

Обзор пакета `mathgraph` системы R.

Краткое описание данного пакета.

Как ясно из названия пакета, он даёт возможности для создания, изменения и визуализации графов.

Создание графа

`Mathgraph` – это функция которая умеет создавать граф, граф бывает двух типов направленный и не направленный, поэтому одно из полей этой функции `directed`, которое соответственно принимает значение `TRUE(T)` и `FALSE(F)`, по дефолту `F`. Граф состоит из рёбер и вершин, поэтому основной параметр `MATHGRAPH` это формула которая показывает как соединять вершины

Символы в формуле:

- `~` - с этого символа должно начинаться
- `+` - для добавления ещё одной части
- `*` - произведение двух векторов, так что рёбрами становятся все возможные пары из элементов левого множителя с элементами из правого множителя.
- `/` - строятся соответствующие рёбра (независимо от равенство от длин векторов(свойство системы R, а не данной функции))(делимое и делитель соответственно должны быть векторами).

`Mathgraph` – соответственно возвращает объект класса `mathgraph`

Что можно делать с графом ещё

`Length(..)` – возвращает количество рёбер в графе.

`s(..)` – метод переопределён для класса `mathgraph`.

`Is.mathgraph(..)` – проверка принадлежность объекта к заданному классу.

`Getpath(x, begin, end)` – возвращает путь идущий от `begin` к `end`, выбирается кратчайший путь в том смысле, что в нём наименьшее число рёбер, если таких путей несколько, то выбирается с наименьшими индексами.

Объект класса `mathgraph[число]` – возвращает ребро с индексом `число`(по сути это тоже объект класса `mathgraph`).

Объект класса `mathgraph` или `print(..)` – просто выводит все рёбра.

`Plot(..)` – рисует `mathgraph`.

`Adjamat(..)` – возвращает матрицу связанности. То есть если существует ребро между `i` и `j` вершинами, то на позиции `[i, j]` стоит `1`, в противном случае ноль. Если граф не направленный то матрица симметричная.

`Alldirected(..)` – возвращает граф, в котором все рёбра стали не направленными.

Incidmat(.., expand = (T|F)) – возвращает матрицу m на n , m – число вершин, n – число рёбер. В позиции i, j стоит 0 это значит что j ребро не связано с i -ым, 1 если из j -ое ребро исходит i -ой вершины. И -1 если входит в i -ую вершину. Expand – расширенный режим или нет, по умолчанию расширенный, это означает, что ненарпавленные рёбру дублируются.

С помощью команды names(..) можно переопределить названия ребёр, и обращаться к рёбрами соответственно по названию.

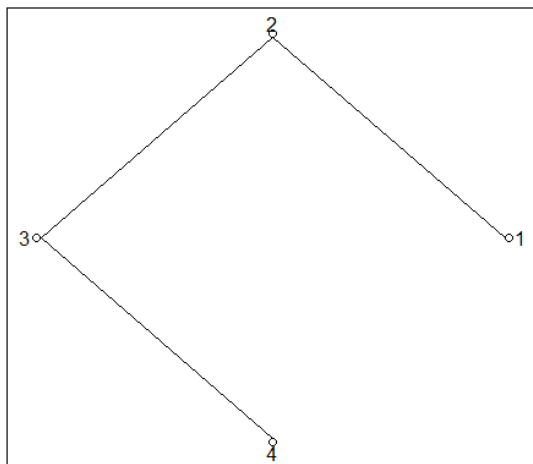
Sortmathgraph(..) – сортирует рёбра объекта mathgraph, так чтобы рёбра были упорядочены: сначала все рёбра исходящие из 1-го, потом их второй вершины и так далее.

Unique(..) – возвращает объект класса mathgraph, в котором удалены одинаковые рёбра.

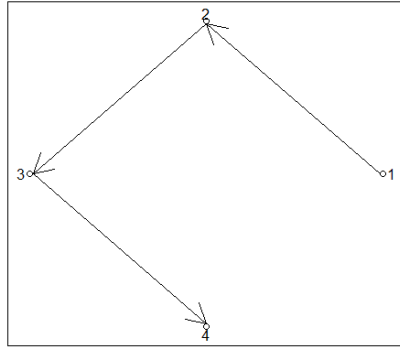
Примеры

```
> jjm <- mathgraph(~ 1:3 / 2:4)
```

```
> plot(jjm)
```



```
> jjm <- mathgraph(~ 1:3 / 2:4, directed = T)
```



```
> plot(jjm)
```

```
> jjm <- mathgraph(~ 1:3 * 2:4, directed = T)
```

```
> plot(jjm)
```

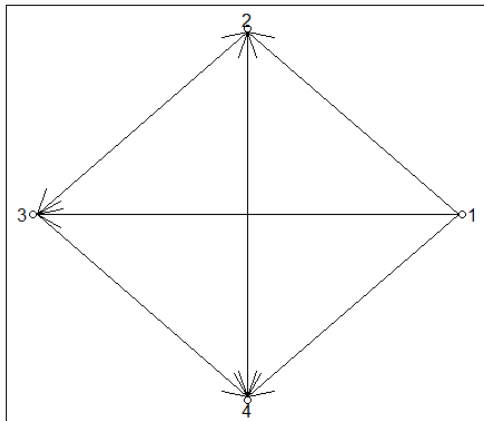
Предупреждения

1: In arrows(px[x[mdir, 1]], py[x[mdir, 1]], px[x[mdir, 2]], py[x[mdir, 2]] :

стрелка нулевой длины имеет неопределенный угол и поэтому пропущена

2: In arrows(px[x[mdir, 1]], py[x[mdir, 1]], px[x[mdir, 2]], py[x[mdir, 2]] :

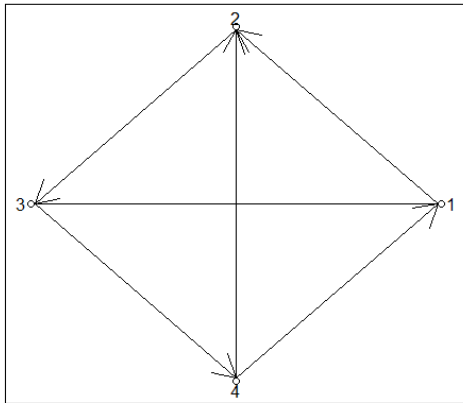
стрелка нулевой длины имеет неопределенный угол и поэтому пропущена



```
> jjm <- mathgraph(~ 1:3 / 2:4, directed = T)
```

```
> jjm <- c(jjm, mathgraph(~ 4 / c(2, 1), directed = T), mathgraph(~ 3 * 1))
```

```
> plot(jjm)
```



>jjm

[1] node 1 -> node 2

[2] node 2 -> node 3

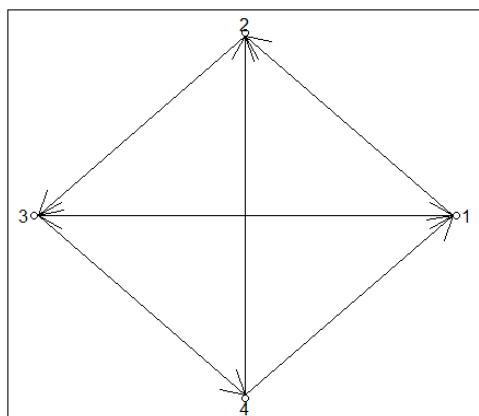
[3] node 3 -> node 4

[4] node 4 -> node 2

[5] node 4 -> node 1

[6] node 3 -- node 1

> plot(alldirected(jjm))



```
> adjamat(jjm)
```

```
node 1 node 2 node 3 node 4
```

```
node 1  1  1  0  0
```

```
node 2  0  0  1  0
```

```
node 3  1  0  0  1
```

```
node 4  1  1  0  0
```

```
> incidmat(jjm)
```

```
arc 1 arc 2 arc 3 arc 4 arc 5 edge 6 edge 6
```

```
node 1  1  0  0  0 -1 -1  1
```

```
node 2 -1  1  0 -1  0  0  0
```

```
node 3  0 -1  1  0  0  1 -1
```

```
node 4  0  0 -1  1  1  0  0
```

здесь видно что 6 ребро не направленное в силу его продублированности

```
> incidmat(jjm, expand = F)
```

```
arc 1 arc 2 arc 3 arc 4 arc 5 edge 6
```

```
node 1  1  0  0  0 -1  1
```

```
node 2 -1  1  0 -1  0  0
```

```
node 3  0 -1  1  0  0  1
```

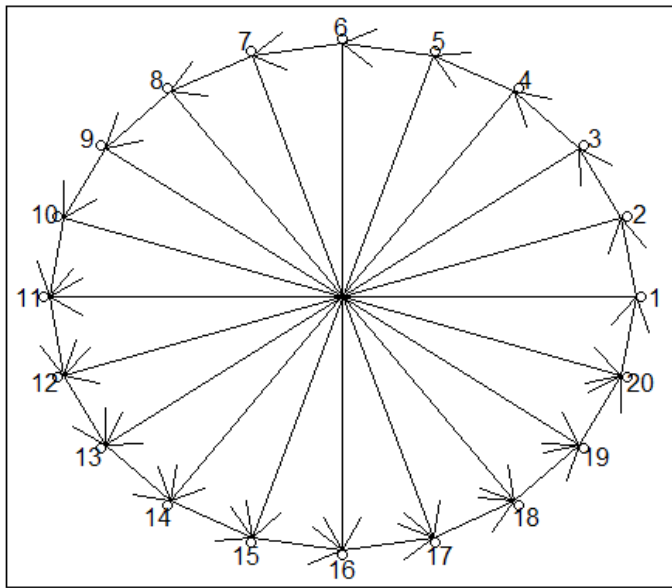
```
node 4  0  0 -1  1  1  0
```

```
> jjm <- mathgraph(~1:10 / 11: 20, directed = T)
```

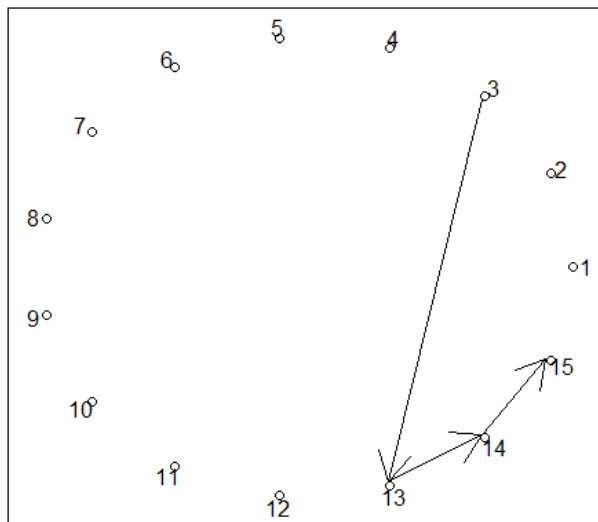
```
> jjm <- c(jjm, mathgraph(~1:19 / 2: 20, directed = T))
```

```
> jjm <- c(jjm, mathgraph(~20 / 1, directed = T))
```

```
> plot(jjm)
```



> plot(getpath(jjm, 3, 15))



как видно plot не рисует вершины о которых ничего не знает.

Спасибо за внимание

Надеюсь теперь вы знаете как работать с пакетом mathgraph.

Зачем это надо

Этот пакет активно используется в других пакетах, где нужно графическое представление графов.