

Additional Material : Run gnuplot from Perl Script

Example perl script to generate outlier datapoint in boxplot and to generate labeled outlier boxplot

```
#!/usr/bin/perl
use strict ;
use warnings;

my @filelist;
my $statfile;
my $label ;
my $supperlimit ;
my @dataarray ;
my @key ;
my @value ;
my @quarts ;
my $PROGRAM ;
my $GP ;
my $add;
my $linearray;

@filelist=('stat1.txt', 'stat2.txt', 'stat3.txt' ) ;
foreach my $n (@filelist)
{
    $statfile = $n.".stat" ;
    open my $PROGRAM, '|-', 'gnuplot'
        or die "Couldn't pipe to gnuplot: $!";

    say {$PROGRAM} "set print '$statfile' ";
    say {$PROGRAM} "stats '$n' using 2 ";
    close $PROGRAM;
}

foreach my $n (@filelist)
{
    $statfile = $n.".stat" ;
    @dataarray = ();
    @key = ();
    @value = ();

    open (Statisticsfile , $statfile) ;
    while (my $limit = <Statisticsfile>) {

        chomp $limit ;
        @dataarray = split(/\t/, $limit);

        push(@key, $dataarray[0]);
        push(@value, $dataarray[1]);
    }

    $supperlimit = $value[21] + 1.5 * ($value[21]-$value[19]) ;
```

```

push(@quarts,$upperlimit);

open (Fileoutlier , $n) ;
while (my $add = <Fileoutlier> ) {
chomp $add ;

if ($add =~ /^$/ or $add =~ /^#/ ) {
    next; }

    my @linearray = split (/t/,$add);
$label= $n.".label";
open (OUT4,'>>', $label) ;
if ($linearray[1] > $upperlimit) {
print OUT4 "$linearray[0] \t $linearray[1] \t $linearray[2] \n " ;
}

}

}

{
open(GP, "| gnuplot") or die "Error while piping to Gnuplot: $! \n";
print GP << "Gnuplot-end";

set term svg
set output 'boxplot-labeled.svg'
set font "times,5"
set xtics ("stat1" 1, "stat2" 2, "stat3" 3)
set style data boxplot
set xrange[0:4]

plot 'stat1.txt' using (1):2 notitle, 'stat1.txt.label' using (1):2:3 with labels notitle offset 2,0, 'stat2.txt'
using (2):2 notitle, "stat2.txt.label" using (2):2:3 with labels notitle offset 2,0, 'stat3.txt' using (3):2
notitle, 'stat3.txt.label' using (3):2:3 with labels notitle offset 3,0

Gnuplot-end
close (GP) ;
}

```