|  |
| --- |
| Санкт-Петербургский Государственнный Университет |
| Спецификация по работе с пакетом аналитических функций для PostgreSQL |
| Первая редакция |

|  |
| --- |
| Акимов Максим Евгеньевич, Чудова Марина Юрьевна  28.12.2017 |

Оглавление

[Utilities Functions 3](#_Toc515319066)

[dots2DimShow 3](#_Toc515319067)

[timeSeriesShow 4](#_Toc515319071)

[dots2DimClusteredShow 5](#_Toc515319075)

[dist 6](#_Toc515319079)

[initClusters 7](#_Toc515319083)

[maxi 8](#_Toc515319087)

[mini 9](#_Toc515319091)

[Determinant 10](#_Toc515319095)

[TransposeMatrix 11](#_Toc515319099)

[InvertMatrix 11](#_Toc515319103)

[MultMatrix 12](#_Toc515319107)

[Cramer 13](#_Toc515319111)

[OLS 14](#_Toc515319115)

[I 15](#_Toc515319119)

[MA 16](#_Toc515319123)

[Random Functions 18](#_Toc515319127)

[random2DimDotsCircles 18](#_Toc515319128)

[random2DimDotsBlobs 19](#_Toc515319132)

[randomTimeSeries 20](#_Toc515319136)

[Forecasting Functions 22](#_Toc515319140)

[expForecast 22](#_Toc515319141)

[expTrendForecast 23](#_Toc515319145)

[holtWintersForecast 24](#_Toc515319149)

[linearRegression 26](#_Toc515319153)

[AR 27](#_Toc515319157)

[ARIMA 28](#_Toc515319161)

[Forecasting Metrics 30](#_Toc515319165)

[MAE 30](#_Toc515319166)

[RMSE 31](#_Toc515319170)

[MPE 31](#_Toc515319174)

[MAPE 32](#_Toc515319178)

[AD 33](#_Toc515319182)

[MAD 34](#_Toc515319186)

[R2 35](#_Toc515319190)

[THEIL 36](#_Toc515319194)

[Pearson 37](#_Toc515319198)

[Clustering Functions 39](#_Toc515319202)

[DBSCAN 39](#_Toc515319203)

[Kmeans 40](#_Toc515319207)

[Kmedians 41](#_Toc515319211)

[FOREL 42](#_Toc515319215)

[Agglomerative 44](#_Toc515319219)

[Clustering Metrics 46](#_Toc515319223)

[Sillhuette 46](#_Toc515319224)

[Dunn 47](#_Toc515319228)

[DB 47](#_Toc515319232)

Utilities Functions

dots2DimShow

FUNCTION anfun.dots2DimShow(\_tbl) RETURNS table(id, X, Y)

Описание: Функция возвращает таблицу по её названию. Её данные представляют собой идентификационные номера (столбец **id**) и координаты точек в двумерном пространстве (столбцы **X** и **Y**). Эта функция была реализована для внутреннего использования в других функциях.

Input

\_tbl: varchar

Название таблицы, столбцы которой функция возвращает.

Output

table(id, X, Y): таблица из трёх столбцов

id: integer

Представляет собой первый столбец с целочисленными значениями в таблице **\_tbl**. Идентификационный номер точки.

X: numeric

Представляет собой второй столбец с вещественными числами в таблице **\_tbl**. Первая координата точки.

Y: numeric

Представляет собой третий столбец с вещественными числами в таблице **\_tbl**. Вторая координата точки.

Example

>> IN  
  
SELECT \* FROM anfun.dots2DimShow('random2DimDotsBlobs(100, 3, 1)');  
  
>> OUT  
  
-------------------------------  
| id | X | Y |  
-------------------------------  
| 1 | 1.015750 | 0.644730 |  
| 2 | 0.967589 | 0.610255 |  
| ... | ... | ... |  
| 300 | 0.805604 | 0.642990 |  
-------------------------------

Total running time of the script: ( 0 minutes 0.243 seconds)

timeSeriesShow

FUNCTION anfun.timeSeriesShow(\_tbl) RETURNS table(X, Y)

Описание: Функция возвращает таблицу по её названию. Её данные представляют собой временной ряд, где столбец **X** – момент времени, а столбец **Y** – значение некоторого признака в данный момент времени. Эта функция была реализована для внутреннего использования в других функциях.

Input

\_tbl: varchar

Название таблицы, столбцы которой функция возвращает.

Output

table(X, Y): таблица из двух столбцов

X: integer

Представляет собой первый столбец с целочисленными значениями в таблице **\_tbl**. Момент времени.

Y: numeric

Представляет собой второй столбец с вещественными числами в таблице **\_tbl**. Значение некоторого признака в данный момент времени.

Example

>> IN  
  
SELECT \* FROM anfun.timeSeriesShow('randomTimeSeries(100, 2, 3)');  
  
>> OUT  
  
--------------------  
| X | Y |  
--------------------  
| 1 | 0.157399 |  
| 2 | 0.093537 |  
| ... | ... |  
| 100 | 1.366820 |  
--------------------

Total running time of the script: ( 0 minutes 0.221 seconds)

dots2DimClusteredShow

FUNCTION anfun.dots2DimClusteredShow(\_tbl) RETURNS table(id, X, Y, C)

Описание: Функция возвращает таблицу по её названию. Её данные представляют собой идентификационные номера (столбец **id**), координаты точек в двумерном пространстве (столбцы **X** и **Y**) и номер кластера, к которому принадлежит данная точка (столбец **C**). Эта функция была реализована для внутреннего использования в других функциях.

Input

\_tbl: varchar

Название таблицы, столбцы которой функция возвращает.

Output

table(id, X, Y, C): таблица из четырёх столбцов

id: integer

Представляет собой первый столбец с целочисленными значениями в таблице **\_tbl**. Идентификационный номер точки.

X: numeric

Представляет собой второй столбец с вещественными числами в таблице **\_tbl**. Первая координата точки.

Y: numeric

Представляет собой третий столбец с вещественными числами в таблице **\_tbl**. Вторая координата точки.

C: integer

Представляет собой четвёртый столбец с целочисленными значениями в таблице **\_tbl**. Номер кластера, в котором находится данная точка.

Example

>> IN

SELECT \* INTO TS FROM anfun.initClusters('random2DimDotsBlobs(100, 3, 1)')

SELECT \* FROM anfun.dots2DimClusteredShow('TS')  
  
>> OUT  
  
---------------------------------------  
| id | X | Y | C |  
---------------------------------------  
| 1 | 0.599849 | 0.509507 | 1 |  
| 2 | 0.803723 | 0.612849 | 2 |  
| ... | ... | ... | ... |  
| 300 | 0.787751 | 0.523696 | 300 |  
---------------------------------------

Total running time of the script: ( 0 minutes 0.184 seconds)

dist

FUNCTION anfun.dist(x1, y1, x2, y2, p=2.0) RETURNS numeric

Описание: Используя параметры функции, вычисляет расстояние Минковского между двумя точками в двумерном пространстве с параметром **p** по формуле: .

Input

x1: numeric

Первая координата первой точки.

x2: numeric

Первая координата второй точки.

y1: numeric

Вторая координата первой точки.

y2: numeric

Вторая координата второй точки.

p: numeric

Параметр расстояния Минковского. По умолчанию устанавливается значение, равное 2.0.

Output

numeric

Результат математического выражения – расстояние между двумя данными точками.

Example

>> IN  
  
SELECT \* FROM anfun.dist(1.0, 20.0, 4.0, 16.0);  
  
>> OUT  
  
-------------------------------  
| dist |

| numeric |  
-------------------------------  
| 5.00000000000000000000|  
-------------------------------

Total running time of the script: ( 0 minutes 0.170 seconds)

initClusters

FUNCTION anfun.initClusters(\_tbl) RETURNS table(id, X, Y, C)

Описание: Функция возвращает таблицу по её названию. Её данные представляют собой идентификационные номера (столбец **id**) и координаты точек в двумерном пространстве (столбцы **X** и **Y**), а последний столбец (**C**) повторяет первый (**id**). Эта функция была реализована для внутреннего использования в других функциях.

Input

\_tbl: varchar

Название таблицы, столбцы которой функция преобразовывает и возвращает.

Output

table(id, X, Y, C): таблица из четырёх столбцов

id: integer

Представляет собой первый столбец с целочисленными значениями в таблице **\_tbl**. Идентификационный номер точки.

X: numeric

Представляет собой второй столбец с вещественными числами в таблице **\_tbl**. Первая координата точки.

Y: numeric

Представляет собой третий столбец с вещественными числами в таблице **\_tbl**. Вторая координата точки.

C: integer

Представляет собой первый столбец с целочисленными значениями в таблице **\_tbl**. Идентификационнй номер точки.

Example

>> IN  
  
SELECT \* FROM anfun.initClusters('random2DimDotsBlobs(100, 3, 1)');  
  
>> OUT  
  
---------------------------------------  
| id | X | Y | C |  
---------------------------------------  
| 1 | 0.823421 | 0.625892 | 1 |  
| 2 | 0.803723 | 0.612849 | 2 |  
| ... | ... | ... | ... |  
| 300 | 0.146234 | 0.213487 | 300 |  
---------------------------------------

Total running time of the script: ( 0 minutes 0.448 seconds)

maxi

FUNCTION anfun.maxi(a, b) RETURNS numeric

Описание: Принимая в качестве параметров пару чисел, возвращает наибольшее из них.

Input

a: numeric

Входное значение первого параметра.

b: numeric

Входное значение второго параметра.

Output

numeric

Максимальное значение, выбранное из пары входных параметров.

Example

>> IN  
  
SELECT \* FROM anfun.maxi(1.0, 20.7);  
  
>> OUT  
  
-------------------------------  
| maxi |

| numeric |  
-------------------------------  
| 20.7000000000000000000|  
-------------------------------

Total running time of the script: ( 0 minutes 0.146 seconds)

mini

FUNCTION anfun.mini(a, b) RETURNS numeric

Описание: Принимая в качестве параметров пару чисел, возвращает наименьшее из них.

Input

a: numeric

Входное значение первого параметра.

b: numeric

Входное значение второго параметра.

Output

numeric

Минимальное значение, выбранное из пары входных параметров.

Example

>> IN  
  
SELECT \* FROM anfun.mini(1.6, 29.7);  
  
>> OUT  
  
-------------------------------  
| mini |

| numeric |  
-------------------------------  
| 1.60000000000000000000|  
-------------------------------

Total running time of the script: ( 0 minutes 0.140 seconds)

Determinant

FUNCTION anfun.Determinant(sys) RETURNS numeric

Описание: Рекурсивная функция, которая находит определитель матрицы, данной как параметр, и возвращает его значение.

Input

sys: numeric[][]

Матрица, определитель которой необходимо найти. Ограничение: матрица должна быть квадратной.

Output

numeric

Значения определителя матрицы.

Example

>> IN  
  
SELECT anfun.Determinant('{{2,2},{3,4}}');  
  
>> OUT  
  
-------------------------------  
| determinant |

| numeric |  
-------------------------------  
| 2.00000000000000000000|  
-------------------------------

Total running time of the script: ( 0 minutes 0.145 seconds)

TransposeMatrix

FUNCTION anfun.TransposeMatrix(sys) RETURNS numeric[][]

Описание: Функция, которая транспонирует матрицу, данную как параметр, и возвращает её как результат.

Input

sys: numeric[][]

Матрица, которую необходимо транспонировать. Ограничение: матрица должна быть квадратной.

Output

numeric[][]

Матрица, представляющая собой транспонированную исходную матрицу.

Example

>> IN  
  
SELECT anfun.TransposeMatrix('{{2,2},{3,4}}');  
  
>> OUT  
  
-------------------------------  
| transposematrix |

| numeric[] |  
-------------------------------  
| {{2,3},{2,4}}|  
-------------------------------

Total running time of the script: ( 0 minutes 0.237 seconds)

InvertMatrix

FUNCTION anfun.InvertMatrix(sys) RETURNS numeric[][]

Описание: Функция, которая находит и возвращает матрицу, обратную данной как параметр.

Input

sys: numeric[][]

Матрица, обратную которой необходимо найти.

Output

numeric[][]

Матрица, обратная исходной.

Example

>> IN  
  
SELECT anfun.InvertMatrix('{{2,2},{3,4}}');  
  
>> OUT  
  
-------------------------------  
| invertmatrix |

| numeric[] |  
-------------------------------  
| {{2,-1},{-1.5,1}}|  
-------------------------------

Total running time of the script: ( 0 minutes 0.206 seconds)

MultMatrix

FUNCTION anfun.MultMatrix(G, H) RETURNS numeric[][]

Описание: Функция, которая вычисляет произведение двух матриц, данных как параметры, и возвращает результирующую матрицу.

Input

G: numeric[][]

Матрица, представляющая собой первый множитель.

H: numeric[][]

Матрица, представляющая собой второй множитель. Ограничение: количество строк матрицы **H** должно совпадать с кольчеством столбцов матрицы **G**.

Output

numeric[][]

Матрица, являющаяся результатом произведения исходных матриц.

Example

>> IN  
  
SELECT anfun.MultMatrix('{{2,2},{3,4}}', '{{2,2},{3,4}}');  
  
>> OUT  
  
-------------------------------  
| multmatrix |

| numeric[] |  
-------------------------------  
| {{10,12},{18,22}}|  
-------------------------------

Total running time of the script: ( 0 minutes 0.210 seconds)

Cramer

FUNCTION anfun.Cramer(sys, r) RETURNS numeric[]

Описание: Функция, представляющая собой метод Крамера. Этот алгоритм решает квадратную систему линейных уравнений с ненулевым определителем основной матрицы. Функция возвращает результирующий вектор.

Input

sys: numeric[][]

Матрица, представляющая коэффициенты из левой части системы линейных уравнений. Ограничение: матрица должна быть квадратной и её определитель не должен равняться нулю.

r: numeric[]

Вектор, представляющий собой правую часть системы линейных уравнений.

Output

numeric[]

Вектор, являющийся решением данной системы уравнений.

Example

>> IN  
  
SELECT anfun.Cramer('{{2,2},{3,4}}', '{5,8}');  
  
>> OUT  
  
-------------------------------  
| cramer |

| numeric[] |  
-------------------------------  
| {2,0.5}|  
-------------------------------

Total running time of the script: ( 0 minutes 0.219 seconds)

OLS

FUNCTION anfun.OLS(A, b) RETURNS numeric[]

Описание: Функция, представляющая собой метод наименьших квадратов. Этот метод может использоваться для решения неопределенных систем уравнений, т.е. когда количество уравнений превышает количество неизвестных, для поиска решений в случае обычных нелинейных систем уравнений, для приближения точечных значений некоторой функции. В данном пакете этот алгоритм используется для нахождения коэффициентов в регрессионном анализе. Функция возвращает результирующий вектор.

Input

A: numeric[][]

Матрица, представляющая коэффициенты из левой части системы уравнений.

b: numeric[]

Вектор, представляющий собой правую часть системы уравнений.

Output

numeric[]

Вектор, являющийся точным или приближенным решением данной системы уравнений.

Example

>> IN  
  
SELECT anfun.OLS('{{2,2},{3,4}}', '{5,8}');  
  
>> OUT  
  
-------------------------------  
| OLS |

| numeric[] |  
-------------------------------  
| {2,0.5}|  
-------------------------------

Total running time of the script: ( 0 minutes 0.204 seconds)

I

FUNCTION anfun.I(\_tbl) RETURNS table(X, Y)

Описание: Иногда удобнее работать не с самим временным рядом, а с изменениями его значений. Такое преобразование называется интеграцией временного ряда. Однако чаще используют интеграцию второго порядка (т.е. вычисляются изменения изменений значений временного ряда). Такая модель называется интеграцией (или интегрированием), т.к. в конечном счет мы будем выполнять обратное преобразование в исходный временной ряд. Данная функция выполняет интеграцию данного временного ряда и возвращает таблицу с его результатами.

Input:

\_tbl: varchar

Название таблицы со значениями временного ряда.

Output:

table( X, Y): таблица из двух столбцов

X: integer

Первый столбец выводимой таблицы. Представляет собой значения временного ряда из таблицы \_**tbl**.

Y: numeric

Второй столбец выводимой таблицы. Представляет собой результаты интеграции.

Example

>> IN

SELECT \* FROM anfun.I('anfun.randomTimeSeries(100, 10, 10)')

>> OUT

---------------------

| X | Y |  
---------------------  
| 1 | 0.000000 |  
| 2 | 1.945126 |  
| ... | ... |  
| 100 | 0.587379 |

---------------------

Total running time of the script: ( 0 minutes 3.113 seconds)

MA

FUNCTION anfun.MA(\_tbl, q) RETURNS table(X, Y)

Описание: Функция представляет собой модель скользящего среднего. Этот алгоритм избавляет временной ряд от выбросов путём его сглаживания. Степень сглаживания определяется параметром **q**: чем его значение больше, тем сильнее будет сглаживание. Данная функция возвращает таблицу со сглаженным и значениями временного ряда.

Input:

\_tbl: varchar

Название таблицы со значениями временного ряда.

q: integer

Степень сглаживания.

Output:

table( X, Y): таблица из двух столбцов

X: integer

Первый столбец выводимой таблицы. Представляет собой значения временного ряда из таблицы \_**tbl**.

Y: numeric

Второй столбец выводимой таблицы. Представляет собой результаты сглаживания с помощью модели скользящего среднего.

Example

>> IN

SELECT \* FROM anfun.I('anfun.randomTimeSeries(100, 10, 15)', 3)

>> OUT

---------------------

| X | Y |  
---------------------  
| 1 | -1.172529 |  
| 2 | 0.412259 |  
| ... | ... |  
| 100 | 4.507527 |

---------------------

Total running time of the script: ( 0 minutes 1.343 seconds)

Random Functions

random2DimDotsCircles

FUNCTION anfun.random2DimDotsCircles(k, h, w) RETURNS table(id, X, Y)

Описание: Результатом данной функции является таблица с **k\*h** строками, где первый столбец (**id**) – порядковый номер записи, а второй и третий (**X** и **Y**) - пара случайных значений (точек в двумерном пространстве распределённых по **h** группам).

Input

k: integer

Количество точек в каждой группе.

h: integer

Количество групп, представляющих собой окружности в двумерном пространстве.

w: numeric

Ширина окружности.

Output

table(id, X, Y): таблица из трёх столбцов

id: integer

Первый столбец выводимой таблицы. Обозначает порядковый номер записи.

X: numeric

Второй столбец выводимой таблицы. Представляет собой случайное значение, обозначающее первую координату точки.

Y: numeric

Третий столбец выводимой таблицы. Представляет собой случайное значение, обозначающее вторую координату точки.

Example

>> IN  
  
SELECT \* FROM anfun.random2DimDotsCircles(3, 100, 0.5);  
  
>> OUT  
  
-------------------------------  
| id | X | Y |  
-------------------------------  
| 1 | 0.846090 | 0.538731 |  
| 2 | -0.77349 | 0.649459 |  
| ... | ... | ... |  
| 300 | -39.3421 | -91.9349 |  
-------------------------------

Total running time of the script: ( 0 minutes 0.357 seconds)

random2DimDotsBlobs

FUNCTION anfun.random2DimDotsBlobs(k, h, w) RETURNS table(id, X, Y)

Описание: Результатом данной функции является таблица с **k\*h** строками, где первый столбец (**id**) – порядковый номер записи, а второй и третий (**X** и **Y**) - пара случайных значений (точек в двумерном пространстве распределённых по **h** группам).

Input

k: integer

Количество точек в каждой группе.

h: integer

Количество групп, представляющих собой сгустки точек в двумерном пространстве.

w: numeric

Размер сгустка точек.

Output

table(id, X, Y): таблица из трёх столбцов

id: integer

Первый столбец выводимой таблицы. Обозначает порядковый номер записи.

X: numeric

Второй столбец выводимой таблицы. Представляет собой случайное значение, обозначающее первую координату точки.

Y: numeric

Третий столбец выводимой таблицы. Представляет собой случайное значение, обозначающее вторую координату точки.

Example

>> IN  
  
SELECT \* FROM anfun.random2DimDotsBlobs(3, 100, 0.5);  
  
>> OUT  
  
-------------------------------  
| id | X | Y |  
-------------------------------  
| 1 | 0.432227 | 0.356194 |  
| 2 | 0.432208 | 0.356218 |  
| ... | ... | ... |  
| 300 | 0.227779 | 0.648105 |  
-------------------------------

Total running time of the script: ( 0 minutes 0.250 seconds)

randomTimeSeries

FUNCTION anfun.randomTimeSeries(k, o, l) RETURNS table(X, Y)

Описание: Результатом данной функции является таблица с **k** строками, где первый столбец – **X** – номер записи, а второй – **Y** – значение записи. Интерпретируемый временной ряд содержит сезонную составляющую с фиксированным количеством сезонов (параметр **o**) за период генерации. Параметр **l** представляет собой шумовой коэффициент: чем больше его значение, тем больше уровень помех в значениях временного ряда.

Input

k: integer

Количество записей.

o: numeric

Количество сезонов.

l: numeric

Шумовой коэффициент.

Output

table(X, Y): таблица из двух столбцов

X: integer

Первый столбец выводимой таблицы. Обозначает порядковый номер записи (момент времени).

Y: numeric

Второй столбец выводимой таблицы. Представляет собой значение записи.

Example

>> IN  
  
SELECT \* FROM anfun.randomTimeSeries(100, 10, 0.5);  
  
>> OUT  
  
--------------------  
| X | Y |  
--------------------  
| 1 | 0.031519 |  
| 2 | 0.039775 |  
| ... | ... |  
| 100 | 64.59560 |  
--------------------

Total running time of the script: ( 0 minutes 0.239 seconds)

Forecasting Functions

expForecast

FUNCTION anfun.expForecast(\_tbl, alpha, l, ll) RETURNS table(X, Y)

Описание: Функция использует модель одинарного экспоненциального сглаживания, чтобы сгладить данные временного ряда и выявить чёткую тенденцию временного ряда и сезонные составляющие, а также спрогнозировать дальнейшее поведение данных. Алгоритм одинарного экспоненциального сглаживания основывается на постоянном пересмотре прогнозных значений. Они складываются из предыдущего сглаженного значения и некоторой доли ошибки предыдущего сглаживания. Величина этой ошибки, которая используется для корректировки прогноза, определяется константой сглаживания (параметр функции **alpha**). Функция возвращает таблицу с результатами сглаживания (первые **l** записей) и результатами прогнозирования (последние **ll** записей).

Input:

\_tbl: varchar

Название таблицы со значениями временного ряда, которые функция сглаживает и прогнозирует.

alpha: numeric

Коэффициент сглаживания ряда, 0≤**alpha**≤1.

l: integer

Количество записей со сглаженными данными.

ll: integer

Количество записей прогноза.

Output:

table( X, Y): таблица из двух столбцов

X: integer

Первый столбец выводимой таблицы. Представляет собой значения временного ряда из таблицы \_**tbl**.

Y: numeric

Второй столбец выводимой таблицы. Представляет собой результаты сглаживания и результаты прогнозирования.

Example

>> IN

SELECT \* FROM anfun.expForecast('anfun.randomTimeSeries(1000, 10, 10)', 0.1, 800, 200)

>> OUT

---------------------

| X | Y |  
---------------------  
| 1 | -0.797841 |  
| 2 | -0.761926 |  
| ... | ... |  
| 1000 |-17.194240 |

---------------------

Total running time of the script: ( 0 minutes 2.140 seconds)

expTrendForecast

FUNCTION anfun.expTrendForecast(\_tbl, alpha, beta, l, ll) RETURNS table(X, Y)

Описание: Функция использует модель двойного экспоненциального сглаживания, чтобы сгладить данные временного ряда и выявить чёткую тенденцию временного ряда и сезонные составляющие, а также спрогнозировать дальнейшее поведение данных. Возвращает таблицу с результатами сглаживания (первые **l** записей) и результатами прогнозирования (последние **ll** записей).

Input:

\_tbl: varchar

Название таблицы со значениями временного ряда, которые функция сглаживает и прогнозирует.

alpha: numeric

Первый коэффициент сглаживания, 0≤**alpha**≤1.

beta: numeric

Второй коэффициент сглаживания, 0≤**beta**≤1.

l: integer

Количество записей со сглаженными данными.

ll: integer

Количество записей прогноза.

Output:

table( X, Y): таблица из двух столбцов

X: integer

Первый столбец выводимой таблицы. Представляет собой значения временного ряда из таблицы \_**tbl**.

Y: numeric

Второй столбец выводимой таблицы. Представляет собой результаты сглаживания и результаты прогнозирования.

Example

>> IN

SELECT \* FROM anfun.expTrendForecast('anfun.randomTimeSeries(1000, 10, 10)', 0.1, 0.1, 800, 200)

>> OUT

---------------------

| X | Y |  
---------------------  
| 1 | -1.280930 |  
| 2 | 10.031670 |  
| ... | ... |  
| 1000 |-159.594789|

---------------------

Total running time of the script: ( 0 minutes 2.265 seconds)

holtWintersForecast

FUNCTION anfun.holtWintersForecast(\_tbl, alpha, beta, gamma, p, l, ll) RETURNS table(X, Y)

Описание: Функция использует модель Хольта-Винтерса, чтобы сгладить данные временного ряда и выявить чёткую тенденцию временного ряда и сезонные составляющие, а также спрогнозировать дальнейшее поведение данных. Возвращает таблицу с результатами сглаживания (первые **l** записей) и результатами прогнозирования (последние **ll** записей).

Input:

\_tbl: varchar

Название таблицы со значениями временного ряда, которые функция сглаживает и прогнозирует.

alpha: numeric

Первый коэффициент сглаживания, 0≤**alpha**≤1.

beta: numeric

Второй коэффициент сглаживания, 0≤**beta**≤1.

gamma: numeric

Третий коэффициент сглаживания, 0≤**gamma**≤1.

P: integer

Количество периодов.

l: integer

Количество записей со сглаженными данными.

ll: integer

Количество записей прогноза.

Output:

table( X, Y): таблица из двух столбцов

X: integer

Первый столбец выводимой таблицы. Представляет собой значения временного ряда из таблицы \_**tbl**.

Y: numeric

Второй столбец выводимой таблицы. Представляет собой результаты сглаживания и результаты прогнозирования.

Example

>> IN

SELECT \* FROM anfun.holtWintersForecast('anfun.randomTimeSeries(1000, 10, 10)', 0.1, 0.1, 0.1, 100, 800, 200)

>> OUT

---------------------

| X | Y |  
---------------------  
| 1 | 21.342371 |  
| 2 | 21.002482 |  
| ... | ... |  
| 1000 | 14.744813 |

---------------------

Total running time of the script: ( 0 minutes 7.226 seconds)

linearRegression

FUNCTION anfun.linearRegression(\_tbl, l, ll) RETURNS table(X, Y)

Описание: Функция использует модель линейной регрессии, чтобы выявить общее направление движения процесса. Для вычисления коэффициентов здесь используется метод наименьших квадратов. Возвращает таблицу с обработанными исходными данными (первые **l** записей) и результатами прогнозирования (последние **ll** записей).

Input:

\_tbl: varchar

Название таблицы со значениями временного ряда, которые функция прогнозирует.

l: integer

Количество записей обработанных исходных данных.

ll: integer

Количество записей прогноза.

Output:

table( X, Y): таблица из двух столбцов

X: integer

Первый столбец выводимой таблицы. Представляет собой значения временного ряда из таблицы \_**tbl**.

Y: numeric

Второй столбец выводимой таблицы. Представляет собой результаты прогнозирования.

Example

>> IN

SELECT \* FROM anfun.linearRegression('anfun.randomTimeSeries(1000, 1, 10)', 800, 200)

>> OUT

---------------------

| X | Y |  
---------------------  
| 1 | 17.639771 |  
| 2 | 18.262286 |  
| ... | ... |  
| 1000 | 11.322812 |

---------------------

Total running time of the script: ( 0 minutes 1.226 seconds)

AR

FUNCTION anfun.AR(\_tbl, p, l, ll) RETURNS table(X, Y)

Описание: Функция использует модель авторегрессии. В данной модели прогнозирования значений процесса в данный момент времени выражается, как линейная комбинация предыдущих значени процесса. Количество предыдущих значений, которые будут взяты для вычисления прогноза, регулируется пользователем с помощью параметра **p.** Для вычисления коэффициентов здесь используется метод наименьших квадратов. Функция возвращает таблицу с исходными данными (первые **l** записей) и результатами прогнозирования (последние **ll** записей).

Input:

\_tbl: varchar

Название таблицы со значениями временного ряда, которые функция прогнозирует.

p: integer

Порядок авторегрессии – количество предыдущих значений процесса, которые будут использованы для вычисления прогноза. Ограничение: **p** < (**l** - 1) / 2.

l: integer

Количество записей обработанных исходных данных.

ll: integer

Количество записей прогноза.

Output:

table( X, Y): таблица из двух столбцов

X: integer

Первый столбец выводимой таблицы. Представляет собой значения временного ряда из таблицы \_**tbl**.

Y: numeric

Второй столбец выводимой таблицы. Представляет собой результаты прогнозирования.

Example

>> IN

SELECT \* FROM anfun.AR('anfun.randomTimeSeries(10, 1, 2)', 3, 8, 2)

>> OUT

---------------------

| X | Y |  
---------------------  
| 1 | -0.314016 |  
| 2 | -0.990943 |  
| ... | ... |  
| 10 | -0.841335 |

---------------------

Total running time of the script: ( 0 minutes 3.016 seconds)

ARIMA

FUNCTION anfun.ARIMA(\_tbl, p, q, l, ll) RETURNS table(X, Y)

Описание: Функция представляет собой модель прогнозирования ARIMA. ARIMA является одним из самых популярных алгоритмов прогнозирования. Эта модель основана на авторегрессии значений, обработанных с помощью интеграции и модели скользящего среднего. Таким образом, эта функция сначала находит значения изменений временного ряда 2-го порядка, затем их сглаживает, прогнозирует, и, наконец, т.к. мы спрогнозировали лишь изменение будущего значения процесса преобразовывает обратно, чтобы получить конечный результат. Функция возвращает таблицу с результатами сглаживания (первые **l** записей) и результатами прогнозирования (последние **ll** записей).

Input:

\_tbl: varchar

Название таблицы со значениями временного ряда, которые функция прогнозирует.

p: integer

Порядок авторегрессии – количество предыдущих значений процесса, которые будут использованы для вычисления прогноза. Ограничение: **p** < (**l** - 1) / 2.

q: integer

Степень сглаживания

l: integer

Количество записей обработанных исходных данных.

ll: integer

Количество записей прогноза.

Output:

table( X, Y): таблица из двух столбцов

X: integer

Первый столбец выводимой таблицы. Представляет собой значения временного ряда из таблицы \_**tbl**.

Y: numeric

Второй столбец выводимой таблицы. Представляет собой результаты прогнозирования.

Example

>> IN

SELECT \* FROM anfun.ARIMA('anfun.randomTimeSeries(10, 1, 2)', 3, 3, 8, 2)

>> OUT

---------------------

| X | Y |  
---------------------  
| 1 | 0.658999 |  
| 2 | 1.578643 |  
| ... | ... |  
| 10 | 2.844895 |

---------------------

Total running time of the script: ( 0 minutes 0.551 seconds)

Forecasting Metrics

MAE

FUNCTION anfun.MAE(\_tbl1, \_tbl2) RETURNS numeric

Описание: Принимая в качестве параметров две таблицы, функция возвращает число - значение средней абсолютной ошибки прогноза. Эта метрика относительна.

Input

\_tbl1: varchar

Название таблицы с начальными данными.

\_tbl2: varchar

Название таблицы с результатами прогнозирования.

Output

numeric

Значение средней абсолютной ошибки.

Example

>> IN

SELECT \* INTO TS FROM anfun.randomTimeSeries(1000, 10, 10);

SELECT \* INTO HWF FROM anfun.holtWintersForecast('TS', 0.1, 0.1, 0.1, 100, 800, 200);

SELECT anfun.MAE('TS', 'HWF')

>> OUT

---------------------------------------  
| mae |

| numeric |  
---------------------------------------  
| 3.0046881497135595 |

---------------------------------------

Total running time of the script: ( 0 minutes 0.096 seconds)

RMSE

FUNCTION anfun.RMSE(\_tbl1, \_tbl2) RETURNS numeric

Описание: Принимая в качестве параметров две таблицы, возвращает число - значение среднеквадратичного отклонения. Эта метрика относительна.

Input

\_tbl1: varchar

Название таблицы с начальными данными.

\_tbl2: varchar

Название таблицы с результатами прогнозирования.

Output

numeric

Значение среднеквадратичного отклонения.

Example

>> IN

SELECT \* INTO TS FROM anfun.randomTimeSeries(1000, 10, 10);

SELECT \* INTO HWF FROM anfun.holtWintersForecast('TS', 0.1, 0.1, 0.1, 100, 800, 200);

SELECT anfun.RMSE('TS', 'HWF')

>> OUT

---------------------------------------  
| rmse |

| numeric |  
---------------------------------------  
| 12.03128778370900196048368060483521|

---------------------------------------

Total running time of the script: ( 0 minutes 0.131 seconds)

MPE

FUNCTION anfun.MPE(\_tbl1, \_tbl2) RETURNS numeric

Описание: Принимая в качестве параметров две таблицы, возвращает число - значение среднего процента ошибки прогноза. Эта метрика относительна.

Input

\_tbl1: varchar

Название таблицы с начальными данными.

\_tbl2: varchar

Название таблицы с результатами прогнозирования.

Output

numeric

Значение среднего процента ошибки прогноза.

Example

>> IN

SELECT \* INTO TS FROM anfun.randomTimeSeries(1000, 10, 10);

SELECT \* INTO HWF FROM anfun.holtWintersForecast('TS', 0.1, 0.1, 0.1, 100, 800, 200);

SELECT anfun.MPE('TS', 'HWF')

>> OUT

---------------------------------------  
| mpe |

| numeric |  
---------------------------------------  
| 22.006435440674266215220600 |

---------------------------------------

Total running time of the script: ( 0 minutes 0.093 seconds)

MAPE

FUNCTION anfun.MAPE(\_tbl1, \_tbl2) RETURNS numeric

Описание: Принимая в качестве параметров две таблицы, возвращает число - значение средней относительной ошибки прогноза. Эта метрика относительна.

Input

\_tbl1: varchar

Название таблицы с начальными данными.

\_tbl2: varchar

Название таблицы с результатами прогнозирования.

Output

numeric

Значение средней относительной ошибки прогноза.

Example

>> IN

SELECT \* INTO TS FROM anfun.randomTimeSeries(1000, 10, 10);

SELECT \* INTO HWF FROM anfun.holtWintersForecast('TS', 0.1, 0.1, 0.1, 100, 800, 200);

SELECT anfun.MAPE('TS', 'HWF')

>> OUT

---------------------------------------  
| mape |

| numeric |  
---------------------------------------  
| 36.189760486552313150113400 |

---------------------------------------

Total running time of the script: ( 0 minutes 0.133 seconds)

AD

FUNCTION anfun.AD(\_tbl1, \_tbl2) RETURNS numeric

Описание: Принимая в качестве параметров две таблицы, возвращает число - значение абсолютного отклонения от среднего. Эта метрика относительна.

Input

\_tbl1: varchar

Название таблицы с начальными данными.

\_tbl2: varchar

Название таблицы с результатами прогнозирования.

Output

numeric

Значение абсолютного отклонения от среднего.

Example

>> IN

SELECT \* INTO TS FROM anfun.randomTimeSeries(1000, 10, 10);

SELECT \* INTO HWF FROM anfun.holtWintersForecast('TS', 0.1, 0.1, 0.1, 100, 800, 200);

SELECT anfun.AD('TS', 'HWF')

>> OUT

---------------------------------------  
| ad |

| numeric |  
---------------------------------------  
| 12654.8824879395006672 |

---------------------------------------

Total running time of the script: ( 0 minutes 0.128 seconds)

MAD

FUNCTION anfun.MAD(\_tbl1, \_tbl2) RETURNS numeric

Описание: Принимая в качестве параметров две таблицы, возвращает число - значение среднего абсолютного отклонения. Эта метрика относительна.

Input

\_tbl1: varchar

Название таблицы с начальными данными.

\_tbl2: varchar

Название таблицы с результатами прогнозирования.

Output

numeric

Значение среднего абсолютного отклонения.

Example

>> IN

SELECT \* INTO TS FROM anfun.randomTimeSeries(1000, 10, 10);

SELECT \* INTO HWF FROM anfun.holtWintersForecast('TS', 0.1, 0.1, 0.1, 100, 800, 200);

SELECT anfun.MAD('TS', 'HWF')

>> OUT

---------------------------------------  
| mad |

| numeric |  
---------------------------------------  
| 12.6548824879395007 |

---------------------------------------

Total running time of the script: ( 0 minutes 0.182 seconds)

R2

FUNCTION anfun.R2(\_tbl1, \_tbl2) RETURNS numeric

Описание: Принимая в качестве параметров две таблицы, возвращает число - значение коэффициента детерминации. Значение этой метрики не зависит от единиц измерения данных и находится в диапазоне от 0 до 1. Чем ближе значение коэффициента к 1 - тем лучше качество подгонки, и наоборот.

Input

\_tbl1: varchar

Название таблицы с начальными данными.

\_tbl2: varchar

Название таблицы с результатами прогнозирования.

Output

numeric

Значение коэффициента детерминации.

Example

>> IN

SELECT \* INTO TS FROM anfun.randomTimeSeries(1000, 10, 10);

SELECT \* INTO HWF FROM anfun.holtWintersForecast('TS', 0.1, 0.1, 0.1, 100, 800, 200);

SELECT anfun.R2('TS', 'HWF')

>> OUT

---------------------------------------  
| r2 |

| numeric |  
---------------------------------------  
| 0.89704672840185367181868157766363 |

---------------------------------------

Total running time of the script: ( 0 minutes 0.119 seconds)

THEIL

FUNCTION anfun.THEIL(\_tbl1, \_tbl2) RETURNS numeric

Описание: Принимая в качестве параметров две таблицы, возвращает число – значение коэффициента несоответствия Тейла. Значение этой метрики не зависит от единиц измерения данных и находится в диапазоне от 0 до бесконечности. Чем ближе значение коэффициента к 0 - тем лучше качество подгонки.

Input

\_tbl1: varchar

Название таблицы с начальными данными.

\_tbl2: varchar

Название таблицы с результатами прогнозирования.

Output

numeric

Значение коэффициента несоответствия Тейла.

Example

>> IN

SELECT \* INTO TS FROM anfun.randomTimeSeries(1000, 10, 10);

SELECT \* INTO HWF FROM anfun.holtWintersForecast('TS', 0.1, 0.1, 0.1, 100, 800, 200);

SELECT anfun.THEIL('TS', 'HWF')

>> OUT

---------------------------------------  
| theil |

| numeric |  
---------------------------------------  
| 0.16016392809032078406497967566201 |

---------------------------------------

Total running time of the script: ( 0 minutes 0.130 seconds)

Pearson

FUNCTION anfun.Pearson(\_tbl1, \_tbl2) RETURNS numeric

Описание: Принимая в качестве параметров две таблицы, возвращает число – значение коэффициента корреляции Пирсона. Значение этой метрики не зависит от единиц измерения данных и находится в диапазоне от 0 до 1. Чем ближе значение коэффициента к 1 - тем лучше качество подгонки.

Input

\_tbl1: varchar

Название таблицы с начальными данными.

\_tbl2: varchar

Название таблицы с результатами прогнозирования.

Output

numeric

Значение коэффициента корреляции Пирсона.

Example

>> IN

SELECT \* INTO TS FROM anfun.randomTimeSeries(1000, 10, 10);

SELECT \* INTO HWF FROM anfun.holtWintersForecast('TS', 0.1, 0.1, 0.1, 100, 800, 200);

SELECT anfun.Pearson('TS', 'HWF')

>> OUT

---------------------------------------  
| pearson |

| numeric |  
---------------------------------------  
| 0.99016392809032078406497967566201 |

---------------------------------------

Total running time of the script: ( 0 minutes 0.130 seconds)

Clustering Functions

DBSCAN

FUNCTION anfun.DBSCAN(\_tbl, n, e, p=2.0) RETURNS table(id, X, Y, C)

Описание: Принимая в качестве параметров таблицу с данными, функция кластеризует их и возвращает таблицу с дополнительным столбцом, в котором определены номера кластеров. Работа алгоритма заключается в том, что для каждой точки кластера её соседство заданного радиуса должно содержать не менее некоторого числа точек, это число точек задаётся пороговым значением.

Input

\_tbl: varchar

Название таблицы, столбцы которой функция возвращает.

n: integer

Минимальное количество соседей, находящихся на расстоянии не большем, чем **e**.

e: numeric

Максимальный радиус до соседа.

p: numeric

Параметр расстояния Минковского. По умолчанию устанавливается значение, равное 2.0.

Output

table(id, X, Y, C): таблица из четырёх столбцов

id: integer

Представляет собой первый столбец с целочисленными значениями в таблице \_tbl.

X: numeric

Представляет собой второй столбец с вещественными числами в таблице \_tbl.

Y: numeric

Представляет собой третий столбец с вещественными числами в таблице \_tbl.

C: integer

Представляет собой четвёртый столбец с целочисленными значениями в таблице \_tbl, которые являются номерами кластеров.

Example

>> IN

SELECT \* FROM anfun.DBSCAN('anfun.random2DimDotsCircles(50,2,1)', 3, 0.5)

>> OUT

-----------------------------------------------------------

| id | X | Y | C |  
-----------------------------------------------------------  
| 66 | 1.21016401631128 | -1.66603106278408 | -1 |  
| 82 | 1.79869858114246 | 1.79869858114246 | -1 |  
| ... | ... | ... | ... |  
| 100 | 1.17477098984541 | 1.79869858114246 | 52 |

-----------------------------------------------------------

Total running time of the script: ( 0 minutes 2.140 seconds)

Kmeans

FUNCTION anfun.Kmeans(\_tbl, k, p=2.0) RETURNS table(id, X, Y, C)

Описание: Принимая в качестве параметров таблицу с данными, функция кластеризует их и возвращает таблицу с дополнительным столбцом, в котором определены номера кластеров. Ее работа основана на вычислении центроидов и последующей минимизации целевой функции. Необходимым условием для работы данного алгоритма является задание количество кластеров.

Input:

\_tbl: varchar

Название таблицы, столбцы которой функция возвращает.

k: integer

Количество кластеров.

p: numeric

Параметр расстояния Минковского. По умолчанию устанавливается значение, равное 2.0.

Output

table(id, X, Y, C): таблица из четырёх столбцов

id: integer

Представляет собой первый столбец с целочисленными значениями в таблице \_tbl.

X: numeric

Представляет собой второй столбец с вещественными числами в таблице \_tbl.

Y: numeric

Представляет собой третий столбец с вещественными числами в таблице \_tbl.

C: integer

Представляет собой четвёртый столбец с целочисленными значениями в таблице \_tbl, которые являются номерами кластеров.

Example

>> IN

SELECT \* FROM anfun.KMEANS('anfun.random2DimDotsBlobs(100,5,0.6)', 5);

>> OUT

-------------------------------------------------------------

| id | X | Y | C |  
-------------------------------------------------------------  
| 1 | -0.886080343466947 | -0.100481258086202 | 3 |  
| 2 | 0.127891907945575 | -0.970856501812635 | 3 |  
| ... | ... | ... | ... |  
| 100 | -1.88230500926938 | -0.457830247602034 | 3 |

-------------------------------------------------------------

Total running time of the script: ( 0 minutes 0.438 seconds)

Kmedians

FUNCTION anfun.Kmedians(\_tbl, k, p=2.0) RETURNS table(id, X, Y, C)

Возвращает таблицу с результатами кластеризации.

Принимая в качестве параметров таблицу с данными, функция кластеризует их и возвращает таблицу с дополнительным столбцом, в котором определены номера кластеров.

Input

\_tbl: varchar

Название таблицы, столбцы которой функция возвращает.

k: integer

Количество кластеров.

p: numeric

Параметр расстояния Минковского. По умолчанию устанавливается значение, равное 2.0.

Output

table(id, X, Y, C): таблица из четырёх столбцов

id: integer

Представляет собой первый столбец с целочисленными значениями в таблице \_tbl.

X: numeric

Представляет собой второй столбец с вещественными числами в таблице \_tbl.

Y: numeric

Представляет собой третий столбец с вещественными числами в таблице \_tbl.

C: integer

Представляет собой четвёртый столбец с вещественными числами в таблице \_tbl, которые являются номерами кластеров.

Example

>> IN

SELECT \* FROM anfun.Kmedians(‘anfun.random2DimDotsBlobs(100,5,0.6)’,5);

>> OUT

-------------------------------------------------------------

| id | X | Y | C |  
-------------------------------------------------------------  
| 1 | -0.886080343466947 | -0.100481258086202 | 3 |  
| 2 | 0.127891907945575 | -0.970856501812635 | 3 |  
| ... | ... | ... | ... |  
| 100 | -1.88230500926938 | -0.457830247602034 | 5 |

-------------------------------------------------------------

Total running time of the script: ( 0 minutes 0.401 seconds)

FOREL

FUNCTION anfun.FOREL(\_tbl, r, p=2.0) RETURNS table(id, X, Y, C)

Возвращает таблицу с результатами кластеризации.

Принимая в качестве параметров таблицу с данными, функция кластеризует их и возвращает таблицу с дополнительным столбцом, в котором определены номера кластеров.

Input:

\_tbl: varchar

Название таблицы, столбцы которой функция возвращает.

r: integer

Радиус на расстоянии которого могут находиться точки кластера.

p: numeric

Параметр расстояния Минковского. По умолчанию устанавливается значение, равное 2.0.

Output

table(id, X, Y, C): таблица из четырёх столбцов

id: integer

Представляет собой первый столбец с целочисленными значениями в таблице \_tbl.

X: numeric

Представляет собой второй столбец с вещественными числами в таблице \_tbl.

Y: numeric

Представляет собой третий столбец с вещественными числами в таблице \_tbl.

C: integer

Представляет собой четвёртый столбец с вещественными числами в таблице \_tbl, которые являются номерами кластеров.

Example

>> IN

SELECT \* FROM anfun.FOREL('anfun.random2DimDotsBlobs(100,5,1)',0.1);

>> OUT

-------------------------------------------------------------

| id | X | Y | C |  
-------------------------------------------------------------  
| 1 | -0.886080343466947 | -0.100481258086202 | 1 |  
| 2 | 0.127891907945575 | -0.970856501812635 | 1 |  
| ... | ... | ... | ... |  
| 100 | -1.88230500926938 | -0.457830247602034 | 5 |

-------------------------------------------------------------

Total running time of the script: ( 0 minutes 0.325 seconds)

Agglomerative

FUNCTION anfun.Aglomerative(\_tbl, metric, p=2.0) RETURNS table(id, X, Y, C)

Возвращает таблицу с результатами кластеризации.

Принимая в качестве параметров таблицу с данными, функция кластеризует их и возвращает таблицу с дополнительным столбцом, в котором определены номера кластеров.

Input:

\_tbl: varchar

Название таблицы, столбцы которой функция возвращает.

metric: integer

Значение метрики на основе которой рассчитывается наилучшее разбиение. Значение 1 соответсвует метрике Sillhouette, значение 2 – метрике Dunn

p: numeric

Параметр расстояния Минковского. По умолчанию устанавливается значение, равное 2.0.

Output

table(id, X, Y, C): таблица из четырёх столбцов

id: integer

Представляет собой первый столбец с целочисленными значениями в таблице \_tbl.

X: numeric

Представляет собой второй столбец с вещественными числами в таблице \_tbl.

Y: numeric

Представляет собой третий столбец с вещественными числами в таблице \_tbl.

C: integer

Представляет собой четвёртый столбец с вещественными числами в таблице \_tbl, которые являются номерами кластеров.

Example

>> IN

SELECT \* FROM anfun.Aglomerative('anfun.random2DimDotsBlobs(100,5,1)',1);

>> OUT

-------------------------------------------------------------

| id | X | Y | C |  
-------------------------------------------------------------  
| 1 | -0.886080343466947 | -0.100481258086202 | 1 |  
| 2 | 0.127891907945575 | -0.970856501812635 | 21 |  
| ... | ... | ... | ... |  
| 100 | -1.88230500926938 | -0.457830247602034 | 81 |

-------------------------------------------------------------

Total running time of the script: ( 0 minutes 24.012 seconds)

Clustering Metrics

Sillhuette

FUNCTION anfun.Sillhuette(\_data, p=2.0) RETURNS numeric

Описание: Принимая в качестве параметров таблицу, возвращает число - значение метрики Силуэт. Данная метрика не предполагает знания истинных меток объектов, и позволяет оценить качество кластеризации, используя только сами данные и результат кластеризации. Sillhuette показывает, насколько среднее расстояние до объектов своего кластера отличается от среднего расстояния до объектов других кластеров.

Input

\_data: varchar

Название таблицы с данными.

p: numeric

Параметр расстояния Минковского. По умолчанию устанавливается значение, равное 2.0.

Output

numeric

Значение метрики Силуэт для данной кластеризации.

Example

>> IN

SELECT \* INTO KMC FROM anfun.KMEANS('anfun.random2DimDotsBlobs(100,5,0.6)', 5);

SELECT anfun.Sillhuette('KMC')

>> OUT

---------------------------------------  
| sillhuette |

| numeric |  
---------------------------------------  
| 0.98712756208706756961518657458291 |

---------------------------------------

Total running time of the script: ( 0 minutes 4.124 seconds)

Dunn

FUNCTION anfun.Dunno(\_data, p=2.0) RETURNS numeric

Возвращает число – значение метрики

Принимая в качестве параметров таблицу, возвращает число - зачение метрики Dunn.

Input

\_data: varchar

Название таблицы с данными.

p: numeric

Параметр расстояния Минковского. По умолчанию устанавливается значение, равное 2.0.

Output

numeric

Значение метрики Dunn для данной кластеризации.

Example

>> IN

SELECT \* INTO KMC FROM anfun.KMEANS(‘anfun.random2DimDotsBlobs(100,5,0.6)’,5);

SELECT anfun.Dunno(‘KMC’)

>> OUT

---------------------------------------  
| Dunno |

| numeric |  
---------------------------------------  
| 12.57133631512548436983416629628562|

---------------------------------------

Total running time of the script: ( 0 minutes 4.124 seconds)

DB

FUNCTION anfun. DavidBoulduin(\_data, p=2.0) RETURNS numeric

Возвращает число – значение метрики

Принимая в качестве параметров таблицу, возвращает число - зачение метрики Дэвиса - Болдуина.

Input

\_data: varchar

Название таблицы с данными.

p: numeric

Параметр расстояния Минковского. По умолчанию устанавливается значение, равное 2.0.

Output

numeric

Значение метрики DB для данной кластеризации.

Example

>> IN

SELECT \* INTO KMC FROM anfun.KMEANS(‘anfun.random2DimDotsBlobs(100,5,0.6)’,5);

SELECT anfun.DavidBoulduin(‘KMC’)

>> OUT

---------------------------------------  
| DB |

| numeric |  
---------------------------------------  
| 0.375658938226 64565198082176842245|

---------------------------------------

Total running time of the script: ( 0 minutes 4.124 seconds)