Bios 6301: Assignment 5

Due Tuesday, 10 November, 1:00 PM

50 points total.

Question 1

24 points

Import the HAART dataset (`haart.csv`) from the GitHub repository into R, and perform the following manipulations: (4 points each)

```
haart <- read.csv("../Bios6301 - master/datasets/haart.csv")
```

1. Convert date columns into a usable (for analysis) format. Use the `table` command to display the counts of the year from `init.date`.

```
haart[,'init.date'] <- as.Date(haart[,'init.date'], format='%m/%d/%y')
haart[,'last.visit'] <- as.Date(haart[,'last.visit'], format='%m/%d/%y')
haart[,'date.death'] <- as.Date(haart[,'date.death'], format='%m/%d/%y')
table(year(haart[, 'init.date']))
```

```
##
## 1998 2000 2001 2002 2003 2004 2005 2006 2007
##    1    5   17   60  270  292  207  104   44
```

2. Create an indicator variable (one which takes the values 0 or 1 only) to represent death within 1 year of the initial visit. How many observations died in year 1?

```
death.1year <- as.numeric(haart[,'date.death'] - haart[,'init.date'] < 365)
sum(death.1year, na.rm=T)
```

```
## [1] 92
```

3. Use the `init.date`, `last.visit` and `death.date` columns to calculate a followup time (in days), which is the difference between the first and either the last visit or a death event (whichever comes first). If these times are longer than 1 year, censor them (this means if the value is above 365, set followup to 365). Print the quantiles for this new variable.

```
followUp <- c()
for (i in 1:length(haart[,'init.date'])) {
  dif <- haart[,'last.visit'][i] - haart[,'init.date'][i]
  if (is.na(haart[, 'last.visit'][i])) {
    dif <- 366
  }
  deathDif <- haart[,'date.death'][i] - haart[,'init.date'][i]
  if (is.na(haart[, 'date.death'][i])) {
    deathDif <- 366
  }
```

```
  if (dif > 365 & deathDif > 365) {
    followUp[i] <- 365
    next
  }
  if (dif > deathDif) {
    followUp[i] <- deathDif
    next
  }
  else followUp[i] <- dif
}
quantile(followUp)
```

```
##     0%    25%    50%    75%   100%
##   0.00 320.75 365.00 365.00 365.00
```

4. Create another indicator variable representing loss to followup; this means the observation is not known to be dead but does not have any followup visits after the first year. How many records are lost-to-followup?

```
lost2followUp <- c()
for (i in 1:length(haart[,'death'])) {
  if (haart[,'death'][i] == 0 & is.na(haart[,'last.visit'][i])) {
    lost2followUp[i] <- 1
  }
  else lost2followUp[i] <- 0
}
sum(lost2followUp)
```

```
## [1] 0
```

5. Recall our work in class, which separated the `init.reg` field into a set of indicator variables, one for each unique drug. Create these fields and append them to the database as new columns. Which drug regimens are found over 100 times?

```
reg_list <- strsplit(as.character(haart[,'init.reg']), ',')

all_drugs <- unique(unlist(reg_list))
reg_drugs <- matrix(nrow=nrow(haart), ncol = length(all_drugs))
for (i in seq_along(all_drugs)) {
  reg_drugs[,i] <- +sapply(reg_list, function(x) all_drugs[i] %in% x)
}
colnames(reg_drugs) <- all_drugs
haart <- cbind(haart, reg_drugs)

which(colSums(reg_drugs) > 100)
```

```
## 3TC AZT EFV NVP D4T
##   1   2   3   4   5
```

** Note: The numbers below the output of the which function are their ranks in descending order, i.e. 1 indicates the drug with the most occurances.

6. The dataset `haart2.csv` contains a few additional observations for the same study. Import these and append them to your master dataset (if you were smart about how you coded the previous steps, cleaning the additional observations should be easy!). Show the first five records and the last five records of the complete (and clean) data set.

```
haart2 <- read.csv("../Bios6301 - master/datasets/haart2.csv")
haart2[,'init.date'] <- as.Date(haart2[,'init.date'], format='%m/%d/%y')
haart2[,'last.visit'] <- as.Date(haart2[,'last.visit'], format='%m/%d/%y')
haart2[,'date.death'] <- as.Date(haart2[,'date.death'], format='%m/%d/%y')

reg_list <- strsplit(as.character(haart2[,'init.reg']), ',')
reg_drugs <- matrix(nrow=nrow(haart), ncol = length(all_drugs))
for (i in seq_along(all_drugs)) {
  reg_drugs[,i] <- +sapply(reg_list, function(x) all_drugs[i] %in% x)
}
colnames(reg_drugs) <- all_drugs
haart2 <- cbind(haart2, reg_drugs)

haartTot <- rbind(haart, haart2)

head(haartTot)
```

```
##   male age aids cd4baseline logvl  weight hemoglobin    init.reg
## 1    1  25    0          NA    NA      NA         NA 3TC,AZT,EFV
## 2    1  49    0         143    NA 58.0608         11 3TC,AZT,EFV
## 3    1  42    1         102    NA 48.0816          1 3TC,AZT,EFV
## 4    0  33    0         107    NA 46.0000         NA 3TC,AZT,NVP
## 5    1  27    0          52     4      NA         NA 3TC,D4T,EFV
## 6    0  34    0         157    NA 54.8856         NA 3TC,AZT,NVP
##    init.date last.visit death date.death 3TC AZT EFV NVP D4T ABC DDI IDV
## 1 2003-07-01 2007-02-26     0       <NA>   1   1   1   0   0   0   0   0
## 2 2004-11-23 2008-02-22     0       <NA>   1   1   1   0   0   0   0   0
## 3 2003-04-30 2005-11-21     1 2006-01-11   1   1   1   0   0   0   0   0
## 4 2006-03-25 2006-05-05     1 2006-05-07   1   1   0   1   0   0   0   0
## 5 2004-09-01 2007-11-13     0       <NA>   1   0   1   0   1   0   0   0
## 6 2003-12-02 2008-02-28     0       <NA>   1   1   0   1   0   0   0   0
##   LPV RTV SQV FTC TDF DDC NFV T20 ATV FPV
## 1   0   0   0   0   0   0   0   0   0   0
## 2   0   0   0   0   0   0   0   0   0   0
## 3   0   0   0   0   0   0   0   0   0   0
## 4   0   0   0   0   0   0   0   0   0   0
## 5   0   0   0   0   0   0   0   0   0   0
## 6   0   0   0   0   0   0   0   0   0   0
```

```
tail(haartTot)
```

```
##      male      age aids cd4baseline    logvl  weight hemoglobin
## 1995    1 23.00000   NA         154 3.995635 65.5000         14
## 1996    0 31.00000    0         236       NA 45.8136         NA
## 1997    0 27.00000    0         232       NA      NA         NA
## 1998    1 38.72142    0         170       NA 84.0000         NA
## 1999    1 23.00000   NA         154 3.995635 65.5000         14
## 2000    0 31.00000    0         236       NA 45.8136         NA
```

```
##         init.reg  init.date last.visit death date.death 3TC AZT EFV NVP
## 1995 3TC,DDI,EFV 2007-01-31 2007-04-16     0       <NA>   1   0   1   0
## 1996 3TC,D4T,NVP 2003-12-03 2007-10-11     0       <NA>   1   0   0   1
## 1997 3TC,AZT,NVP 2003-12-01 2004-01-05     0       <NA>   1   1   0   1
## 1998 3TC,AZT,NVP 2002-09-26 2004-03-29     0       <NA>   1   1   0   1
## 1999 3TC,DDI,EFV 2007-01-31 2007-04-16     0       <NA>   1   0   1   0
## 2000 3TC,D4T,NVP 2003-12-03 2007-10-11     0       <NA>   1   0   0   1
##      D4T ABC DDI IDV LPV RTV SQV FTC TDF DDC NFV T20 ATV FPV
## 1995   0   0   1   0   0   0   0   0   0   0   0   0   0   0
## 1996   1   0   0   0   0   0   0   0   0   0   0   0   0   0
## 1997   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 1998   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 1999   0   0   1   0   0   0   0   0   0   0   0   0   0   0
## 2000   1   0   0   0   0   0   0   0   0   0   0   0   0   0
```

Question 2

10 points

Obtain the code for using Newton's Method to estimate logistic regression parameters (`logistic.r`) and modify it to predict `death` from `weight`, `hemoglobin` and `cd4baseline` in the HAART dataset. Use complete cases only. Report the estimates for each parameter, including the intercept.

Note: The original script `logistic_debug.r` is in the exercises folder. It needs modification, specifically, the logistic function should be defined:

```
haartFull <- haart[!is.na(haart['weight']),]
haartFull <- haartFull[!is.na(haartFull['hemoglobin']),]
haartFull <- haartFull[!is.na(haartFull['cd4baseline']),]
haartFull <- haartFull[!is.na(haartFull['death']),]
logistic <- function(x) 1 / (1 + exp(-x))

estimate_logistic <- function(x, y, MAX_ITER=10) {

    n <- dim(x)[1]
    k <- dim(x)[2]

    x <- as.matrix(cbind(rep(1, n), x))
    y <- as.matrix(y)

    # Initialize fitting parameters
    theta <- rep(0, k+1)

    J <- rep(0, MAX_ITER)

    for (i in 1:MAX_ITER) {

        # Calculate linear predictor
        z <- x %*% theta
        # Apply logit function
        h <- logistic(z)

        # Calculate gradient
        grad <- t((1/n)*x) %*% as.matrix(h - y)
```

```
        # Calculate Hessian
        H <- t((1/n)*x) %*% diag(array(h)) %*% diag(array(1-h)) %*% x

        # Calculate log likelihood
        J[i] <- (1/n) %*% sum(-y * log(h) - (1-y) * log(1-h))

        # Newton's method
        theta <- theta - solve(H) %*% grad
    }

    return(theta)
}

estimate_logistic(haartFull[,c('weight','hemoglobin','cd4baseline')],
                  haartFull[,'death'])
```

```
##                      [,1]
## rep(1, n)    3.576411744
## weight      -0.046210552
## hemoglobin  -0.350642786
## cd4baseline  0.002092582
```

```
# Compare with R's built-in linear regression
g <- glm(death ~ weight + hemoglobin + cd4baseline, data=haartFull, family=binomial(logit))
print(g$coefficients)
```

```
##  (Intercept)       weight   hemoglobin  cd4baseline
##  3.576411744 -0.046210552 -0.350642786  0.002092582
```

Question 3

14 points

Import the `addr.txt` file from the GitHub repository. This file contains a listing of names and addresses (thanks google). Parse each line to create a data.frame with the following columns: lastname, firstname, streetno, streetname, city, state, zip. Keep middle initials or abbreviated names in the firstname column. Print out the entire data.frame.

```
addr <- read.table("../Bios6301 - master/datasets/addr.txt", header=F, sep='\t',
                   stringsAsFactors = F)
addr <- sapply(addr, as.character)
addr_temp <- unlist(strsplit(addr[,1], split= '  '))
addr_list <- c()
for (i in addr_temp) {
  if (nchar(i) > 0) {
    trimmed <- str_trim(i)
    addr_list <- c(addr_list, trimmed)
  }
}

lnames <- c()
fnames <- c()
```

```
strno <- c()
strna <- c()
city <- c()
sta <- c()
zip <- c()
for (i in seq(1,length(addr_list),6)) {
  lnames <- c(lnames, addr_list[i])
  fnames <- c(fnames, addr_list[i+1])
  strno <- c(strno, strsplit(addr_list[i+2],' ')[[1]][1])
  strna <- c(strna, sub('[^A-Z]+', '', addr_list[i+2]))
  city <- c(city, addr_list[i+3])
  sta <- c(sta, addr_list[i+4])
  zip <- c(zip, addr_list[i+5])
}

(address <- data.frame(Last.Name=lnames, First.Name=fnames, Street.Num=strno, Street.Name=strna,
                       City=city, State=sta, Zip=zip))
```

```
##      Last.Name First.Name Street.Num          Street.Name       City State
## 1        Bania  Thomas M.        725    Commonwealth Ave.     Boston    MA
## 2      Barnaby      David        373        W. Geneva St.   Wms. Bay    WI
## 3       Bausch       Judy        373        W. Geneva St.   Wms. Bay    WI
## 4      Bolatto    Alberto        725    Commonwealth Ave.     Boston    MA
## 5    Carlstrom       John        933          E. 56th St.    Chicago    IL
## 6   Chamberlin Richard A.        111           Nowelo St.       Hilo    HI
## 7        Chuss       Dave       2145          Sheridan Rd   Evanston    IL
## 8        Davis      E. J.        933          E. 56th St.    Chicago    IL
## 9        Depoy     Darren        174         W. 18th Ave.   Columbus    OH
## 10     Griffin       Greg       5000          Forbes Ave. Pittsburgh    PA
## 11   Halvorsen       Nils        933          E. 56th St.    Chicago    IL
## 12      Harper         Al        373        W. Geneva St.   Wms. Bay    WI
## 13       Huang     Maohai        725 W. Commonwealth Ave.     Boston    MA
## 14     Ingalls   James G.        725 W. Commonwealth Ave.     Boston    MA
## 15     Jackson   James M.        725 W. Commonwealth Ave.     Boston    MA
## 16     Knudsen      Scott        373        W. Geneva St.   Wms. Bay    WI
## 17       Kovac       John       5640        S. Ellis Ave.    Chicago    IL
## 18   Landsberg      Randy       5640        S. Ellis Ave.    Chicago    IL
## 19          Lo  Kwok-Yung       1002         W. Green St.     Urbana    IL
## 20 Loewenstein  Robert F.        373        W. Geneva St.   Wms. Bay    WI
## 21       Lynch       John       4201          Wilson Blvd  Arlington    VA
## 22     Martini       Paul        174         W. 18th Ave.   Columbus    OH
## 23       Meyer    Stephan        933          E. 56th St.    Chicago    IL
## 24      Mrozek       Fred        373        W. Geneva St.   Wms. Bay    WI
## 25     Newcomb       Matt       5000          Forbes Ave. Pittsburgh    PA
## 26       Novak      Giles       2145          Sheridan Rd   Evanston    IL
## 27      Odalen      Nancy        373        W. Geneva St.   Wms. Bay    WI
## 28      Pernic       Dave        373        W. Geneva St.   Wms. Bay    WI
## 29      Pernic        Bob        373        W. Geneva St.   Wms. Bay    WI
## 30    Peterson    Jeffrey       5000          Forbes Ave. Pittsburgh    PA
## 31       Pryke       Clem        933          E. 56th St.    Chicago    IL
## 32      Rebull      Luisa       5640        S. Ellis Ave.    Chicago    IL
## 33   Renbarger     Thomas       2145          Sheridan Rd   Evanston    IL
## 34     Rottman        Joe       8730  W. Mountain View Ln  Littleton    CO
```

```
## 35   Schartman      Ethan        933          E. 56th St.    Chicago    IL
## 36       Spotz        Bob        373       W. Geneva St.   Wms. Bay    WI
## 37       Thoma       Mark        373       W. Geneva St.   Wms. Bay    WI
## 38      Walker      Chris        933       N. Cherry St.     Tucson    AZ
## 39      Wehrer     Cheryl       5000         Forbes Ave. Pittsburgh    PA
## 40       Wirth      Jesse        373       W. Geneva St.   Wms. Bay    WI
## 41      Wright       Greg        791 Holmdel-Keyport Rd.    Holmdel    NY
## 42     Zingale    Michael       5640       S. Ellis Ave.    Chicago    IL
##           Zip
## 1       02215
## 2       53191
## 3       53191
## 4       02215
## 5       60637
## 6       96720
## 7  60208-3112
## 8       60637
## 9       43210
## 10      15213
## 11      60637
## 12      53191
## 13      02215
## 14      02215
## 15      02215
## 16      53191
## 17      60637
## 18      60637
## 19      61801
## 20      53191
## 21      22230
## 22      43210
## 23      60637
## 24      53191
## 25      15213
## 26 60208-3112
## 27      53191
## 28      53191
## 29      53191
## 30      15213
## 31      60637
## 32      60637
## 33 60208-3112
## 34      80125
## 35      60637
## 36      53191
## 37      53191
## 38      85721
## 39      15213
## 40      53191
## 41 07733-1988
## 42      60637
```

Question 4

2 points

The first argument to most functions that fit linear models are formulas. The following example defines the response variable `death` and allows the model to incorporate all other variables as terms. `.` is used to mean all columns not otherwise in the formula.

```
url <- "https://github.com/fonnesbeck/Bios6301/raw/master/datasets/haart.csv"
haart_df <- read.csv(url)[,c('death','weight','hemoglobin','cd4baseline')]
coef(summary(glm(death ~ ., data=haart_df, family=binomial(logit))))
```

```
##                 Estimate  Std. Error   z value     Pr(>|z|)
## (Intercept)  3.576411744 1.226870535  2.915069 0.0035561039
## weight      -0.046210552 0.022556001 -2.048703 0.0404911395
## hemoglobin  -0.350642786 0.105064078 -3.337418 0.0008456055
## cd4baseline  0.002092582 0.001811959  1.154872 0.2481427160
```

Now imagine running the above several times, but with a different response and data set each time. Here's a function:

```
myfun <- function(dat, response) {
  form <- as.formula(response ~ .)
  coef(summary(glm(response ~ ., data=dat, family=binomial(logit))))
}
```

Unfortunately, it doesn't work. `tryCatch` is "catching" the error so that this file can be knit to PDF.

```
tryCatch(myfun(haart_df, death), error = function(e) e)
```

```
## <simpleError in eval(expr, envir, enclos): object 'death' not found>
```

What do you think is going on? Consider using `debug` to trace the problem.

When you plug death in for response and run the funciton, the function looks for the stored object `death` which does not exist.

5 bonus points I couldn't find a way for the function run in the environment and also call the variable. The code below is the closest I got, but it now inputs `death` as a character which still poses an error.

```
myfun2 <- function(dat) {
  attach(dat)
  response <- readline(prompt= "What is the dependent variable: ")
  myfun(dat, response)
  detach(dat)
}
```