Subject: Re: Box-Whisker plots in IDL
Posted by mankoff on Mon, 20 Aug 2007 17:04:11 GMT

On Aug 20, 12:42 pm, te...@atmsci.msrc.sunysb.edu wrote:
> Hi,
>
> Does anyone know of a way for IDL to make box-whisker plots? I know
> IDL get calculate the median, but how about the 25th and 75th
> percentiles which are needed for such a plot?
>
> Thanks,
>
> Howard

Percentile algorithm is described in this post:
 http://groups.google.com/group/comp.lang.idl-pvwave/browse_f
rm/thread/20da3d30998010e9/db2b0217422ad887?lnk=gst&q=hi
stogram+percentile&rnum=1#db2b0217422ad887
a.k.a http://tinyurl.com/32pbsh

  -k.

Subject: Re: Box-Whisker plots in IDL
Posted by David Fanning on Mon, 20 Aug 2007 17:10:47 GMT

teich@atmsci.msrc.sunysb.edu writes:

> Does anyone know of a way for IDL to make box-whisker plots? I know
> IDL get calculate the median, but how about the 25th and 75th
> percentiles which are needed for such a plot?

See the histogram tutorial for a fast way to find the
quartiles of a number distribution, which is what you
will need to calculate the 25th and 75th percentiles.
That is, the 25th percentile will be the median of the
lower two quartiles of values, and the 75 percentile will
be the median of the upper two quartiles.

  http://www.dfanning.com/tips/histogram_tutorial.html

The rest is just PLOTS commands. :-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Box-Whisker plots in IDL
Posted by teich on Mon, 20 Aug 2007 17:27:22 GMT
View Forum Message <> Reply to Message

On Aug 20, 1:10 pm, David Fanning <n...@dfanning.com> wrote:
> te...@atmsci.msrc.sunysb.edu writes:
>> Does anyone know of a way for IDL to make box-whisker plots? I know
>> IDL get calculate the median, but how about the 25th and 75th
>> percentiles which are needed for such a plot?
>
> See the histogram tutorial for a fast way to find the
> quartiles of a number distribution, which is what you
> will need to calculate the 25th and 75th percentiles.
> That is, the 25th percentile will be the median of the
> lower two quartiles of values, and the 75 percentile will
> be the median of the upper two quartiles.
>
> http://www.dfanning.com/tips/histogram_tutorial.html
>
> The rest is just PLOTS commands. :-)
>
> Cheers,
>
> David
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming:http://www.dfanning.com/
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Thanks for the info!!!

Subject: Re: Box-Whisker plots in IDL
Posted by Brian Larsen on Mon, 20 Aug 2007 19:01:50 GMT
View Forum Message <> Reply to Message

When you solve this problem if you wouldn't mind posting the function/
procedure you come up with I would love to have a copy as I sometimes
do those and haven't had the time/patience to implement them in idl

yet.

Cheers,

Brian

--- --------------------
Brian Larsen
Boston University
Center for Space Physics

Subject: Re: Box-Whisker plots in IDL
Posted by teich on Mon, 20 Aug 2007 19:18:35 GMT
View Forum Message <> Reply to Message

On Aug 20, 3:01 pm, Brian Larsen <balar...@gmail.com> wrote:
> When you solve this problem if you wouldn't mind posting the function/
> procedure you come up with I would love to have a copy as I sometimes
> do those and haven't had the time/patience to implement them in idl
> yet.
>
> Cheers,
>
> Brian
>
> --- -------------- ------
> Brian Larsen
> Boston University
> Center for Space Physics

Well, I am looking into the histogram procedure, but I am not getting
what I think the 25th and 75th quartiles should be. It seems
histogram is not so easy to master. What I am looking into is doing
the following:

data=randomu(sd,100)*100
box plot needs min, max, median which are straight forward:

min(data)
max(data)
median(data,/even)

For the quartiles I am trying:

lower_ind=where(data lt median(data,/even))

upper_ind=where(data gt median(data,/even))
 qtr_25th=median(data[lower_ind(0):lower_ind(n_elements(lower _ind)-1)],/
even)
 qtr_75th=median(data[upper_ind(0):upper_ind(n_elements(upper _ind)-1)],/
even)

However, I think this would work only for a monotonically increasing
array. I am not sure how to get 'data' like that. If anyone wants to
add to this, feel free.

How

Subject: Re: Box-Whisker plots in IDL
Posted by David Fanning on Mon, 20 Aug 2007 19:40:23 GMT
View Forum Message <> Reply to Message

teich@atmsci.msrc.sunysb.edu writes:

> Well, I am looking into the histogram procedure, but I am not getting
> what I think the 25th and 75th quartiles should be. It seems
> histogram is not so easy to master. What I am looking into is doing
> the following:
>
>
> data=randomu(sd,100)*100
> box plot needs min, max, median which are straight forward:
>
> min(data)
> max(data)
> median(data,/even)
>
> For the quartiles I am trying:
>
> lower_ind=where(data lt median(data,/even))
> upper_ind=where(data gt median(data,/even))
> qtr_25th=median(data[lower_ind(0):lower_ind(n_elements(lower _ind)-1)],/
> even)
> qtr_75th=median(data[upper_ind(0):upper_ind(n_elements(upper _ind)-1)],/
> even)
>
> However, I think this would work only for a monotonically increasing
> array. I am not sure how to get 'data' like that. If anyone wants to
> add to this, feel free.

I calculate it like this:

  data=randomu(sd,100)*100

```
minVal = min(data)
maxVal = max(data)
medianVal = median(data,/even)

; Find the quartiles.
binsize = (maxVal - minVal) / 4.0
h = Histogram(data, BINSIZE=binsize, REVERSE_INDICES=ri)
qtr_25th = Median(data[ri[ri[0]:ri[2]-1]])
qtr_75th = Median(data[ri[ri[2]:ri[4]-1]])

Print, minVal, maxVal, medianVal, qtr_25th, qtr_75th
END
```

With 100 values I get this:

    0.401314   98.0063   58.9402   20.0477   73.3419

With 10000 values, I get this, which leads me to think the
algorithm might be correct:

    0.0249010   99.9960   49.9658   25.0268   74.8059

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Box-Whisker plots in IDL
Posted by teich on Mon, 20 Aug 2007 20:22:56 GMT
View Forum Message <> Reply to Message

On Aug 20, 3:40 pm, David Fanning <n...@dfanning.com> wrote:
> te...@atmsci.msrc.sunysb.edu writes:
>> Well, I am looking into the histogram procedure, but I am not getting
>> what I think the 25th and 75th quartiles should be. It seems
>> histogram is not so easy to master. What I am looking into is doing
>> the following:
>
>> data=randomu(sd,100)*100
>> box plot needs min, max, median which are straight forward:
>

```
>>  min(data)
>>  max(data)
>>  median(data,/even)
>
>>  For the quartiles I am trying:
>
>>  lower_ind=where(data lt median(data,/even))
>>  upper_ind=where(data gt median(data,/even))
>>   qtr_25th=median(data[lower_ind(0):lower_ind(n_elements(lower _ind)-1)],/
>>  even)
>>   qtr_75th=median(data[upper_ind(0):upper_ind(n_elements(upper _ind)-1)],/
>>  even)
>
>>  However, I think this would work only for a monotonically increasing
>>  array.  I am not sure how to get 'data' like that.  If anyone wants to
>>  add to this, feel free.
>
>  I calculate it like this:
>
>     data=randomu(sd,100)*100
>
>     minVal = min(data)
>     maxVal = max(data)
>     medianVal = median(data,/even)
>
>     ; Find the quartiles.
>     binsize = (maxVal - minVal) / 4.0
>     h = Histogram(data, BINSIZE=binsize, REVERSE_INDICES=ri)
>     qtr_25th = Median(data[ri[ri[0]:ri[2]-1]])
>     qtr_75th = Median(data[ri[ri[2]:ri[4]-1]])
>
>     Print, minVal, maxVal, medianVal, qtr_25th, qtr_75th
>     END
>
>  With 100 values I get this:
>
>       0.401314    98.0063    58.9402    20.0477    73.3419
>
>  With 10000 values, I get this, which leads me to think the
>  algorithm might be correct:
>
>       0.0249010    99.9960    49.9658    25.0268    74.8059
>
>  Cheers,
>
>  David
>
>  --
```

> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming:http://www.dfanning.com/
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Hi, Suppose data is something simple like:

data=[2,3,5,7,7,10,11,11,12,15,16,17,17]

I get a 75th quartile of 11.0. Shouldn't I get around 15? Maybe I
made a mistake. However, what about if I revise what I wrote before
to:

sorted_data=data(sort(data))

lower_ind=where(sorted_data lt median(sorted_data,/even))
upper_ind=where(sorted_data gt median(sorted_data,/even))
 qtr_25th=median(sorted_data[lower_ind(0):lower_ind(n_element s(lower_ind)-1)],/
even)
 qtr_75th=median(sorted_data[upper_ind(0):upper_ind(n_element s(upper_ind)-1)],/
even)

Howie

Subject: Re: Box-Whisker plots in IDL
Posted by David Fanning on Mon, 20 Aug 2007 20:56:08 GMT
View Forum Message <> Reply to Message

teich@atmsci.msrc.sunysb.edu writes:

> Hi, Suppose data is something simple like:
>
> data=[2,3,5,7,7,10,11,11,12,15,16,17,17]
>
> I get a 75th quartile of 11.0. Shouldn't I get around 15?

JD will have to explain the difference between BINSIZE
and NBINS to us again. (And I think he is in China for
a couple of weeks.) But I got strange results with my
HISTOGRAM method, too. Here is a slightly revised program:

```
  data=[2,3,5,7,7,10,11,11,12,15,16,17,17]
  ;box plot needs min, max, median which are straight forward:

  minVal = min(data)
  maxVal = max(data)
  medianVal = median(data,/even)
```

```
; Find the quartiles.
h = Histogram(data, NBINS=4, REVERSE_INDICES=ri, $
    MIN=minVal, MAX=maxVal)
qtr_25th = Median(data[ri[ri[0]:ri[2]-1]])
qtr_75th = Median(data[ri[ri[2]:ri[4]-1]])


Print, minVal, maxVal, medianVal, qtr_25th, qtr_75th
END
```

And the result I get with the new data:

```
  2     17    11.0000    7.00000    16.0000
```


--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Box-Whisker plots in IDL
Posted by teich on Mon, 20 Aug 2007 21:03:52 GMT
View Forum Message <> Reply to Message

On Aug 20, 4:56 pm, David Fanning <n...@dfanning.com> wrote:
> te...@atmsci.msrc.sunysb.edu writes:
>> Hi, Suppose data is something simple like:
>
>> data=[2,3,5,7,7,10,11,11,12,15,16,17,17]
>
>> I get a 75th quartile of 11.0. Shouldn't I get around 15?
>
> JD will have to explain the difference between BINSIZE
> and NBINS to us again. (And I think he is in China for
> a couple of weeks.) But I got strange results with my
> HISTOGRAM method, too. Here is a slightly revised program:
>
>   data=[2,3,5,7,7,10,11,11,12,15,16,17,17]
>   ;box plot needs min, max, median which are straight forward:
>
>   minVal = min(data)
>   maxVal = max(data)
>   medianVal = median(data,/even)
>

>   ; Find the quartiles.
>   h = Histogram(data, NBINS=4, REVERSE_INDICES=ri, $
>   MIN=minVal, MAX=maxVal)
>   qtr_25th = Median(data[ri[ri[0]:ri[2]-1]])
>   qtr_75th = Median(data[ri[ri[2]:ri[4]-1]])
>
>   Print, minVal, maxVal, medianVal, qtr_25th, qtr_75th
>   END
>
> And the result I get with the new data:
>
>   2   17   11.0000   7.00000   16.0000
>
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming:http://www.dfanning.com/
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Thanks,

I think I will go with yours!

Howie

Subject: Re: Box-Whisker plots in IDL
Posted by jschwab@gmail.com on Mon, 20 Aug 2007 23:04:14 GMT
View Forum Message <> Reply to Message

Pardon me if I'm mistaken, but I think these "quartiles with
histogram" examples, including the one that's in JD's histogram
tutorial are fundamentally incorrect.

You are assuming "Equal bin widths" ==> "Equal #'s in each bin" !

When HISTOGRAM splits a data list into N bins, it does so such that
the *width* of the bins are equal. In no way does it somehow create a
situation in which the *number of points* in each bin is equal (which
is what would be required to find quartiles in such a manner).

The given examples have only "worked" because you're either dealing
with uniform distributions (in which case equal bin widths do imply
equal numbers in each bin) or because the example data happens to be
roughly uniform.

If you want to convince yourself, try one of those codes with
data = randomu(seed, 1000) * 100.

and then with
data2 = data * data
The quartiles in the 2nd case should simply be the squares of the
quartiles from the first.

Cheers,
Josiah
--
Josiah Schwab
MIT, Course VIII

Subject: Re: Box-Whisker plots in IDL
Posted by David Fanning on Mon, 20 Aug 2007 23:47:28 GMT
View Forum Message <> Reply to Message

jschwab@gmail.com writes:

> Pardon me if I'm mistaken, but I think these "quartiles with
> histogram" examples, including the one that's in JD's histogram
> tutorial are fundamentally incorrect.
>
> You are assuming "Equal bin widths" ==> "Equal #'s in each bin" !
>
> When HISTOGRAM splits a data list into N bins, it does so such that
> the *width* of the bins are equal. In no way does it somehow create a
> situation in which the *number of points* in each bin is equal (which
> is what would be required to find quartiles in such a manner).
>
> The given examples have only "worked" because you're either dealing
> with uniform distributions (in which case equal bin widths do imply
> equal numbers in each bin) or because the example data happens to be
> roughly uniform.
>
> If you want to convince yourself, try one of those codes with
> data = randomu(seed, 1000) * 100.
> and then with
> data2 = data * data
> The quartiles in the 2nd case should simply be the squares of the
> quartiles from the first.

Humm. Maybe you are right. (Isn't it odd that math types
never hit the SEND button until someone else has made
a fool of themselves?)

OK, how about this:

```
data=randomu(sd,100)*100
minVal = min(data)
maxVal = max(data)
medianVal = median(data,/even)

; Find the quartiles.
qtr_25th = Median(data[Where(data LE medianVal, countlowerhalf)])
qtr_75th = Median(data[Where(data GT medianVal, countupperhalf)])
void = Where(data LT qtr_25th, countlowerquarter)
void = Where(data GE qtr_75th, countupperquarter)

Print, minVal, maxVal, medianVal, qtr_25th, qtr_75th
Print, countlowerquarter, countlowerhalf-countlowerquarter, $
      countupperhalf-countupperquarter, countupperquarter
END
```

Which gives me:

```
    1.74060    99.8840    53.5631    31.7422    73.8378
        25        25       25        25
```

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Box-Whisker plots in IDL
Posted by jschwab@gmail.com on Tue, 21 Aug 2007 00:06:31 GMT
View Forum Message <> Reply to Message

On Aug 20, 7:47 pm, David Fanning <n...@dfanning.com> wrote:
> jsch...@gmail.com writes:
>> Pardon me if I'm mistaken, but I think these "quartiles with
>> histogram" examples, including the one that's in JD's histogram
>> tutorial are fundamentally incorrect.
>
>> You are assuming "Equal bin widths" ==> "Equal #'s in each bin" !
>
>> When HISTOGRAM splits a data list into N bins, it does so such that
>> the *width* of the bins are equal. In no way does it somehow create a
>> situation in which the *number of points* in each bin is equal (which
>> is what would be required to find quartiles in such a manner).
>

>> The given examples have only "worked" because you're either dealing
>> with uniform distributions (in which case equal bin widths do imply
>> equal numbers in each bin) or because the example data happens to be
>> roughly uniform.
>
>> If you want to convince yourself, try one of those codes with
>> data = randomu(seed, 1000) * 100.
>> and then with
>> data2 = data * data
>> The quartiles in the 2nd case should simply be the squares of the
>> quartiles from the first.
>
> Humm. Maybe you are right. (Isn't it odd that math types
> never hit the SEND button until someone else has made
> a fool of themselves?)
>
> OK, how about this:
>
>   data=randomu(sd,100)*100
>   minVal = min(data)
>   maxVal = max(data)
>   medianVal = median(data,/even)
>
>   ; Find the quartiles.
>   qtr_25th = Median(data[Where(data LE medianVal, countlowerhalf)])
>   qtr_75th = Median(data[Where(data GT medianVal, countupperhalf)])
>   void = Where(data LT qtr_25th, countlowerquarter)
>   void = Where(data GE qtr_75th, countupperquarter)
>
>   Print, minVal, maxVal, medianVal, qtr_25th, qtr_75th
>   Print, countlowerquarter, countlowerhalf-countlowerquarter, $
>   countupperhalf-countupperquarter, countupperquarter
>   END
>
> Which gives me:
>
>   1.74060   99.8840   53.5631   31.7422   73.8378
>   25   25   25   25
>
> Cheers,
>
> David
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming:http://www.dfanning.com/
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

This looks good to me. I wrote a routine to find the energy quartiles
of some x-ray data a few weeks back and that was the way I ended up
doing it. Not that speed is an issue, but I'd be curious to see how
this method compares with a SORT, or some other (yet undiscussed)
method. Maybe I'll play around with that tonight if I have some extra
time.

Cheers,
Josiah
--
Josiah Schwab
MIT, Course VIII

Subject: Re: Box-Whisker plots in IDL
Posted by Marshall Perrin on Tue, 21 Aug 2007 00:37:53 GMT
View Forum Message <> Reply to Message

teich@atmsci.msrc.sunysb.edu <teich@atmsci.msrc.sunysb.edu> wrote:
> lower_ind=where(sorted_data lt median(sorted_data,/even))
> upper_ind=where(sorted_data gt median(sorted_data,/even))
> qtr_25th=median(sorted_data[lower_ind(0):lower_ind(n_element s(lower_ind)-1)],/
> even)
> qtr_75th=median(sorted_data[upper_ind(0):upper_ind(n_element s(upper_ind)-1)],/
> even)

Pardon me, but I think your calculation for the 25th and 75th percentiles here
is in deep trouble. lower_ind is a set of discrete and discontinuous indices,
so it's not correct to subscript sorted_data using the range construct
like that! You really want just

 qtr_25th=median(sorted_data[lower_ind])
 qtr_75th=median(sorted_data[upper_ind])

Cheers,

 - Marshall

Subject: Re: Box-Whisker plots in IDL
Posted by David Fanning on Tue, 21 Aug 2007 04:01:17 GMT
View Forum Message <> Reply to Message

jschwab@gmail.com writes:

> This looks good to me. I wrote a routine to find the energy quartiles
> of some x-ray data a few weeks back and that was the way I ended up

> doing it. Not that speed is an issue, but I'd be curious to see how
> this method compares with a SORT, or some other (yet undiscussed)
> method. Maybe I'll play around with that tonight if I have some extra
> time.

I wrote a short article to illustrate how I would go
about creating a box and whisker plot in IDL:

  http://www.dfanning.com/graphics_tips/box&whisker.html

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Box-Whisker plots in IDL
Posted by jschwab@gmail.com on Tue, 21 Aug 2007 04:46:51 GMT
View Forum Message <> Reply to Message

On Aug 21, 12:01 am, David Fanning <n...@dfanning.com> wrote:
> jsch...@gmail.com writes:
>> This looks good to me. I wrote a routine to find the energy quartiles
>> of some x-ray data a few weeks back and that was the way I ended up
>> doing it. Not that speed is an issue, but I'd be curious to see how
>> this method compares with a SORT, or some other (yet undiscussed)
>> method. Maybe I'll play around with that tonight if I have some extra
>> time.
>
> I wrote a short article to illustrate how I would go
> about creating a box and whisker plot in IDL:
>
>   http://www.dfanning.com/graphics_tips/box&whisker.html
>
> Cheers,
>
> David
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming:http://www.dfanning.com/
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Well I see David beat me to it, but I was also playing around writing

some box-and-whisker plotting code. I think David's is nicer than mine
(big surprise), but I'll post what I wrote so anyone else can play
with it if they like.

Cheers,
Josiah

--
```
;+
; NAME: BWPLOT
;
; PURPOSE: Draw a box-and-whisker plot
;
; CATEGORY: Plotting, Graphics
;
; CALLING SEQUENCE: BWPLOT, data
;
; INPUTS: data = data to be plotted
;
; OPTIONAL INPUTS: None
;
; KEYWORD PARAMETERS:
;
;    OUTLIERS - if set, plots outliers (points which are > 1.5x
;          the interquartile range away from Q25 or Q75
;
;    BOXWIDTH - height of the box as a percentage of the screen
;          height; defaults to 10%
;
;    WHISKWIDTH - height of the whiskers as a percentage of the
;           screen height; defaults to half of the box width
;
;    BOXCOLOR - color to make the box portion of the plot; specify in
;          the same manner that one would set COLOR when using
;          PLOT
;
;    WHISKCOLOR - color to make the whisker portion of the plot;
;          specify in the same manner that one would set
;          COLOR when using PLOT
;
;    OUTSYM - plot symbol to use for outliers; only relavent when the
;          OUTLIERS keyword is set; specify in the same manner
;          that one would set PSYM when using PLOT
;
;    OUTCOLOR - color to make the outliers in the plot; only relavent
;          when the OUTLIERS keyword is set; specify in the
;          same manner that one would set COLOR when using PLOT
;
```

```
;     QUARTILES - variable to contain the 5 values used to contruct
;              the plot; a 5 element array
;              [min, q25, median, q75, max]
;
;     IQR - variable to contain the value of the interquartile range
;
; OUTPUTS: None (see keywords QUARTILES and IQR)
;
; OPTIONAL OUTPUTS: None
;
; COMMON BLOCKS: None
;
; SIDE EFFECTS: None
;
; RESTRICTIONS:
;     Does not produce vertical plots
;     Does not produce multiple plots
;     Does not explictly label quartiles
;
; PROCEDURE: Straightforward
;
; EXAMPLE:
;    Make a box-and-whisker plot of some random data
;
;    random_data = randomu(seed, 1000) * 100.
;    bwplot, random_data
;
; MODIFICATION HISTORY:
;
;   Created:
;       Mon Aug 20, Josiah Schwab
;
; LICENSE
;    Copyright (c) 2007 Josiah Schwab
;
;Permission is hereby granted, free of charge, to any person obtaining
a copy
;of this software and associated documentation files (the "Software"),
to deal
;in the Software without restriction, including without limitation the
rights
;to use, copy, modify, merge, publish, distribute, sublicense, and/or
sell
;copies of the Software, and to permit persons to whom the Software is
;furnished to do so, subject to the following conditions:
;
;The above copyright notice and this permission notice shall be
included in
```

```
;all copies or substantial portions of the Software.
;
;THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR
;IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY,
;FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT
SHALL THE
;AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR
OTHER
;LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,
ARISING FROM,
;OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
DEALINGS IN
;THE SOFTWARE.
;
;-

PRO bwplot, data, $
        OUTLIERS = OUTLIERS, $
        BOXWIDTH = BOXWIDTH, WHISKWIDTH = WHISKWIDTH, $
        BOXCOLOR = BOXCOLOR, WHISKCOLOR = WHISKCOLOR, $
        QUARTILES = QUARTILES, IQR = IQR, $
        OUTSYM = OUTSYM, OUTCOLOR = OUTCOLOR

COMPILE_OPT IDL2
ON_ERROR, 2

;test for at least 5 pts
if n_elements(data) lt 5 then message, "Must have at least 5 points"

;; set keywords
if not keyword_set(outliers) then outliers = 0
if not keyword_set(boxwidth) then boxwidth = 0.1
if not keyword_set(whiskwidth) then whiskwidth = boxwidth / 2.
if not keyword_set(boxcolor) then boxcolor = !P.color
if not keyword_set(whiskcolor) then whiskcolor = !P.color
if not keyword_set(outsym) then outsym = 1
if not keyword_set(outcolor) then outcolor = !P.color

;; calculate quartiles
;; they are returned in variable "Quartiles"
minVal = min(data, max = maxVal)
medVal = median(data,/EVEN)
q25Val = median(data[where(data LE medVal)], /even)
q75Val = median(data[where(data GT medVal)], /even)

quartiles = [minVal, q25Val, medVal, q75Val, maxVal]
```

```
;; calculate interquartile range
IQR = q75Val - q25Val

;; set up plot
left = floor(minVal - 0.1 * IQR)
right = ceil(maxVal + 0.1 * IQR)
plot, data, data, /nodata, $
    xrange = [left, right], yrange = [-1, 1], $
    xstyle = 1, ystyle = 1, $
    yticks = 1, ytickname = [' ', ' ']

;; OUTLIERS == 1 --> PLOT OUTLIERS SEPERATELY

if outliers eq 1 then begin

  low = q25Val - 1.5 * IQR
  high = q75Val + 1.5 * IQR

  out = where( (data LT low) OR (data GT high) , out_count, $
           complement = not_out )
  if out_count gt 0 then $
    oplot, data[out], rebin([0], out_count), $
        psym = outsym, color = outcolor
  whisk_min = min(data[not_out], max = whisk_max)

endif else begin

  whisk_min = minVal
  whisk_max = maxVal

endelse

;draw box
plots, [q25Val, q75Val, q75Val, q25Val, q25val], $
    boxwidth * [1, 1, -1, -1, 1], $
    color = boxcolor
plots, [medVal, medVal], boxwidth * [1, -1], color = boxcolor

;draw whiskers
plots, [q75val, whisk_max], [0, 0], color = whiskcolor
plots, [q25val, whisk_min], [0, 0], color = whiskcolor
plots, [whisk_max, whisk_max], whiskwidth * [-1, 1], color =
whiskcolor
plots, [whisk_min, whisk_min], whiskwidth * [-1, 1], color =
whiskcolor
```

end

Subject: Re: Box-Whisker plots in IDL
Posted by JD Smith on Mon, 27 Aug 2007 01:50:45 GMT
View Forum Message <> Reply to Message

On Mon, 20 Aug 2007 23:04:14 +0000, jschwab@gmail.com wrote:

> Pardon me if I'm mistaken, but I think these "quartiles with
> histogram" examples, including the one that's in JD's histogram
> tutorial are fundamentally incorrect.
>
> You are assuming "Equal bin widths" ==> "Equal #'s in each bin" !

I probably shouldn't have called them "quartiles", as they are really
quarter range bins (data quartiles). The data quartile is of course
as useful as the ordered quartile, but not for this problem.

One straightforward option for the ordered quartile is to use SORT,
picking out only the elements at nel/4 and 3*nel/4, e.g.:

```
n=n_elements(data)
s=sort(data)
qval=data[s[3*n/4]]
```

Unfortunately, this is fairly slow for large data sets. Another
faster but approximate option is to form a cumulative total of
HISTOGRAM's output with an appropriate bin size, and find where it
reaches 25% and 75% of the total count of data points.

Depending on your needs and bin width, you may want to dive into the
individual bin using REVERSE_INDICES to find the *exact* quartile
value itself. This isn't as hard as it sounds:

```
bs=0.05 ;something appropriate for bin size
h=histogram(data,REVERSE_INDICES=r,BINSIZE=bs,OMIN=om)
cum=total(h,/CUMULATIVE,/PRESERVE_TYPE)
quart=3*n/4
v=value_locate(cum,quart)
vals=data[r[r[v+1]:r[v+2]-1]]
qval=vals[(sort(vals))[quart-cum[v]]]
```

You'll find this is roughly 10x faster than using SORT by itself. And
if you only need the approximate value (good to the histogram bin
width), simply replace the last two lines with:

```
qval=om+bs*(v+1.5)
```

for a modest additional speed-up. All the usual caveats with
HISTOGRAM apply (e.g. beware when dealt overly sparse data).

This problem reminds me of the one quote I always remember from
Numerical Recipes: "Selection is Sorting's austere sister."

JD