A How-To Guide

Janzen and Stern published a paper in Evolution (1998, 52:1564-1571) describing the use of logistic regression for analyzing selection and a new method for transforming the logistic regression coefficients into selection gradients that can be used in equations describing microevolutionary change. It should be noted that the former is *not* a novel contribution to the field (others have also used logistic regression for this purpose) but the latter is. The point of this web page is simply to make it easier for users to implement these statistical methods in their own work on selection. We provide a SAS program that performs the logistic regression analysis and calculates the transformed selection gradients. We also provide the dataset from Bumpus' classic study of sparrow mortality and the output from the SAS logistic regression analysis of a subset (the males) of those data.

One important point in performing logistic regression in SAS is to be sure that your model predicts the probability of survival rather than the probability of mortality. The program below achieves this goal by having the data sorted with survivors first and using the order=data option in PROC LOGISTIC. Another option, assuming the traditional 0=dead/1=survived coding, is to use the DESCENDING option.

You will likely need to modify the SAS program to analyze your own datasets. It is especially important to note where the survival variable and the predictors appear in the input statement as this arrangement affects some later parts of the program. Please contact Fred Janzen *email:* fjanzen@iastate.edu if you have thoughts, comments, or suggestions. Note that an S-PLUS program is also available (and considerably simpler). If interested, please contact Hal Stern *email:* sternh@uci.edu

Implementation:

- SAS Program for Selection Analysis using Logistic Regression
- Bumpus' Sparrow Data
- Output from SAS Logistic Regression Analysis of Bumpus' Data

---

SAS Program for Selection Analysis using Logistic Regression

```
/* Program to apply Janzen-Stern logistic regression average gradient approach*/
/* Input is the data set */
/* Output includes the logistic regression output and */
/* for each variable: logistic regression coefficient and standard error */
/* standardized logistic regression coefficient and standard error */
/* average gradient coefficient and standard error */
/* plus constant AG used in average gradient calculation */
/* proportion of survivors used for relative fitness */

options linesize=80;
```

```
/* Read in Bumpus data, transform variables, use only males */
/* Note that: surv = survival is the second input variable */
/* variables 3 through 11 are the predictors for which selection */
/* gradients are to be obtained */

filename bumpus 'bump.dat';
data bumpm;

        infile bumpus;
        input sex surv len ext wt head humer femur tibio skull stern;
        len = log(len);
        ext = log(ext);
        wt = log(wt)/3;
        head = log(head);
        humer = log(humer);
        femur = log(femur);
        tibio = log(tibio);
        skull = log(skull);
        stern = log(stern);
        if sex=1;


/* Save standard deviations for use in standardizing coeffients */
/* Note: only save standard deviations for the predictors */

proc means data = bumpm noprint;

        var len ext wt head humer femur tibio skull stern;
        output out=stdevs std = sd1 sd2 sd3 sd4 sd5 sd6 sd7 sd8 sd9;

proc transpose prefix = sdev data=work.stdevs out=sds;
data stdv;

        set work.sds;
        if _N_ > 2;
        dummy = 1;


/* Logistic regression --- saving estimates and variance/covariance */

proc logistic data = bumpm order = data outest=coef covout;

        model surv = len ext wt head humer femur tibio skull stern;
        output out=fit predicted=phat;


/* Create SAS data sets containing coefficients and standard errors */
/* The approach used here is kind of clumsy but works. */
/* The use of dummy = 1 is needed for later combination of variables */
```

```
/* For other datasets: */
/* change the 11 below to (2 + number of predictors) */
/* add one line for each coefficient standard error to logistse data step */

data logistc;

        set work.coef;
        if _N_ = 1;

proc transpose prefix = coef data = work.logistc out=logistcf;
data logcoef;

        set work.logistcf;
        dummy = 1;
        if _N_ > 1 and _N_ < 11;


data logistse;

        set work.coef;
        if _N_ = 3 then serr = sqrt(len);
        if _N_ = 4 then serr = sqrt(ext);
        if _N_ = 5 then serr = sqrt(wt);
        if _N_ = 6 then serr = sqrt(head);
        if _N_ = 7 then serr = sqrt(humer);
        if _N_ = 8 then serr = sqrt(femur);
        if _N_ = 9 then serr = sqrt(tibio);
        if _N_ = 10 then serr = sqrt(skull);
        if _N_ = 11 then serr = sqrt(stern);
        if _N_ > 2;
        dummy = 1;


/* Compute constant needed for average gradient and relative fitness
calculations */

data fit2;

        set work.fit;
        agconst = phat*(1-phat);

proc univariate data=fit2 noprint;

        var surv agconst;
        output out=avggrad mean=relfit agmean;

data ag;

        set work.avggrad;
        dummy = 1;
```

```
/* Combine the various pieces to produce output and relative fitness
calculations */

data final;

    merge work.stdv work.logcoef work.logistse work.ag; by dummy;
    ag = agmean;
    rf = relfit;
    alpha = coef1*sdev1;
    se = serr*sdev1;
    bavgrad = coef1*sdev1*ag/rf;
    bavgrads = serr*sdev1*ag/rf;

proc print;

    var coef1 serr alpha se bavgrad bavgrads ag relfit;

run;
```

Bumpus' Sparrow Data

```
1 1 154 241 24.5 31.2 .687 .668 1.022 .587 .830
1 1 160 252 26.9 30.8 .736 .709 1.180 .602 .841
1 1 155 243 26.9 30.6 .733 .704 1.151 .602 .846
1 1 154 245 24.3 31.7 .741 .688 1.146 .584 .839
1 1 156 247 24.1 31.5 .715 .706 1.129 .575 .821
1 1 161 253 26.5 31.8 .780 .743 1.144 .607 .893
1 1 157 251 24.6 31.1 .741 .736 1.153 .610 .862
1 1 159 247 24.2 31.4 .728 .718 1.126 .609 .793
1 1 158 247 23.6 29.8 .703 .673 1.079 .602 .820
1 1 158 252 26.2 32.0 .749 .739 1.153 .614 .857
1 1 160 252 26.2 32.0 .741 .723 1.129 .624 .892
1 1 162 253 24.8 32.3 .766 .752 1.134 .633 .923
1 1 161 243 25.4 31.8 .721 .722 1.126 .597 .891
1 1 160 250 23.7 29.8 .730 .703 1.103 .590 .820
1 1 159 247 25.7 31.4 .729 .717 1.141 .592 .927
1 1 158 253 25.7 31.9 .743 .699 1.150 .600 .860
1 1 159 247 26.5 31.6 .733 .714 1.155 .611 .923
1 1 166 253 26.7 32.5 .767 .765 1.230 .600 .878
1 1 159 247 23.9 31.4 .752 .723 1.113 .602 .825
1 1 160 248 24.7 31.3 .752 .737 1.176 .603 .803
1 1 161 252 28.0 31.8 .770 .731 1.190 .590 .885
1 1 163 251 27.9 31.9 .769 .745 1.168 .622 .860
1 1 156 242 25.9 32.0 .723 .711 1.116 .609 .886
1 1 165 251 25.7 32.2 .751 .742 1.161 .613 .865
```

1 1 160 247 26.6 32.4 .728 .707 1.108 .590 .836
1 1 158 244 23.2 31.6 .730 .713 1.142 .585 .888
1 1 160 242 25.7 31.6 .709 .705 1.124 .620 .788
1 1 157 245 26.3 32.2 .741 .726 1.143 .595 .850
1 1 159 244 24.3 31.5 .723 .698 1.107 .615 .847
1 1 160 253 26.7 32.1 .739 .714 1.117 .592 .864
1 1 158 245 24.9 31.4 .726 .703 1.119 .580 .854
1 1 161 247 23.8 31.4 .735 .694 1.101 .602 .789
1 1 160 247 25.6 32.3 .756 .745 1.135 .607 .902
1 1 160 247 27.0 32.0 .755 .736 1.174 .631 .873
1 1 153 241 24.7 32.2 .728 .680 1.092 .592 .884
1 0 165 249 26.5 31.0 .738 .701 1.095 .606 .847
1 0 160 245 26.1 32.0 .736 .709 1.109 .611 .842
1 0 161 249 25.6 32.3 .743 .718 1.128 .602 .828
1 0 162 246 25.9 32.3 .738 .709 1.135 .607 .869
1 0 163 250 25.5 32.5 .752 .731 1.197 .623 .888
1 0 162 247 27.6 31.8 .731 .719 1.113 .597 .869
1 0 163 246 25.8 31.4 .689 .662 1.073 .604 .836
1 0 161 246 24.9 30.5 .739 .726 1.138 .580 .803
1 0 160 242 26.0 31.0 .745 .713 1.105 .600 .803
1 0 162 246 26.5 31.5 .720 .696 1.092 .606 .809
1 0 160 249 26.0 31.4 .726 .689 1.097 .602 .850
1 0 161 250 27.1 31.6 .737 .711 1.120 .631 .852
1 0 162 248 25.1 31.9 .744 .722 1.154 .591 .839
1 0 165 252 26.0 32.3 .726 .710 1.145 .609 .887
1 0 161 243 25.6 32.5 .709 .707 1.122 .607 .832
1 0 161 244 25.0 31.3 .702 .685 1.082 .595 .874
1 0 162 248 24.6 31.0 .713 .700 1.086 .590 .837
1 0 164 244 25.0 31.2 .703 .690 1.074 .608 .795
1 0 158 247 26.0 32.0 .729 .710 1.145 .607 .803
1 0 162 253 28.3 31.8 .752 .718 1.152 .600 .857
1 0 156 239 24.6 30.5 .659 .658 1.042 .570 .810
1 0 166 251 27.5 31.5 .720 .691 1.118 .612 .847
1 0 165 253 31.0 32.4 .765 .750 1.183 .613 .905
1 0 166 250 28.3 32.4 .754 .718 1.179 .607 .916
1 1 156 246 24.6 32.0 .741 .735 1.167 .592 .849
1 1 156 245 25.5 32.1 .761 .717 1.147 .620 .816
1 1 163 248 24.8 32.2 .742 .733 1.165 .606 .854
1 1 163 248 26.3 33.0 .736 .704 1.148 .609 .839
1 1 160 250 24.4 31.5 .746 .715 1.173 .604 .893
1 1 156 237 23.3 30.6 .692 .664 1.011 .588 .774
1 1 162 253 26.7 32.0 .759 .734 1.197 .630 .878
1 1 163 254 26.4 32.0 .766 .750 1.165 .605 .886
1 1 164 251 26.9 32.0 .755 .742 1.171 .620 .886
1 1 163 244 24.3 31.3 .718 .680 1.082 .610 .892
1 1 160 247 27.0 31.5 .764 .732 1.177 .617 .846
1 1 160 250 26.8 32.5 .764 .729 1.123 .635 .842
1 1 158 247 24.9 32.4 .745 .724 1.139 .588 .865

1 1 158 249 26.1 32.2 .742 .736 1.148 .602 .817
1 1 158 243 26.6 32.4 .747 .711 1.163 .612 .891
1 1 155 237 23.3 30.2 .685 .653 1.011 .587 .794
1 0 160 249 24.2 30.4 .740 .717 1.130 .620 .840
1 0 156 236 26.8 30.0 .690 .671 1.067 .563 .832
1 0 158 240 23.5 31.0 .715 .702 1.113 .595 .805
1 0 166 245 26.9 31.7 .715 .695 1.107 .601 .847
1 0 165 255 28.6 31.5 .766 .744 1.175 .613 .854
1 0 157 238 24.7 31.2 .680 .677 1.156 .599 .769
1 0 164 250 27.3 31.8 .764 .726 1.171 .588 .860
1 0 166 256 25.7 31.7 .752 .751 1.187 .595 .858
1 0 167 255 29.0 32.2 .765 .745 1.197 .638 .855
1 0 161 246 25.0 31.5 .739 .707 1.123 .587 .850
1 0 166 254 27.5 31.4 .760 .742 1.124 .604 .914
1 0 161 251 26.0 31.5 .731 .707 1.122 .589 .828
0 1 156 245 25.3 31.6 .729 .710 1.152 .620 .809
0 1 154 240 22.6 30.4 .705 .686 1.103 .584 .770
0 1 153 240 25.1 31.0 .724 .713 1.123 .585 .812
0 1 153 236 23.2 30.9 .698 .678 1.132 .596 .795
0 1 155 243 24.4 31.5 .734 .736 1.170 .596 .801
0 1 163 247 25.1 32.0 .748 .734 1.166 .602 .821
0 1 157 238 24.6 30.9 .726 .727 1.175 .588 .797
0 1 155 239 24.0 32.8 .732 .742 1.175 .601 .835
0 1 164 248 24.2 32.7 .752 .752 1.201 .604 .830
0 1 158 238 24.9 31.0 .741 .689 1.091 .592 .866
0 1 158 240 24.1 31.3 .733 .706 1.107 .591 .867
0 1 160 244 24.0 31.1 .731 .730 1.152 .589 .808
0 1 161 246 26.0 32.3 .758 .732 1.154 .623 .859
0 1 157 245 24.9 32.0 .752 .740 1.186 .593 .787
0 1 157 235 25.5 31.5 .712 .704 1.132 .611 .781
0 1 156 237 23.4 30.9 .708 .691 1.123 .613 .798
0 1 158 244 25.9 31.4 .729 .705 1.146 .597 .851
0 1 153 238 24.2 30.5 .715 .707 1.116 .595 .821
0 1 155 236 24.2 30.3 .727 .705 1.120 .585 .790
0 1 163 246 27.4 32.5 .732 .711 1.163 .630 .862
0 1 159 236 24.0 31.5 .709 .713 1.129 .607 .845
0 0 155 240 26.3 31.4 .709 .710 1.125 .614 .815
0 0 156 240 25.8 31.5 .715 .678 1.127 .597 .812
0 0 160 242 26.0 32.6 .710 .732 1.157 .597 .854
0 0 152 232 23.2 30.3 .676 .683 1.048 .590 .780
0 0 160 250 26.5 31.7 .741 .731 1.187 .615 .886
0 0 155 237 24.2 31.0 .727 .723 1.118 .610 .787
0 0 157 245 26.9 32.2 .766 .751 1.227 .620 .841
0 0 165 245 27.7 33.1 .780 .757 1.195 .633 .895
0 0 153 231 23.9 30.1 .680 .662 1.042 .592 .781
0 0 162 239 26.1 30.3 .709 .685 1.092 .587 .911
0 0 162 243 24.6 31.6 .741 .729 1.162 .605 .840
0 0 159 245 23.6 31.8 .727 .700 1.129 .610 .855

0 0 159 247 26.0 30.9 .711 .666 1.098 .580 .749
0 0 155 243 25.0 30.9 .730 .711 1.127 .598 .839
0 0 162 252 24.8 31.9 .752 .738 1.180 .615 .875
0 0 152 230 22.8 30.4 .682 .664 1.042 .551 .734
0 0 159 242 24.8 30.8 .717 .667 1.090 .575 .809
0 0 155 238 24.6 31.2 .706 .702 1.102 .588 .758
0 0 163 249 30.5 33.4 .767 .767 1.207 .640 .896
0 0 163 242 24.8 31.0 .713 .713 1.128 .607 .813
0 0 156 237 23.9 31.7 .718 .716 1.090 .611 .800
0 0 159 238 24.7 31.5 .726 .701 1.145 .600 .800
0 0 161 245 26.9 32.1 .751 .704 1.142 .607 .819
0 0 155 235 22.6 30.7 .695 .692 1.119 .584 .771
0 0 162 247 26.1 31.9 .751 .735 1.157 .618 .802
0 0 153 237 24.8 30.6 .732 .718 1.172 .594 .802
0 0 162 245 26.2 32.5 .728 .731 1.102 .614 .832
0 0 164 248 26.1 32.3 .739 .707 1.159 .592 .823

Output from SAS Logistic Regression Analysis of Bumpus' Data

The LOGISTIC Procedure

Data Set: WORK.BUMPM
Response Variable: SURV
Response Levels: 2
Number of Observations: 87
Link Function: Logit

Response Profile

Ordered
Value SURV Count

1 1 51
2 0 36

Model Fitting Information and Testing Global Null Hypothesis BETA=0

Intercept
Intercept and
Criterion Only Covariates Chi-Square for Covariates

AIC 120.008 85.432 .
SC 122.474 110.091 .
-2 LOG L 118.008 65.432 52.577 with 9 DF (p=0.0001)
Score . . 39.767 with 9 DF (p=0.0001)

Analysis of Maximum Likelihood Estimates

Parameter Standard Wald Pr > Standardized Odds
Variable DF Estimate Error Chi-Square Chi-Square Estimate Ratio

INTERCPT 1 524.5 203.9 6.6178 0.0101 . .
LEN 1 -121.5 31.4041 14.9709 0.0001 -1.339057 0.000
EXT 1 23.9622 31.7505 0.5696 0.4504 0.237328 999.000
WT 1 -67.4383 25.9018 6.7788 0.0092 -0.674262 0.000
HEAD 1 17.4699 19.1313 0.8339 0.3612 0.198370 999.000
HUMER 1 23.3579 24.7035 0.8940 0.3444 0.413056 999.000
FEMUR 1 18.5443 25.7978 0.5167 0.4722 0.342805 999.000
TIBIO 1 -3.1706 16.4709 0.0371 0.8474 -0.064029 0.042
SKULL 1 18.0801 16.4902 1.2021 0.2729 0.237809 999.000
STERN 1 19.8310 10.4105 3.6286 0.0568 0.458132 999.000

Association of Predicted Probabilities and Observed Responses

Concordant = 90.2% Somers' D = 0.805
Discordant = 9.7% Gamma = 0.806
Tied = 0.1% Tau-a = 0.395
(1836 pairs) c = 0.903

OBS COEF1 SERR ALPHA SE BAVGRAD BAVGRADS AG RELFIT

1 -121.509 31.4041 -2.42878 0.62772 -0.49958 0.12912 0.12058 0.58621
2 23.962 31.7505 0.43046 0.57038 0.08854 0.11732 0.12058 0.58621
3 -67.438 25.9018 -1.22298 0.46972 -0.25156 0.09662 0.12058 0.58621
4 17.470 19.1313 0.35980 0.39402 0.07401 0.08105 0.12058 0.58621
5 23.358 24.7035 0.74920 0.79236 0.15411 0.16298 0.12058 0.58621
6 18.544 25.7978 0.62178 0.86498 0.12790 0.17792 0.12058 0.58621
7 -3.171 16.4709 -0.11614 0.60331 -0.02389 0.12410 0.12058 0.58621
8 18.080 16.4902 0.43134 0.39341 0.08872 0.08092 0.12058 0.58621
9 19.831 10.4105 0.83096 0.43622 0.17092 0.08973 0.12058 0.58621

Last Updated 9/2/99