w271: Homework 4 (Due: Week 5)

Due: 4pm Pacific Time on the Day of the Live Session of Week 5

Instructions (Please Read it Carefully!):

- **Page limit of the pdf report: None, but please be reasonable**
- Page setup:
  - Use the following font size, margin, and linespace:
    - fontsize=11pt
    - margin=1in
    - line_spacing=single
- Submission:
  - Each student submits his/her homework to the course github repo by the deadline; submission and revision made after the deadline will not be graded
  - Submit 2 files:
    1. A pdf file that details your answers. Include all the R codes used to produce the answers. *Please do not suppress the codes in your pdf file.*
    2. R markdown file used to produce the pdf file
  - Use the following file-naming convensation; fail to do so will receive 10% reduction in the grade:
    - StudentFirstNameLastName_HWNumber.fileExtension
    - For example, if the student's name is Kyle Cartman for homework 1, name your files as
      - KyleCartman_HW1.Rmd
      - KyleCartman_HW1.pdf
  - Although it sounds obvious, please write your name on page 1 of your pdf and Rmd files.
  - For statistical methods that we cover in this course, use only the R libraries and functions that are covered in this course. If you use libraries and functions for statistical modeling that we have not covered, you have to (1) provide an explanation of why such libraries and functions are used instead and (2) reference to the library documentation. **Lacking the explanation and reference to the documentation will result in a score of zero for the corresponding question.** For data wrangling and data visualization, you are free to use other libraries, such as dplyr, ggplot2, etc.
  - For mathematical formulae, type them in your R markdown file. **Do not write them on a piece of paper, snap a photo, and either insert the image file or sumbit the image file separately. Doing so will receive a 0 for that whole question.**
  - Students are expected to act with regards to UC Berkeley Academic Integrity.

Question 18 a and b of Chapter 3 (page 192,193)

For the wheat kernel data (*wheat.csv*), consider a model to estimate the kernel condition using the density explanatory variable as a linear term.

a. Write an R function that computes the log-likelihood function for the multinomial regression model. Evaluate the function at the parameter estimates produced by multinom(), and verify that your computed value is the

same as that produced by logLik() (use the object saved from multinom() within this function).

```
library(package = nnet)

wheat <- read.csv('wheat.csv')

wheat$Healthy <- ifelse(test = wheat$type == "Healthy" , yes = 1 , no = 0)
wheat$Scab <- ifelse(test = wheat$type == "Scab" , yes = 1 , no = 0)
wheat$Sprout <- ifelse(test = wheat$type == "Sprout" , yes = 1 , no = 0)

logL<-function(beta,x,v1,v2,v3) { pi_healthy = 1/(1 + exp(beta[1] + beta[3]*x) + exp(beta[2] + b
eta[4]*x))
pi_scab = exp(beta[1] + beta[3]*x)/(1 + exp(beta[1] + beta[3]*x) + exp(beta[2] + beta[4]*x))
pi_sprout = exp(beta[2] + beta[4]*x)/(1 + exp(beta[1] + beta[3]*x) + exp(beta[2] + beta[4]*x))
sum(v1*log(pi_healthy) + v2*log(pi_scab) + v3*log(pi_sprout))
}

mod.fit <- multinom(formula = type ~ density, data = wheat )
```

```
## # weights:  9 (4 variable)
## initial  value 302.118379
## iter  10 value 229.769334
## iter  20 value 229.712304
## final  value 229.712290
## converged
```

```
logL(beta = coefficients(mod.fit), x = wheat$density, v1 = wheat$Healthy, v2 = wheat$Scab, v3 =
wheat$Sprout)
```

```
## [1] -229.7123
```

```
logLik(mod.fit)
```

```
## 'log Lik.' -229.7123 (df=4)
```

b. Maximize the log-likelihood function using optim() to obtain the MLEs and the estimated covariance matrix. Compare your answers to what is obtained by multinom(). Note that to obtain starting values for optim(), one approach is to estimate separate logistic regression models for $log\left(\frac{\pi_2}{\pi_1}\right)$ and $log\left(\frac{\pi_3}{\pi_1}\right)$. These models are estimated only for those observations that have the corresponding responses (e.g., a $Y = 1$ or $Y = 2$ for $log\left(\frac{\pi_2}{\pi_1}\right)$).

```
Sprout_Data <- wheat[which(wheat$type!='Sprout'),]
Scab_Data <- wheat[which(wheat$type!='Scab'),]

mod.fit.binary2 <- glm(formula=type~density, family=binomial(link="logit" ), data=Sprout_Data)

mod.fit.binary3 <- glm(formula=type~density, family=binomial(link="logit" ), data=Scab_Data)

binary.coef <- matrix(c(mod.fit.binary2$coefficients[1], mod.fit.binary3$coefficients[1], mod.fi
t.binary2$coefficients[2],mod.fit.binary3$coefficients[2]), nrow=2, ncol=2)


mod.fit.optim<-optim(par = binary.coef, fn = logL, hessian = TRUE, x = wheat$density, v1 = wheat
$Healthy, v2 = wheat$Scab, v3 = wheat$Sprout, control = list(fnscale = -1), method = "BFGS")


mod.fit.optim$par
```

```
##          [,1]      [,2]
## [1,] 29.39074 -24.57235
## [2,] 19.13215 -15.48479
```

```
coefficients(mod.fit)
```

```
##        (Intercept)   density
## Scab      29.37827 -24.56215
## Sprout    19.12165 -15.47633
```

```
-solve(mod.fit.optim$hessian)
```

```
##            [,1]      [,2]       [,3]      [,4]
## [1,]  13.528400 10.332699 -11.084134 -8.278666
## [2,]  10.332699 11.143630  -8.334220 -8.976677
## [3,] -11.084134 -8.334220   9.113216  6.686857
## [4,]  -8.278666 -8.976677   6.686857  7.248590
```

```
vcov(mod.fit)
```

```
##                    Scab:(Intercept) Scab:density Sprout:(Intercept)
## Scab:(Intercept)          13.519536   -11.076940          10.325093
## Scab:density             -11.076940     9.107371          -8.328101
## Sprout:(Intercept)        10.325093    -8.328101          11.136180
## Sprout:density            -8.272576     6.681955          -8.970700
##                    Sprout:density
## Scab:(Intercept)        -8.272576
## Scab:density             6.681955
## Sprout:(Intercept)      -8.970700
## Sprout:density           7.243792
```