

Computing Guide

SAS commands used in labs (SasCode.out)

A full listing of SAS commands can be found in the manuals for this package, at the computer centre in the Henrietta Harvey building. SAS also provides on-line help for commands.

SAS routines are built out of commands, these routines are modified to suite the situation and goals of computation. Assignments in this course can be accomplished by modifying one of the following routines, depending on the situation or goals of computation.

SAS routines are saved in control files (with extension .sas) which are turned over to the package for processing.

Sections of the routines have been assigned names, which are used in other parts of the lab manual. These names occur in *boldface italics* to the right of the routine box.

```
Title 'Daphnia longispina data.';
Options linesize=72;
Data A; infile 'srbx9_5.dat' obs = 7;
      Input SeriesI SeriesII;
Proc print;
Proc means;
      Var SeriesI SeriesII;
```

*Define Data
from file*

*Output
to log file*

*Compute
a variance*

```
Title 'Daphnia longispina data.';
Options linesize=72;
Data A; infile 'srbx9_5.dat' obs = 7;
      Input SeriesI SeriesII;
Data B; Set A;
      Age = SeriesI; Series = 'I'; output;
      Age = SeriesII; Series = 'II'; output;
Proc print;
Proc means; By Series;
      Var Age;
```

*Define data
from a file*

Reorganize

*Output
to log file*

*Compute
a variance*

SAS commands

SAS commands used in labs

```
Title 'Daphnia longispina data.';
Options linesize=72;
Data A; infile 'srbx9_5.dat' obs = 7;
  Input SeriesI SeriesII;
Data B; Set A;
  Age = SeriesI; Series = 'I'; output;
  Age = SeriesII; Series = 'II'; output;
Data C(Keep = Age Series);
  Set B;
  Filename srOUT 'srbx9_5b.dat';
  File srOUT;
  Put @6 Series @2 Age;
Proc Print;
```

***Define data
from a file***

Reorganize

***Define data
to a file***

```
proc glm data=a outstat=z0;
  class d;
  model r=d / ss3;
  output out=out1 r=res p=pred;
  proc plot data=out1; plot res*pred/vref=0;
  proc univariate data=out1 normal; var res;
```

***Run GLM
(Anova)***

Note that the glm command in SAS assumes that all explanatory variables are regression variables. ANOVA (classification variables are defined by the 'class' statement.

```
proc glm data=a outstat=z0;
  model r=d / ss3;
  Etc..
```

***Run GLM
(Regression)***

```
proc glm data=a outstat=z0;
  class da;
  model r=da db da*db / ss3;
  Etc...
```

***Run GLM
(Ancova)***

On the next page is a program written by Tammo Bult, to compute p-values by randomization, for the general linear model, when the assumptions concerning residuals are not met.

```

options nocenter nodate linesize=75 pagesize=35;

/* READ DATA */
  data a;      infile datxd2 missover;
              input d 2 r 1;

/* SPLIT UP DATASET IN 2 SEP. DATASETS ON EXPL.VAR.(1) AND RESP.VAR.(2) */
  data ar; set a; keep r; /* response variable */
  data ae; set a; drop r; /* explanatory variables */

/* PERFORM INITIAL ANALYSIS */
  proc glm data=a outstat=z0; class d;
    model r=d / ss3;
    output out=out1 r=res p=pred;
  proc plot data=out1; plot res*pred/vref=0;
  proc univariate data=out1 normal; var res;

/* SET NUMBER OF RANDOMIZATIONS AND NUMBER OF OBSERVATIONS */
  %let nrand=434;
  %let nobs1=20;

/* MISCELLANEOUS */
  data y1; input _SOURCE_ $ F; cards;
              ;
              ;

/* MACRO THAT PERFORMS RANDOMIZATION SCHEME */
  %macro loop;
  %do l=1 %to &nrand;
    data b; retain seed &l; do i=1 to &nobs1;
      v1=ranuni(seed); output; end; keep v1;
    data c; merge b ar;
    proc sort data=c; by v1;
    data c; set c; keep r;
    data d; merge c ae;
  /* analysis on randomized data */
    proc glm data=d outstat=z1 noprint;
      class d;
      model r=d / ss3;
  /* organize F values calculated via this analysis */
    data z1; set z1;
      keep _SOURCE_ F;
      if F=. then delete;
    proc append base=y1 data=z1 force;
  %end;
  %mend loop;
  %loop;

/* CALCULATE P VALUES */
  data z0; set z0; F0=F;
              if F=. then delete;
              keep _SOURCE_ F0;

  data y1; set y1;
              if F=. then delete;
  proc sort data=z0; by _SOURCE_;
  proc sort data=y1; by _SOURCE_;
  data p0; merge y1 z0; by _SOURCE_;
  data p0; set p0; if F > F0 then do; nh=1; end;
              if F <= F0 then do; nh=0; end;
  proc means data=p0 noprint; var nh; by _SOURCE_;
    output out=p1 n=nobs sum=nh;
  data p1; set p1; pc=nh/nobs;
  proc print data=p1; var _SOURCE_ pc;
endsas;

```