Week 9 Logistic Regression Clinic

Instructions: Your instructor will supply a CSV file (with the file name Week9hiringData.csv) which you can read into R. It is recommended that you assign the imported data set to a data frame called “hiredata.” This data set contains a list of n=295 survey responses from raters who participated in an employee hiring process. The ultimate dependent variable, “hired,” is a binary variable with 0 for a candidate who was subsequently not hired and 1 for a candidate who was. The “recommend” variable is each survey participant’s recommendation of whether to hire, with 1 = “Definitely Hire,” 2 = “Possibly Hire,” and 3 = “Do Not Hire.” In addition, there are six belief questions all on 1 to 4 scales (with 1 as most favorable and 4 as least favorable) with assessments of the candidates on issues like leadership and collaboration. The ultimate research question is to understand the connection between survey participant responses and the ultimate hiring decisions. Can survey respondents accurately assess who will be hired?

This clinic unfolds in three phases: 1) creating an initial logistic regression model using the recommend variable as a predictor; 2) running the Bayesian model of logistic regression; and 3) finding additional predictor(s) that may improve the model.

**Phase 1 Instructions**: In this first phase of the clinic, read the data into R, inspect the data, and develop a basic logistic regression model with one predictor – namely the hire recommendation. The dependent variable will be the actual hiring decision.

Use the “Import Dataset” dialog to import the data into R. It is recommend to used read.csv() (the first option on the drop down menu: “From Text (base) . . .” Assign the result to a data frame called “hiredata.” Run summary(hiredata) and past in the results below. Take special note of the min and max of the survey variables: recommend, vision, issues, trends, consult, lead, and collab.  
 row hired rater recommend vision issues trends consult

Min. : 19.0 Min. :0.0000 Length:295 Min. :1.000 Min. :1.000 Min. :1.000 Min. :1.000 Min. :1.000

1st Qu.: 97.5 1st Qu.:0.0000 Class :character 1st Qu.:1.000 1st Qu.:1.000 1st Qu.:1.000 1st Qu.:1.000 1st Qu.:1.000

Median :181.0 Median :0.0000 Mode :character Median :2.000 Median :2.000 Median :2.000 Median :2.000 Median :2.000

Mean :181.3 Mean :0.2508 Mean :1.973 Mean :1.708 Mean :1.742 Mean :1.753 Mean :1.759

3rd Qu.:264.5 3rd Qu.:0.5000 3rd Qu.:3.000 3rd Qu.:2.000 3rd Qu.:2.000 3rd Qu.:2.000 3rd Qu.:2.000

Max. :353.0 Max. :1.0000 Max. :3.000 Max. :4.000 Max. :4.000 Max. :4.000 Max. :4.000

lead collab

Min. :1.000 Min. :1.00

1st Qu.:1.000 1st Qu.:1.00

Median :2.000 Median :2.00

Mean :1.942 Mean :1.78

3rd Qu.:2.000 3rd Qu.:2.00

Max. :4.000 Max. :4.00



Create a correlation matrix of the data and paste it below. The cor() procedure will not work on text or factor data, so you need to select the subset of numerical values. This statement should work: cor(hiredata[,4:10]). Add any comments that you may have about the values you see in the correlation matrix.  
 recommend vision issues trends consult lead collab

recommend 1.0000000 0.5221602 0.5127583 0.3393823 0.4936638 0.4880820 0.6038339

vision 0.5221602 1.0000000 0.7596677 0.5800789 0.5175185 0.5294269 0.6517940

issues 0.5127583 0.7596677 1.0000000 0.6357136 0.5216559 0.5649790 0.6042700

trends 0.3393823 0.5800789 0.6357136 1.0000000 0.3544414 0.5212734 0.4457242

consult 0.4936638 0.5175185 0.5216559 0.3544414 1.0000000 0.5407321 0.6477603

lead 0.4880820 0.5294269 0.5649790 0.5212734 0.5407321 1.0000000 0.5635504

collab 0.6038339 0.6517940 0.6042700 0.4457242 0.6477603 0.5635504 1.0000000

Means, standard deviations, and correlations with confidence intervals

Variable M SD 1 2 3 4 5 6

1. recommend 1.97 0.74

2. vision 1.71 0.73 .52\*\*

[.43, .60]

3. issues 1.74 0.81 .51\*\* .76\*\*

[.42, .59] [.71, .80]

4. trends 1.75 0.80 .34\*\* .58\*\* .64\*\*

[.23, .44] [.50, .65] [.56, .70]

5. consult 1.76 0.80 .49\*\* .52\*\* .52\*\* .35\*\*

[.40, .58] [.43, .60] [.43, .60] [.25, .45]

6. lead 1.94 0.84 .49\*\* .53\*\* .56\*\* .52\*\* .54\*\*

[.40, .57] [.44, .61] [.48, .64] [.43, .60] [.45, .62]

7. collab 1.78 0.78 .60\*\* .65\*\* .60\*\* .45\*\* .65\*\* .56\*\*

[.53, .67] [.58, .71] [.53, .67] [.35, .53] [.58, .71] [.48, .64]

Note. M and SD are used to represent mean and standard deviation, respectively.

Values in square brackets indicate the 95% confidence interval.

The confidence interval is a plausible range of population correlations

that could have caused the sample correlation (Cumming, 2014).

\* indicates p < .05. \*\* indicates p < .01.

1. Run histograms on each of the numeric variables to ascertain the shape of their distributions. Note any anomalies below:  
   all tend to have outliers on the high end

Run and interpret a basic logistic regression model using glm(). The formula should specify the dependent variable (hired) and the predictor (recommend). Here’s a command that should work:  
  
glmOut <- glm(formula = hired ~ recommend, family = binomial(link="logit"), data = hiredata)  
  
Run summary() on the glmOut object and paste in the results below. Write a brief statement summarizing the results. Is the predictor statistically significant?

Call:

glm(formula = hired ~ recommend, family = binomial(link = "logit"),

data = hiredata)

Deviance Residuals:

Min 1Q Median 3Q Max

-1.1793 -0.6710 -0.3508 0.4124 2.3744

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) 1.3854 0.3992 3.470 0.00052 \*\*\*

recommend -1.3809 0.2266 -6.094 1.1e-09 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 332.33 on 294 degrees of freedom

Residual deviance: 286.34 on 293 degrees of freedom

AIC: 290.34

YESSSSS

Number of Fisher Scoring iterations: 4

Use the exp() and confint() commands as described on page 225 of the textbook to convert the log odds for the coefficient on the predictor into regular odds. Paste the results below along with a one sentence interpretation:  
> exp(coef(glmOut)) # Convert log odds to odds

(Intercept) recommend

3.9962895 0.2513514

> exp(confint(glmOut)) # Look at confidence intervals

Waiting for profiling to be done...

2.5 % 97.5 %

(Intercept) 1.8465086 8.8680474

recommend 0.1582932 0.3856539

recommend is < 1 which means as recommend increases, the y variable hired decreases. Recommendations stink.

1. You will note that the plain odds version of the coefficient on the predictor is fractional. This can make interpretation of the results messier, particularly for non-statisticians to whom you may wish to communicate your results. You can invert the sense of the recommend variable with this simple statement:  
     
   hiredata$recInv <- 4 - hiredata$recommend  
     
   Try the math in your head by plugging in the minimum value of recommend (1) and the maximum value of recommend (3). Note that this command adds a new variable to your existing data set. To cross check your results, you could correlate the new variable with the old one. Report and briefly explain the result below:  
     
   they are exact opposites (perfect negative correlation as would be desired/expected)

Rerun the code for Questions 4 and 5 using the new version of the predictor variable. Provide a new interpretation of the plain odds ratio, based on the output you get from applying exp() and confint() to the coefficient.  
  
Call:

glm(formula = hired ~ recInv, family = binomial(link = "logit"),

data = hiredata)

Deviance Residuals:

Min 1Q Median 3Q Max

-1.1793 -0.6710 -0.3508 0.4124 2.3744

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -4.1382 0.5544 -7.464 8.37e-14 \*\*\*

recInv 1.3809 0.2266 6.094 1.10e-09 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 332.33 on 294 degrees of freedom

Residual deviance: 286.34 on 293 degrees of freedom

AIC: 290.34

Number of Fisher Scoring iterations: 4

Just kidding recommendations ROCK… YAYYY reverse coding 😊  
  
  
> exp(coef(glmOut)) # Convert log odds to odds

(Intercept) recInv

0.0159508 3.9784934

> exp(confint(glmOut)) # Look at confidence intervals

Waiting for profiling to be done...

2.5 % 97.5 %

(Intercept) 0.00506101 0.04473355

recInv 2.59299854 6.31738889

1. Produce, paste in below, and interpret a Nagelkerke pseudo-R-squared using this code:  
     
   install.packages("BaylorEdPsych")

library(BaylorEdPsych)

PseudoR2(glmOut)   
  
Then write a brief paragraph that reports the full set of results you have obtained so far.

PseudoR2(glmOut) # Get Pseudo R-squared values

McFadden Adj.McFadden Cox.Snell Nagelkerke McKelvey.Zavoina Effron Count

0.13838469 0.12033013 0.14435037 0.21358554 0.24120380 0.14355150 0.73898305

Adj.Count AIC Corrected.AIC

-0.04054054 290.33709354 290.37818943

**Phase 2 Instructions**: Conduct a Bayesian logistic regression analysis, using the facilities in the MCMCpack package.

Install the MCMCpack package and library it using the appropriate menus or command. Paste in the output produced by the library() command below:  
  
> library(MCMCpack) # Load the package

Run the MCMClogit() function using the same model as for Question 7 above. The following code should work:  
  
bayesLogitOut <- MCMClogit(formula = hired ~ recInv, data = hiredata)  
  
Run summary() on the output object and paste the results below. Comment on how the Bayesian (MCMC) mean of the coefficient parameter on the predictor compares with the corresponding result from the conventional glm() analysis.   
summary(bayesLogitOut) # Show model summary

Iterations = 1001:11000

Thinning interval = 1

Number of chains = 1

Sample size per chain = 10000

1. Empirical mean and standard deviation for each variable,

plus standard error of the mean:

Mean SD Naive SE Time-series SE

(Intercept) -4.198 0.5619 0.005619 0.017321

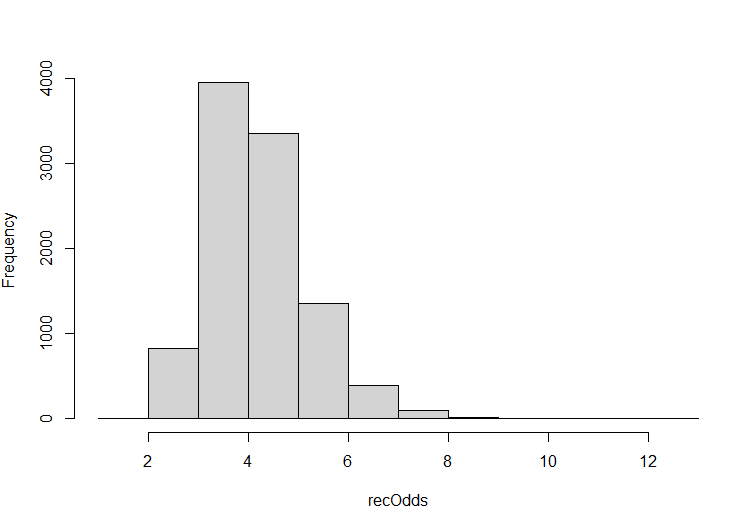
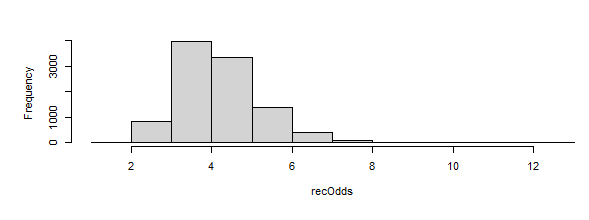
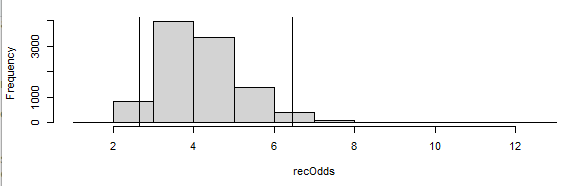
recInv 1.405 0.2274 0.002274 0.006926

2. Quantiles for each variable:

2.5% 25% 50% 75% 97.5%

(Intercept) -5.3816 -4.566 -4.167 -3.822 -3.163

recInv 0.9766 1.250 1.398 1.555 1.866

2. Create a plot of the MCMC output by running plot(bayesLogitOut). Take note of any anomalies in the trace plot. Does the distribution of parameter estimates on the predictor appear to overlap with zero?  
   **No Go !!**  
   
3. We can improve our view of the parameter estimates of the coefficient by converting the distribution from log odds to plain odds. The following code develops a histogram of the posterior distribution of plain odds:  
     
   recLogOdds <- as.matrix(bayesLogitOut[,"recInv"])  
   recOdds <- apply(recLogOdds,1,exp)   
   hist(recOdds, main=NULL)   
     
   
4. Add details to your histogram by marking off the HDI:  
     
   abline(v=quantile(recOdds,c(0.025)),col="black")   
   abline(v=quantile(recOdds,c(0.975)),col="black")   
     
   
5. Write an interpretation of all of the Bayesian output. Obtain and report specific values for the mean of the posterior distribution as well as the upper and lower bounds of its HDI. Hint: The code to get the upper and lower bounds is already embedded in the code provided for Question 13.

The population mean falls within the lower bound 2.5 and upper bound 6.5 HDI.

**Phase 3 Instructions**: Everything we have done so far has focused on the recommend variable provided by the survey respondents. Conceptually, that is the most proximal variable to the actual hiring decision that we have used as our dependent variable. There are six additional attitude/belief variables in the data set, however, and it would be interesting to know if any of them could add to our predictive capability.

1. Create an lm() model that predicts recommend from the six attitude/belief variables: vision, issues, trends, consult, lead, and collab. Paste the results of the lm model below:
2. Here’s a curious bit of logic: We know that if two predictors are highly correlated, that they will “fight” each other in a linear prediction equation. Because the two predictors have so much in common, once you have placed either one of them in the prediction model, the other one does not have much to contribute when added. As a result, what we are looking for in the output of Question 15, is one or more of the six attitude/belief variables that is a *poor* predictor of recommend. If it is a poor predictor of recommend, that means that it is substantially independent of recommend, and as such it may have a shot at being a workable additional predictor of hired. Based on the results of Question 15, name one, two or three *poor* predictors of recommend
3. Create a series of glm() models with two predictors, namely recommend and something else. For example, the following glm() model would be for recommend and trends:  
     
   glmOut <- glm(formula = hired ~ recommend + trends, family = binomial(link="logit"), data = hiredata)  
     
   If you find a model where both recommend and the other variable are significant predictors, paste in the output of the model(s) below:
4. Using your favorite model from Question 17, create the other output we need to interpret the model using exp(), confint(), and PseudoR2(). Paste in the output below and provide a brief interpretation.
5. If time permits, repeat Questions 17 and 18 using the inverted version of recommend. It will also be helpful to invert your other predictor variable. Pay close attention to the min and max of that other variable as you develop a line of code to invert it.
6. If time permits, create a Bayesian version of your final model from Question 19. Paste the results below. It is especially important to include the posterior distribution of plain odds, with numeric statements of the mean of the distribution and the lower and upper bounds of the HDI.
7. If time permits, write an integration interpretation of Questions 19 and 20. Try to answer the original research question: Can survey respondents accurately assess who will be hired?