

## 直系同源预测方法

现在的位置: [首页](#) > [Script](#) > 正文

[RSS](#)

[小中大](#)

使用Perl绘制统计图

2011年10月22日 / [Script](#) / [暂无评论](#) / 被围观 158 views+

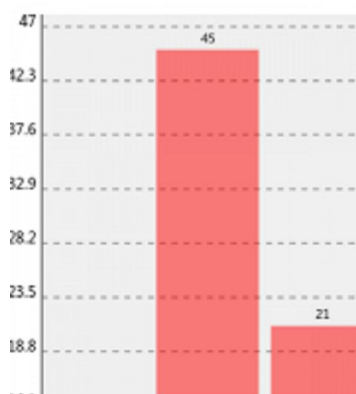
■

### Bar

代码如下:

```
01 #!/use/bin/perl
02
03 use SVG::TT::Graph::Bar;
04
05 my @fields      = qw(Jan Feb Mar);
06 my @data_sales_02 = qw(12 45 21);
07
08 my $graph = SVG::TT::Graph::Bar->new(
09     {
10         'height' => '500',
11         'width'  => '300',
12         'fields' => \@fields,
13     }
14 );
15
16 $graph->add_data(
17     {
18         'data'  => \@data_sales_02,
19         'title' => 'Sales 2002',
20     }
21 );
22
23 open( my $fh, '>', "bar.svg" );
24 select $fh;
25 binmode $fh;
26 print $graph->burn();
27 close($fh);
```

结果图片:





■

### BarHorizontal

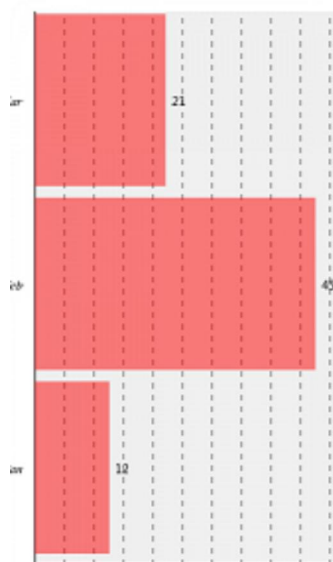
代码如下:

```

01 #!/use/bin/perl
02 use SVG::TT::Graph::BarHorizontal;
03
04 my @fields      = qw(Jan Feb Mar);
05 my @data_sales_02 = qw(12 45 21);
06
07 my $graph = SVG::TT::Graph::BarHorizontal->new(
08     {
09         'height' => '500',
10         'width'  => '300',
11         'fields' => \@fields,
12     }
13 );
14 $graph->add_data(
15     {
16         'data'  => \@data_sales_02,
17         'title' => 'Sales 2002',
18     }
19 );
20
21 open( my $fh, '>', "barhorizontal.svg" );
22 select $fh;
23 binmode $fh;
24 print $graph->burn();
25 close($fh);

```

运行结果:



## BarLine

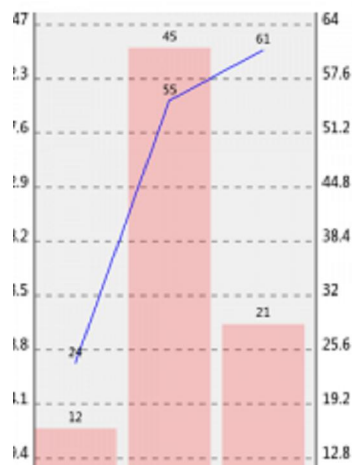
代码:

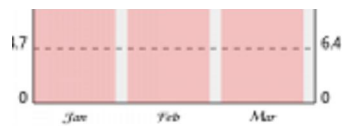
```

01 #!/use/bin/perl
02
03 use SVG::TT::Graph::BarLine;
04
05 my @fields      = qw(Jan Feb Mar);
06 my @data_sales_02 = qw(12 45 21);
07 my @data_sales_03 = ( 24, 55, 61 );
08
09 my $graph = SVG::TT::Graph::BarLine->new(
10     {
11         'height' => '500',
12         'width'  => '300',
13         'fields' => \@fields,
14     }
15 );
16 $graph->add_data(
17     {
18         'data' => \@data_sales_02,
19         'title' => 'Sales 2002',
20     }
21 );
22 $graph->add_data(
23     {
24         'data' => \@data_sales_03,
25         'title' => 'Sales 2003',
26     }
27 );
28
29 open( my $fh, '>', "barline.svg" );
30 select $fh;
31 binmode $fh;
32 print $graph->burn();
33 close($fh);

```

运行结果:



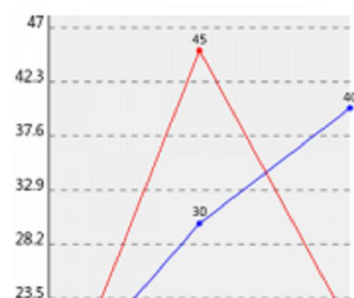


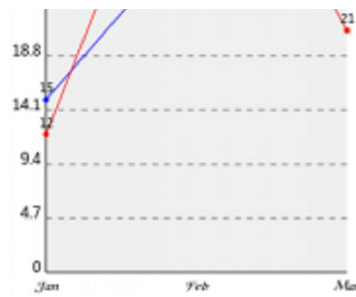
## LINE

代码:

```
01 #!/use/bin/perl
02
03 use SVG::TT::Graph::Line;
04
05 my @fields      = qw(Jan Feb Mar);
06 my @data_sales_02 = qw(12 45 21);
07 my @data_sales_03 = qw(15 30 40);
08
09 my $graph = SVG::TT::Graph::Line->new(
10     {
11         'height' => '500',
12         'width'  => '300',
13         'fields' => \@fields,
14     }
15 );
16
17 $graph->add_data(
18     {
19         'data'  => \@data_sales_02,
20         'title' => 'Sales 2002',
21     }
22 );
23
24 $graph->add_data(
25     {
26         'data'  => \@data_sales_03,
27         'title' => 'Sales 2003',
28     }
29 );
30
31 open( my $fh, '>', "line.svg" );
32 select $fh;
33 binmode $fh;
34 print $graph->burn();
35 close($fh);
```

运行结果:





■

## Pie

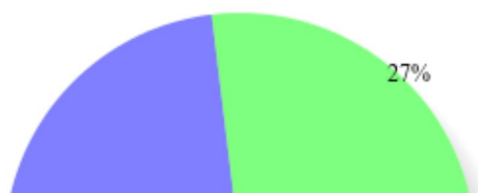
代码:

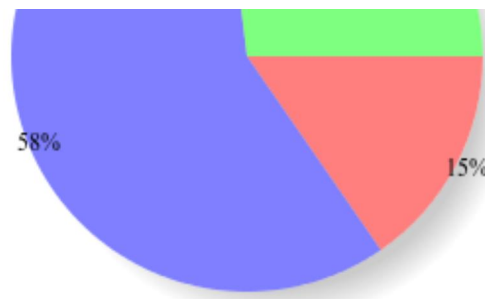
```

01 #!/use/bin/perl
02
03 use SVG::TT::Graph::Pie;
04
05 my @fields      = qw(Jan Feb Mar);
06 my @data_sales_02 = qw(12 45 21);
07
08 my $graph = SVG::TT::Graph::Pie->new(
09     {
10         'height' => '500',
11         'width'  => '300',
12         'fields' => \@fields,
13     }
14 );
15 $graph->add_data(
16     {
17         'data'  => \@data_sales_02,
18         'title' => 'Sales 2002',
19     }
20 );
21
22 open( my $fh, '>', "pie.svg" );
23 select $fh;
24 binmode $fh;
25 print $graph->burn();
26 close($fh);

```

运行结果:





■

### TimeSeries

代码:

```

01 #!/use/bin/perl
02
03 use SVG::TT::Graph::TimeSeries;
04
05 my @data_cpu = (
06     '2003-09-03 09:30:00', 23, '2003-09-03 09:45:00', 54,
07     '2003-09-03 10:00:00', 67, '2003-09-03 10:15:00', 12
08 );
09 my @data_disk = (
10     '2003-09-03 09:00:00', 12, '2003-09-03 10:00:00', 26,
11     '2003-09-03 11:00:00', 23
12 );
13
14 my $graph = SVG::TT::Graph::TimeSeries->new(
15     {
16         'height' => '500',
17         'width'  => '300',
18     }
19 );
20
21 $graph->add_data(
22     {
23         'data'  => \@data_cpu,
24         'title' => 'CPU',
25     }
26 );
27
28 $graph->add_data(
29     {
30         'data'  => \@data_disk,
31         'title' => 'Disk',
32     }
33 );
34
35 open( my $fh, '>', "timeseries.svg" );
36 select $fh;

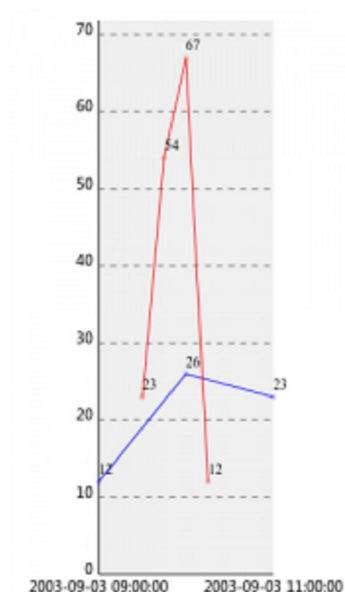
```

```

37 binmode $fh;
38 print $graph->burn();
39 close($fh);

```

运行结果:



■

## Venn

```

01 #!/usr/bin/perl
02 use warnings;
03 use Carp;
04 use strict;
05
06 use Venn::Chart;
07
08 # Create the Venn::Chart constructor
09 my $VennChart = new Venn::Chart( 400, 400 ) or die("error : $!");
10
11 # Set a title and a legend for our chart
12 $VennChart->set( -title => 'Venn diagram' );
13 $VennChart->set_legends( 'Team 1', 'Team 2', 'Team 3' );
14
15 # 3 lists for the Venn diagram
16 my @Team1 = qw/abel edward momo albert jack julien chris/;
17 my @Team2 = qw/edward isabel antonio delta albert kevin jake/;
18 my @Team3 = qw/gerald jake kevin lucia john edward/;
19
20 # Create a diagram with gd object
21 my $gd_venn = $VennChart->plot( \@Team1, \@Team2, \@Team3 );
22
23 # Create a Venn diagram image in png, gif and jpeg format
24 open( my $fh_venn, '>', "VennChart.png" );
25 binmode $fh_venn;
26 print { $fh_venn } $gd_venn->png;
27 close($fh_venn);
28
29 # Create an histogram image of Venn diagram (png, gif and jpeg format).

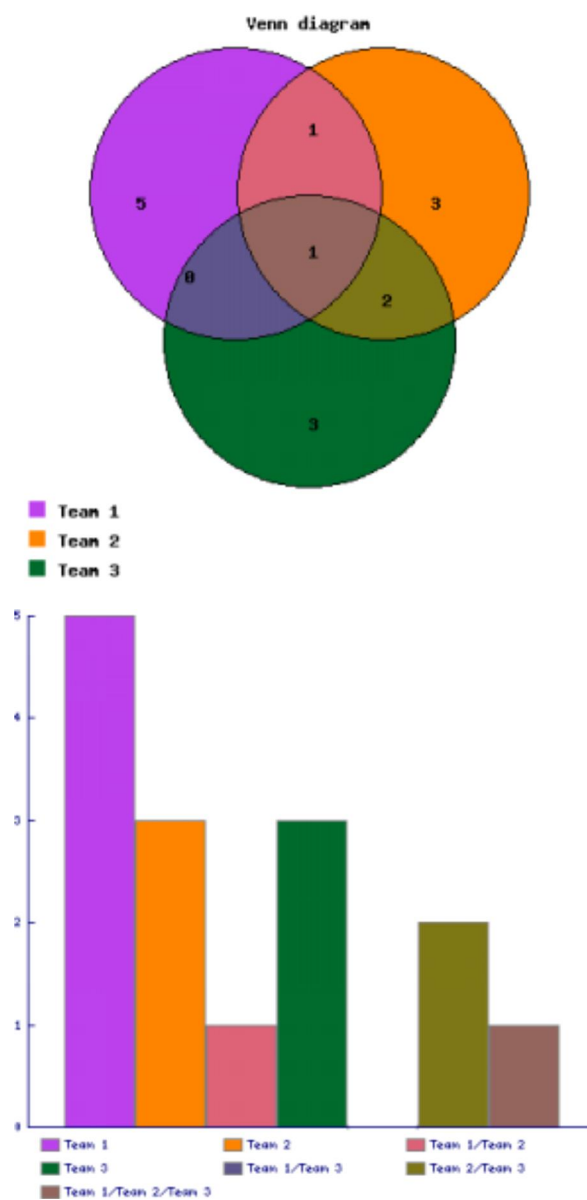
```

```

30 my $gd_histogram = $VennChart->plot_histogram;
31 open( my $fh_histo, '>', "VennHistogram.png" );
32 binmode $fh_histo;
33 print {$fh_histo} $gd_histogram->png;
34 close($fh_histo);

```

运行结果：



转载自：Yixf's blog: <http://http://yixf.name>

分享到：



做学术拒绝沉默

我也来参与分享互动 🔍

[返回](#)