Subject: most probable value of a data set
Posted by gunvicsin11 on Sat, 19 Oct 2013 12:35:11 GMT
View Forum Message <> Reply to Message

Hello everyone,
Instead of finding the mean value of a data set, I want to find the most probable value of a data set. How can I find that.
Please do help me in this regard
thanking you in advance
sid

Subject: Re: most probable value of a data set
Posted by Phillip Bitzer on Sat, 19 Oct 2013 22:21:38 GMT
View Forum Message <> Reply to Message

I assume by "most probable value", you mean the mode. Ultimately you'll want to use HISTOGRAM, but how exactly you would calculate the histogram, and hence the mode, really depends on the data set (e.g., what is the "correct" bin size for your data).

Some hints here:
http://www.idlcoyote.com/code_tips/mode.html

Subject: Re: most probable value of a data set
Posted by Gordon Farquharson on Mon, 21 Oct 2013 23:22:01 GMT
View Forum Message <> Reply to Message

On Saturday, October 19, 2013 5:35:11 AM UTC-7, sid wrote:

> Instead of finding the mean value of a data set, I want to find the most probable value of a data set. How can I find that.

Here is an IDL implementation for the half sample mode function found in the modeest package for R [1]. It is a fairly crude direct copy, but it works well for me. (Not sure what Google is going to do to the formatting of the code - you may have to fix some lines.)

BTW, if you interested in mode estimation techniques, look at the paper by Bickel and Frühwirth.

Gordon

;; Author: Wolfgang Huber and Ligia Pedroso Bras (coauthors of package 'genefilter')
;; Modifications: P. Poncet

FUNCTION _deal_ties, ny, i, tie_action, tie_limit

```
    ;; ny,         : length of the data
    ;; i,          : index
    ;; tie_action, : action to be taken
    ;; tie_limit)  : limit

    compile_opt IDL2, LOGICAL_PREDICATE, STRICTARRSUBS, HIDDEN

    ;; Deal with ties
    maxi = max(i)
    mini = min(i)

    IF (maxi - mini GT tie_limit * ny) THEN BEGIN
       message, "encountered a tie, and the difference between minimal and maximal value is >
length('x') * 'tie.limit', so the distribution could be multimodal", /INFO
    ENDIF

    ;; Take the action specified in "tie.action"
    CASE tie_action OF
       'mean'   : return, mean(i)
       'median' : return, median(i)
       'max'    : return, maxi
       'min'    : return, mini
       ELSE     : message, "invalid value" + tie_action + " for argument 'tie.action'"
    ENDCASE

END

;;#######################################################
;;# Robertson and Cryer's / FSM / HSM mode estimator
;;# FSM = fraction-of-sample mode
;;# HSM = half-sample mode
;;#######################################################

;; Author: D.R. Bickel
;; Modications: P. Poncet

FUNCTION hsm, x, bw, k, TIEACTION=tie_action, TIELIMIT=tie_limit

    ;; x  : sample (the data)
    ;; bw : bandwidth (fraction of the observations to consider)
    ;; k  : length of the intervals

    compile_opt IDL2, LOGICAL_PREDICATE, STRICTARRSUBS

    IF ~keyword_set(tie_action) THEN tie_action = "mean"
    IF ~keyword_set(tie_limit) THEN tie_limit = 0.05
```

```
if (n_elements(k) NE 0 AND n_elements(bw) EQ 0) THEN BEGIN
   bw = (k+1) / n_elements(x)
ENDIF ELSE IF (n_elements(k) EQ 0 AND n_elements(bw) EQ 0) THEN BEGIN
   bw = 0.5
ENDIF

IF (bw LE 0 OR bw GT 1) THEN BEGIN
   message, "argument 'bw' must belong to (0, 1]", /INFO
   return, !values.f_nan
ENDIF

y = x[sort(x)]

WHILE n_elements(y) GE 4 DO BEGIN

   ny = n_elements(y)
   k = ceil(bw*ny) - 1

   inf = y[0:(ny-k)-1]
   sup = y[k:ny-1]
   diffs = sup - inf
   i = where(diffs EQ min(diffs))

   ;; Ties?
   IF (n_elements(i) gt 1) THEN BEGIN
      i = _deal_ties(ny, i, tie_action, tie_limit)
   ENDIF

   if (diffs[i] EQ 0) THEN BEGIN
      y = y[i]
   ENDIF ELSE BEGIN
      y = y[i:(i+k)]
   ENDELSE

ENDWHILE

IF n_elements(y) EQ 3 THEN BEGIN
   z = 2*y[1] - y[0] - y[2]
   IF z LT 0 THEN BEGIN
      M = mean(y[0:1])
   ENDIF ELSE IF z GT 0 THEN BEGIN
      M = mean(y[1:2])
   ENDIF ELSE IF z EQ 0 THEN BEGIN
      M = y[1]
   ENDIF
ENDIF ELSE BEGIN
   M = mean(y)
ENDELSE
```

```
    return, M

END
```
---

[1] http://cran.r-project.org/web/packages/modeest/index.html