Implementation of Raincloud Plot with SAS and its USE for Clinical Trial Data

Yutaka Morioka
EPS Corporation
Abstract
A boxplot is a standardized way of displaying the dataset based on the five-number summary.

**Minimum** (the lowest data point in the data set excluding any outliers)
**Maximum** (the highest data point in the data set excluding any outliers)
**First quartile** (Q1 or 25th percentile)
**Median** (Q2 or 50th percentile)
**Third quartile** (Q3 or 75th percentile)
We cannot always grasp the data distribution from Boxplot.
Violin plot

Kernel density estimation
Raincloud plot

Cloud [half violin plot]

Umbrella [box plot]

Rain [strip plot]
Bee swarm

PROC SGPLOT DATA=sample;
  SCATTER X=x1 Y=y1 / JITTER JITTERWIDTH=0.5;
RUN;

Jittering strip plot
Raincloud plots: a multi-platform tool for robust data visualization

(Micah Allen et al.)

https://wellcomeopenresearch.org/articles/4-63

Raincloud plot was proposed by Micah Allen et al. at Aarhus University. Many examples of implementation in R, Python, and Matlab were presented. In the spirit of Open Science, they have introduced their technology widely on the Web and SNS, and We have made the most of the feedback.
Test data

data test;
call streaminit(1080);
GROUP="A`````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````
Kernel density estimation (Violin plot)

```sas
proc kde data = test;
  univar val / out = kde;
  by GROUP;
run;

data wk2;
  set kde test;
run;
proc sort data = wk2;
  by group;
run;

data val; table work.kde
  GROUP    var       value    density    count
  A        val     -8.263235935  8.04848898E-6  0
  A        val     -8.110429818  0.0000101993  0
  A        val     -7.958538641  0.0000126557  0
  A        val     -7.806637464  0.000161299  0
  A        val     -7.654741286  0.000201201  0
  A        val     -7.502845109  0.000245785  0
  A        val     -7.350948932  0.000308575  0
  A        val     -7.198052755  0.000379311  0
  A        val     -7.047155678  0.000463941  0
  A        val     -6.8952604  0.000546476  0
  A        val     -6.743364223  0.00098394  0
  A        val     -6.591468046  0.001034418  0
  A        val     -6.439571869  0.001099013  0
  A        val     -6.287676631  0.001160816  0
  A        val     -6.135778514  0.001403207  0
  A        val     -5.98388537  0.001595718  0
  A        val     -5.83198716  0.001594079  0
  A        val     -5.680090983  0.00229017  0
  A        val     -5.529194805  0.002571973  0
  A        val     -5.37629628  0.003103506  0
  A        val     -5.224402451  0.003589113  0
  A        val     -5.072506274  0.004132724  0
  A        val     -4.920610096  0.00478502  0
  A        val     -4.76871319  0.005410375  0
  A        val     -4.61817742  0.006152243  0
  A        val     -4.464921565  0.006867776  0
  A        val     -4.313025388  0.007360382  0
  A        val     -4.16112921  0.00893236  0
  A        val     -4.029230303  0.009888336  0
  A        val     -3.857338958  0.011031163  0
  A        val     -3.705440679  0.012281084  0
  A        val     -3.553544502  0.013580864  0
  A        val     -3.401643824  0.014892357  0
```
ods graphics on / height = 3in width = 12in;
proc template;
  define statgraph RCP;
  begingraph;
    layout overlay
    / xaxisopts = (label = '' type= linear
       linearopts = (viewmin = -20 viewmax = 60
       tickvalu sequence = (start = -20 end = 60 increment = 10))
    yaxisopts = (label = '');

    bandplot x = value limitlower = 0 limitupper = density/ display = all;

    boxplot y = val x = eval(-0.02 + coalesce(0, val))
       / orient = horizontal
       boxwidth = 0.3;

    scatterplot x = val
       y = eval(-0.05 + 0.01*cdf('NORMAL', rannor(1234)) + coalesce(0, val))
       / markerattrs = (symbol = circle size = 8 transparency = 0.4);

  endlayout;
  endgraph;
end;
run;
ods html path = "&outpath" file = 'test.html';

proc sgrender data = wk2 template = RCP;
  by group;
run;
ods html close;
Kernel density estimation also estimates points beyond the range of real data

Proc KDE + truncate option  ⇒  Cut beyond the range of the actual data.

```plaintext
proc kde data = test;
   univar val / out = kde truncate;
   by GROUP;
run;
```
Kernel density estimation – simulation 1

If N is less than 50 or so, the accuracy of determining single/multiple peaks does not seem to be good.
**Kernel density estimation**

– simulation 2

Differences in methods used to compute the bandwidth.

```知情
proc kde data = test;
   univar val / out = kde method = os;
run;
```

Oversmoothed Estimation (More smoothing)
Sheather-Jones Plug In (default)
Multi Panel Raincloud plot
Tips

ods _all_ close;
ods output SGPlot=box;
proc sgplot data=test;
   vbox val/group=group;
run;
ods html;

Get only the parameters of boxplot as a data set. No plot is created.
data wk2;
set kde(in=ina) /*violin*/
  box(in=inb where=(^missing(BOX_VAL_GROUP_GROUP____Y))) /*box*/
  test(in=inc) /*sprit*/;
call streaminit(20220901);
if ina then low=0;
if missing(BOX_VAL_GROUP_GROUP____ST) then BOX_VAL_GROUP_GROUP____ST="DUMMY";
if ^missing(BOX_VAL_GROUP_GROUP____GP) then group=BOX_VAL_GROUP_GROUP____GP;
if inb then dummy_x=-0.01;
if inc then do;
  dummy_y=-0.05;
  random=rand("uniform")*0.01;
  if ranuni(777)<0.5 then dummy_y=dummy_y - random; /
  else dummy_y=dummy_y + random;
end;
run;
Graph Template --- Multi Panel Raincloud plot

ods graphics on /imagemap=on tipmax=5000;
ods html path="xxxxxxxxxxxxxxxx" file="test.html";
proc template;
  define statgraph RCP;
  begingraph;
    entrytitle "Rain Cloud Plot";
    layout datalattice rowvar=group/columnaxisopts=(label="Value") rowaxisopts=(display=none);
    layout prototype;
      bandplot x=value limitupper=density limitlower=low;
      boxplotparm y=BOX_VAL_GROUP_GROUP____Y x=dummy_x
        stat=BOX_VAL_GROUP_GROUP___ST /boxwidth=0.3 orient = horizontal;
      scatterplot x=val y=dummy_y/ jitter=auto jitteropts=(axis=Y width=1)
        markerattrs=(symbol=circle size=8 transparency=0.4)
        roldename=(tip1=ID tip2=VAL) tip=(tip1 tip2) tiplabel=(tip1="ID" tip2="Value")
        ;
    endlayout;
  endgraph;
end;
run;
proc sgrender data=wk2 template=RCP;
run;
ods html close;
Imagemap – Graph Cursor Action

ods graphics on /imagemap=on tipmax=5000;

scatterplot x=val y=dummy_y/ jitter=auto jitteropts=(axis=Y width=1) markerattrs=(symbol=circle size=8 transparency=0.4) rolename=(tip1=ID tip2=VAL) tip=(tip1 tip2) tiplabel=(tip1="ID" tip2="Value");

When the mouse cursor is hovered over the scatterplot, the data can be popped out according to the settings in rolename and tip. Useful for tracing each data.
Raincloud plot for Repeated Measure
Paired Raincloud plot

```plaintext
bandplot y=value  limitupper=-0.2  limitlower=density1 / group=TREAT fillattrs=(transparency=0.4) tip=none;
bandplot y=value  limitupper=density2  limitlower=0.2 / group=TREAT fillattrs=(transparency=0.4) tip=none;
boxplot y = val x =box_x /boxwidth = 1 group=TREAT groupdisplay=cluster clusterwidth=0.1 name="box";
seriesplot x=series_x y=val /display=all group=TREAT_ID markerattrs=(symbol=circlefilled size=8 transparency=0.4) lineattrs=(thickness=1 pattern=solid ) linecolorgroup=TREAT markercolorgroup=TREAT;
```
Programs already released by SAS users


Binning

ods output mapping=bin;
proc hpbin
data=test numbin=10 bucket ;
    input val;
run;

QUANTILE Binning

(Max – Min) / (number of bin)
Binning

ods output mapping=bin;
proc hpbin data=test numbin=10 quantile ;
   input val;
run;

Quantile Binning

Use Quantile
Faded raincloud plots

```
bandplot x = value1 limitlower = 0 limitupper = density
   / display = all fillattrs=(transparency=0.3);
bandplot x = value2 limitlower = 0 limitupper = density
   / display = all fillattrs=(transparency=0);
```
Bivariable Raincloud plot

proc kde data=xxx ;
  bivar Var1 Var2/ out=kde;
run;

x =var1
y =var2
z= kde

Rain z-axis is dummy adjustment with random

Located on the bottom is Quantile binning(10)

All code:
http://sas-tumesas.blogspot.com/2022/10/23d-raincloud-plot.html
REFERENCES


   https://wellcomeopenresearch.org/articles/4-63 (Accessed Dec 05, 2022)

   (Accessed Dec 05, 2022)

Thank you for your attention