНЕЦ					

2.1.12. Алгоритм факторизации неотрицательных матриц (Lee-Seung)

Описание

Алгоритм[4] позволяет получить для неотрицательной матрицы A такие матрицы W и H, что $A \sim WH$, где число столбцов W может быть намного меньше числа столбцов A.

Параметры директивы

→ Файл, содержащий матрицу A.

→ Целое положительное число, задающее размерности W и H. W - $n \times r$, H - $r \times m$ матрицы.

← Имя выходного файла для W.

← Имя выходного файла для H.

Вызов из командной строки

Вызов директивы из командной строки для файла in.csv:
nnmf in.csv 10 w.csv h.csv

Вызов из пользовательского скрипта

НАЧАЛО				
nnmf	in.csv	10	w.csv	h.csv
КОНЕЦ				

[4]Lee D. D, Seung H. S. Learning the parts of objects by non-negative matrix factorization // Nature, 1999. - № 401. - P. 788–791. doi:10.1038/44565

2.1.13. Ускоренный алгоритм факторизации неотрицательных матриц (Lee-Seung)

Описание

Алгоритм[5] позволяет получить для неотрицательной матрицы A такие матрицы W и H, что $A \sim WH$, где число столбцов W может быть намного меньше числа столбцов A.

Параметры директивы

→ Файл, содержащий матрицу A.

→ Целое положительное число, задающее размерности W и H. W - $n \times r$, H - $r \times m$ матрицы.

← Имя выходного файла для W.

← Имя выходного файла для H.

Вызов из командной строки

Вызов директивы из командной строки для файла in.csv:
nnmf_sum in.csv 10 w.csv h.csv

Вызов из пользовательского скрипта

НАЧАЛО				
nnmf_sum	in.csv	10	w.csv	h.csv
КОНЕЦ				

[5]Gonzalez E. F., Zhang Y. Accelerating the Lee-Seung Algorithm for Nonnegative Matrix Factorization // Department of Computational and Applied Mathematics Rice University, Houston, Texas 2005. - 13 p.

2.1.14. Метод ближайшего соседа (метод одиночной связи)

Описание

В данном алгоритме[6] измерение расстояния между кластерами производится следующим образом:

$$\rho_{st} = \min_{i \in K_s, j \in K_t} l_{ij} \quad (s = 1, 2, \ldots, p, t = 1, 2, \ldots, p)$$

Параметры директивы

→ имя входного файла.

← имя выходного файла, который будет содержать дерево объединения в формате *.nwk.

← имя выходного файла.

Вызов из командной строки

Вызов директивы из командной строки для файла in.csv:
single_linkage in.csv tree.nwk out.csv

Вызов из пользовательского скрипта

НАЧАЛО			
single_linkage	in.csv	tree.nwk	out.csv
КОНЕЦ			

[6]Калинина В.Н., Соловьев В.И. Введение в многомерный статистический анализ: Учебное пособие / ГУУ .- М., 2003. (стр. 40)

2.2 Метрики

2.2.1. Евклидова метрика

Описание

Директива производит вычисление евклидовых расстояний между строками матрицы:

$$d_{jk} = \sqrt{\sum_{i=1}^{I}(x_{ji} - x_{ki})^2}$$

Если x_{ji} или x_{ki} не содержит значения, то используется среднее между суммой квадратов имеющихся элементов.

Если указано два входных файла, то будут вычислены расстояния для каждой строки из первого файла с каждой строкой из второго файла.

Параметры директивы

→ Имя входного файла.

→ (опционально) Имя второго входного файла.

← Имя выходного файла.

Выход

Таблица евклидовых расстояний.

Вызов из командной строки

euclidean_metric in.csv out.csv
euclidean_metric in.csv in2.csv out.csv

Вызов из пользовательского скрипта

НАЧАЛО			
euclidean_metric	in.csv	out.csv	
euclidean_metric	in.csv	in2.csv	out.csv
КОНЕЦ			

2.2.2. Метрика Минковского

Описание

Директива производит вычисление расстояний по метрике минковского между строками матрицы:

$$d_{jk} = \sqrt[r]{\sum_{i=1}^{I}(x_{ji} - x_{ki})^r}$$

Если x_{ji} или x_{ki} не содержит значения, то используется среднее между суммой степеней имеющихся элементов.

Если указано два входных файла, то будут вычислены расстояния для каждой строки из первого файла с каждой строкой из второго файла.

Параметры директивы

→ Имя входного файла.

→ (опционально) Имя второго входного файла.

← Имя выходного файла.

→ Параметр r.

Выход

Таблица расстояний по метрике минковского.

Вызов из командной строки

minkowski_metric in.csv out.csv 4
minkowski_metric in.csv in2.csv out.csv

Вызов из пользовательского скрипта

НАЧАЛО				
minkowski_metric	in.csv	out.csv	4	
minkowski_metric	in.csv	in2.csv	out.csv	4
КОНЕЦ				

2.2.3. Коэффициент Жаккара

Описание

Директива производит вычисление бинарной меры сходства по строкам матрицы:

$$K_J = \frac{c}{a + b - c}$$

где a - количество не нулевых элементов в первом векторе, b - количество не нулевых элементов во втором векторе, c - количество общих элементов в первом и втором векторе.

Если указано два входных файла, то будут вычислены коэффициенты для каждой строки из первого файла с каждой строкой из второго файла.

Параметры директивы

→ Имя входного файла.

→ (опционально) Имя второго входного файла.

← Имя выходного файла.

Выход

Таблица с коэффициентами Жаккара.

Вызов из командной строки

jaccard in.csv out.csv
jaccard in.csv in2.csv out.csv

Вызов из пользовательского скрипта

НАЧАЛО			
jaccard	in.csv	out.csv	
jaccard	in.csv	in2.csv	out.csv
КОНЕЦ			

2.2.4. Коэффициент Жаккара-Наумова

Описание

Директива производит вычисление индекса Жаккара-Наумова[7] по строкам матрицы, в котором расстояние между двумя вектор-строками x_i и y_j вычисляется как:

$$K_{ij} = \frac{\sum_{l=1}^{L} \min(x_{il}, y_{jl})}{\sum_{l=1}^{L} \max(x_{il}, y_{jl})}$$

где L - число элементов в векторе, K_{ij} - значение элемента матрицы с индексом ij.

Если указано два входных файла, то будут вычислены коэффициенты для каждой строки из первого файла с каждой строкой из второго файла.

Параметры директивы

→ Имя входного файла.

→ (опционально) Имя второго входного файла.

← Имя выходного файла.

Выход

Таблица с коэффициентами Жаккара-Наумова.

Вызов из командной строки

jaccardNaumov in.csv out.csv
jaccardNaumov in.csv in2.csv out.csv

Вызов из пользовательского скрипта

НАЧАЛО			
jaccardSteinhaus	in.csv	out.csv	
jaccardSteinhaus	in.csv	in2.csv	out.csv
jaccardRuzicka	in.csv	out.csv	
jaccardRuzicka	in.csv	in2.csv	out.csv
jaccardNaumov	in.csv	out.csv	
jaccardNaumov	in.csv	in2.csv	out.csv
КОНЕЦ			

[7] Также известный как индекс Жаккара-Рюжечки, Жаккара-Штайнхауса.

2.2.5. Расстояние Хэмминга

Описание

Директива производит вычисление расстояния Хэмминга по строкам матрицы. Расстояние между строками определяется как число позиций, в которых соответствующие элементы строк не совпадают.

Если указано два входных файла, то будут вычислены расстояния для каждой строки из первого файла с каждой строкой из второго файла.

Параметры директивы

→ Имя входного файла.

→ (опционально) Имя второго входного файла.

← Имя выходного файла.

Выход

Таблица с расстояниями Хэмминга.

Вызов из командной строки

hamming in.csv out.csv
hamming in.csv in2.csv out.csv

Вызов из пользовательского скрипта

НАЧАЛО			
hamming	in.csv	out.csv	
hamming	in.csv	in2.csv	out.csv
КОНЕЦ			

2.2.6. Манхэттенское расстояние

Описание

Директива производит вычисление манхэттенского расстояния по строкам матрицы:

$$d_{jk} = \sum_{i=1}^{I} |x_{ji} - x_{ki}|$$

Если x_{ji} или x_{ki} не содержит значения, то используется среднее между суммой модулей имеющихся элементов.

Если указано два входных файла, то будут вычислены расстояния для каждой строки из первого файла с каждой строкой из второго файла.

Параметры директивы

→ Имя входного файла.

→ (опционально) Имя второго входного файла.

← Имя выходного файла.

Выход

Таблица с расстояниями по манхэттенской метрике.

Вызов из командной строки

manhattan in.csv out.csv
manhattan in.csv in2.csv out.csv

Вызов из пользовательского скрипта

НАЧАЛО			
manhattan	in.csv	out.csv	
manhattan	in.csv	in2.csv	out.csv
КОНЕЦ			

2.2.7. p-distance

Описание

Директива производит вычисление расстояний между генетическими последовательностями по строкам таблицы. Расстояние между строками определяется как число позиций, в которых соответствующие элементы строки не совпадают, нормированных на длину вектора.

Параметры директивы

→ Имя входного файла.

← Имя выходного файла.

Выход

Таблица с расстояниями.

Вызов из командной строки

p_distance in.csv out.csv

Вызов из пользовательского скрипта

НАЧАЛО		
p_distance	in.csv	out.csv
КОНЕЦ		

Пример входного файла

in.csv

seq1	CAGACAGTCC
seq2	CACACTGCCA

out.csv

	o1	o2
o1	0	0.4
o2	0.4	0

2.2.8. Расстояние Джукса-Кантора

Описание

Директива производит вычисление расстояний между генетическими последовательностями по строкам таблицы. Расстояние Джукса-Кантора похоже на p-distance, но учитывает вероятность инверсий:

$$d = -\frac{3}{4}\ln(1 - \frac{4}{3}p)$$

Параметры директивы

→ Имя входного файла.

← Имя выходного файла.

Выход

Таблица с расстояниями.

Вызов из командной строки

jukes_cantor in.csv out.csv

Вызов из пользовательского скрипта

НАЧАЛО		
jukes_cantor	in.csv	out.csv
КОНЕЦ		

Пример входного файла

in.csv

seq1	CAGACAGTCC
seq2	CACACTGCCA

out.csv

	o1	o2
o1	0	0.571605039035173
o2	0.571605039035173	0

2.2.9. Расстояние Кимуры

Описание

Директива производит вычисление расстояний между генетическими последовательностями по строкам таблицы. Расстояние Кимуры похоже на Расстояние Джукса-Кантора, но учитывает вероятности нуклеотидных транзиций и трансверсий:

$$d = -\frac{1}{2}\ln(1 - 2P - Q) - \frac{1}{4}\ln(1 - 2Q),$$

где P - наблюдаемая часть транзиций, Q - наблюдаемая часть трансверсий.

Параметры директивы

→ Имя входного файла.

← Имя выходного файла.

Выход

Таблица с расстояниями.

Вызов из командной строки

kimura_distance in.csv out.csv

Вызов из пользовательского скрипта

НАЧАЛО		
kimura_distance	in.csv	out.csv
КОНЕЦ		

Пример входного файла

in.csv

seq1	CAGACAGTCC
seq2	CACACTGCCA

out.csv

	o1	o2
o1	0	0.575646273248511
o2	0.575646273248511	0

2.2.10. Объединение спектров

Описание

Директива производит объединение спектров — множество пар столбцов координата, интенсивность сигнала — разной длинны из набора входных файлов формата CSV.

Шаг сетки h и полуширина окна w вычисляются из входных параметров:

$$h = \frac{X_{end} - X_{begin}}{N}$$
$$w_i = \left(K_s + i * \frac{K_e - K_s}{N} \right) * h$$

где i - номер узла, K_s - начальный множитель полуширины окна, K_e - конечный множитель полуширины окна.

Минимальное значение X_{min} и максимальное значение X_{max} x-координаты каждого спектра должны удовлетворять условиям:

$$X_{begin} < X_{min} - w_0; \qquad X_{end} > X_{max} + w_0;$$
$$X_{begin} < X_{min} - w_{N-1}; \qquad X_{end} > X_{max} + w_{N-1}.$$

Алгоритм

Для всех спектров задаются начальное и конечное значение x-координаты (X_{begin}, X_{end}). Для каждого узла сетки i находятся все x-координаты x_j в интервале $[ih - w_i$, $ih + w_i]$ с ненулевыми интенсивностями сигнала y_j. По каждому сигналу вычисляется его влияние на узел i по формуле:

$$z(i, x_j) = y_j * f\left(\frac{|ih - x_j|}{w_i} \right);$$
$$f(x) = 1 - 3x^2 + 2x^3.$$

Суммарное влияние z_i на узел i вычисляется по формуле $z_i = S\{z(i, x_j)\}$, по всем x_j, попавшим в окно $[ih - w_i, ih + w_i]$, где S – способ объединения (сумма, усреднение, максимум).

Результат работы алгоритма: матрица вектор-столбцов z одинаковой длины для всех спектров в формате *.csv, дополненная номерами узлов сетки в первом столбце и заголовками столбцов в первой строке.

Параметры директивы

→ Количество входных файлов F.

→ Имя входного файла 1.

→ ...

→ Имя входного файла F.

← Имя выходного файла.

→ X_{begin}.

→ X_{end}.

→ Количество узлов сетки N.

→ Множитель K_s начальной полуширины окна w_i.

→ (опционально) Множитель K_e конечной полуширины окна w_i. Если параметр не задан, то $K_e = K_s$.

→ (опционально) Функция объединения: average, sum, max. Если параметр не задан, то используется average.

Выход

Таблица объединенных спектров.

Вызов из командной строки

spectrumDistance 1 in.csv out.csv 1000 3000 10 1

Вызов из пользовательского скрипта

НАЧАЛО									
spectrumDistance	1	i.csv	o.csv	1000	3000	10	1		
spectrumDistance	2	i.csv	i2.csv	o.csv	1000	3000	10	1	
spectrumDistance	2	i.csv	i2.csv	o.csv	1000	3000	10	1	max
КОНЕЦ									

Пример использования

Для входного файла in.csv:

in.csv

CON	spectrum1	vals	spectrum2	vals
1	2536.6	1012.14	1536.26	1422.06
2	2610.6	1277.94	1596.51	1348.26
3			1680.86	1440.43
4			1745.31	1493.17

При запуске директивы с параметрами:

spectrumDistance	1	in.csv	out.csv	1000	3000	10	1

Будут вычислены параметры h и w:

$$h = 200,$$
$$w = 200.$$

Выходной файл out.csv будет иметь вид:

out.csv

in.csv	spectrum1	spectrum2
1000	0	0
1200	0	0
1400	0	171.233496556762
1600	0	906.552348547181
1800	0	867.621806348876
2000	0	0
2200	0	0
2400	240.643343664361	0
2600	1019.5239838512	0
2800	10.38868863324	0

2.2.11. Тест Мантеля (Mantel test)

Описание

Тест Мантеля используется для вычисления корреляции между двумя матрицами.

Параметры директивы

→ Имя входного файла, содержащего таблицу A.

→ Имя входного файла, содержащего таблицу B.

← Имя выходного файла.

→ Количество итераций N.

Выход

Файл, содержащий N, z, p-value и сумму квадратов отклонений для каждого элемента матрицы, следующей формы:

variable	N	z	p-value	square error
value				

Вызов из командной строки

Для входных файлов A.csv, B.csv:
mantel_test A.csv B.csv out.csv 1000000

Вызов из пользовательского скрипта

НАЧАЛО				
mantel_test	A.csv	B.csv	out.csv	1000000
КОНЕЦ				

2.2.12. Ранговый тест Мантеля (Rank Mantel test)

Описание

Ранговый тест Мантеля используется для вычисления корреляции между двумя матрицами.

Параметры директивы

→ Имя входного файла, содержащего таблицу A.

→ Имя входного файла, содержащего таблицу B.

← Имя выходного файла.

→ Количество итераций N.

Выход

Файл, содержащий N, z, p-value и сумму квадратов отклонений для каждого элемента матрицы, следующей формы:

variable	N	z	p-value	square error
value				

Вызов из командной строки

Для входных файлов A.csv, B.csv:
rank_mantel_test A.csv B.csv out.csv 1000000

Вызов из пользовательского скрипта

НАЧАЛО				
rank_mantel_test	A.csv	B.csv	out.csv	1000000
КОНЕЦ				

2.2.13. Корреляция

Описание

Директива производит вычисление коэффициентов корреляции между каждой строкой первого файла и каждой строкой второго файла.

Параметры директивы

→ Имя первого входного файла.

→ Имя второго входного файла.

← Имя выходного файла.

→ (опционально) Опции по транспонированию входных матриц:

- t - транспонировать первую входную матрицу;
- nt - транспонировать вторую входную матрицу;
- tt - транспонировать первую и вторую входную матрицу.

Вызов из командной строки

Для входных файлов A.csv, B.csv:
correlation A.csv B.csv out.csv
correlation A.csv B.csv out.csv tt

Вызов из пользовательского скрипта

НАЧАЛО				
correlation	A.csv	B.csv	out.csv	
correlation	A.csv	B.csv	out.csv	tt
КОНЕЦ				

2.2.14. Метрика сжатия

Описание

Директива производит вычисление меры сходства объектов через оценку степени сжатия текстового описания объектов:

combined_to_sum:

$$d_{jk} = 2\frac{c_{jk}}{c_{jj} + c_{kk}} - 1,$$

union:

$$d_{jk} = \frac{c_{jj} + c_{kk} - c_{jk}}{c_{jk}},$$

normalized_compression_distance:

$$d_{jk} = \frac{c_{jk} - min(c_{jj}, c_{kk})}{max(c_{jj}, c_{kk})},$$

где c_{jk}, c_{jj} и c_{kk} - размер сжатия описания двух объектов с индексами j и k, как если бы это была одна строка.

Для модуля compress_metric входной файл должен содержать набор строк - объектов, для которых нужно вычислить меру сходства, один столбец - текстовое описание объектов.

Для модуля compress_metric_for_files входной файл должен содержать столбец с именами файлов, для содержимого которых нужно вычислить меру сходства. Входной файл дожен иметь ключи строк - названия объектов - и ключи столбцов.

Параметры директивы

→ Имя входного файла.

← Имя выходного файла, содержащего матрицу $D = d_{jk}$.

← Имя выходного файла, содержащего матрицу $O = c_{jj}$.

← Имя выходного файла, содержащего матрицу $C = c_{jk}$.

→ Тип метрики: combined_to_sum, union или normalized_compression_distance.

Выход

Таблица с мерой сходства объектов.

Вызов из командной строки

compress_metric in.csv out.csv obj.csv combined.csv combined_to_sum
compress_metric_for_files in.csv out.csv obj.csv combined.csv union

Вызов из пользовательского скрипта

НАЧАЛО					
compress_metric	in.csv	d.csv	o.csv	c.csv	union
compress_metric_for_files	files.csv	d.csv	o.csv	c.csv	union
КОНЕЦ					

Пример входных файлов

in.csv

c	data
o1	description for object 1
o2	data
o3	another description

files.csv

c	file names
o1	object_1.csv
o2	object_2.csv
o3	object_3.csv

d.csv

c	o1	o2	o3
o1	1	0.233333333333333	0.432432432432432
o2	0.233333333333333	1	0.28
o3	0.472222222222222	0.28	1

o.csv

c	compressed size
o1	29
o2	8
o3	24

c.csv

c	o1	o2	o3
o1	29	30	37
o2	30	8	25
o3	36	25	24

2.3 Работа с матрицами

2.3.1. Центрирование

Описание

Директива производит центрирование строк матрицы: для каждой строки вычисляется среднее и вычитается из каждого элемента соответствующей строки.

Параметры директивы

→ Имя входного файла.

← Имя выходного файла.

→ (опционально) Опции по транспонированию входной матрицы:

- t - транспонировать входную матрицу.

Выход

Файл, содержащий таблицу, в которой строки матрицы центрированы.

Вызов из командной строки

Для входного файла in.csv:
centre in.csv out.csv

Вызов из пользовательского скрипта

НАЧАЛО			
centre	in.csv	out.csv	
centre	in.csv	out.csv	t
КОНЕЦ			

2.3.2. Двойное центрирование

Описание

Директива производит двойное центрирование:

$$g_{ij} = a_{ij} - \overline{a_i} - \overline{a_j} + \overline{a},$$

где $\overline{a_i}$ - среднее по строкам, $\overline{a_j}$ - среднее по столбцам, $\overline{a}$ - среднее по всем элементам матрицы $A = a_{ij}$.

Параметры директивы

→ Имя входного файла.

← Имя выходного файла.

→ (опционально) Опции по транспонированию входной матрицы:

.

– t - транспонировать входную матрицу.

Выход

Файл, содержащий таблицу, в которой строки матрицы центрированы.

Вызов из командной строки

Для входного файла in.csv:
centreDouble in.csv out.csv

Вызов из пользовательского скрипта

НАЧАЛО			
centreDouble	in.csv	out.csv	
centreDouble	in.csv	out.csv	t
КОНЕЦ			

2.3.3. Нормирование на длину

Описание

Директива производит нормирование строк матрицы на длину. Для этого каждый элемент строки делится на свою длину:

$$s_j = \sqrt{\sum_{i=1}^{I} x_{ij}^2},$$

где I - число столбцов входной матрицы.

Параметры директивы

→ Имя входного файла.

← Имя выходного файла.

→ (опционально) Опции по транспонированию входной матрицы:

- t - транспонировать входную матрицу;

Выход

Файл, содержащий таблицу, в которой строки нормированы на длину.

Вызов из командной строки

Для входного файла in.csv:
normalizeLength in.csv out.csv

Вызов из пользовательского скрипта

НАЧАЛО			
normalizeLength	in.csv	out.csv	
normalizeLength	in.csv	out.csv	t
КОНЕЦ			

2.3.4. Нормирование на сигму

Описание

Директива производит нормирование строк матрицы на сигму. Для этого каждый элемент строки x_j делится на свое стандартное отклонение s_j:

$$s_j = \frac{\sqrt{\sum_{i=1}^{I} x_{ij}^2}}{\sqrt{I}},$$

где I - число столбцов входной матрицы.

Параметры директивы

→ Имя входного файла.

← Имя выходного файла.

→ (опционально) Опции по транспонированию входной матрицы:

- t - транспонировать входную матрицу;

Выход

Файл, содержащий таблицу, в которой строки нормированы на сигму.

Вызов из командной строки

Для входного файла in.csv:
normalizeSigma in.csv out.csv

Вызов из пользовательского скрипта

НАЧАЛО			
normalizeSigma	in.csv	out.csv	
normalizeSigma	in.csv	out.csv	t
КОНЕЦ			

2.3.5. Нормирование на сумму

Описание

Директива производит нормирование строк матрицы на сумму. Для этого каждый элемент строки делится на s_j:

$$s_j = \sum_{i=1}^{I} x_{ij},$$

где I - число столбцов входной матрицы.

Параметры директивы

→ Имя входного файла.

← Имя выходного файла.

→ (опционально) Опции по транспонированию входной матрицы:

- t - транспонировать входную матрицу;

Выход

Файл, содержащий таблицу, в которой строки нормированы на сумму.

Вызов из командной строки

Для входного файла in.csv:
normalizeSum in.csv out.csv

Вызов из пользовательского скрипта

НАЧАЛО			
normalizeSum	in.csv	out.csv	
normalizeSum	in.csv	out.csv	t
КОНЕЦ			

2.3.6. Нормирование на сумму квадратов

Описание

Директива производит нормирование элементов матрицы на сумму квадратов. Для этого каждый элемент x_{ij} делится на s:

$$s = \sqrt{\sum_{i=1}^{I} \sum_{j=1}^{J} x_{ij}^2},$$

где I - число столбцов входной матрицы, J - число строк входной матрицы.

Параметры директивы

→ Имя входного файла.

← Имя выходного файла.

→ (опционально) Опции по транспонированию входной матрицы:

- t - транспонировать входную матрицу;

Выход

Файл, содержащий таблицу, в которой строки нормированы на сумму квадратов.

Вызов из командной строки

Для входного файла in.csv:
normalizeSquare in.csv out.csv

Вызов из пользовательского скрипта

НАЧАЛО			
normalizeSquare	in.csv	out.csv	
normalizeSquare	in.csv	out.csv	t
КОНЕЦ			

2.3.7. Нормирование на максимальное значение в строке

Описание

Директива производит нормирование каждой строки матрицы на максимальное значение в этой строке.

Параметры директивы

→ Имя входного файла.

← Имя выходного файла.

→ (опционально) Опции по транспонированию входной матрицы:

- t - транспонировать входную матрицу;

Выход

Файл, содержащий таблицу, в которой каждая строка нормирована на свое максимальное значение.

Вызов из командной строки

Для входного файла in.csv:
normalizeMax in.csv out.csv

Вызов из пользовательского скрипта

НАЧАЛО			
normalizeMax	in.csv	out.csv	
normalizeMax	in.csv	out.csv	t
КОНЕЦ			

2.3.8. Квантильное выравнивание (quantile normalization)

Описание

Основной целью метода квантильного выравнивания (quantile normalization) является приведение распределений признаков объектов к одному распределению.

Алгоритм включает следующие этапы:

1. Во входной матрицы A, в каждом столбце вычисляются ранги элементов (к каждому элементу добавляется случайное число порядка 10^{-7}, чтобы исключить повторяющиеся элементы). Таким образом формируется матрица рангов R.
2. Каждый столбец матрицы A сортируется по возрастанию.
3. Вычисляется вектор V, каждый элемент которого является средним соответствующей строки отсортированной матрицы A.
4. Формируется выходная матрица O, которая является матрицей R, в которую подставили вместо каждого ранга, соответствующее значение из вектора V.

Параметры директивы

→ Имя входного файла.

← Имя выходного файла.

Выход

Квантильно выровненная таблица.

Вызов из командной строки

Для входного файла in.csv:
normalizeQuantile in.csv out.csv

Вызов из пользовательского скрипта

НАЧАЛО		
normalizeQuantile	in.csv	out.csv
КОНЕЦ		

2.3.9. Преобразование Бокса-Кокса

Описание

Директива производит преобразование Бокса-Кокса[8] для каждой строки y входной матрицы:

$$y^{(\lambda)} = \begin{cases} \dfrac{(y+s)^{\lambda}-1}{\lambda}, & \text{если } \lambda \neq 0, \\ ln(y+s), & \text{если } \lambda = 0. \end{cases}$$

Для вычисления значения λ производится поиск максимума логарифмической функции правдоподобия по профилю:

$$L(y_p) = (\lambda - 1)\sum_{i=1}^{N} ln(y_p) - \frac{N}{2} ln\left(\sum_{i=1}^{N} \frac{\left(y_{pi}^{(\lambda)} - \overline{y_p^{(\lambda)}}\right)^2}{N}\right),$$

где y_p - профиль, N - число объектов, $\overline{y_p^{(\lambda)}}$ - среднее по профилю, к которому применено преобразование Бокса-Кокса; s - сдвиг, минимизирующий стандартное отклонение профиля после применения преобразования Бокса-Кокса.

Параметры директивы

→ Имя входного файла.

← Имя выходного файла.

← Имя выходного файла для вычисленных значений λ.

← Имя выходного файла для значений сдвигов, применённых ко входной последовательности.

Вызов из командной строки

Для входного файла in.csv:
normalizeBoxCox in.csv out.csv l.csv shifts.csv

Вызов из пользовательского скрипта

НАЧАЛО				
normalizeBoxCox	in.csv	out.csv	l.csv	shifts.csv
КОНЕЦ				

[8] Box G. E. P., Cox D. R. An Analysis of Transformations // Journal of the Royal Statistical Society. Series B (Methodological), Vol. 26, No. 2. (1964), pp. 211-252.

2.3.10. Обратное преобразование Бокса-Кокса

Описание

Директива производит обратное преобразование Бокса-Кокса для каждой строки y входной матрицы:

$$x = \begin{cases} \sqrt[\lambda]{\lambda y + 1} - s, & \text{если } \lambda \neq 0, \\ e^y - s, & \text{если } \lambda = 0. \end{cases}$$

Параметры директивы

→ Имя входного файла.

← Имя выходного файла.

← Имя выходного файла для вычисленных значений λ.

← Имя выходного файла для значений сдвигов, применённых ко входной последовательности.

Вызов из командной строки

Для входного файла in.csv:
inverseBoxCox in.csv out.csv l.csv shifts.csv

Вызов из пользовательского скрипта

НАЧАЛО				
inverseBoxCox	in.csv	out.csv	l.csv	shifts.csv
КОНЕЦ				

2.3.11. Логарифмирование

Описание

Директива производит логарифмирование всех значений входной матрицы по заданному основанию матрицы.

Параметры директивы

→ Имя входного файла.

← Имя выходного файла.

→ Основание логарифма.

Выход

Файл, содержащий таблицу, в которой все значения прологарифмированы. Всякая ячейка входной матрицы, содержащая строку или число меньше или равное нуля, будет иметь значение NaN в выходном файле.

Вызов из командной строки

Для входного файла in.csv:
log in.csv out.csv 2

Вызов из пользовательского скрипта

НАЧАЛО			
log	in.csv	out.csv	2
КОНЕЦ			

2.3.12. Ортогонализация по модифицированной схеме Грама-Шмидта

Описание

Директива производит ортогонализацию вектор-строк матрицы по модифицированной схеме Грама-Шмидта:

1. для всех $i = 1..k$
    1. для всех $j = i + 1..k$:

$$v_j = v_j - proj_{v_i}(v_j)$$

Параметры директивы

→ Имя входного файла.

← Имя выходного файла.

Выход

Файл, содержащий таблицу, с ортогональными вектор-строками.

Вызов из командной строки

Для входного файла in.csv:
gramSchmidtProcess in.csv out.csv

Вызов из пользовательского скрипта

НАЧАЛО		
gramSchmidtProcess	in.csv	out.csv
КОНЕЦ		

2.3.13. Преобразование Фишера

Описание

Директива производит преобразование Фишера. Преобразование определено как:

$$z = \frac{1}{2} \ln \frac{1+r}{1-r},$$

и применяется к матрице корреляций.

Параметры директивы

→ Имя входного файла, содержащего таблицу коррелций.

← Имя выходного файла.

Вызов из командной строки

Для входного файла in.csv:
fisherTransformation in.csv out.csv

Вызов из пользовательского скрипта

НАЧАЛО		
fisherTransformation	in.csv	out.csv
КОНЕЦ		

2.3.14. Угловое преобразование Фишера

Описание

Директива производит угловое преобразование Фишера. Преобразование определено как:

$$z = arcsin\sqrt{p},$$

и применяется к матрице частот.

Параметры директивы

→ Имя входного файла, содержащего таблицу частот.

← Имя выходного файла.

Вызов из командной строки

Для входного файла in.csv:
angularTransformation in.csv out.csv

Вызов из пользовательского скрипта

НАЧАЛО		
angularTransformation	in.csv	out.csv
КОНЕЦ		

2.3.15. Перемножение матриц

Описание

Директива производит перемножение матриц, при этом происходит проверка совпадения ключей столбцов первой таблицы с ключами строк второй таблицы.

Параметры директивы

→ Имя входного файла с матрицей A.

→ Имя входного файла с матрицей B.

← Имя выходного файла.

Выход

Файл, содержащий результат перемножения матриц C: $C = AB$.

Вызов из командной строки

Для входных файлов A.csv, B.csv:
prod A.csv B.csv out.csv

Вызов из пользовательского скрипта

НАЧАЛО			
prod	A.csv	B.csv	out.csv
КОНЕЦ			

2.3.16. Операции над элементами матрицы

Описание

Директива производит вычисление функций от каждого элемента входной матрицы. Если результатом операции является нечисловое значение, то оно заменяется на -1.

Арифметические функции:

add <x>, sub <x>, mul <x>, div <x>.

Тригонометрические функции:

cos, sin, tg, ctg, arccos, arcsin, arctg, arcctg.

Гиперболические функции:

cosh, sinh, tgh, ctgh, arccosh, arcsinh, arctgh, arcctgh.

Экспоненциальные и логарифмические функции:

exp - вычисление экспоненциальной функции;

log <base> - вычисление логарифма по основанию base;

exp2 - вычисление бинарной экспоненты: 2^x.

Степенные функции:

pow <power> - возведение в степень power;

sqrt - вычисление квадратного корня;

cbrt - вычисление кубического корня.

Функции ошибок и гамма-функции:

erf - вычисление функции ошибки;

erfc - вычисление дополнительной функции ошибки;

gamma - вычисление гамма-функции;

lgamma - вычисление логарифма гамма-функции.

Функции округления и остатков:

ceil - округление вверх;

floor - округление вниз;

mod <x> - вычисление остатка от деления на x;

trunc - округление к целому в сторону нуля;

round - округление к ближайшему целому.

Функции сравнения:

isgreater <x> - больше, чем x;

isgreaterequal <x> - больше или равно x;

isless <x> - меньше, чем x;

islessequal <x> - меньше или равно x;

islessgreater <x> - не равно x;

isequal <x> - равно x;

isunordered - не число.

Другие функции:

abs - вычисление модуля;

max <x> - вычисление максимума с x;

min <x> - вычисление минимума с x;

sg, sign - вычисление знака.

entropy - вычисление $-v * ln(v)$ для каждого элемента матрицы v.

Параметры директивы

→ Имя входного файла.

← Имя выходного файла.

→ Название функции.

→ Параметр, если он необходим.

Выход

Файл, содержащий результат применения заданной функции.

Вызов из командной строки

Для входного файла in.csv:
matrixOnElementsOperation in.csv out.csv sin
matrixOnElementsOperation in.csv out.csv pow -1

Вызов из пользовательского скрипта

НАЧАЛО				
matrixOnElementsOperation	in.csv	out.csv	sin	
matrixOnElementsOperation	in.csv	out.csv	pow	-1
КОНЕЦ				

2.3.17. Поэлементные операции с матрицами

Описание

Директива производит поэлементное сложение, вычитание или умножение матриц.

Параметры директивы

→ Имя файла с первой матрицой.

→ Имя файла со второй матрицей.

← Имя выходного файла.

→ Идентификатор операции: add, sub, mul

Выход

Файл, содержащий результат применения заданной функции.

Вызов из командной строки

Для входных файлов a.csv, b.csv:
matrixElementsOperation a.csv b.csv out.csv add

Вызов из пользовательского скрипта

НАЧАЛО				
matrixElementsOperation	a.csv	b.csv	out.csv	add
КОНЕЦ				

2.3.18. Вычисление модуля каждого элемента

Описание

Директива вычисляет модуль каждого элемента входной матрицы.

Параметры директивы

→ Имя входного файла.

← Имя выходного файла.

Выход

Файл, содержащий результат применения модуля.

Вызов из командной строки

Для входного файла in.csv:
abs in.csv out.csv

Вызов из пользовательского скрипта

НАЧАЛО		
abs	in.csv	out.csv
КОНЕЦ		

2.3.19. Вычислить базовые статистики

Описание

Директива производит вычисление базовых статистик матрицы по столбцам:

1. число элементов столбца;
2. среднее;
3. стандартное отклонение;
4. дисперсию;
5. асимметрию;
6. коэффициент эксцесса;
7. сумму элементов;
8. сумму квадратов элементов;
9. минимум и максимум.
10. доверительный интервал.

Параметры директивы

→ Имя входного файла.

← Имя выходного файла.

→ (опционально) Уровень риска, по умолчанию 0.05.

Выход

Результат вычисления будет записан в выходной файл вида:

ID	N	mean	standard deviation	variance	skewness	excess kurtosis	sum	sum of squares	min	max	confidence interval min	confidence interval max

Вызов из командной строки

Для входного файла in.csv:
stat in.csv out.csv

Вызов из пользовательского скрипта

НАЧАЛО			
stat	in.csv	out.csv	
stat	in.csv	out.csv	0.05
КОНЕЦ			

2.3.20. Вычислить функцию распределения

Описание

Директива производит вычисление значения функции распределения каждого элемента входной матрицы x, равное вероятности события $\{X \geq x\}$, для нормального распределения или распределения Стьюдента.

Параметры директивы

→ Имя входного файла.

← Имя выходного файла.

→ Название распределения: normal или student.

→ Среднее для нормального распределения; число степеней свободы для распределения Стьюдента.

→ Стандартное отклонение (необходимо указывать только для нормального распределения).

Выход

Таблица значений функции распределения будет записана в выходной файл.

Вызов из командной строки

Для входного файла in.csv:
distribution in.csv out.csv normal 0 1

Вызов из пользовательского скрипта

НАЧАЛО					
distribution	in.csv	out.csv	normal	0	1
КОНЕЦ					

2.3.21. Вычислить дисперсию столбцов матрицы

Описание

Директива производит вычисление дисперсии каждого столбца матрицы.

Параметры директивы

→ Имя входного файла.

← Имя выходного файла.

Выход

Таблица значений дисперсии каждого столбца входной матрицы.

Вызов из командной строки

Для входного файла in.csv:
columnVariance in.csv out.csv

Вызов из пользовательского скрипта

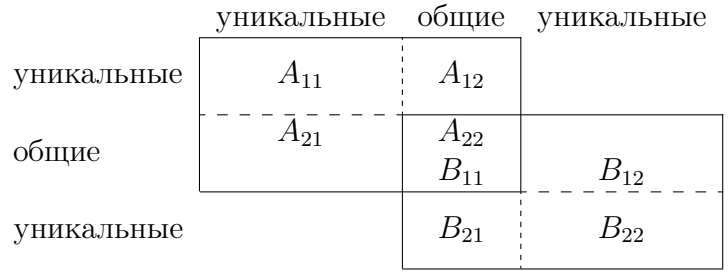
НАЧАЛО		
columnVariance	in.csv	out.csv
КОНЕЦ		

2.4 Подготовка данных

2.4.1. Разделение матриц

Описание

Директива производит разделение двух матриц на 8 по значениям ключей-строк и ключей-столбцов так, что:



Параметры директивы

→ Имя файла, содержащего таблицу A.

→ Имя файла, содержащего таблицу B.

← Имя файла, содержащего таблицу A_{11}.

← Имя файла, содержащего таблицу A_{12}.

← Имя файла, содержащего таблицу A_{21}.

← Имя файла, содержащего таблицу A_{22}.

← Имя файла, содержащего таблицу B_{11}.

← Имя файла, содержащего таблицу B_{12}.

← Имя файла, содержащего таблицу B_{21}.

← Имя файла, содержащего таблицу B_{22}.

Выход

Таблицы $A_{11}, \ldots, A_{22}$, содержащие значения из таблицы A, таблицы $B_{11}, \ldots, B_{22}$, содержащие значения из таблицы B.

Вызов из командной строки

Для входных файлов A.csv, B.csv:
8matrix A.csv B.csv A11.csv A12.csv A21.csv A22.csv B11.csv B12.csv B21.csv B22.csv

Вызов из пользовательского скрипта

НАЧАЛО										
8matrix	A.csv	B.csv	A11.csv	A12.csv	A21.csv	A22.csv	B11.csv	B12.csv	B21.csv	B22.csv
КОНЕЦ										

Пример работы директивы

Для входных файлов A.csv, B.csv:

A.csv

A_CON	p3	p4	p5	p6
o3	1	2	2	4
o4	2	2	2	5
o3	3	5	5	0
o6	1	2	2	4

B.csv

B_CON	p1	p2	p3	p4
o1	1	2	2	4
o2	2	3	4	5
o3	3	7	8	0
o4	1	2	2	9

Таблицы A_{11}, A_{12}, A_{21}, A_{22} будут иметь вид:

A11.csv

A_CON	p3	p4
o3	1	2
o4	2	2

A12.csv

A_CON	p5	p6
o3	2	4
o4	2	5

A21.csv

A_CON	p3	p4
o3	3	5
o6	1	2

A22.csv

A_CON	p5	p6
o3	5	0
o6	2	4

Таблицы B_{11}, B_{12}, B_{21}, B_{22} будут иметь вид:

B11.csv

B_CON	p1	p2
o1	1	2
o2	2	3

B12.csv

B_CON	p3	p4
o1	2	4
o2	4	5

B21.csv

B_CON	p1	p2
o3	3	7
o4	1	2

B22.csv

B_CON	p3	p4
o3	8	0
o4	2	9

2.4.2. Слияние матриц

Описание

Директива, обратная к разделению матриц, производит слияние произвольного числа таблиц по значениям ключей строк и ключей столбцов. При этом первая таблица является базовой, к ней последовательно добавляются все последующие таблицы. Рассмотрим две таблицы A_1 и A_2, которые необходимо объединить. Для каждой ячейки из таблицы A_2 производится поиск подходящей позиции в таблице A_1 по алгоритму:

1. Если в таблице A_1 существует столбец и существует строка с соответствующими значениями ключей и значение найденной ячейки пусто, то в неё записывается значение из таблицы A_2.

2. Если в таблице A_1 существует столбец, существует строка с соответствующими значениями ключей, но значение найденной ячейки не пусто и в A_1 содержится меньше столбцов с текущим значением ключа, чем в таблице A_2, то в A_1 будет добавлен столбец, каждая ячейка которого будет иметь пустое значение, после чего в соответствующую ячейку вставляется значение ячейки из A_2.

3. Если в таблице A_1 существует столбец и не найдена строка с соответствующим значением ключа, то в таблицу A_1 добавляется строка ключом из A_2, каждая ячейка которой будет иметь пустое значение, после чего в соответствующую ячейку вставляется значение ячейки из A_2.

4. Если в таблице A_1 не существует столбец с нужным значением ключа, то в A_1 добавляется новый столбец, каждая ячейка которого будет иметь пустое значение, переходим на шаг 1.

Параметры директивы

→ Количество таблиц для слияния N.

→ Таблица 1.

→ ...

→ Таблица N.

← Имя выходного файла.

Выход

Файл, содержащий объединенную таблицу.

Вызов из командной строки

Для входных файлов A1.csv, A2.csv, A3.csv:
matrixMerger 3 A1.csv A2.csv A3.csv out.csv

Вызов из пользовательского скрипта

НАЧАЛО					
matrixMerger	3	A1.csv	A2.csv	A3.csv	out.csv
КОНЕЦ					

Пример работы директивы

Для входных файлов A1.csv, A2.csv, A3.csv:

A1.csv

A_CON	p5	p6
o6	2	4

A2.csv

A_CON	p3	p4
o6	1	2

A3.csv

A_CON	p5	p6
o3	2	4
o4	2	5
o3	5	0

Тогда out.csv будет иметь вид:

out.csv

A_CON	p5	p6	p3	p4
o6	2	4	1	2
o3	2	4	NaN	NaN
o4	2	5	NaN	NaN
o3	5	0	NaN	NaN

2.4.3. Объединение столбцов матрицы

Описание

Директива производит строковое объединение значений столбцов входной матрицы в один.

Параметры директивы

→ Имя входного файла.

← Имя выходного файла.

Выход

Таблица, содержащая два столбца: ключи строк и результат объединения столбцов входной матрицы.

Вызов из командной строки

mergeColumns in.csv out.csv

Вызов из пользовательского скрипта

НАЧАЛО		
mergeColumns	in.csv	out.csv
КОНЕЦ		

Пример работы директивы

Пример входного файла in.csv и выходного out.csv:

in.csv

CON	c1	c2	c3
r1	1	2	3
r2	a	b	c
r3	4	-5	6.78

out.csv

CON	c1c2c3
r1	123
r2	abc
r3	4-56.78

2.4.4. Выравнивание таблиц

Описание

Директива производит удаление минимального количества строк так, чтобы число строк с одинаковыми метками в таблицах совпадало.

Параметры директивы

→ Имя входного файла A.

→ Имя входного файла B.

← Имя выравненного файла A.

← Имя выравненного файла B.

← Имя файла с удаленными строками из A.

← Имя файла с удаленными строками из B.

Выход

Выровненные матрицы.

Вызов из командной строки

Для входных файлов A1.csv, A2.csv, A3.csv:
trim A.csv B.csv A.trimmed.csv B.trimmed.csv A.deleted.csv B.deleted.csv

Вызов из пользовательского скрипта

НАЧАЛО						
trim	A.csv	B.csv	A.cmn.csv	B.cmn.csv	A.rm.csv	B.rm.csv
КОНЕЦ						

2.4.5. Замена значений ячеек значениями из указанного файла

Описание

Директива производит замену значений ячеек входной таблицы значениями из файла, содержащего замены. Замены осуществляется в указанной части матрицы в том случае, если в ячейке содержится строка для замены. Допустимые значения для указания где нужно осуществлять замены:

a - угловой элемент.

r - ключи строк.

c - ключи столбцов.

d - матрица данных.

Файл замен должен иметь следующую структуру:

proto.csv

#	from	to
1	GSM86857	GSM86787
...	...	...
N	GSM87190	GSM87123

Ключи строк и ключи столбцов в файле замен могут быть произвольными.

Параметры директивы

→ Имя входного файла.

→ Имя файла, содержащего замены.

← Имя выходного файла.

→ Указание, в какой части (возможно нескольких частей) матрицы осуществлять замены.

Выход

Таблица с измененными значениями.

Вызов из командной строки

Для входного файла in.csv:
replaceCellValuesByValuesFromAnotherFile in.csv groups.csv out.csv car

Вызов из пользовательского скрипта

НАЧАЛО				
replaceCellValuesByValuesFromAnotherFile	in.csv	grp.csv	out.csv	car
КОНЕЦ				

Пример работы директивы

Пусть входной файл in.csv, ex.csv имеют следующий вид:

in.csv

CON	object 1	object 2	object 3	object 4
p1	1	2	3	4
p2	2	3	4	5
p3	3	4	5	6
p4	4	5	6	7
p5	5	6	7	8
p6	6	7	8	9

ex.csv

N	from	to
1	p1	parameter 1
2	p3	parameter 3
3	5	not used
4	CON	new con

При вызове директивы с параметрами:

replaceCellValuesByValuesFromAnotherFile	in.csv	ex.csv	out.csv	r

Файл out.csv будет иметь вид:

out.csv

CON	object 1	object 2	object 3	object 4
parameter 1	1	2	3	4
p2	2	3	4	5
parameter 3	3	4	5	6
p4	4	5	6	7
p5	5	6	7	8
p6	6	7	8	9

Параметр r указывает, в какой части таблицы производить замены. В данном случае - в столбце, содержащем ключи строк.

2.4.6. Замена значений ячеек заданным значением

Описание

Директива производит замену значений ячеек входной таблицы заданным значением. Замена осуществляется в указанной части матрицы в том случае, если в ячейке содержится строка для замены. Допустимые значения для указания где нужно осуществлять замены:

a - угловой элемент.

r - ключи строк.

c - ключи столбцов.

d - матрица данных.

Параметры директивы

→ Указание, в какой части (возможно нескольких частей) матрицы осуществлять замены.

→ Строка поиска.

→ Строка, на которую будет произведены замены.

→ Имя входного файла.

← Имя выходного файла.

Выход

Таблица с измененными значениями.

Вызов из командной строки

Для входного файла in.csv:
replace_if_contains cd “age 20” 20 in.csv out.csv

Вызов из пользовательского скрипта

НАЧАЛО					
replace_if_contains	cd	age 20	20	in.csv	out.csv
КОНЕЦ					

2.4.7. Замена значений ячеек матрицы

Описание

Директива производит замену значений ячеек входной таблицы заданным значением. Замены осуществляется в указанной части матрицы. Допустимые значения для указания где нужно осуществлять замены:

a - угловой элемент.

r - ключи строк.

c - ключи столбцов.

Параметры директивы

→ Имя входного файла.

← Имя выходного файла.

→ Указание, в какой части матрицы осуществлять замены.

→ Текстовая строка - где заменять (Семантическое описание запроса).

→ Текстовая строка - на что заменять.

Выход

Файл, содержащий матрицу с заданной заменой.

Вызов из командной строки

Для входного файла in.csv:
replaceLabels in.csv out.csv r [$-3] value

Вызов из пользовательского скрипта

НАЧАЛО					
replaceLabels	in.csv	out.csv	r	[$-3]	value
КОНЕЦ					

2.4.8. Замена значений ячеек матрицы по регулярному выражению

Описание

Директива производит замену значений ячеек входной таблицы, которые удовлетворяют регулярному выражению, значением заданного формата. Результат соответствия фрагмента регулярного выражения в круглых скобках доступен в строке формата по своему номеру. (Пример: строка Z-1101 удовлетворяет регулярному выражению (\w)-([01]*), при этом для получения результата соответствия \w в строке формата, нужно использовать $1, а для [01]* - $2. Таким образом, для строки формата $2->$1 результат будет выглядеть как 1101->Z.)

Параметры директивы

→ Имя входного файла.

← Имя выходного файла.

→ Регулярное выражение.

→ Формат значения.

Выход

Файл, содержащий матрицу с измененными значениями.

Вызов из командной строки

Для входного файла in.csv:
regex in.csv out.csv ([a-zA-Z])-([01]*) $2->$1

Вызов из пользовательского скрипта

НАЧАЛО				
regex	in.csv	out.csv	([a-zA-Z])-([01]*)	$2->$1
КОНЕЦ				

2.4.9. Преобразовать вектор пар объект-признак в матрицу признаков

Описание

Директива преобразует вектор пар объект-признак в матрицу признаков, которая используется в nnbp. Допускается повторение объектов для указания нескольких признаков. В выходном файле имена признаков - идентификаторы столбцов, имена объектов - идентификаторы строк; каждой паре объект-признак сопоставляется единица в выходной матрице.

Параметры директивы

→ Имя входного файла.

← Имя выходного файла.

Выход

Таблица соответствия объектов признакам.

Вызов из командной строки

Для входного файла in.csv:
convertGroupVectorToMatrix in.csv out.csv

Вызов из пользовательского скрипта

НАЧАЛО		
convertGroupVectorToMatrix	in.csv	out.csv
КОНЕЦ		

Пример работы директивы

Для входного файла in.csv:
in.csv

object	group
o1	g1
o1	g4
o2	g2
o3	g3
o4	g2

Файл out.csv будет иметь вид:

out.csv

obj/grp	g1	g4	g2	g3
o1	1	1	0	0
o1	1	1	0	0
o2	0	0	1	0
o3	0	0	0	1
o4	0	0	1	0

2.4.10. Транспонирование

Описание

Директива производит транспонирование входной таблицы.

Параметры директивы

→ Имя входного файла.

← Имя выходного файла.

Выход

Файл, содержащий транспонированную таблицу.

Вызов из командной строки

Для входного файла in.csv:
transpose in.csv out.csv

Вызов из пользовательского скрипта

НАЧАЛО		
transpose	in.csv	out.csv
КОНЕЦ		

2.4.11. Сортировка по строке

Описание

Директива производит сортировку входной таблицы по значениям из заданной строки по возрастанию. Если строка содержит нечисловые значения, то производится лексикографическое упорядочивание.

Для сортировки по ключам столбцов следует указать нулевую строку ([$0]) или угловой ключ.

Параметры директивы

→ Имя входного файла.

← Имя выходного файла.

→ Номер строки, по значениям из которой будет произведена сортировка (Семантическое описание запроса).

→ Порядок сортировки:

inc - по возрастанию;

dec - по убыванию.

Выход

Файл, содержащий отсортированную таблицу по заданной строке.

Вызов из командной строки

Для входного файла in.csv:
sortByRow in.csv out.csv [$5] inc

Вызов из пользовательского скрипта

НАЧАЛО				
sortByRow	in.csv	out.csv	[$5]	inc
КОНЕЦ				

2.4.12. Сортировка по столбцу

Описание

Директива производит сортировку входной таблицы по значениям из заданного столбца по возрастанию. Если столбец содержит нечисловые значения, то производится лексикографическое упорядочивание.

Для сортировки по ключам столбцов следует указать нулевую строку ([$0]) или угловой ключ.

Параметры директивы

→ Имя входного файла.

← Имя выходного файла.

→ Номер столбца, по значениям из которого будет произведена сортировка (Семантическое описание запроса).

→ Порядок сортировки:

inc - по возрастанию;

dec - по убыванию.

Выход

Файл, содержащий отсортированную таблицу по заданному столбцу.

Вызов из командной строки

Для входного файла in.csv:
sortByColumn in.csv out.csv [$5] dec

Вызов из пользовательского скрипта

НАЧАЛО				
sortByColumn	in.csv	out.csv	[$5]	dec
КОНЕЦ				

2.4.13. Выборка строк таблицы

Описание

Директива позволяет производить выборку строк таблицы.

Семантическое описание запроса

Каждый запрос начинается с указания его типа - запрос на включение строк (Пример: [label3]), либо запрос на исключение строк (Пример:]label3[). Допускается три формы запроса:

1. указание шаблона колонки (Пример: [label3] - будут включены все строки с меткой label3);
2. лексико-графическое перечисление (Пример: [label1:label3] - будут включены все строки, лежащие лексико-графически между метками label1 и label3);
3. перечисление по абсолютным значениям (Пример: [label1..label3] - в данном случае будут включены все строки, лежащие между первой строкой с меткой label1 и первой строкой с меткой label3, встречающейся после метки label1).

Помимо указания метки, можно указать номер её вхождения в группе повторяющихся меток, при этом отрицательное число указывает на то, что нужно считать с конца (Пример: [label1#4..label3#-2] - в этом случае первые три строки с меткой label1 будут пропущены, будут включены строки, начиная с четвертой строки с этой меткой и до второй с конца строки с меткой label3).

Также можно указать абсолютный номер строки, используя арабские цифры либо буквенную форму, используемую в excel (Пример: [$3..$-4] - будут включены строки начиная с третьей и заканчивая четвертой с конца; [$C..$AF] - будут включены все строки, начиная со строки с номером C, заканчивая строкой с номером AF).

Параметры директивы

→ Имя входного файла.

← Имя выходного файла, содержащего строки, удовлетворяющие запросу.

← Имя выходного файла, содержащего строки, не удовлетворяющие запросу.

→ Запрос.

Выход

Файл, содержащий выборку строк, удовлетворяющих запросу и файл, содержащий выборку строк, не и удовлетворяющих запросу.

Вызов из командной строки

Для входного файла in.csv:
copyRows in.csv out.included.csv out.excluded.csv [$B..$AR]

Вызов из пользовательского скрипта

НАЧАЛО				
copyRows	in.csv	out.included.csv	out.excluded.csv	[$B..$AR]
КОНЕЦ				

Примеры запросов

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
	1	2	a	b	c	2	4	6	b	e	d	c		3.1	a	e	1	d
[b:d]	0	0	0	1	1	0	0	0	1	0	1	1	0	0	0	0	0	1
[b:d#2]	0	0	0	1	1	0	0	0	1	0	1	1	0	0	0	0	0	1
[b:d#-2]	0	0	0	1	1	0	0	0	1	0	1	1	0	0	0	0	0	1
[b:3.14]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
[b:$9]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
[b:$H]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
[b:~]	0	0	0	1	1	0	0	0	1	1	1	1	0	0	0	1	0	1
[b:]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
[b..d]	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
[b..d#2]	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
[b..d#-2]	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
[b..3.14]	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
[b..$9]	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
[b..$H]	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
[b..~]	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
[b..]	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
[2:e]	0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	1
[2:e#2]	0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	1
[2:e#-2]	0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	1
[2:4]	0	1	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0
[2:$9]	0	1	0	0	0	1	1	1	0	0	0	0	0	1	0	0	0	0
[2:$H]	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
[2:~]	0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	1
[2:]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
[2..e]	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
[2..e#2]	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
[2..e#-2]	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
[2..4]	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
[2..$9]	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
[2..$H]	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
[2..~]	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
[2..]	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
[$5:c]	0	0	0	1	1	0	0	0	1	0	0	1	0	0	0	0	0	0
[$5:c#2]	0	0	0	1	1	0	0	0	1	0	0	1	0	0	0	0	0	0
[$5:c#-2]	0	0	0	1	1	0	0	0	1	0	0	1	0	0	0	0	0	0
[$5:4]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[$5:$7]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
[$5:$H]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
[$5:~]	0	0	0	1	1	0	0	0	1	1	1	1	0	0	0	1	0	1
[$5:]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
[$5..c]	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
[$5..c#2]	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
[$5..c#-2]	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
[$5..4]	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
[$5..$7]	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
[$5..$H]	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
[$5..~]	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
[$5..]	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
[~:b]	1	1	1	1	0	1	1	1	1	0	0	0	1	1	1	0	1	0
[~:b#2]	1	1	1	1	0	1	1	1	1	0	0	0	1	1	1	0	1	0
[~:b#-2]	1	1	1	1	0	1	1	1	1	0	0	0	1	1	1	0	1	0
[~:4]	1	1	0	0	0	1	1	0	0	0	0	0	1	1	0	0	1	0
[~:$5]	1	1	1	1	0	1	1	1	1	0	0	0	1	1	1	0	1	0
[~:$H]	1	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0
[~:~]	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
[~:]	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
[~..b]	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
[~..b#2]	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
[~..b#-2]	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
[~..4]	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
[~..$5]	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
[~..$H]	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
[~..~]	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
[~..]	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
[:b]	1	1	1	1	0	1	1	1	1	0	0	0	1	1	1	0	1	0
[:b#2]	1	1	1	1	0	1	1	1	1	0	0	0	1	1	1	0	1	0
[:b#-2]	1	1	1	1	0	1	1	1	1	0	0	0	1	1	1	0	1	0
[:4]	1	1	0	0	0	1	1	0	0	0	0	0	1	1	0	0	1	0
[:$5]	1	1	1	1	0	1	1	1	1	0	0	0	1	1	1	0	1	0
[:$H]	1	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0
[:~]	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
[:]	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
[..b]	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
[..4]	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
[..~]	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
[..]	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

2.4.14. Выборка столбцов таблицы

Описание

Директива позволяет производить выборку столбцов таблицы.

Семантическое описание запроса

Документации для директивы "Выборка строк таблицы" содержит семантическое описание запроса в разделе "Семантическое описание запроса".

Параметры директивы

→ Имя входного файла.

← Имя выходного файла, содержащего столбцы, удовлетворяющие запросу.

← Имя выходного файла, содержащего столбцы, не удовлетворяющие запросу.

→ Запрос.

Выход

Файл, содержащий выборку столбцов, удовлетворяющих запросу и файл, содержащий выборку столбцов, не и удовлетворяющих запросу.

Вызов из командной строки

Для входного файла in.csv:
copyColumns in.csv out.included.csv out.excluded.csv [$B..$AR]

Вызов из пользовательского скрипта

НАЧАЛО				
copyColumns	in.csv	out.included.csv	out.excluded.csv	[$B..$AR]
КОНЕЦ				

2.4.15. Вставить диагональ в матрицу

Описание

Директива позволяет вставить диагональ в матрицу. Диагональ хранится в отдельном файле в виде вектор-столбца.

Параметры директивы

→ Имя входного файла с диагональю.

→ Имя входного файла с таблицей.

← Имя выходного файла.

Выход

Файл, содержащий таблицу с новой диагональю.

Вызов из командной строки

Для входного файла in.csv и файла с диагональю diagonal.csv:
insertDiagonal diagonal.csv in.csv out.csv

Вызов из пользовательского скрипта

НАЧАЛО			
insertDiagonal	diagonal.csv	in.csv	out.csv
КОНЕЦ			

2.4.16. Преобразовать таблицу в вектор

Описание

Директива позволяет преобразовать таблицу в вектор-столбец, в которой ключи строк вектора будут иметь вид:

row_<=>_col,

где row - ключ строки, col - ключ столбца входной таблицы.

Параметры директивы

→ Имя входного файла.

← Имя выходного файла.

← Имя выходного файла для ключей строк и столбцов.

→ Тип матрицы, которую нужно преобразовать в вектор:

upper_triangular, triangular - верхнетреугольная;

lower_triangular - нижнетреугольная;

diagonal - диагональная;

rectangular - прямоугольная.

Выход

Файл, содержащий вектор.

Вызов из командной строки

Для входного файла in.csv:
convertMatrixToLine in.csv out.csv labels.csv triangular

Вызов из пользовательского скрипта

НАЧАЛО				
convertMatrixToLine	in.csv	out.csv	labels.csv	triangular
КОНЕЦ				

2.4.17. Преобразовать вектор в таблицу

Описание

Директива позволяет преобразовать вектор в треугольную, прямоугольную или диагональную таблицу, при этом каждый элемент вектора должен иметь ключ вида:

row_<=>_col,

где row - ключ строки, col - ключ столбца.

Параметры директивы

→ Имя входного файла.

← Имя выходного файла.

→ Тип матрицы, в которую нужно преобразовать:

upper_triangular, triangular - верхнетреугольная;

lower_triangular - нижнетреугольная;

diagonal - диагональная;

rectangular - прямоугольная.

Выход

Файл, содержащий таблицу, построенную из вектора.

Вызов из командной строки

Для входного файла in.csv:
convertLineToMatrix in.csv out.csv rectangular

Вызов из пользовательского скрипта

НАЧАЛО			
convertLineToMatrix	in.csv	out.csv	rectangular
КОНЕЦ			

2.4.18. Переместить строку вверх

Описание

Директива делает строку с указанным ключом первой строкой в выходном файле. Строки, которые находились выше перемещаемой строки сдвигаются вниз.

Параметры директивы

→ Имя входного файла.

← Имя выходного файла.

→ Ключ строки, которую необходимо поместить вверх (Семантическое описание запроса).

Выход

Файл, с переставленной строкой.

Вызов из командной строки

Для входного файла in.csv:
moveLinesUP in.csv out.csv [A]

Вызов из пользовательского скрипта

НАЧАЛО			
moveLinesUP	in.csv	out.csv	[A]
КОНЕЦ			

Пример работы директивы

Пусть таблица имеет следующие ключи столбцов:

in.csv

CON	obj_1	obj_2	obj_3
0	1	2	3
A	B	C	D

Тогда, если указать ключ A, то выходной файл будет иметь вид:

out.csv

A	B	C	D
CON	obj_1	obj_2	obj_3
0	1	2	3

2.4.19. Поменять местами строки по ключу

Описание

Директива меняет местами две строки с указанными ключами.

Параметры директивы

→ Имя входного файла.

← Имя выходного файла.

→ Ключ первой строки.

→ Ключ второй строки.

Выход

Файл, переставленными строками.

Вызов из командной строки

Для входного файла in.csv:
exchangeRowsByKeyValue in.csv out.csv A B

Вызов из пользовательского скрипта

НАЧАЛО				
exchangeRowsByKeyValue	in.csv	out.csv	A	B
КОНЕЦ				

2.4.20. Поменять местами столбцы по ключу

Описание

Директива меняет местами два столбца с указанными ключами.

Параметры директивы

→ Имя входного файла.

← Имя выходного файла.

→ Ключ первой столбца.

→ Ключ второй столбца.

Выход

Файл, переставленными столбцами.

Вызов из командной строки

Для входного файла in.csv:
exchangeColumnsByKeyValue in.csv out.csv A B

Вызов из пользовательского скрипта

НАЧАЛО				
exchangeColumnsByKeyValue	in.csv	out.csv	A	B
КОНЕЦ				

2.4.21. Удаление строк с нечисловыми значениями

Описание

Директива удаляет все строки таблицы, содержащие нечисловые значения. Ключи строк и ключи столбцов не учитываются при поиске.

Параметры директивы

→ Имя входного файла.

← Имя выходного файла, не содержащего нечисловые значения в строках.

← Имя выходного файла, содержащего удаленные строки.

Выход

Файл, содержащий таблицу с числовыми значенимя.

Вызов из командной строки

Для входного файла in.csv:
deleteAllRowsWithNotNumberValues in.csv included.csv excluded.csv

Вызов из пользовательского скрипта

НАЧАЛО			
deleteAllRowsWithNotNumberValues	in.csv	included.csv	excluded.csv
КОНЕЦ			

2.4.22. Удаление строк, которые не содержат указанное значение в указанном столбце

Описание

Директива удаляет все строки таблицы, которые не содержат указанное значение в указанном столбце.

Параметры директивы

→ Имя входного файла.

← Имя выходного файла, содержащего таблицу с отфильтрованными строками.

← Имя выходного файла, содержащего удаленные строки.

→ Ключ столбца.

→ Значение, по которому будет произведено удаление строк.

Выход

Файл, содержащий таблицу с отфильтрованными строками.

Вызов из командной строки

Для входного файла in.csv:
deleteAllRowsByColumnValues in.csv inc.csv exc.csv obj_3 2

Вызов из пользовательского скрипта

НАЧАЛО					
deleteAllRowsByColumnValues	in.csv	inc.csv	exc.csv	obj_3	2
КОНЕЦ					

Пример работы директивы

Пусть входной файл in.csv имеет вид:

in.csv

CON	obj_1	obj_2	obj_3
p1	1	7	1
p2	2	2	2
p3	3	4	3
p4	4	9	2
p5	5	0	1

Тогда при запуске директивы с параметрами:

deleteAllRowsByColumnValues	in.csv	inc.csv	exc.csv	obj_3	2

Выходные файлы будет иметь вид:

inc.csv

CON	obj_1	obj_2	obj_3
p1	1	7	1
p3	3	4	3
p5	5	0	1

exc.csv

CON	obj_1	obj_2	obj_3
p2	2	2	2
p4	4	9	2

2.4.23. Замена ключей-строк в таблице

Описание

Директива производит замену ключей-строк таблицы to значениями ключей-строк таблицы from. При этом число строк в таблицах должно быть одинаковым.

Параметры директивы

→ Имя файла, содержащего таблицу from.

↔ Имя файла, содержащего таблицу to.

Выход

Результат замены будет записан в файл, содержащий таблицу to.

Вызов из командной строки

insertRowLabels from.csv to.csv

Вызов из пользовательского скрипта

НАЧАЛО		
insertRowLabels	from.csv	to.csv
КОНЕЦ		

2.4.24. Замена ключей-столбцов в таблице

Описание

Директива производит замену ключей-столбцов таблицы to значениями ключей-столбцов таблицы from. При этом число столбцов в таблицах должно быть одинаковым.

Параметры директивы

→ Имя файла, содержащего таблицу from.

↔ Имя файла, содержащего таблицу to.

Выход

Результат замены будет записан в файл, содержащий таблицу to.

Вызов из командной строки

insertColumnLabels from.csv to.csv

Вызов из пользовательского скрипта

НАЧАЛО		
insertColumnLabels	from.csv	to.csv
КОНЕЦ		

2.4.25. Замена ключей строк/столбцов числовыми значениями

Описание

Директива производит замену ключей строк либо ключей столбцов последовательными либо случайными числовыми значениями.

Параметры директивы

→ Имя входного файла.

← Имя выходного файла.

→ Тип генерируемых чисел:

sequential, s - для последовательных;

random, r - для случайных.

→ Указание, где нужно производить замену:

row, r - замена ключей строк;

column, c - замена ключей столбцов.

→ (опционально) Префикс ключа.

Выход

Результат замены ключей будет записан в выходной файл.

Вызов из командной строки

labelReplacer in.csv out.csv s r

Вызов из пользовательского скрипта

НАЧАЛО					
labelReplacer	in.csv	out.csv	sequential	row	
labelReplacer	in.csv	out.csv	s	c	COL_
КОНЕЦ					

2.4.26. Дописать таблицу справа

Описание

Директива производит объединение таблиц без учета значений ключей строк путем дописывания таблицы справа. Если вторая таблица содержит больше строк, то недостающие ключи строк будут взяты из второй таблицы.

Параметры директивы

→ Имя входного файла, содержащего таблицу A1.

→ Имя входного файла, содержащего таблицу A2.

← Имя выходного файла.

Выход

Результат объединения таблиц будет записан в выходной файл.

Вызов из командной строки

appendRight A1.csv A2.csv out.csv

Вызов из пользовательского скрипта

НАЧАЛО			
appendRight	A1.csv	A2.csv	out.csv
дописатьСправа	A1.csv	A2.csv	out.csv
КОНЕЦ			

Пример работы директивы

Пример входных файлов A1.csv, A2.csv и выходного out.csv:

A1.csv

A	c1	c2	c3
r1	1	2	a
r2	b	3	4

A2.csv

B	c4	c5
r3	5	6
r4	c	d
r5	7	e

out.csv

A	c1	c2	c3	c4	c5
r1	1	2	a	5	6
r2	b	3	4	c	d
r5				7	e

2.4.27. Дописать таблицу вниз

Описание

Директива производит объединение таблиц без учета значений ключей столбцов путем дописывания таблицы вниз. Если вторая таблица содержит больше столбцов, то недостающие ключи столбцов будут взяты из второй таблицы.

Параметры директивы

→ Имя входного файла, содержащего таблицу A_1.

→ Имя входного файла, содержащего таблицу A_2.

← Имя выходного файла.

Выход

Результат объединения таблиц будет записан в выходной файл.

Вызов из командной строки

appendBottom A1.csv A2.csv out.csv

Вызов из пользовательского скрипта

НАЧАЛО			
appendBottom	A1.csv	A2.csv	out.csv
дописатьСнизу	A1.csv	A2.csv	out.csv
КОНЕЦ			

Пример работы директивы

Пример входных файлов A1.csv, A2.csv и выходного out.csv:

A1.csv

A	c1	c2
r1	1	2
r2	3	str
r3	5	6

A2.csv

B	c3	c4	c5
r4	str2	8	a
r5	9	10	b
r6	11	12	c

out.csv

A	c1	c2	c5
r1	1	2	
r2	3	str	
r3	5	6	
r4	str2	8	a
r5	9	10	b
r6	11	12	c

2.4.28. Дописать файл

Описание

Директива дописывает второй файл к концу первого и записывает результат в выходной файл.

Параметры директивы

→ Имя первого входного файла.

→ Имя второго входного файла.

← Имя выходного файла.

Выход

Результат объединения файлов будет записан в выходной файл.

Вызов из командной строки

appendFile a.csv b.csv out.csv

Вызов из пользовательского скрипта

НАЧАЛО			
appendFile	a.csv	b.csv	out.csv
КОНЕЦ			

Пример работы директивы

Пример входных файлов A.csv, B.csv и выходного out.csv:

A.csv

A	c1	c2
r1	1	2
r2	3	str
r3	5	6

B.csv

B	c3	c4	c5
r4	str2	8	a
r5	9	10	b
r6	11	12	c

out.csv

A	c1	c2	
r1	1	2	
r2	3	str	
r3	5	6	
B	c3	c4	c5
r4	str2	8	a
r5	9	10	b
r6	11	12	c

2.4.29. Развертка матрицы в двоичные признаки

Описание

Директива строит бинарную матрицу для каждого набора из N столбцов по значениям матрицы из этого набора.

Параметры директивы

→ Имя входного файла.

← Имя выходного файла.

→ Размер блока N.

Выход

Для каждого набора из N столбцов будет записана бинарная матрица в выходной файл.

Вызов из командной строки

makeBinMatrix in.csv out.csv 3

Вызов из пользовательского скрипта

НАЧАЛО			
makeBinMatrix	in.csv	out.csv	3
КОНЕЦ			

Пример работы директивы

Для входного файла in.csv:

in.csv

IN	p1	p2	p3	p4	p5	p6
o1	3	4	1	2	5	5
o2	1	2	1	1	1	2
o3	1	1	1	1	3	4

При запуске директивы с параметрами:

makeBinMatrix	in.csv	out.csv	3

Файл out.csv будет иметь вид:

out.csv

IN	p1_1	p1_2	p1_3	p1_4	p4_1	p4_2	p4_3	p4_4	p4_5
o1	1	0	1	1	0	1	0	0	2
o2	2	1	0	0	2	1	0	0	0
o3	3	0	0	0	1	0	1	1	0

2.4.30. Разделить таблицу по значениям в заданном столбце

Описание

Директива производит разделение входной таблицы на две по числовым значениям в заданном столбце и критерием разделения:

- больше указанного значения;
- меньше указанного значения;
- равно указанному значению.

Параметры директивы

→ Имя входного файла.

← Имя выходного файла со строками, удовлетворяющими критерию.

← Имя выходного файла со строками, не удовлетворяющими критерию.

→ Шаблон столбца (Семантическое описание запроса).

→ Критерий сравнения значений:

gt - больше;

ls - меньше;

eq - равно.

→ Значение, относительно которого проводится разбиение.

Вызов из командной строки

splitByColumn in.csv inc.csv ex.csv [p1] gt 5.1

Вызов из пользовательского скрипта

НАЧАЛО						
splitByColumn	in.csv	inc.csv	ex.csv	[p1]	gt	5.1
КОНЕЦ						

2.4.31. Разделить таблицу по значениям в заданной строке

Описание

Директива производит разделение входной таблицы на две по числовым значениям в заданной строке и критерием разделения:

- больше указанного значения;
- меньше указанного значения;
- равно указанному значению.

Параметры директивы

→ Имя входного файла.

← Имя выходного файла со столбцами, удовлетворяющими критерию.

← Имя выходного файла со столбцами, не удовлетворяющими критерию.

→ Шаблон строки (Семантическое описание запроса).

→ Критерий сравнения значений:

gt - больше;

ls - меньше;

eq - равно.

→ Значение, относительно которого проводится разбиение.

Вызов из командной строки

splitByRow in.csv inc.csv ex.csv [p1] gt 5.1

Вызов из пользовательского скрипта

НАЧАЛО						
splitByRow	in.csv	inc.csv	ex.csv	[p1]	gt	5.1
КОНЕЦ						

2.4.32. Разделить таблицу по подстроке в ключах строк

Описание

Директива производит разделение входной таблицы на две по наличию указанной подстроки в ключах строк.

Параметры директивы

→ Имя входного файла.

← Имя выходного файла со строками, ключ которых содержит указанную подстроку.

← Имя выходного файла со строками, ключ которых не содержит указанную подстроку.

→ Подстрока.

Вызов из командной строки

splitColumnsBySubstring in.csv inc.csv exc.csv substring

Вызов из пользовательского скрипта

НАЧАЛО				
splitColumnsBySubstring	in.csv	inc.csv	exc.csv	substring
КОНЕЦ				

2.4.33. Разделить таблицу по подстроке в ключах столбцов

Описание

Директива производит разделение входной таблицы на две по наличию указанной подстроки в ключах столбцов.

Параметры директивы

→ Имя входного файла.

← Имя выходного файла со столбцами, ключ которых содержит указанную подстроку.

← Имя выходного файла со столбцами, ключ которых не содержит указанную подстроку.

→ Подстрока.

Вызов из командной строки

splitRowsBySubstring in.csv inc.csv exc.csv substring

Вызов из пользовательского скрипта

НАЧАЛО				
splitRowsBySubstring	in.csv	inc.csv	exc.csv	substring
КОНЕЦ				

2.4.34. Разделить значения по столбцам

Описание

Директива производит разделение каждого значения входной матрицы по одному символу по строкам.

Параметры директивы

→ Имя входного файла.

← Имя выходного файла.

Вызов из командной строки

splitColumnValuesIntoRows in.csv out.csv

Вызов из пользовательского скрипта

НАЧАЛО		
splitColumnValuesIntoRows	in.csv	out.csv
КОНЕЦ		

2.4.35. Извлечь числовое значение

Описание

Директива extract позволяет извлекать числовые значения из матрицы и записывать их в файл. Для выходного файла можно указать номер строки и номер столбца, куда нужно вставить значение из исходной матрицы. Если номер строки и номер столбца не указаны, то будет создан файл с таблицей 1×1, в которую будет записано значение.

Параметры директивы

→ Имя входного файла.

→ Номер строки.

→ Номер столбца.

← Имя выходного файла.

→ (опционально) Номер строки.

→ (опционально) Номер столбца.

Выход

В выходной файл будет записано число по указанному номеру строки, столбца либо NaN, если в матрице по этому адресу записано не число или значение находится за границами матрицы.

Вызов из командной строки

```
extract in.csv 2 5 out.csv
extract in.csv 2 5 out.csv 4 2
```

Вызов из пользовательского скрипта

НАЧАЛО						
extract	in.csv	2	5	out.csv		
extract	in.csv	2	5	out.csv	4	2
КОНЕЦ						

2.4.36. Вставить элемент

Описание

Директива позволяет вставить матрицу или элемент в таблицу.

Параметры директивы

↔ Имя файла, в который нужно вставить элемент.

→ Номер строки.

→ Номер столбца.

→ Тип значения: file или value.

→ Значение, которое необходимо вставить, либо имя файла с таблицей.

Выход

Для значения с типом value во входную таблицу будет записан элемент по указанному номеру строки, столбца.

Для значения с типом file во входную таблицу будет записана матрица, левый верхний угол которой будет расположен по указанному номеру строки, столбца.

Вызов из командной строки

insert in.csv 2 5 value 3.14
insert in.csv 2 5 file sub.csv

Вызов из пользовательского скрипта

НАЧАЛО					
insert	in.csv	2	5	value	3.14
insert	in.csv	2	5	file	sub.csv
КОНЕЦ					

2.4.37. Заполнить матрицу случайными значениями

Описание

Директива читает входной файл и создает матрицу такого же размера со случайными значениями.

Параметры директивы

→ Имя входного файла.

← Имя выходного файла.

Выход

Матрица со случайными значенимя.

Вызов из командной строки

initializeMatrix in.csv out.csv

Вызов из пользовательского скрипта

НАЧАЛО		
initializeMatrix	in.csv	out.csv
КОНЕЦ		

2.4.38. Замена значений столбца на соответствующий ему ранг

Описание

Директива производит замену значений в столбце на соответствующий ранг для указанного набора столбцов.

Параметры директивы

→ Имя входного файла.

← Имя выходного файла.

→ Шаблон (Семантическое описание запроса).

Вызов из командной строки

replaceColumnsByRanks in.csv out.csv [$3..$7]

Вызов из пользовательского скрипта

НАЧАЛО			
replaceColumnsByRanks	in.csv	out.csv	[$3..$7]
КОНЕЦ			

2.4.39. Подсчет количества повторов ключей строк

Описание

Директива по производит по заданному списку ключей строк подсчет числа встречаемости этих ключей в заданном файле.

Параметры директивы

→ Имя входного файла.

→ Имя файла со списком ключей.

← Имя выходного файла.

Вызов из командной строки

labelCount in.csv labelList.csv out.csv

Вызов из пользовательского скрипта

НАЧАЛО			
labelCount	in.csv	labelList.csv	out.csv
КОНЕЦ			

Пример работы директивы

Для входных файлов in.csv и labelList.csv:

in.csv

CON	col1	col2	col3
row1	1	2	3
row2	4	5	6
row3	7	8	9
row3	0	1	2

labelList.csv

CON	labels
1	row1
2	abc
3	row3

Выходной файл будет иметь вид:

out.csv

CON	label	occurrences
1	row1	1
2	abc	0
3	row3	2

2.4.40. Сдвиг матрицы

Описание

Директива производит сдвиг матрицы на заданное количество строк вниз.

Параметры директивы

→ Имя входного файла.

← Имя выходного файла.

→ Глубина сдвига.

→ Количество сдвигов.

Вызов из командной строки

matrixShifter in.csv out.csv 1 3

Вызов из пользовательского скрипта

НАЧАЛО				
matrixShifter	in.csv	out.csv	1	3
КОНЕЦ				

Пример работы директивы

Для входного файла:

in.csv

CON	p1	p2
o1	1	2
o2	3	str
o3	5	6

При запуске директивы с параметрами:

matrixShifter	in.csv	out.csv	1	3

Выходной файл будет иметь вид:

out.csv

CON	p1	p2	p1	p2	p1	p2
o1	1	2				
o2	3	str	1	2		
o3	5	6	3	str	1	2
o1			5	6	3	str
o2					5	6

2.4.41. Генерация данных (bootstrap)

Описание

Директива bootstrap генерирует новую выборку из существующей рандомизирующим алгоритмом.

Параметры директивы

→ Имя входного файла.

← Имя выходного файла.

→ (опционально) Значение для инициализации генератора случайных чисел.

← (опционально) Имя файла, в который будет записано случайное значение [9].

Вызов из командной строки

bootstrap in.csv out.csv

Вызов из пользовательского скрипта

НАЧАЛО				NULL
bootstrap	in.csv	out.csv		NULL
bootstrap	in.csv	out.csv	123	NULL
bootstrap	in.csv	out.csv	123	next_seed.csv
бутстреп	in.csv	out.csv		NULL
КОНЕЦ				NULL

Пример работы директивы

Пример входного файла in.csv и возможного выходного out.csv:

in.csv

IN	c1	c2	c3
r1	1	2	3
r2	a	b	c
r3	3	4	5
r4	4	5	6
r5	5	d	e

out.csv

IN	c1	c2	c3
r3	3	4	5
r5	5	d	e
r3	3	4	5
r2	a	b	c
r4	4	5	6

[9] Это случайное значение может быть использовано для инициализации генератора случайных чисел при следующем вызове модуля.

2.4.42. Генерация данных ключей (bootstrap)

Описание

Модуль генерирует целочисленные веса для ключей строк входной матрицы так, что их сумма равна числу строк матрицы.

Параметры директивы

→ Имя входного файла.

← Имя выходного файла.

→ Число повторов.

→ (опционально) Значение для инициализации генератора случайных чисел.

← (опционально) Имя файла, в который будет записано случайное значение [10].

Вызов из командной строки

bootstrapLabels in.csv out.csv 5

Вызов из пользовательского скрипта

НАЧАЛО					
bootstrapLabels	in.csv	out.csv	5		
bootstrapLabels	in.csv	out.csv	5	123	
bootstrapLabels	in.csv	out.csv	5	123	next_seed.csv
КОНЕЦ					

Пример работы директивы

Пример входного файла in.csv и возможного выходного out.csv:

[10] Это случайное значение может быть использовано для инициализации генератора случайных чисел при следующем вызове модуля.

in.csv

IN	c1	c2	c3
r1	1	2	3
r2	a	b	c
r3	3	4	5
r4	4	5	6
r5	5	d	e

out.csv

IN	weight_1	weight_2	weight_3	weight_4	weight_5
r1	0	1	2	2	0
r2	1	2	3	0	1
r3	2	1	0	0	2
r4	1	1	0	1	1
r5	1	0	0	2	1

2.4.43. Перестановка строк

Описание

Директива генерирует новую выборку из существующей осуществляя перестановку строк рандомизирующим алгоритмом.

Параметры директивы

→ Имя входного файла.

← Имя выходного файла.

→ (опционально) Значение для инициализации генератора случайных чисел.

← (опционально) Имя файла, в который будет записано случайное значение [11].

Вызов из командной строки

permutation in.csv out.csv

Вызов из пользовательского скрипта

НАЧАЛО				NULL
permutation	in.csv	out.csv		NULL
permutation	in.csv	out.csv	123	NULL
permutation	in.csv	out.csv	123	next_seed.csv
КОНЕЦ				NULL

Пример работы директивы

in.csv

IN	c1	c2	c3
r1	1	2	3
r2	a	b	c
r3	3	4	5
r4	4	5	6
r5	5	d	e

out.csv

IN	c1	c2	c3
r3	3	4	5
r2	a	b	c
r5	5	d	e
r4	4	5	6
r1	1	2	3

[11] Это случайное значение может быть использовано для инициализации генератора случайных чисел при следующем вызове модуля.

2.4.44. Преобразование дерева кластеризации в 0,1-матрицу

Описание

Директива для каждой группы объектов во входном файле строит столбец, содержащий еденицы для объектов, входящих в группу.

Параметры директивы

→ Имя входного файла.

← Имя выходного файла.

Вызов из командной строки

treeToMatrix in.nwk out.csv

Вызов из пользовательского скрипта

НАЧАЛО		
trim	in.nwk	out.csv
КОНЕЦ		

2.5 Файловые операции

2.5.1. Создать пустой файл

Описание

Директива создает пустой файл с указанным именем, в случае, если такой файл уже существует, его содержимое удаляется.

Параметры директивы

→ Имя входного файла.

Выход

Пустой файл.

Вызов из командной строки

createFile log.txt

Вызов из пользовательского скрипта

НАЧАЛО	
createFile	log.txt
КОНЕЦ	

2.5.2. Скопировать файл

Описание

Директива создает копию входного файла.

Параметры директивы

→ Имя входного файла.

← Имя выходного файла.

Выход

Копия входного файла.

Вызов из командной строки

copy in.csv out.csv

Вызов из пользовательского скрипта

НАЧАЛО		
copy	in.csv	out.csv
КОНЕЦ		

2.5.3. Удалить файл

Описание

Директива удаляет файл. Если не указан путь, то будет удален файл из рабочей директории. В качестве имени файла можно указать конкретный файл (Пример: in.csv) или маску (Пример: in*.csv).

Параметры директивы

→ Имя файла.

Вызов из командной строки

delete in.csv

Вызов из пользовательского скрипта

НАЧАЛО	
delete	in.csv
КОНЕЦ	

2.5.4. Преобразовать CSV в TXT

Описание

Директива преобразует файл в формате CSV в TXT файл, в котором в качестве разделителя столбцов используется таб.

Параметры директивы

→ Имя входного файла.

← Имя выходного файла.

Вызов из командной строки

csvToTxt in.csv out.csv

Вызов из пользовательского скрипта

НАЧАЛО		
csvToTxt	in.csv	out.csv
КОНЕЦ		

2.5.5. Изменить разделитель CSV файла

Описание

Директива производит замену разделителя в CSV файле в соответствии с указанными параметрами. В качестве разделителя допускаются следующие значения:

- space;
- tab;
- semicolon;
- colon;
- comma.

Если в качестве параметра разделителя указан один символ, то он будет использован как разделитель столбцов.

Параметры директивы

→ Имя входного файла.

← Имя выходного файла.

→ Разделитель входного файла.

→ Разделитель входного файла.

Вызов из командной строки

convertCSV in.csv out.csv comma semicolon

Вызов из пользовательского скрипта

НАЧАЛО				
convertCSV	in.csv	out.csv	comma	semicolon
convertCSV	in.csv	out.csv	,	;
КОНЕЦ				

2.6 Другие

2.6.1. Сравнение матриц

Описание

Директива производит сравнение двух матриц. Модуль завершается с ошибкой, если матрицы не совпадают.

Параметры директивы

→ Первый входной файл для сравнения.

→ Второй входной файл для сравнения.

→ Тип сравниваемых матриц: string или numeric.

→ (опционально) Поведение при несовпадении матриц: abort или report.

Вызов из командной строки

Для входных файлов lhs.csv и rhs.csv:
assertEqual lhs.csv rhs.csv numeric

Вызов из пользовательского скрипта

НАЧАЛО			
assertEqual	lhs.csv	rhs.csv	numeric
КОНЕЦ			

2.6.2. Напечатать текст

Описание

Директива позволяет напечатать произвольный текст в процессе работы скрипта.

Параметры директивы

→ Текст, который необходимо напечатать

Вызов из командной строки

echo text

Вызов из пользовательского скрипта

НАЧАЛО	
echo	text
КОНЕЦ	

2.6.3. Перекодировка файла

Описание

Директива производит конвертацию кодировки входного файла. При запуске модуля с неправильным названием кодировки будет выведен список всех допустимых значений.

Параметры директивы

→ Кодировка входного файла.

→ Кодировка выходного файла.

→ Входной файл.

← Выходной файл.

Вызов из командной строки

Для входного файла in.csv:
encodingConverter CP1251 UTF8 in.csv out.csv

Вызов из пользовательского скрипта

НАЧАЛО				
encodingConverter	CP1251	UTF8	in.csv	out.csv
КОНЕЦ				

2.6.4. Визуализация данных

Описание

Директива производит визуализацию двумерных данных.

Матрица со входными данными должна содержать два столбца - координаты объектов. Квадратная матрица связей позволяет соединить объекты дугами. Её размерность должна совпадать с числом объектов входной матрицы. При построении дуг используется порядковый индекс объекта без учёта ключей строк/столбцов таблицы связей, поэтому порядок объектов входной матрицы и матрицы связей должен совпадать. Дуга, соединяющая два объекта, будет нарисована, если целая часть соответствующего значения матрицы связей не равна нулю. Значения на диагонали игнорируются.

Таблица групп объектов определяет принадлежность объектов группе. Каждая строка состоит из имени объекта, имени группы и, опционально, цвета. Если цвет не указан, он назначается автоматически[12]. Если для одной группы указано несколько разных значений цвета, то будет использовано первое. Примеры таблиц групп:

groups1.csv

object	group
object1	group1
object2	group2
object3	group1

groups2.csv

object	group	color
object1	group1	0xFFFF0000
object2	group2	0xFF0000FF
object3	group1	

Настройки отображения задаются в файле plot_defaults.csv в директории исполнения скрипта. Поддерживаются следующие параметры:

- connection_file - имя файла таблицы связей (по умолчанию: не используется);
- group_file - имя файла таблицы групп объектов (по умолчанию: не используется);
- configuration_file - имя файла с настройками отображения. Относительный путь разрешается относительно текущей рабочей директории. Настройки из указанного файла могут переопределять ранее указанные значения, последующие параметры могут переопделять значения из конфигурационного файла (по умолчанию: не используется);
- width, height - ширина и высота изображения в пикселях (по умолчанию: 512);
- draw_area_offset_top,
  draw_area_offset_bottom,
  draw_area_offset_left,
  draw_area_offset_right - отступ от краёв в пикселях (по умолчанию: 20);
- data_space_top_offset_factor, data_space_bottom_offset_factor,
  data_space_left_offset_factor, data_space_right_offset_factor -
  расширение объемлющего прямоугольника (по умолчанию: 0.02);

[12]Цвет является функцией от имени группы, не гарантирующей уникальность значения.

- data_scaling - параметр растягивания данных по осям, допустимые значения: stretch, none (по умолчанию: none);
- background_color - цвет фона в формате ARGB - прозрачность, красный, зеленый, синий цвета (по умолчанию: 0xFFFFFFFF);
- max_axis_label_decimal_count - максимальное количество знаков после запятой значений подписей осей (по умолчанию: 3);
- tick_density - желаемое расстояние между штрихами в пикселях (по умолчанию: 50);
- tick_size - длина штриха в пикселях (по умолчанию: 5);
- grid_line_width - толщина линии объемлющего прямоугольника объектов в пикселях (по умолчанию: 1);
- title_text_size - размер текста заголовка в пикселях (по умолчанию: 12);
- typeface - гарнитура (по умолчанию: Arial);
- font_weight - насыщенность шрифта (по умолчанию: 400);
- font_width - ширина шрифта (по умолчанию: 5);
- font_slant - наклон шрифта: upright, italic, oblique (по умолчанию: upright);
- point_circle_radius - радиус точки в пикселях, соответствующей объекту входных данных (по умолчанию: 2);
- label_placement - стратегия размещения подписей объектов, допустимые значения: AsIs, Coulomb, None (по умолчанию: None);
- coulomb_max_shift - максимальный сдвиг подписи объекта в пикселях (по умолчанию: 10);
- coulomb_iteration_count - количество итераций при вычислении сдвигов подписей объектов (по умолчанию: 10);
- with_convex_hull - построение выпуклой оболочки по контурам групп (по умолчанию: false);
- grid_type - тип сетки, допустимые значения: none, dashed, line (по умолчанию: line);
- grid_color - цвет сетки (по умолчанию: 0xFF808080);
- grid_primary_tick_size - длина штриха для сетки типа dashed (по умолчанию: 10);
- grid_secondary_tick_size - длина промежутка между штрихами для сетки типа dashed (по умолчанию: 10);
- with_anti_aliasing - включить сглаживание (по умолчанию: true);

- image_quality - метрика, задающая соотношение между размером файла и качеством изображения от 0 до 100. Параметр специфичен для используемого формата и может быть проигнорирован (по умолчанию: 100);

- image_output_format - формат изображения: png, jpg (по умолчанию: png).

Параметры директивы

→ Входной файл.

← Выходной файл.

→ (опционально) Имя параметра.

→ (опционально) Значение параметра.

→ (опционально) ...

→ (опционально) Имя параметра.

→ (опционально) Значение параметра.

Вызов из командной строки

Для входного файла in.csv:
plot in.csv out.png

Вызов из пользовательского скрипта

НАЧАЛО						
plot	in.csv	out.png				
график	in.csv	out.png	width	1024	height	512
КОНЕЦ						

2.6.5. Записать заголовок

Описание

Директива для каждого ключа столбцов записывает его позицию и номер в группе одинаковых элементов.

Параметры директивы

→ Имя входного файла.

← Имя выходного файла.

Вызов из командной строки

Для входного файла in.csv:
printHeader in.csv out.csv

Вызов из пользовательского скрипта

НАЧАЛО		
printHeader	in.csv	out.csv
КОНЕЦ		

Пример работы директивы

Пусть таблица имеет следующие ключи столбцов:

in.csv

CON	obj_1	obj_2	obj_3	obj_2	obj_4	obj_2

Тогда выходной файл будет иметь вид:

out.csv

A	B	C	D	E	F	G
0	1	2	3	4	5	6
CON	obj_1	obj_2	obj_3	obj_2	obj_4	obj_2
	1	1	1	2	1	3

2.6.6. Подпрограмма

Описание

Директива производит исполнение скрипта. Доступ к аргументам скрипта осуществляется через переменные вида @1, @2, . . . ; или argument_1, argument_2, . . . Аргумент с индексом 0 содержит имя файла подпрограммы:

НАЧАЛО	
echo	Начато исполнение ≪@0≫ (aka ≪argument_0≫)
echo	Параметр 1: ≪@1≫ (≪argument_1≫)
echo	Параметр 2: ≪@2≫ (≪argument_2≫)
КОНЕЦ	

Параметры директивы

→ Имя файла скрипта.

→ (опционально) Параметр 1.

→ (опционально) . . .

→ (опционально) Параметр N.

Вызов из командной строки

Для входного файла script.csv:
subroutine script.csv arg1 3.14 arg3

Вызов из пользовательского скрипта

НАЧАЛО				
subroutine	script.csv	arg1	3.14	arg3
КОНЕЦ				

Глава 3

Методы

3.1 Метод главных координат

Обычный метод нахождения главных компонент заключается в следующем[1]. Пусть X – центрированная матрица координат объектов в некотором евклидовом пространстве. К ней применяется singular value decomposition (SVD):

$$X = PSV^T,$$

где P, V^T – ортогональные матрицы, а S – диагональная матрица сингулярных чисел матрицы X. Тогда матрица

$$U = XV = PS$$

есть матрица главных компонент X. Можно применять SVD к симметричной матрице XX^T:

$$XX^T = P\Lambda P^T,$$

где P – та же ортогональная матрица, что и для X, а Λ – диагональная матрица сингулярных чисел матрицы XX^T. Однако,

$$XX^T = PSV^T * VSP^T = PSSP^T = PS^2P^T.$$

Следовательно,

$$S^2 = \Lambda, \quad S = \Lambda^{1/2}.$$

То есть, матрица сингулярных чисел матрицы XX^T есть матрица собственных значений матрицы X. Поэтому вычислять главные компоненты надо по формуле

$$U = P\Lambda^{1/2}.$$

Это имеет большой практический смысл, если число объектов существенно меньше, чем число признаков, что встречается все чаще в биологических исследованиях.

[1] Vadim M. Efimov, Kirill V. Efimov, Vera Yu. Kovaleva Principal component analysis and its generalizations for any type sequence (pca-seq) // bioRxiv 535112, doi: 10.1101/535112

Более полувека назад Гауэр (Gower, 1966) нашел, что если вычислить матрицу D евклидовых расстояний между строками X, возвести расстояния в квадрат, применить двойное центрирование и умножение на $-\frac{1}{2}$, то получится матрица XX^T. Применяя к ней SVD, получим главные компоненты. Следовательно, сама матрица X не нужна и может даже не существовать в числовом виде, для вычисления главных компонент некоторого множества объектов достаточно иметь матрицу евклидовых расстояний между ними, полученную любым способом. Именно поэтому Гауэр назвал свой метод методом главных координат (PCo). Если вычислить евклидову матрицу расстояний между строками матрицы главных компонент, то она совпадет с исходной матрицей евклидовых расстояний D. Это свойство можно использовать для контроля правильности расчетов.

PCo нередко применяется для матриц различия, для которых неизвестно, являются ли они евклидовыми расстояниями между объектами или нет. В случае неевклидовости некоторые диагональные числа матрицы Λ будут отрицательными. Иногда малые отрицательные числа могут возникнуть вследствие накопления ошибок компьютерных вычислений. Все такие “компоненты”, а также нулевые надо исключить из рассмотрения.

Документация JACOBI 4

Вадим Ефимов Денис Полунин Ирина Штайгер

25 октября 2020 г.

Словарь терминов

Principal component analysis

Метод главных компонент, 主成分分析.

Principal coordinate analysis

Multidimensional scaling, Метод главных координат, 多维标度.

Singular-value decomposition

Сингулярное разложение матрицы, 奇异值分解.

Two-Block Partial Least-Squares

2B-PLS, 2B-偏最小二乘法.

bootstrap

引导程序.

Предметный указатель