# STAT511HW6

*Ben Straub*

*November 5, 2015*

## 1) MISTLETOE

## EXPLORATORY DATA ANALYSIS

- 17 variables and 25431 Observations in the Mistletoe Data Set
- 5 categorical variables: csize, cdense, usize, udense, phys
- 196 complete cases from the USU survey data.
- 103 infected values that match with infected values from the USU and MNDNR survey.
- This would imply that the MNDNR only has a 52% successful identification rate of infected trees
- 93 values that are coded as false-positive or false-negatives.
- 8 values coded as false positive, i.e. MNDNR coded trees as finding an infection when it was not infected
- 85 values coded as false negatives, i.e. MNDNR coded trees as not finding an infection when in fact there was an infection.

## Analysis of the infected.MNDNR Response Variable of the Mistletoe Data Set

| | Estimate | Std. Error | z value | Pr(>\|z\|) |
|---|---|---|---|---|
| (Intercept) | -3.71059 | 0.23532 | -15.76853 | 0.00000 |
| csize | 0.83532 | 0.06999 | 11.93509 | 0.00000 |
| cdense | 0.00256 | 0.01601 | 0.15978 | 0.87306 |
| usize | 0.14392 | 0.03123 | 4.60873 | 0.00000 |
| udense | 0.06452 | 0.01432 | 4.50604 | 0.00001 |
| si | -0.05554 | 0.00351 | -15.83944 | 0.00000 |
| phys | 0.17848 | 0.04179 | 4.27059 | 0.00002 |
| age | -0.00278 | 0.00099 | -2.79878 | 0.00513 |
| ba | -0.00107 | 0.00077 | -1.39314 | 0.16358 |
| dbh | -0.62686 | 0.02818 | -22.24149 | 0.00000 |
| height | 0.06428 | 0.00311 | 20.66085 | 0.00000 |
| volume | 0.00129 | 0.00056 | 2.31600 | 0.02056 |
| mortal | 1.20296 | 0.03160 | 38.07426 | 0.00000 |
| dense | 0.16738 | 0.02519 | 6.64581 | 0.00000 |
| x | 0.00000 | 0.00000 | 3.92484 | 0.00009 |
| y | 0.00000 | 0.00000 | 0.24493 | 0.80651 |

*OBSERVATIONS:* I created a GLM with infected.mndnr as the response variable and I found that variables *csize, usize, udense, si, phys, age, dbh, volume, height, dense, x and mortal* to be of significance for this model. There are also several coefficients that are negative.

**Model Equation:**

## Analysis of the infected.USU Response Variable as a Subset of the Mistletoe Data Set

| | Estimate | Std. Error | z value | Pr(>\|z\|) |
|---|---|---|---|---|
| (Intercept) | -5.84509 | 2.16038 | -2.70559 | 0.00682 |
| csize | 0.92134 | 0.59021 | 1.56103 | 0.11852 |
| cdense | -0.00167 | 0.10982 | -0.01521 | 0.98786 |
| usize | 0.24312 | 0.26392 | 0.92121 | 0.35694 |
| udense | -0.27264 | 0.16264 | -1.67639 | 0.09366 |
| si | 0.03427 | 0.02029 | 1.68864 | 0.09129 |
| phys | 0.98367 | 0.37865 | 2.59782 | 0.00938 |
| age | 0.00435 | 0.00799 | 0.54463 | 0.58601 |
| ba | -0.00694 | 0.00615 | -1.12947 | 0.25870 |
| dbh | -0.22072 | 0.23189 | -0.95184 | 0.34118 |
| height | -0.03543 | 0.01171 | -3.02596 | 0.00248 |
| volume | 0.00164 | 0.00386 | 0.42555 | 0.67043 |
| mortal | 0.98041 | 0.31579 | 3.10464 | 0.00191 |
| dense | -0.39949 | 0.16592 | -2.40772 | 0.01605 |
| x | 0.00001 | 0.00000 | 1.64139 | 0.10072 |
| y | -0.00001 | 0.00001 | -0.72242 | 0.47004 |

*OBSERVATIONS:* We are assuming that the USU survey data is of more correctness than the MNDNR survey data. This would imply that variables found in this GLM to be of great importance to predicting whether mistletoe is present or not present. For this GLM I used infected.USU as the response variable and found that the variables *udense, si, phys, height, mortal and dense* to be of significance for this model.

**Model Equation:**

## Comparison of USU to MNDNR Data with an Error Term Response Variable

- I created a new Error Term Response Variable that captures the difference between the identifcation of infection between the USU and MNDNR Surveys. When the variable Error is coded as 1, then that represents that an error in the MNDRN survey data. A coding of 0 in our error term means they are coded correctly.

| Est | imate Std | . Error z | value Pr( | >\|z\|) |
|---|---|---|---|---|
| (Intercept) | -5.29864 | 2.03965 | -2.59782 | 0.00938 |
| csize | 0.36341 | 0.56803 | 0.63977 | 0.52232 |
| cdense | -0.12622 | 0.11182 | -1.12878 | 0.25899 |
| usize | -0.16916 | 0.25845 | -0.65451 | 0.51278 |
| udense | -0.18405 | 0.15847 | -1.16141 | 0.24547 |
| si | 0.04878 | 0.02021 | 2.41384 | 0.01579 |
| phys | 0.93343 | 0.35630 | 2.61976 | 0.00880 |
| age | 0.00307 | 0.00743 | 0.41352 | 0.67922 |
| ba | 0.00033 | 0.00595 | 0.05475 | 0.95634 |

| Est | imate Std | . Error z | value Pr( | >\|z\|) |
|---|---|---|---|---|
| dbh | 0.07857 | 0.22120 | 0.35522 | 0.72242 |
| height | -0.03210 | 0.01138 | -2.82158 | 0.00478 |
| volume | -0.00010 | 0.00378 | -0.02542 | 0.97972 |
| mortal | -0.16078 | 0.25727 | -0.62496 | 0.53200 |
| dense | -0.32972 | 0.15782 | -2.08927 | 0.03668 |
| x | 0.00000 | 0.00000 | 0.59860 | 0.54944 |
| y | 0.00000 | 0.00001 | -0.20656 | 0.83636 |

*OBSERVATIONS:* I found the following variables to be of significance for determining if a tree was mislabled as infected or not infected from the MNDNR survery: Dense, Si, Phys, csize and height.

### Model Equation:

## Analysis of False Positives

| | Estimate | Std. Error | z value | Pr(>\|z\|) |
|---|---|---|---|---|
| (Intercept) | -2.3160227 | 0.7062678 | -3.279241 | 0.0010409 |
| cdense | -0.5650148 | 0.4007258 | -1.409979 | 0.1585460 |
| volume | 0.0138385 | 0.0077084 | 1.795242 | 0.0726151 |
| x | -0.0000080 | 0.0000054 | -1.493107 | 0.1354093 |

### Model Equation:

*OBSERVATIONS:* There are only 8 false positives in the data set. I used the USU data set coupled with the AIC function to whittle down the predictor variables and found that the variables that are significant are just *volume*. It appears that Volume is significant in labeling a tree that is not actually infected as infected.

## Analysis of False Negatives

| | Estimate | Std. Error | z value | Pr(>\|z\|) |
|---|---|---|---|---|
| (Intercept) | -7.1225334 | 1.9806395 | -3.596078 | 0.0003231 |
| csize | 0.5798426 | 0.1970247 | 2.942994 | 0.0032506 |
| udense | -0.2352297 | 0.1514561 | -1.553121 | 0.1203942 |
| si | 0.0501047 | 0.0184824 | 2.710942 | 0.0067092 |
| phys | 1.1752576 | 0.3473228 | 3.383761 | 0.0007150 |
| height | -0.0412864 | 0.0108787 | -3.795152 | 0.0001476 |
| dense | -0.2872787 | 0.1553759 | -1.848927 | 0.0644684 |
| x | 0.0000051 | 0.0000025 | 2.078975 | 0.0376196 |

### Model Equation:

*OBSERVATIONS:* The False Negative GLM gave the following variables as significant: csize, si, phys, height, dense and x. The False Negatives are where the MNDNR did not observe the actual mistletoe when in fact it was there. Folks out in the field should pay attention to the predictor variables found to be significant in this

model to not mis-identify infected tree as not infected.

## 2) Categorical versus Continuous Predictor Variables for Mistletoe

```
## The following objects are masked from mis (pos = 3):
##
##     age, ba, cdense, csize, dbh, dense, height, infected.mndnr,
##     infected.USU, mortal, phys, si, udense, usize, volume, x, y
```

```
# Subsetting data for MNDNR study
mis_MNDNR <- mis[c(-17)]
## set aside a test set (20% of data)
n=length(mis_MNDNR$infected.mndnr)
n.test=round(n*.2)
n.train=n-n.test
n
n.test
n.train
test.idx=sample(1:n,size=n.test)
train.idx=(1:n)[-test.idx]
train=mis_MNDNR[train.idx,]
test=mis_MNDNR[test.idx,]
```

| Tables | MSPE | CVMSPE | Difference |
|---|---|---|---|
| All Cont.s | 0.0805383 | 0.0830262 | 0.0024878 |
| 1 Factor | 0.0797674 | 0.0814086 | 0.0016412 |
| 2 Factor | 0.0792361 | 0.0810676 | 0.0018314 |
| 3 Factors | 0.0792291 | 0.0811346 | 0.0019055 |
| 3 Factors A | 0.0787265 | 0.080678 | 0.0019515 |
| 4 Factors | 0.0782471 | 0.0861343 | 0.0078872 |
| 5 Factors | 0.0782471 | 0.0800619 | 0.0018148 |

### Model Form:

*CONCLUSION:* I played around with the factor variables adding in each factor to see what happens to the CVMSPE and MSPE. The above form is how I added factors into the model, starting from left to right (not scientific). I noticed that making usize a factor did something to rate of difference between CVMSPE and MSPE. I decided to leave it out for two vairations. The alternative 3 Factor model has a better CVMSPE, then the one that included usize. I kept on with my method and noticed that the 4 factor model has a much higher CVMSPE, then previous variations. I reintroduced usize back into the 5 factor model and it increased MSPE and CVMSPE. In conclusion, it appears that a model with three factors in it, which are phys, udense and cdense, prodcues a better model than models without any variables as factors or ones with less than 3 or more than 3.

## 3) *COVARIANCE REGRESSION*

```
## 'data.frame':    300 obs. of  3 variables:
##  $ d  : num  -0.54 1.987 7.593 0.181 2.702 ...
```

```
##  $ t  : num  2.93902 1.27833 3.52696 0.00754 1.67332 ...
##  $ sex: Factor w/ 2 levels "F","M": 1 1 1 1 1 1 1 2 1 1 ...
```

```
##        d                    t           sex
##  Min.   :-11.75917   Min.   :0.00754   F:202
##  1st Qu.: -1.00378   1st Qu.:0.33796   M: 98
##  Median : -0.13894   Median :0.75706
##  Mean   : -0.01699   Mean   :1.06754
##  3rd Qu.:  0.86427   3rd Qu.:1.51109
##  Max.   : 11.31424   Max.   :6.77898
```

- 300 Elk Observation in the Data Set
- 202 Female Elk
- 98 Male Elk

```
##########################################
## Numerical MLEs for Regression
##########################################
#trans <- d/sqrt(t)
#fit <- lm(trans~t+factor(sex))

## read in data
## view the first few rows of the data
head(dispersal)
```

```
##            d         t sex
## 1 -0.5402292 2.93902309   F
## 2  1.9868964 1.27833351   F
## 3  7.5929682 3.52696103   F
## 4  0.1808510 0.00753970   F
## 5  2.7022184 1.67332190   F
## 6  0.1323802 0.04295983   F
```

```
attach(dispersal)

dis_M <- subset(dispersal, sex=="M")
dis_F <- subset(dispersal, sex=="F")

y_M = dis_M$d
y_F = dis_F$d
X_M=as.matrix(cbind(1,dis_M[,2:3]))
X_F=as.matrix(cbind(1,dis_F[,2:3]))

# MLE sigma for Men
#normal.lik1<-function(theta,y,X){
#mu<-theta[1]
#sigma2<-theta[2]
#n<-nrow(y)
#logl<- -.5*n*log(2*pi) -.5*n*log(sigma2) -
#(1/(2*sigma2))*sum((y-mu)**2)
#return(-logl)
#}
```

```
#beta.start=c(0,0)
#s2.start=1
#out <- optim(c(beta.start,s2.start), normal.lik1, y=y_M, X=X_M, hessian=T)

#out$par
#OI<-solve(out$hessian)
#se_M<-sqrt(diag(OI))

# MLE sigma for Women
#normal.lik1<-function(theta,y,X){
#mu<-theta[1]
#sigma2<-theta[2]
#n<-nrow(y)
#logl<- -.5*n*log(2*pi) -.5*n*log(sigma2) -
#(1/(2*sigma2))*sum((y-mu)**2)
#return(-logl)
#}

#out <- optim(c(0,1), normal.lik1, y=y_F, data=dis_F, hessian=T)

#out$par
#OI<-solve(out$hessian)
#sigma_F<-diag(OI)
#sigma_F

############## START OF NEXT TRY ############
nll.reg <- function(beta.s2,y,X){
    ## get parameters
    p=length(beta.s2)-1
    n=length(y)
    beta=beta.s2[1:p]
    s2=beta.s2[p+1]
    ## calculate loglikelihood
    loglik=sum(dnorm(y,X%*%beta,sqrt(s2),log=T))
    ## return negative loglikelihood
    -loglik
}

##
## read in some data and estimate parameters using "lm"
##

y=dis_M$d
X=cbind(1,dis_M$t)

##
## find MLE using "optim" numerical optimization
##

beta.start=c(0,0)
s2.start=1
out=optim(c(beta.start,s2.start),nll.reg,y=y,X=X,control=list(trace=10),hessian=T)
```

```
##   Nelder-Mead direct search function minimizer
## function value for initial parameters = 188.382568
##   Scaled convergence tolerance is 2.80712e-06
## Stepsize computed as 0.100000
## BUILD              4 193.120410 184.113986
## EXTENSION          6 192.100870 181.144271
## EXTENSION          8 188.382568 175.829530
## EXTENSION         10 184.113986 173.460951
## LO-REDUCTION      12 181.144271 173.460951
## LO-REDUCTION      14 175.829530 173.460951
## LO-REDUCTION      16 174.977186 173.460951
## LO-REDUCTION      18 174.539832 173.460951
## EXTENSION         20 174.079631 172.352235
## EXTENSION         22 173.613581 171.319334
## LO-REDUCTION      24 173.460951 171.319334
## EXTENSION         26 172.352235 170.692807
## REFLECTION        28 172.130374 170.612149
## REFLECTION        30 171.319334 170.099777
## HI-REDUCTION      32 170.692807 170.099777
## LO-REDUCTION      34 170.612149 170.099777
## REFLECTION        36 170.539710 170.050319
## LO-REDUCTION      38 170.329888 170.024879
## LO-REDUCTION      40 170.099777 170.010520
## REFLECTION        42 170.050319 170.004112
## LO-REDUCTION      44 170.024879 169.988564
## LO-REDUCTION      46 170.010520 169.983523
## LO-REDUCTION      48 170.004112 169.981591
## HI-REDUCTION      50 169.988564 169.979037
## HI-REDUCTION      52 169.983523 169.972154
## HI-REDUCTION      54 169.981591 169.972154
## LO-REDUCTION      56 169.979037 169.972154
## HI-REDUCTION      58 169.975424 169.972154
## LO-REDUCTION      60 169.973680 169.971897
## HI-REDUCTION      62 169.973601 169.971897
## REFLECTION        64 169.972154 169.971573
## LO-REDUCTION      66 169.972010 169.971509
## HI-REDUCTION      68 169.971897 169.971469
## REFLECTION        70 169.971573 169.971285
## HI-REDUCTION      72 169.971509 169.971239
## REFLECTION        74 169.971469 169.971074
## HI-REDUCTION      76 169.971285 169.971074
## HI-REDUCTION      78 169.971239 169.971074
## LO-REDUCTION      80 169.971195 169.971074
## LO-REDUCTION      82 169.971166 169.971074
## EXTENSION         84 169.971112 169.970987
## REFLECTION        86 169.971087 169.970985
## LO-REDUCTION      88 169.971074 169.970985
## REFLECTION        90 169.971015 169.970956
## EXTENSION         92 169.970987 169.970931
## LO-REDUCTION      94 169.970985 169.970931
## LO-REDUCTION      96 169.970956 169.970927
## HI-REDUCTION      98 169.970943 169.970927
## REFLECTION       100 169.970933 169.970920
## LO-REDUCTION     102 169.970931 169.970920
```

```
## HI-REDUCTION     104 169.970927 169.970920
## LO-REDUCTION     106 169.970923 169.970920
## LO-REDUCTION     108 169.970922 169.970919
## Exiting from Nelder Mead minimizer
##     110 function evaluations used
```

```
## get parameter estimates (order is the same as in beta.s2)
out$par
```

```
## [1] -0.1685161 -0.1749353  1.8796088
```

```
## compare to estimates from lm:

fit=lm(y~0+X)
summary(fit)
```

```
##
## Call:
## lm(formula = y ~ 0 + X)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7528 -0.5965  0.0593  0.8639  3.2616
##
## Coefficients:
##    Estimate Std. Error t value Pr(>|t|)
## X1  -0.1685     0.2156  -0.782    0.436
## X2  -0.1750     0.1785  -0.980    0.329
##
## Residual standard error: 1.385 on 96 degrees of freedom
## Multiple R-squared:  0.06345,    Adjusted R-squared:  0.04394
## F-statistic: 3.252 on 2 and 96 DF,  p-value: 0.04299
```

```
coef(fit)
```

```
##         X1         X2
## -0.1685475 -0.1749542
```

```
## get standard errors from optim
H=out$hessian
S=solve(H)
se=sqrt(diag(S))
se
```

```
## [1] 0.2134495 0.1766413 0.2685547
```

```
y=dis_F$d
X=cbind(1,dis_F$t)

##
## find MLE using "optim" numerical optimization
```

```
##

beta.start=c(0,0)
s2.start=1
out=optim(c(beta.start,s2.start),nll.reg,y=y,X=X,control=list(trace=10),hessian=T)
```

```
##   Nelder-Mead direct search function minimizer
## function value for initial parameters = 923.314823
##   Scaled convergence tolerance is 1.37585e-05
## Stepsize computed as 0.100000
## BUILD              4 923.805710 865.878493
## LO-REDUCTION       6 923.314823 865.878493
## EXTENSION          8 921.606126 832.568657
## EXTENSION         10 885.206183 776.639114
## EXTENSION         12 865.878493 728.353044
## EXTENSION         14 832.568657 673.727455
## EXTENSION         16 776.639114 627.268283
## EXTENSION         18 728.353044 574.808224
## LO-REDUCTION      20 673.727455 574.808224
## REFLECTION        22 627.268283 572.649848
## EXTENSION         24 588.974507 560.738964
## EXTENSION         26 574.808224 535.782038
## LO-REDUCTION      28 572.649848 535.782038
## LO-REDUCTION      30 560.738964 535.782038
## LO-REDUCTION      32 559.848610 535.782038
## EXTENSION         34 551.064873 515.841002
## EXTENSION         36 539.463261 498.925345
## EXTENSION         38 535.782038 493.499364
## REFLECTION        40 515.841002 488.344540
## LO-REDUCTION      42 498.925345 488.344540
## HI-REDUCTION      44 493.499364 488.344540
## REFLECTION        46 492.545760 487.433594
## LO-REDUCTION      48 491.078205 487.433594
## LO-REDUCTION      50 488.344540 487.433594
## REFLECTION        52 488.252047 487.335480
## HI-REDUCTION      54 487.680006 487.335480
## HI-REDUCTION      56 487.433594 487.291774
## HI-REDUCTION      58 487.429363 487.233300
## LO-REDUCTION      60 487.335480 487.180350
## HI-REDUCTION      62 487.291774 487.180350
## LO-REDUCTION      64 487.233300 487.180350
## HI-REDUCTION      66 487.188873 487.180350
## LO-REDUCTION      68 487.184504 487.172580
## HI-REDUCTION      70 487.184223 487.169747
## LO-REDUCTION      72 487.180350 487.169747
## LO-REDUCTION      74 487.172580 487.169747
## LO-REDUCTION      76 487.171139 487.168830
## LO-REDUCTION      78 487.170978 487.168043
## HI-REDUCTION      80 487.169747 487.168043
## HI-REDUCTION      82 487.168830 487.168043
## LO-REDUCTION      84 487.168144 487.167874
## LO-REDUCTION      86 487.168095 487.167714
## HI-REDUCTION      88 487.168043 487.167689
```

```
## LO-REDUCTION      90 487.167874 487.167627
## HI-REDUCTION      92 487.167714 487.167627
## LO-REDUCTION      94 487.167689 487.167613
## LO-REDUCTION      96 487.167650 487.167613
## Exiting from Nelder Mead minimizer
##     98 function evaluations used
```

```
## get parameter estimates (order is the same as in beta.s2)
out$par
```

```
## [1]  0.1889501 -0.0475832  7.2848219
```

```
## compare to estimates from lm:

fit=lm(y~0+X)
summary(fit)
```

```
##
## Call:
## lm(formula = y ~ 0 + X)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.6277  -1.2049  -0.2647   1.0283  11.3118
##
## Coefficients:
##    Estimate Std. Error t value Pr(>|t|)
## X1  0.18833    0.27778   0.678    0.499
## X2 -0.04717    0.17716  -0.266    0.790
##
## Residual standard error: 2.712 on 200 degrees of freedom
## Multiple R-squared:  0.002834,   Adjusted R-squared:  -0.007138
## F-statistic: 0.2842 on 2 and 200 DF,  p-value: 0.753
```

```
coef(fit)
```

```
##          X1          X2
##  0.18833091 -0.04716878
```

```
## get standard errors from optim
H=out$hessian
S=solve(H)
se=sqrt(diag(S))
se
```

```
## [1] 0.2764286 0.1763028 0.7250324
```

*CONCLUSION:* I got really bogged down on this part of the assignment. I tried to get my MLE code to work, but I failed at it a bunch. Anyways, I put one iteration in and cut and pasted the code that was provided in the example. I looked at an article at http://polisci2.ucsd.edu/dhughes/teaching/MLE_in_R.pdf to try and figure out my issues, but it was to no avail.