Workshop on the
R Statistical Package

Instructor: Dr. David Lieske, Mount Allison University
Time: 10:00am to 10:50pm
Date: Wednesday, March 19
Format: Kenneth C. Rowe Management Building

Workshop Overview

Topics

- Exploratory Data Analysis (EDA).
- Logistic Regression and Generalized Linear Models (GLM).
- Goodness-of-fit and Accuracy Assessment.
- Model Selection.
- Mapping Model Predictions.

Companion Files

- `amcr.txt`
- `rand.txt`

PART 1 – Exploration of the American Crow (AMCR) Distributional Data.

- Open up a session of the R Statistical Package and set your working directory by choosing "File → Change Dir…". We've stored workshop files in the root of the "c:\" drive.

- To get things going, we need to use a function in R that will allow us to read in the data for the American Crow (AMCR):

```
df <- read.table("amcr.txt", sep="\t", header=TRUE)
```

Let's double check the data frame to make sure everything's where it's supposed to be:

Q. What's a "dataframe"?

A. In R, a dataframe is analagous to an Excel spreadsheet -- in otherwords, it's a database table that R understands is an object with variables (=columns) and data (=rows).

Q. What's a "function"?

A. An object to which you pass data and arguments -- where "something gets done" -- and which returns information to you in the form of a result, or an object.

```
> dim(df)
[1] 2799   13

> str(df)
`data.frame':   2799 obs. of  13 variables:
 $ xcoord  : num  1621035 1620894 1620757 1620629 1616523 ...
 $ ycoord  : num  4973929 4974728 4975528 4976330 4976492 ...
 $ CROPVEG : int  1 1 0 0 0 0 0 0 0 0 ...
 $ CONIFER : int  0 0 0 0 0 0 0 0 0 0 ...
 $ DECID   : int  0 0 0 0 0 0 0 0 0 0 ...
 $ MIXEDF  : int  0 0 1 1 0 1 1 1 1 1 ...
 $ EVIMEAN : num  5226 5226 4800 4270 5189 ...
 $ EVISD   : num  188 188 502 718 424 ...
 $ ELEV    : int  328 322 322 323 336 335 328 326 322 320 ...
 $ DTR     : num  11.1 11.1 11.1 11.1 11.1 ...
 $ PRECIP  : num  730 730 730 730 730 ...
 $ TEMP    : num  7.24 7.24 7.24 7.24 7.24 7.24 7.24 7.24 7.24 7.24 ...
 $ presence: int  1 1 1 0 1 0 1 1 1 1 ...
```

- We're interested in modeling the effect of land cover ("CROPVEG", "CONIFER", "DECID", "MIXEDF", "EVIMEAN", "EVISD"), climactic ("DTR", "PRECIP", and "TEMP"), and topographic ("ELEV") variables on the probability of occurrence for the AMCR.

- At the pre-modelling stage, you should have some working hypotheses linking the environmental predictors and the response of your species of interest. It can also

be helpful to use graphical summaries to explore these relationships (=EDA, or Exploratory Data Analysis).

- We'll use a lowess function bundled with a very helpful package developed by Dr. Frank E. Harrell Jr. called "plsmo".

```
library(Hmisc)
```

- But first we'll create a single variable to hold the different landcover classes (referred to as a "factor" in the R language):

```
LANDCOV <- rep("OTHER",dim(df)[1])
LANDCOV <- ifelse(df$CROPVEG == 1,"CROPVEG",LANDCOV)
LANDCOV <- ifelse(df$CONIFER == 1,"CONIFER",LANDCOV)
LANDCOV <- ifelse(df$DECID == 1,"DECID",LANDCOV)
LANDCOV <- ifelse(df$MIXEDF == 1,"MIXEDF",LANDCOV)
LANDCOV <- as.factor(LANDCOV)
df <- cbind(df,LANDCOV)

str(df)
```

- Let's compare the observed usage with that based on a set of randomly selected landcover types:

```
par(mfrow=c(2,2))

# First, the observed usage:

t <- table(df$LANDCOV,df$presence)
label <- c("CONIFER","CROPVEG","DECID","MIXEDF","OTHER")
barplot(c(t[6]/(t[1]+t[6]),t[7]/(t[2]+t[7]),t[8]/(t[3]+t[8]),
  t[9]/(t[4]+t[9]),t[10]/(t[5]+t[10])),axes=F,main="Used
  Habitat",ylim=c(0,0.60))
axis(side=1, labels=label, at=c(0.50,1.85,3.00,4.25,5.55),lwd=1,
```

```
 font=1)
axis(side=2, lwd=1, font=1)



# Now the available:

df.rand <- read.table("rand.txt", sep="\t", header=TRUE)
LANDCOV <- rep("OTHER",dim(df)[1])
LANDCOV <- ifelse((df.rand$LCOVER == 12 | df.rand$LCOVER ==14),
  "CROPVEG",LANDCOV)
LANDCOV <- ifelse((df.rand$LCOVER == 1 | df.rand$LCOVER == 3),
  "CONIFER",LANDCOV)
LANDCOV <- ifelse(df.rand$LCOVER == 4,"DECID",LANDCOV)
LANDCOV <- ifelse(df.rand$LCOVER == 5,"MIXEDF",LANDCOV)
LANDCOV <- as.factor(LANDCOV)
df.rand <- cbind(df.rand,LANDCOV)

t <- table(df.rand$LANDCOV)
label <- c("CONIFER","CROPVEG","DECID","MIXEDF","OTHER")
total <- sum(t)
barplot(c(t[1]/total,t[2]/total,t[3]/total,
  t[4]/total,t[5]/total),axes=FALSE,main="Available
Habitat",ylim=c(0,0.60))
axis(side=2, lwd=1, font=1)



par(mfrow=c(3,3))

# EVIMEAN
plsmo(x=df$EVIMEAN,y=df$presence,datadensity=T,xlab="EVIMEAN",
  ylab="Prob. of Occurrence")

# EVISD
plsmo(x=df$EVISD,y=df$presence,datadensity=T,xlab="EVISD",
  ylab="Prob. of Occurrence")
```

```
# DTR
plsmo(x=df$DTR,y=df$presence,datadensity=T,xlab="DTR",
  ylab="Prob. of Occurrence")

# PRECIP
plsmo(x=df$PRECIP,y=df$presence,datadensity=T,xlab="PRECIP",
  ylab="Prob. of Occurrence")

# TEMP
plsmo(x=df$TEMP,y=df$presence,datadensity=T,xlab="TEMP",
  ylab="Prob. of Occurrence")

# ELEV
plsmo(x=df$ELEV,y=df$presence,datadensity=T,xlab="ELEV",
  ylab="Prob. of Occurrence")
```

PART 2 – Specification of the Base Logistic Regression for the American Crow (AMCR)

- We're interested in modeling the effect of land cover ("LANDCOV", "EVIMEAN", "EVISD"), climactic ("DTR", "PRECIP", and "TEMP"), and topographic ("ELEV") variables on the probability of occurrence for the AMCR.

```
presence ~ LANDCOV + EVIMEAN + EVISD + DTR + PRECIP +
           TEMP + ELEV
```

The *R* function for calculating logistic regressions ("glm()") is part of the Generalized Linear Models family with a logistic-link. Let's determine what arguments it takes:

```
?glm

Fitting Generalized Linear Models

Description:

     'glm' is used to fit generalized linear models, specified by
     giving a symbolic description of the linear predictor and a
```

```
     description of the error distribution.

Usage:

     glm(formula, family = gaussian, data, weights, subset,
         na.action, start = NULL, etastart, mustart,
         offset, control = glm.control(...), model = TRUE,
         method = "glm.fit", x = FALSE, y = TRUE, contrasts =
NULL, ...)
```

As with the Analysis of Variance example above, when we call the function "glm()" we want to assign the result to an object so that we can view it and manipulate it.

```
df.glm <- glm(formula=presence ~ LANDCOV + EVIMEAN +
  EVISD + DTR + PRECIP + TEMP + ELEV, family=binomial,
  data=df)
```

```
> df.glm

Call:  glm(formula = presence ~ LANDCOV + EVIMEAN + EVISD + DTR + PRECIP +
TEMP + ELEV, family = binomial, data = df)

Coefficients:
   (Intercept)  LANDCOVCROPVEG    LANDCOVDECID   LANDCOVMIXEDF    LANDCOVOTHER
    -8.164e+00       1.910e-02      -1.060e+01      -3.273e-01      -1.572e-01
      EVIMEAN           EVISD             DTR          PRECIP            TEMP
     2.377e-04       3.586e-04       3.007e-01       1.384e-04       4.851e-01
          ELEV
     1.946e-03

Degrees of Freedom: 2798 Total (i.e. Null);  2788 Residual
Null Deviance:      3833
Residual Deviance: 3472         AIC: 3494
```

```
> summary(df.glm)

Call:
glm(formula = presence ~ LANDCOV + EVIMEAN + EVISD + DTR + PRECIP +
    TEMP + ELEV, family = binomial, data = df)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6988  -0.9611  -0.6490   1.0952   2.0552

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)    -8.164e+00  9.015e-01  -9.056  < 2e-16 ***
LANDCOVCROPVEG  1.910e-02  2.304e-01   0.083 0.933915
LANDCOVDECID   -1.060e+01  1.970e+02  -0.054 0.957069
LANDCOVMIXEDF  -3.273e-01  1.976e-01  -1.656 0.097715 .
LANDCOVOTHER   -1.572e-01  2.117e-01  -0.743 0.457566
EVIMEAN         2.377e-04  6.157e-05   3.861 0.000113 ***
EVISD           3.586e-04  1.169e-04   3.066 0.002167 **
DTR             3.007e-01  7.376e-02   4.077 4.56e-05 ***
PRECIP          1.384e-04  4.059e-04   0.341 0.733181
```

```
TEMP            4.851e-01  3.841e-02  12.629  < 2e-16 ***
ELEV            1.946e-03  4.856e-04   4.008 6.13e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 3833.0  on 2798  degrees of freedom
Residual deviance: 3471.8  on 2788  degrees of freedom
AIC: 3493.8
```

```
> anova(df.glm)
```

```
Analysis of Deviance Table
Model: binomial, link: logit
Response: presence
Terms added sequentially (first to last)

          Df Deviance Resid. Df Resid. Dev
NULL                      2798     3833.0
LANDCOV    4    107.6      2794     3725.4
EVIMEAN    1     49.1      2793     3676.3
EVISD      1     19.4      2792     3656.9
DTR        1      0.1      2791     3656.8
PRECIP     1      6.3      2790     3650.5
TEMP       1    162.6      2789     3487.9
ELEV       1     16.1      2788     3471.8
```

Interesting results. Of the land cover variables, EVIMEAN and EVISD appear to be positive predictors for AMCR occurrence. Of the climactic variables, DTR and TEMP appear to also be positive predictors. ELEV appears as a positive predictor as well.

But how can we assess the goodness of fit for the linear model? One measure of fit is predictive accuracy, which we can assess using the area under the curve (AUC) of a receiver operating characteristic curve (ROC). We’ll load a library off the WWW that contains a nice function for calculating AUC :

```
library(Epi)
ROC(test=df.glm$fitted.values, stat=df$presence)$AUC
```

We find that the accuracy is: 0.703.

There’s another measure of fit that can be applied to logistic (binary) measurements, based on the Hosmer-Lemeshow test (see APPENDIX):

```
df.glm.hl <- hltest(resp=df$presence, prob=df.glm$fitted.values, g=5)
```

```
> df.glm.hl
```

```
$table
   group
      1   2   3   4   5
  0 424 455 271 227 204
  1 134 106 290 334 354

$Chat
[1] 65.56132

$df
[1] 3

$sig
[1] 3.808065e-14
```

Model Selection using AIC

In the previous analysis with forced all of the variables to be added to the logistic regression. Another choice would be to use an automated model selection algorithm such as "step()" which uses forward or backward stepwise addition of variables and a stopping rule based on Akaike Information Criterion (AIC) values.

What is the AIC? AIC = -2*ln(likelihood) + 2*p*

Let's repeat the previous model estimation procedure, but this time introduce the function "step()":

```
df.glm.step <- step(df.glm)

Start:  AIC=3493.81
presence ~ LANDCOV + EVIMEAN + EVISD + DTR + PRECIP + TEMP +
    ELEV

          Df Deviance    AIC
- PRECIP   1   3471.9 3491.9
<none>         3471.8 3493.8
- LANDCOV  4   3483.0 3497.0
- EVISD    1   3481.2 3501.2
```

```
- EVIMEAN  1   3486.9 3506.9
- ELEV     1   3487.9 3507.9
- DTR      1   3488.8 3508.8
- TEMP     1   3648.3 3668.3

Step:  AIC=3491.93
presence ~ LANDCOV + EVIMEAN + EVISD + DTR + TEMP + ELEV

          Df Deviance    AIC
<none>         3471.9 3491.9
- LANDCOV  4   3483.1 3495.1
- EVISD    1   3481.8 3499.8
- EVIMEAN  1   3487.6 3505.6
- ELEV     1   3487.9 3505.9
- DTR      1   3489.0 3507.0
- TEMP     1   3654.4 3672.4
…


summary(df.glm.step)

Call:
glm(formula = presence ~ MIXEDF + EVIMEAN + EVISD + DTR + TEMP +
    ELEV, family = binomial, data = df)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6714  -0.9649  -0.6479   1.0953   2.0816

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -8.123e+00  8.536e-01  -9.517  < 2e-16 ***
MIXEDF      -2.517e-01  8.434e-02  -2.985  0.00284 **
EVIMEAN      2.319e-04  5.908e-05   3.926 8.65e-05 ***
EVISD        3.288e-04  1.124e-04   2.925  0.00344 **
DTR          3.027e-01  7.302e-02   4.145 3.40e-05 ***
TEMP         4.898e-01  3.494e-02  14.016  < 2e-16 ***
ELEV         1.939e-03  4.829e-04   4.015 5.93e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 3833.0  on 2798  degrees of freedom
Residual deviance: 3474.2  on 2792  degrees of freedom
AIC: 3488.2

Number of Fisher Scoring iterations: 4
```

Displaying Predicted Probabilities of Occurrence

- Let’s load a library that will help us produce some maps:

```
library(geoR)
```

- We need to "attach" the predicted probabilities of occurrence (from the object `df.glm.step` above) to the "df" dataframe that holds our American Crow data:

```
prob <- df.glm.step$fitted.values
df <- cbind(df,prob)
```

- Now we need to produce an object that the "geoR" package can utilise:

```
df.geodata <- as.geodata(df,coords.col=1:2, data.col=15)
plot.geodata(df.geodata)
```

APPENDIX A

- Here's the code for the function to determine the Hosmer & Lemeshow goodness-of-fit (just copy and paste it into the *R* command window, and call it using "hltest()" and three arguments):

```
hltest <- function(resp, prob, g) {

  # Function for computing C-hat = approx. distributed as a chi-square
  # with g-2 degrees of freedom (Hosmer & Lemeshow p.148)
  # This is an omnibus goodness-of-fit measure for binary (as compared
  # to grouped, binomial data) models.

  # Arguments:
  # resp = "response" = a vector of dummy variables (coded 1/0)
  #        indicating the presence of the effect
  # prob = a vector of expected probabilities for each observation, based
  #        on the model
  # g    = scalar indicating the number of groups to use (e.g. 4, 10)


  # extract the binary values and expected probabilities for the
  # data set

  hl.mat <- cbind(resp,prob)
```

```
  # sort the matrix from lowest to highest expected probabilities

  hl.mat <- hl.mat[order(hl.mat[,2]),]
  key <- seq(1:length(prob))
  hl.mat <- cbind(hl.mat, key)


  # cut the predicted probabilities into “g” equal-sized groups
  group <- factor(cut(hl.mat[,3],g))

  # label the cutoff categories as "1" through "g"
  levels(group) <- c(1:g)

  #
  # Now attach the groupings to the "hl.mat" matrix
  #

  hl.mat <- cbind(hl.mat, group)


  #
  # Hosmer & Lemeshow's Goodness-of-fit (p.148)
  #
  # n = number of observations in the quantile
  # pi-bar = average of the expected probabilities for that quantile
  # o = number of observed positive (=1) outcomes
  # y = number of "positive" responses
  #

  # Count up the number of “resp” == 1, by group
  y <- table(hl.mat[,1], hl.mat[,4])[2,]

  # Note usage of “tapply”: tapply(data,categories,function)
  n <- tapply(hl.mat[,1],hl.mat[,4],length)

  # “pi.bar” = average probability of a “resp”=1, by group
  pi.bar <- tapply(hl.mat[,2], hl.mat[,4],mean)

  c <- (y-n*pi.bar)^2/((n*pi.bar)*(1-pi.bar))


  # Return the results as a list...
  hl.list <- NULL
  hl.list <- list(table=table(hl.mat[,1], group), Chat = sum(c), df=g-2,
    sig = (1-pchisq(sum(c),g-2)))


}
```

- Oftentimes we might want to take a random subset of our data points, for example, to perform model testing. Here’s some simple code -- applied to the “df” dataframe from the AMCR – that will enable you to randomly sample 100 records.

```
# Step 1. Create a vector of rowid numbers the same size as your
# data set to be sampled.
> size <- dim(df)[1]
> rowid <- seq(1,size,by=1)

# Step 2. Sample the rowid numbers without replacement; store the
# result in “row.sample”
```

```
> row.sample <- sample(rowid,size=100,replace=FALSE)

# Step 3. Grab the subset of your original dataframe and call it
# “df.sample”
> df.sample <- df[row.sample,]

# Step 4. Double check to confirm it worked…
> dim(df.sample)
[1] 100  15
```

- Looping in *R* and an “if else” control structure:

```
for (i in 1:100) {

  cat(paste(“Entering “, i, “th loop”, sep=””),”\n”)

  if (i > 95 & i < 100) {

    cat(“*** (Almost done)”,”\n”)

  }else if (i > 99) {

    cat(“!! Done !!”,”\n”)

  }

}
```

- Another model selection approach, but one involving all possible combinations of predictor variables – from a script called “allcombo.R”

```
#
# Name:        "allcombo.R"
# Date:        Aug.10, 2006
# Author:      David Lieske
# Purpose:     This script calls the leaps library to build an
#              "all combinations" matrix of fields for GLM model
#              building.
# Arguments:   Assumes a pre-existing dataframe called "df" exists
#              in the global environment.
# Value:       A matrix called "allcombo.out", sorted by AIC values,
#              with three fields: (1) aic; (2) auc; (3) predictor
#              variable combination
#

library(leaps)
library(Epi)


#
#
# First: define the x variables dataframe
#
#
```

```
allcombo.out <- allcombo.out[sort.list(allcombo.out$aic),]
```

Appendix B

Suggested References

- Bailey, T.C., and A.C. Gatrell. 1995. Interactive spatial data analysis. Prentice Hall, Toronto.
- Bivand, R. 2006. Implementing spatial data analysis software tools in R. Geographical Analysis 38:23-40.
- Chambers, J.M., and T.J. Hastie. 1993. Statistical Models in S. Chapman & Hall, London.
- Fotheringham, A.S., C. Brunsdon, and M. Charlton. 2000. Quantitative geography: perspectives on spatial data analysis. Sage Publications, Ltd. London.
- Fotheringham, A.S., C. Brunsdon, and M. Charlton. 2002. Geographically weighted regression: the analysis of spatially varying relationships. John Wiley & Sons, Ltd.
- Fox,
- Venables, W.N., and B.D. Ripley. Modern Applied Statistics with S. 4th ed. Springer, New York.

- Rey, S.J., and L. Anselin. 2006. Recent advances in software for spatial analysis in the social sciences. Geographical Analysis 38: 1-4.
- Selvin, S. 1998. Modern applied biostatistical methods using S-Plus. Oxford University Press, New York.