Assessing Trends in Wild Turkey Populations Using Hunter Surveys

Jacob Harrell

4/9/2021

INTRODUCTION

Since being re-established within the state, the West Virginia Division of Natural REsources (WVDNR) has conducted an annual hunter survey to coincide with the Spring Gobbler Harvest. The WVDNR uses this data primarily to gage harvest totalswith respect to hunter satisfaction. I was curious about what trends may exist within the data. The overall objective of this report is to assess the hunter survey data for any trends in wild turkey populations observed across the state.

STUDY AREA

The full scope of the study encompassed each of the 55 counties of West Virginia. These 55 counties spanned across six different eco-regions. All participants of the survey hunted on private property throughout the state. Most hunters hunted in one county, but there were a number of hunters that hunted in more than one county.

STATISTICAL MODEL DESCRIPTION

Considering the nature of the data, it lent itself quite well to be fitted within a poisson distribution model. The total number of turkeys called in during a hunt constituted the response variable. Expected counts of turkeys per hunting visit were modeled as a function of two distinct predictor variable : temperature and ecoregion. Other predictor variables had been tested, such as hours spent in woods, private versus public lands, and weather, but these two variables seemed the most appropriate given their relatively low AIC value when modeled. An offset was applied to the model using the hours spent in the woods hunting.

```r
#setwd(choose.dir('Gobbler'))
SG2019 <- read.csv('2019SGRaw.csv')

#Setting up a model for total turkeys using an offset
thouroff <- log(SG2019$Hours) #setting hours as an offset
SG2019$Temp <- factor(SG2019$Temp, levels = c('1', '2', '3', '4')) #setting the categorical variable as
SG2019$Region <- factor(SG2019$Region, levels = c('1', '2', '3', '4', '5', '6')) #setting the categoric
tfitoff <- glm(TotCalled ~ Temp + Region,
               family = poisson,
               data = SG2019,
               offset = thouroff) #poisson model where total number of turkeys called in is a function o
summary(tfitoff)
```

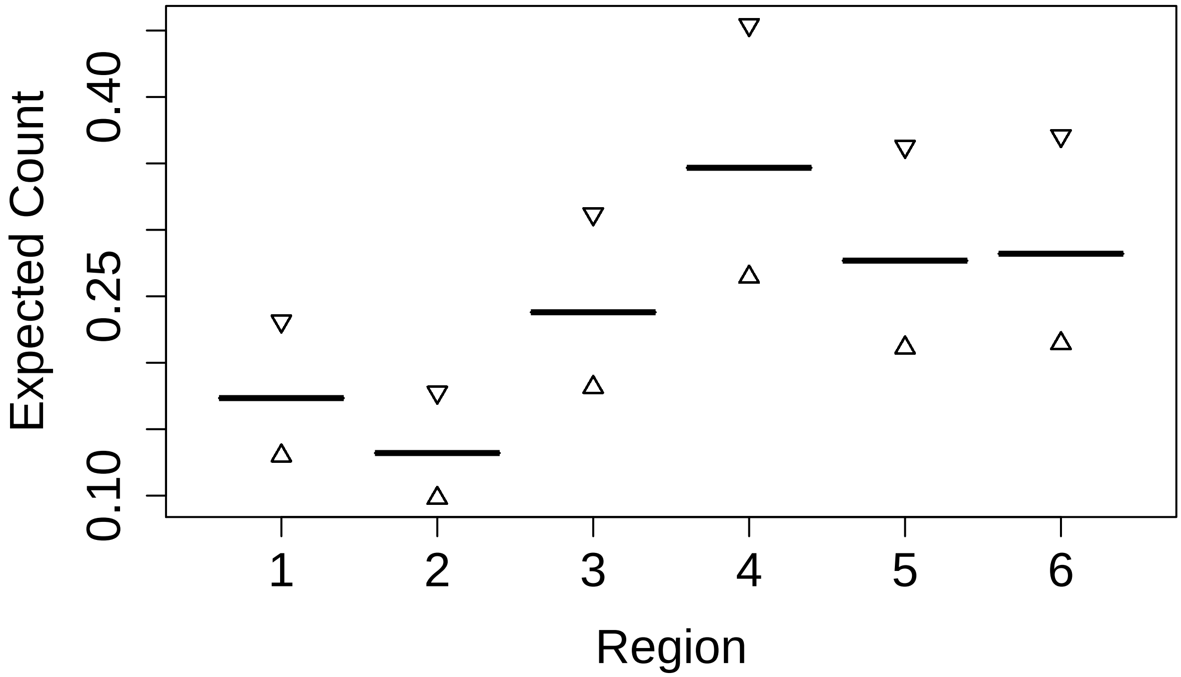
```
##
## Call:
## glm(formula = TotCalled ~ Temp + Region, family = poisson, data = SG2019,
##     offset = thouroff)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0760  -1.2789  -0.9609   0.3574   8.2118
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.75208    0.14689 -11.928  < 2e-16 ***
## Temp2       -0.30317    0.13670  -2.218  0.02657 *
## Temp3       -0.25246    0.13747  -1.836  0.06628 .
## Temp4       -0.00531    0.17206  -0.031  0.97538
## Region2     -0.27238    0.08744  -3.115  0.00184 **
## Region3      0.31652    0.06541   4.839 1.30e-06 ***
## Region4      0.69282    0.06499  10.661  < 2e-16 ***
## Region5      0.46782    0.06510   7.187 6.64e-13 ***
## Region6      0.48644    0.06834   7.118 1.10e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 11030  on 5082  degrees of freedom
## Residual deviance: 10752  on 5074  degrees of freedom
##   (224 observations deleted due to missingness)
## AIC: 15046
##
## Number of Fisher Scoring iterations: 6
```

```
#setting up a data frame for predicted values
TOnd <- data.frame(
  Temp = factor(c('1'), levels = c('1', '2', '3', '4')),
  Region = factor(SG2019$Region),
  thouroff = rep(log(1)))
toprd <- predict.glm(object = tfitoff, newdata = TOnd, type = 'link', se.fit = T)
summary(toprd)
```

```
##                Length Class  Mode
## fit            5307   -none- numeric
## se.fit         5307   -none- numeric
## residual.scale    1   -none- numeric
```

```
#setting up confidence intervals
tolow <- exp(toprd$fit - qnorm(0.975) * toprd$se.fit)
tohigh <- exp(toprd$fit + qnorm(0.975) * toprd$se.fit)

#plotting the expected count per hunting trip between ecoregions when morning temperatures are below 30
plot(y = exp(toprd$fit), x = TOnd$Region, xlab = 'Region',
     ylab = 'Expected Count', cex.axis = 1.5, cex.lab = 1.5,
     ylim = c(min(tolow), max(tohigh)), type = 'l'); points(x=TOnd$Region, y = tohigh, lty = 2, pch = 6
```


Expected Count
0.10
0.25
0.40
1
2
3
4
5
6
Region