

Paper 155-2007

Visualizing Statistical Inference Using SAS®

Barbara Schneider, Medical University of Vienna, Austria

Kristan Schneider, University of Vienna, Austria

ABSTRACT

It is a demanding intention to teach statistics to medical students or physicians. The reason is that they are more firmly rooted in applications rather than in an abstract theoretical framework. Visualization is an excellent tool to improve comprehensibility when explaining dry concepts such as statistical inference. If one desires to draw conclusion from data one has to be familiar with various statistical terms, e.g., p-values, significance level, test statistics, power etc., where at least an intuitive understanding is required. Our presentation suggests an approach for teaching statistical inference using simulated data together with animated graphics. In particular, the GIFANIM device driver that facilitates animated graphics is discussed and the respective SAS code is presented.

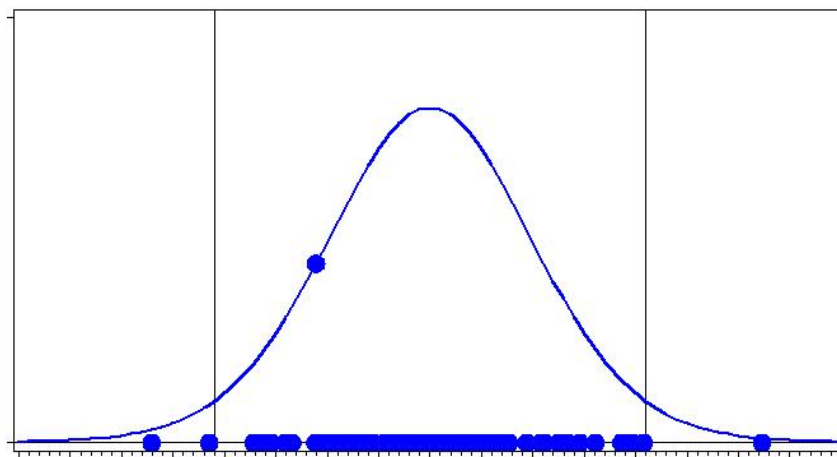
We address to everyone who wants to use visualization as a didactical tool or is interested in the technical background itself. SAS software refers to Version 9.1.3.

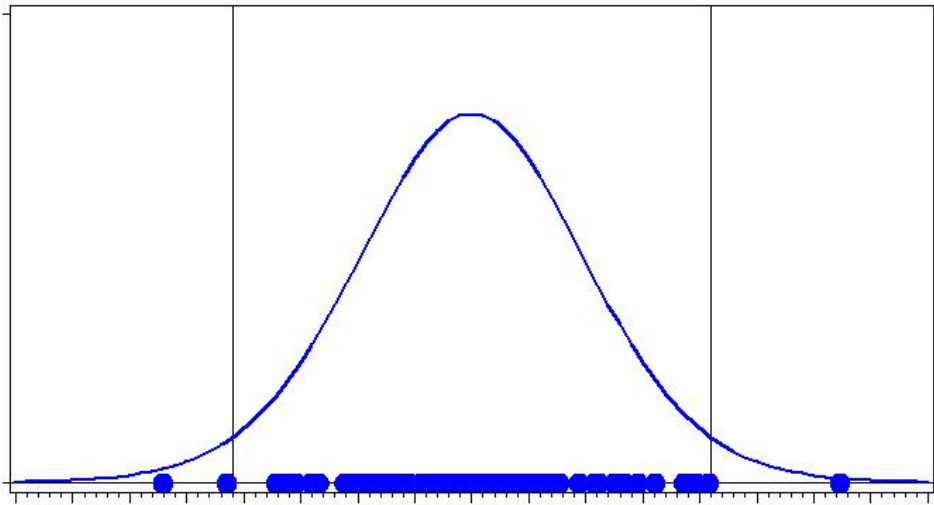
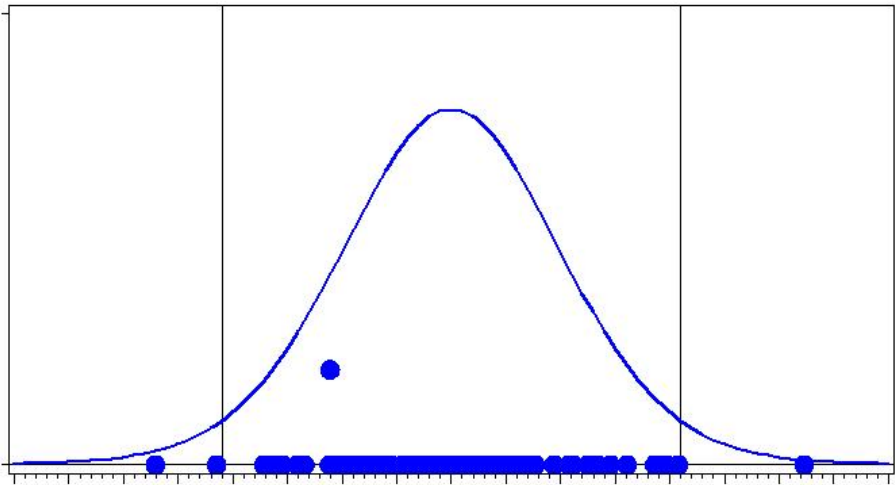
INTRODUCTION

Making medical students or physicians understand statistics is quite a challenge. They are typically not educated to think in abstract terms, so visualization is an excellent tool to obtain more insight on basic but abstract statistical concepts. Physicians want to draw conclusions from data. Therefore, they have to understand statistical testing and should be able to interpret output from statistical software packages. In particular, the meaning of significance levels, and the differences between them and p-values are essential. They must also be aware of power and the importance of sample size considerations. Bridging the gap between physicians' "real world" and statisticians' imaginations, simulations together with visualization are of great value. In the following, we demonstrate how to use visualizations using animated graphs to improve the understanding of the items mentioned above.

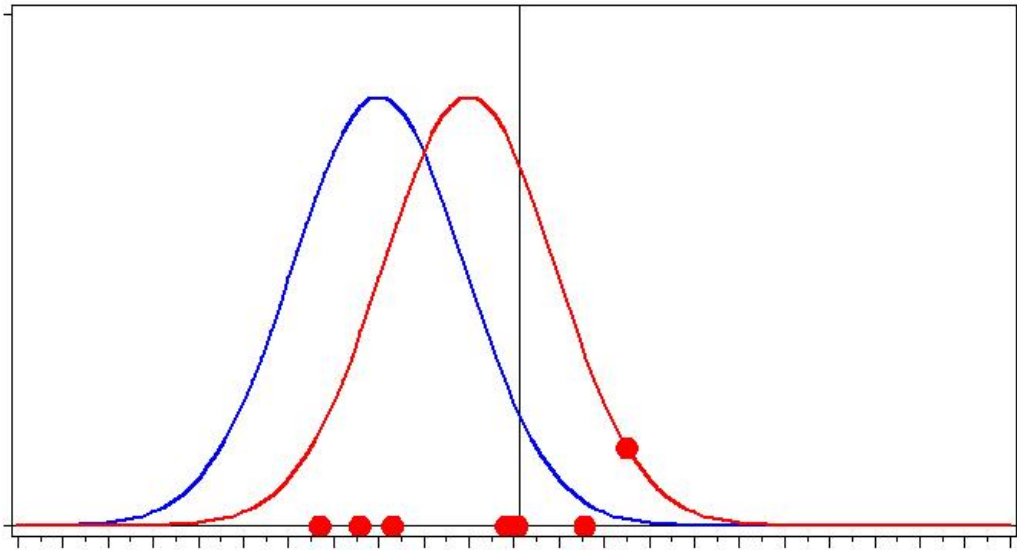
THE ANIMATION

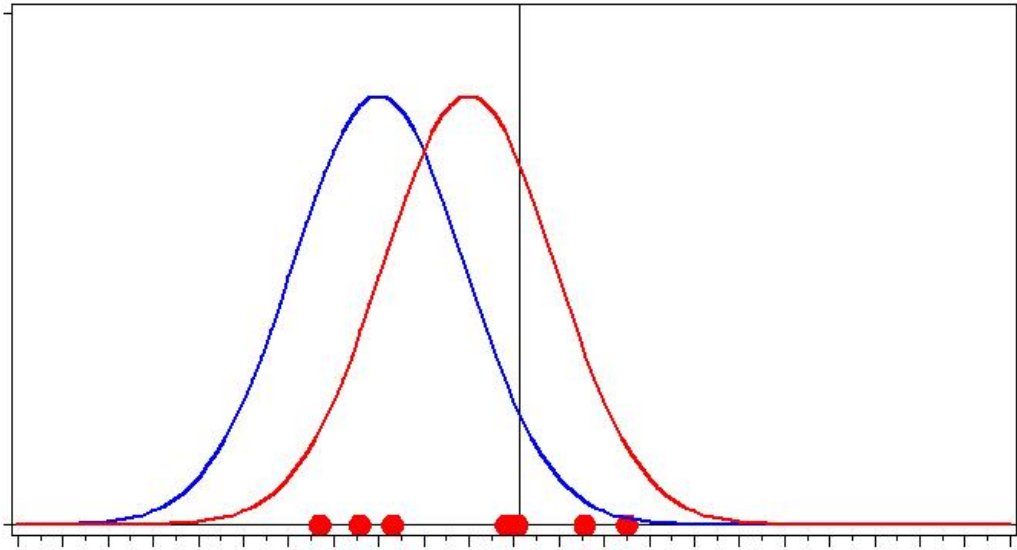
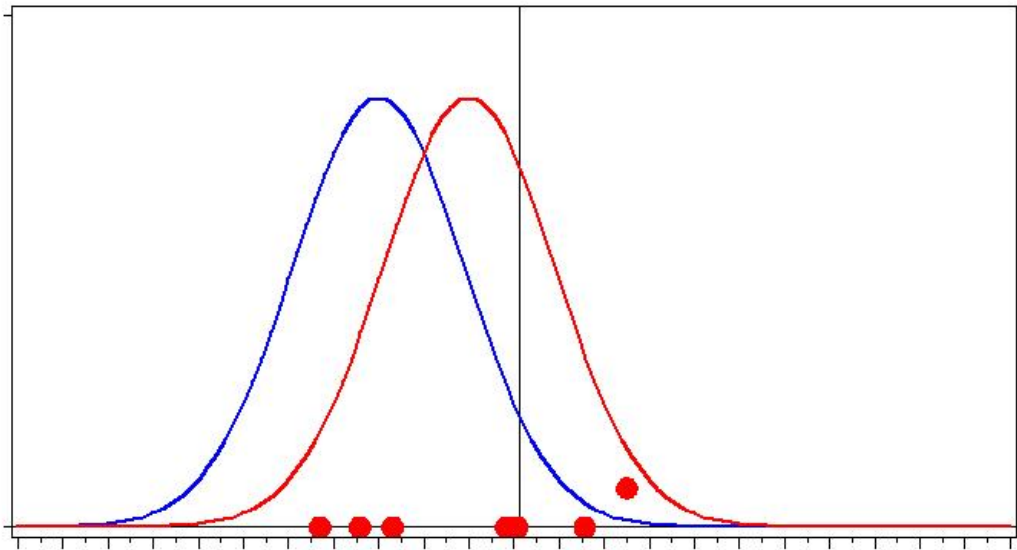
When performing any statistical test, all information of sample values is concentrated in a single value called realisation (value) of the test statistic. The test statistic is a random variable following a specific distribution. To explain significance level we use a two-sided one-sample t-test at a significance level of 5%. We assume that a study is repeated hundred times. Under the assumption that the null hypothesis holds the values of the test statistic can be associated with dropping dots from the shape of the corresponding probability density. The next figures demonstrate this approach using a t-distribution with 19 degrees of freedom (20 observations per sample). Proc gplot is used to create the following sequence of images.



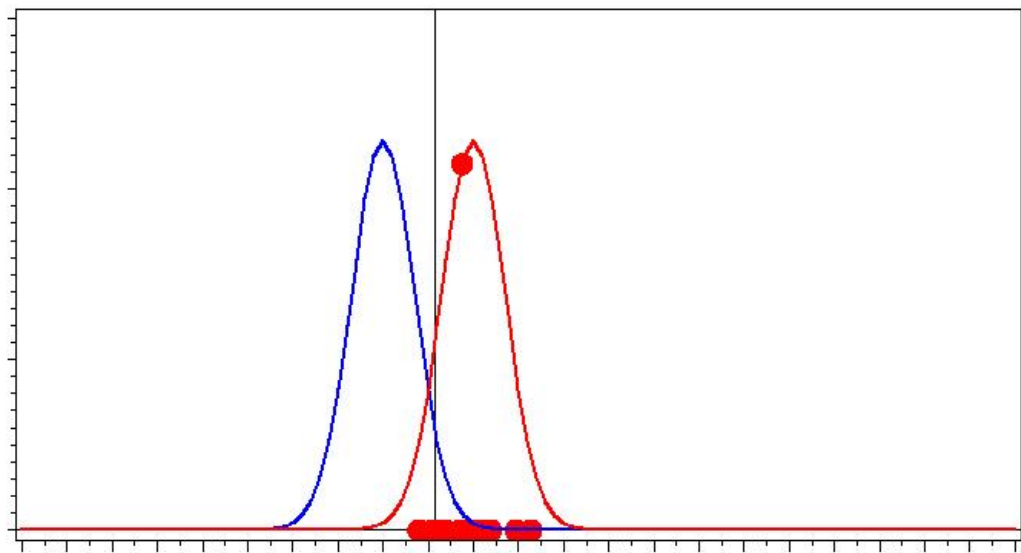


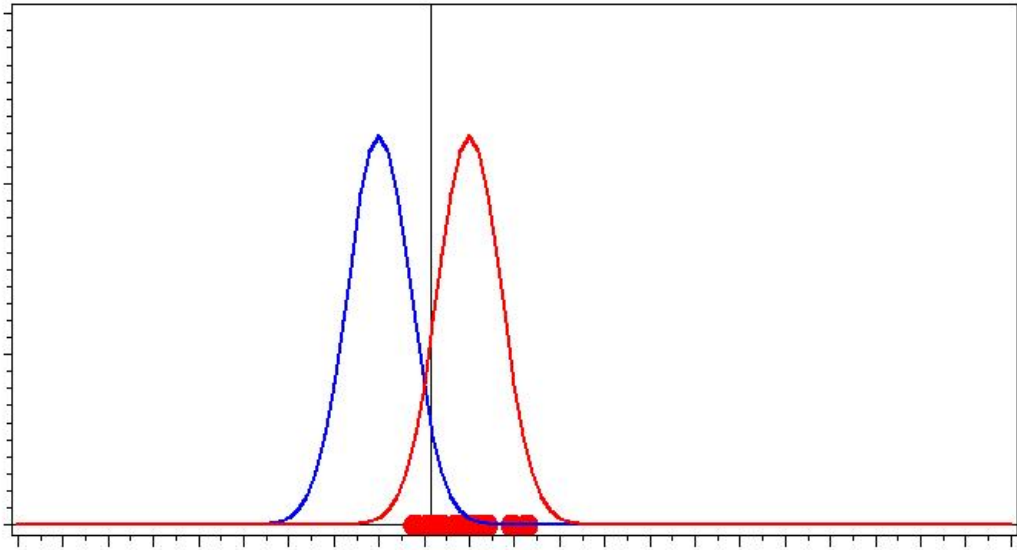
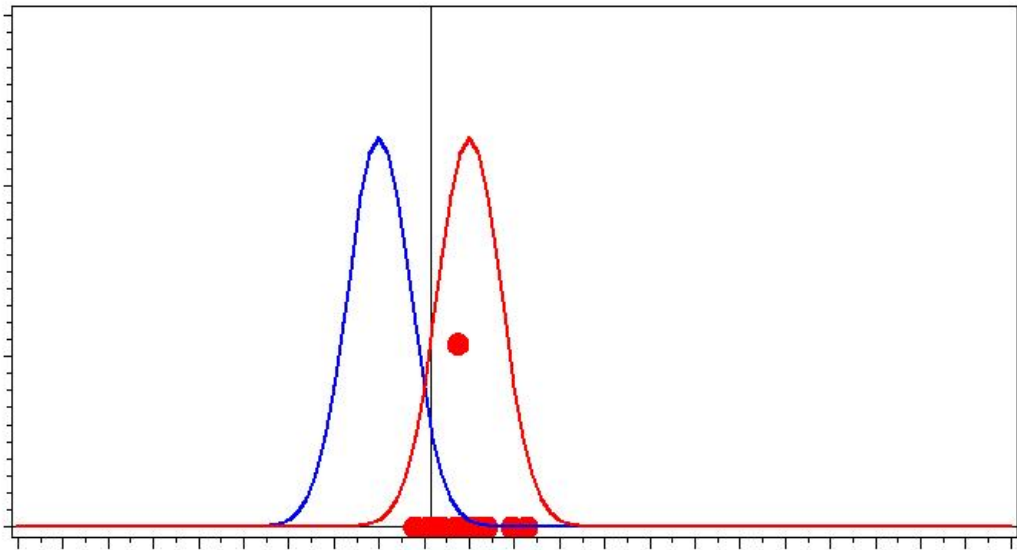
To demonstrate power and sample size considerations we use the one-sided one-sample z-test. The test should detect a difference of one third of the standard deviation. Performing the test with a sample size of ten and a significance level of 5% we get these simulations (blue: null hypothesis blue; red: alternative hypothesis):





A sample size of 75 observations will result in the following pictures:





Finally, this sequence of images is linked together to produce the animation. In the next paragraph, we show how to perform this task using the GIFANIM device driver. (Note that the presentation of the paper is a video poster, so you can watch the entire animation.)

THE GIFANIM DRIVER

The GIFANIM driver provides a mechanism for combining GIF images created with SAS/GRAPH procedures, allowing you to create animated GIFs. The behaviour of the driver is controlled by graphics options that enable you to specify, for instance, delay time, iteration count, transparency, or disposal methods. Before using the GIFANIM device driver, you should become familiar with the animation process controls. The process involved with creating an animated GIF with the GIFANIM driver requires that you, the animator, take control of the job sequence and ensure that the resulting data stream is constructed properly. The GIFANIM data stream has three parts: Header, Body, and Trailer. Each portion of the data stream is equally important and must be present. Otherwise, an incomplete or unreadable animation sequence will result.

PREPARING THE HEADER

When creating a new animated GIF data stream, you must issue GOPTIONS GSFMODE=REPLACE prior to the invocation of the first SAS/GRAPH procedure in the SAS job. The driver will then construct a new data stream by writing a valid GIF header and graphic data from the first procedure.

PREPARING THE BODY

After the first SAS/GRAPH procedure has been executed, you must construct the body of the GIF animation. You can think of the Body as all the graphic images between the first and last image. Set GOPTIONS GSFMODE=APPEND to signal the GIFANIM driver to suppress the header information and to begin appending graphic data to the current data stream. The GOPTIONS GSFMODE=APPEND statement must appear somewhere between the first and second SAS/GRAPH procedures.

Note: If you use BY-group processing on the first graphics procedure to generate multiple graphs, they are automatically appended to the same GIF file. Thus, you do not need GSFMODE=APPEND for that first procedure. If you do not use a second graphics procedure to append additional graphs to the GIF file, you do not need this Body section in your program.

PREPARING THE TRAILER

The final step is to mark the end of the animation by appending a GIF trailer ('3B'x) to the data stream. The way to do this depends on whether the last procedure uses BY-group processing.

- Without BY-group processing, set GOPTIONS GEPILOG='3B'X before the last SAS/GRAPH procedure.
- With BY-group processing, do not assign a value to GEPILOG; otherwise your GIF animation sequence will be incomplete. Because a GEPILOG is written after each graph in a BY-group, the GIF decoder will interpret the first '3B'x as the end of the animation. Instead, you should use a DATA step to add the trailer to the data stream:

```
data _null_;
  file out recfm=n mod;
  put '3B'x;
run;
```

THE CODE

We present the code for the power animation. There are always several possibilities to write code for a specific problem. The code presented in this paper should be taken as a suggestion .

```
/* assign the destination for the output file*/
filename out 'my_file.gif';

/*set the graphics environment and assign the appropriate
graphics options for the animation */
goptions reset=all
          device=gifanim
          gsfname=out
          gsfmde=replace
```

```

        iteration=0
        delay=80
        ftext=swissb
        cback=white
        ctext=white
        htext=1
    ;
%let n=100; /* number of studies = number of random numbers */

/*The following data steps create data for the sequence of graphs*/

data a;
call streaminit(111);
do i=1 to &n;

    x=RAND( 'NORMAL',1,.95) ;
    y=PDF( 'NORMAL',x,1,.95 ) ;
    output;
end;
run;

data b;
set a;
do k=1 to &n+1-i;
    l=&n+1-k;
    output;
end;
run;

proc sort data=b;
by l;
run;

data c;
set b;
by l;
z=0;
w=1;
output;
z=0;
w=2;
output;
z=0;
w=3;
output;
run;
proc sort data=c;
by l w;
run;
data d;
set c;
if i=1 and w=1 then z=y;
if i=1 and w=2 then z=y/2;
run;

data norm;
do x=-4 to 7 by .1;
    z1=PDF( 'NORMAL',x,0,.95);

```



```

zal=PDF('NORMAL',x,1,.95);

    output;
end;
run;
data norm1;
    set norm;
    do l=1 to &n;
        do w=1 to 3;
            output;
        end;
    end;
run;
data plot;
    set norm1 d;
run;
proc sort data=plot;
    by l w;
run;

/* The following code sets the symbol statements for the
   graphs and uses proc gplot to produce the series of plots
   that will be animated by the GIFANIM driver */

symbol1 v=none c= blue i=join w=2;
symbol2 v=dot c= red h=2 ;
symbol3 v=none c= red i=join w=2 ;

proc gplot data=plot uniform;
    plot z*x=2 z1*x=1 zal*x=3 / overlay vref=0 href=    1.56

    nolegend
vaxis=0 to .5 by 0.5
haxis=-4 to 7 by 0.5
;
by l w;
run;
quit;

/* end the animation */
data _null_;
    file out recfm=n mod;
    put '3B'x;
run;

/*disassociates the currently assigned filerefs*/
filename out clear;

```

CONCLUSION

The GIFANIM device driver can be used to create a graphic display as an animated GIF file. Animations are of great value for teaching and presentations in general. As you can see there is no more effort in producing an animated GIF file than a static graphic display.

REFERENCES

SAS/GRAPH 9.1 Reference,

<<http://support.sas.com/documentation/onlinedoc/sas9doc.html>>

CONTACT INFORMATION

Ass.Prof.Dipl.Ing.Dr.
Barbara Schneider
Medical University of Vienna
Section for Medical Statistics
Spitalgasse 23
A - 1090 Vienna
Austria
barbara.schneider@meduniwien.ac.at

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.