

Paper 174-30

SAS® Graphics in Motion

Barbara Schneider Medical University of Vienna, Austria

Kristan Schneider University of Vienna, Austria

ABSTRACT

Animated graphics become more and more popular to enhance the effectiveness of presentations. SAS Software provides the GIFANIM device driver to create sequences of images that are displayed from a single GIF file e.g. stored on a Web page or Microsoft® Office PowerPoint® slide. These linked sequences can result in slide shows or in movie like performances. The poster focuses on an example that we use in a biostatistical course for medical students. An intuitive understanding of random phenomena should be induced by the discussed animation. Animated graphics are of great value to transfer didactic concepts. SAS code is provided for technical details. SAS software refers to Version 9.1.3. The paper might be interesting for teachers in statistics as well as for presenters in general.

INTRODUCTION

Teaching statistics to medical students or physicians is a big challenge. Physicians do not think in abstract terms, so visualization is an excellent tool to obtain more insight on basic but abstract statistical concepts. First it is very important that physicians understand what “random” means. They must be aware of the difference between random and deterministic phenomena. When a physician takes systolic blood pressure measurements from several different patients the values can be viewed as resulting from a normal distribution – the statistician uses the term normally distributed random variable. In the following we demonstrate how this scenario can be visualized using animated graphs.

THE ANIMATION

Taking measurements from one patient after the other can be associated with dropping dots from the shape of the corresponding distribution function of the random variable. The next figures demonstrate this approach using a normal distribution as an example. Proc gplot is used to create the following sequence of images.

Figure 1
l=1: first patient
w=1: first position

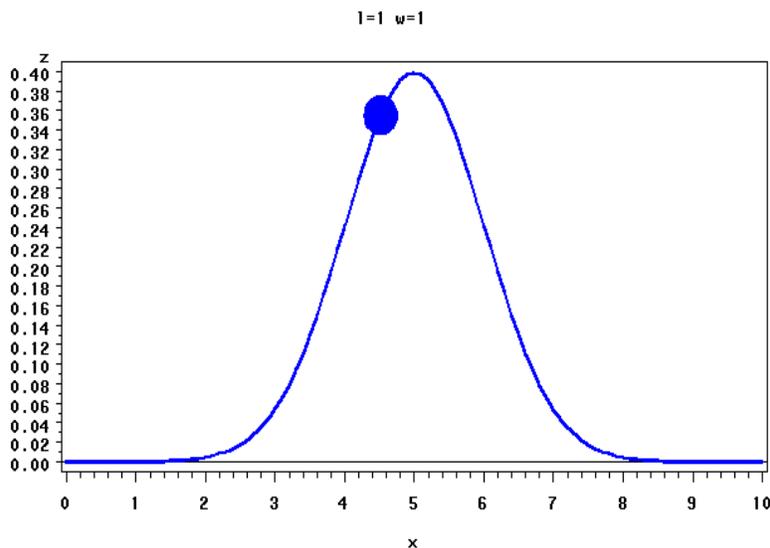


Figure 2
l=1: first patient
w=2: second position

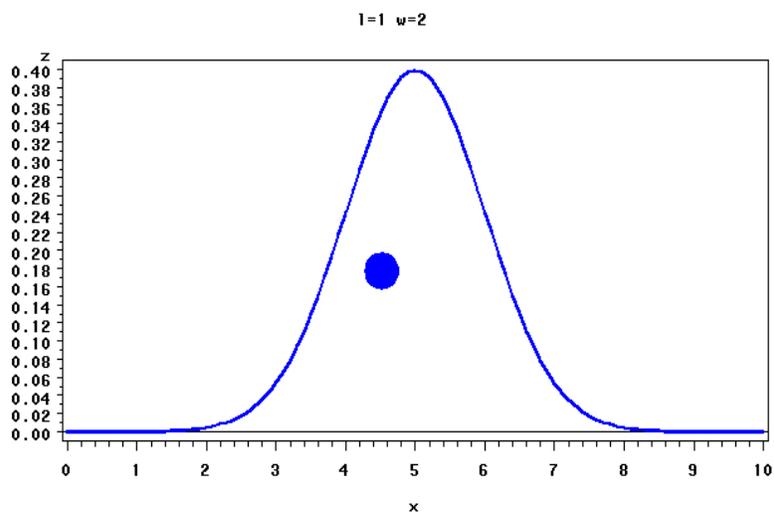


Figure 3
l=1: first patient
w=3: third position

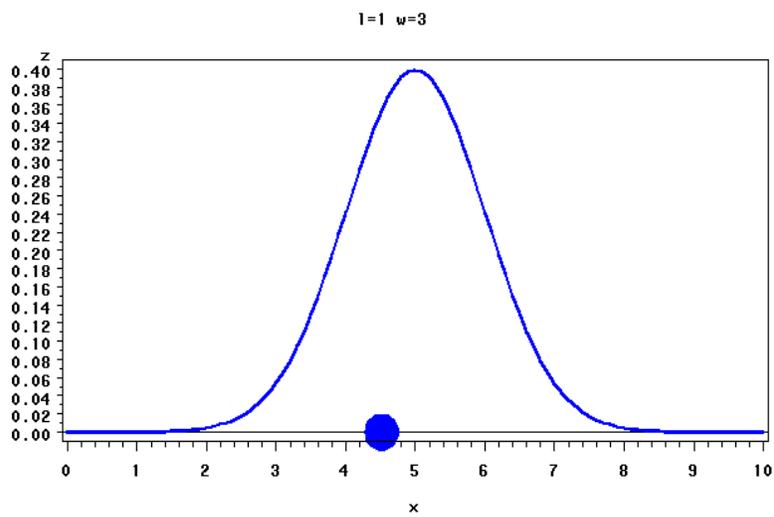


Figure 4
l=2: second patient
w=1: first position

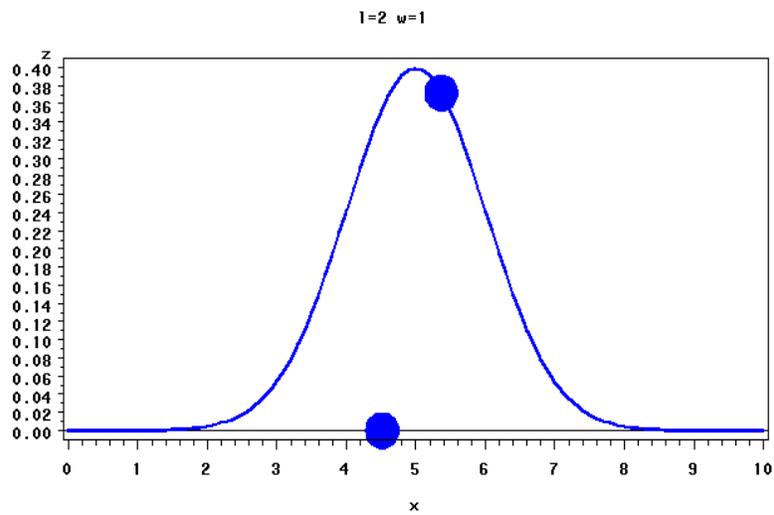


Figure 5
 $l=2$: second patient
 $w=2$: second position

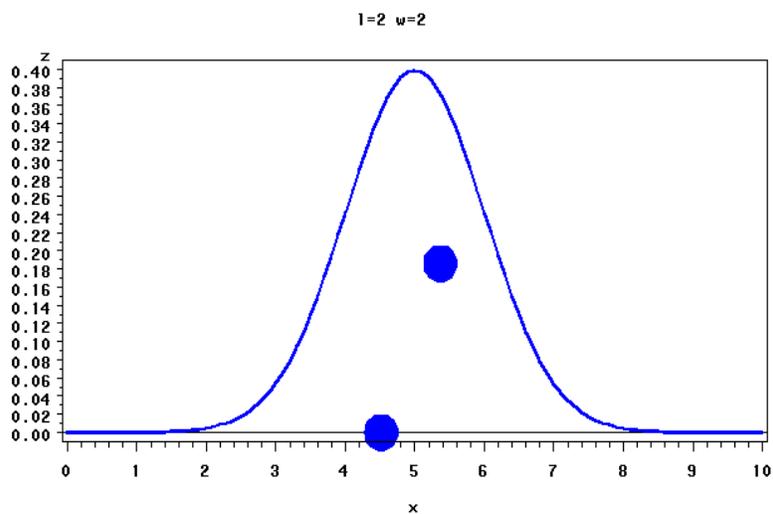


Figure 6
 $l=2$: second patient
 $w=3$: third position

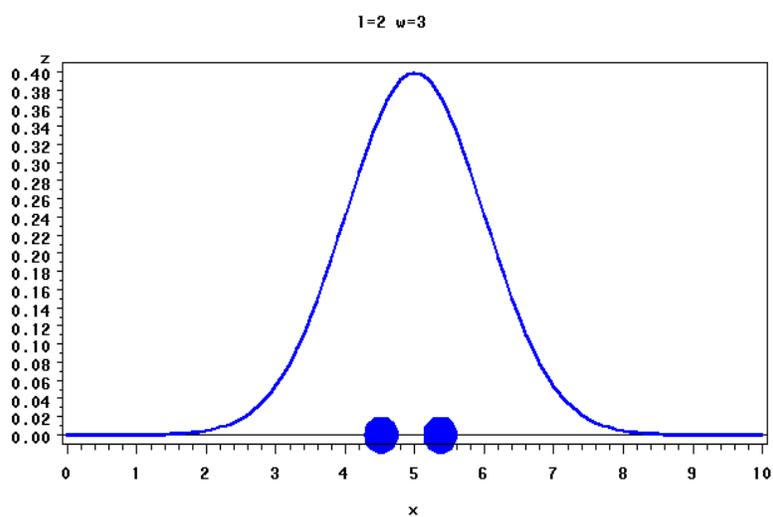


Figure 7
 $l=6$: sixth patient
 $w=1$: first position

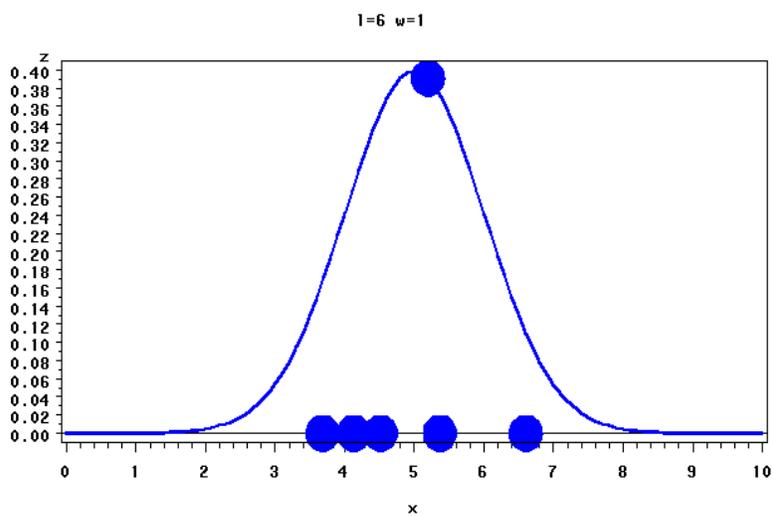


Figure 8
 $l=6$: sixth patient
 $w=2$: second position

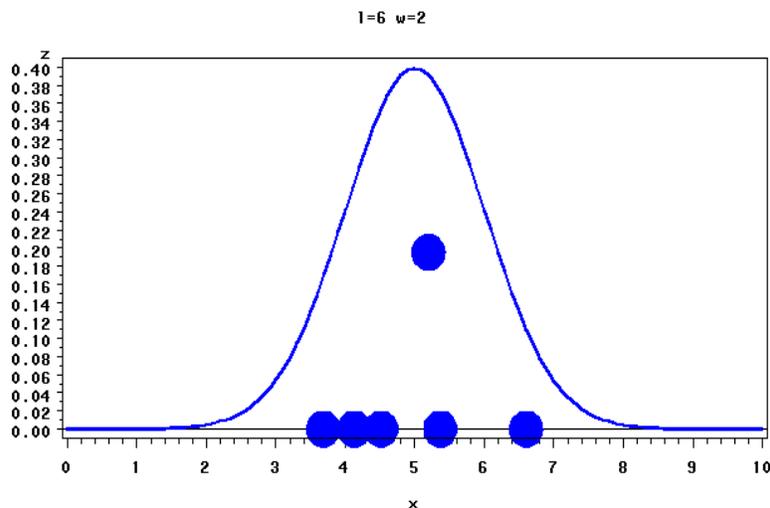
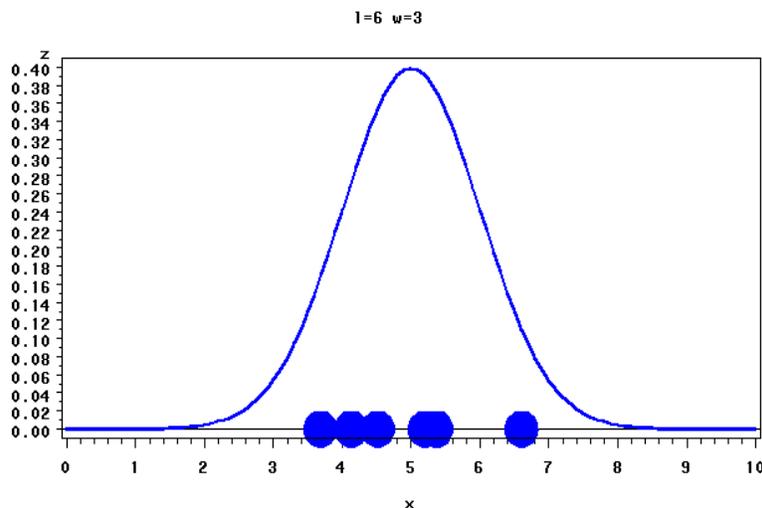


Figure 9
 $l=6$: sixth patient
 $w=3$: third position



Finally this sequence of images has to be linked together to produce the animation. In the next paragraph, we show how to perform this task using the GIFANIM device driver.

THE GIFANIM DRIVER

The GIFANIM driver provides a mechanism for combining GIF images created with SAS/GRAPH procedures that allow you to create GIF animations. The behaviour of the driver is controlled by graphics options that enable you to set such things as delay time, iteration count, transparency, and 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 file 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

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

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

data a;
do i=1 to &n;
```

```
x=5+rannor(111);
y=pdf("normal",x,5,1);
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=l and w=1 then z=y;
if i=l and w=2 then z=y/2;
run;

data norm;
do x=0 to 10 by .1;
z1=pdf("normal",x,5,1);
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=3;
symbol2 v=dot c= blue h=4 ;

proc gplot data=plot uniform;
plot z*x=2 z1*x=1 / overlay vref=0 nolegend ;
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.