Bios 6301: Assignment 5

Hannah Weeks

Due Tuesday, 10 November, 1:00 PM

$5^{n=day}$ points taken off for each day late.

50 points total.

Submit a single knitr file (named `homework5.rmd`), along with a valid PDF output file. Inside the file, clearly indicate which parts of your responses go with which problems (you may use the original homework document as a template). Add your name as `author` to the file's metadata section. Raw R code/output or word processor files are not acceptable.

Failure to name file `homework5.rmd` or include author name may result in 5 points taken off.

Question 1

24 points

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

```
#Load in data and lubridate package
setwd("~/Documents/BIOS 6301/Homework")
library(lubridate)

haart <- read.csv('haart.csv', header=TRUE)
head(haart)
```

```
##   male age aids cd4baseline logvl  weight hemoglobin    init.reg init.date
## 1    1  25    0          NA    NA      NA         NA 3TC,AZT,EFV    7/1/03
## 2    1  49    0         143    NA 58.0608         11 3TC,AZT,EFV  11/23/04
## 3    1  42    1         102    NA 48.0816          1 3TC,AZT,EFV   4/30/03
## 4    0  33    0         107    NA 46.0000         NA 3TC,AZT,NVP   3/25/06
## 5    1  27    0          52     4      NA         NA 3TC,D4T,EFV    9/1/04
## 6    0  34    0         157    NA 54.8856         NA 3TC,AZT,NVP   12/2/03
##   last.visit death date.death
## 1    2/26/07     0       <NA>
## 2    2/22/08     0       <NA>
## 3   11/21/05     1    1/11/06
## 4     5/5/06     1     5/7/06
## 5   11/13/07     0       <NA>
## 6    2/28/08     0       <NA>
```

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

```
names(haart)
```

```
##  [1] "male"        "age"         "aids"        "cd4baseline" "logvl"
##  [6] "weight"      "hemoglobin"  "init.reg"    "init.date"   "last.visit"
## [11] "death"       "date.death"
```

The columns containing dates that must be reformatted are `init.date`, `last.visit`, and `date.death`.

```
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")
#Display first few rows to make sure changes were made correctly
head(haart)
```

```
##   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
## 1 2003-07-01 2007-02-26     0       <NA>
## 2 2004-11-23 2008-02-22     0       <NA>
## 3 2003-04-30 2005-11-21     1 2006-01-11
## 4 2006-03-25 2006-05-05     1 2006-05-07
## 5 2004-09-01 2007-11-13     0       <NA>
## 6 2003-12-02 2008-02-28     0       <NA>
```

```
class(haart[,'init.date'])
```

```
## [1] "Date"
```

Now, display the counts by year for `init.date`.

```
#Using lubridate
table(year(haart[,'init.date']))
```

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

```
#Alternatively, could use format function
#table(format(haart[,'init.date'], '%Y'))
```

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?

Here, we specify 365 days from the initial date to be the one year followup period.

```
deathOneYear <- numeric(length=nrow(haart))
for(i in 1:nrow(haart)){
  x <- difftime(haart[i,'date.death'], haart[i,'init.date'], units='days')
  ifelse(x <= 365, deathOneYear[i] <- 1, deathOneYear[i] <- 0)
}
haart[,'deathInOneYear'] <- deathOneYear

sum(deathOneYear)
```

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

So there were 92 deaths within the first year.

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 quantile for this new variable.

A note on the "whichever comes first" requirement: If the patient died and has a death date, that patient cannot have a last visit *after* their date of death. So, if they died, then I looked to see if a last visit date was entered earlier than th.

```
followup <- numeric(length=nrow(haart))

#Run for each patient record
for(i in 1:nrow(haart)){
  #If patient is dead and last.visit date is missing...
  if(haart[i,'death'] == 1 & is.na(haart[i,'last.visit'])){
    #...define followup using death date
    followup[i] <- difftime(haart[i,'date.death'], haart[i,'init.date'], units='days')
  #If patient is dead and last.visit date is not missing...
  } else if(haart[i,'death'] == 1 & !is.na(haart[i,'last.visit'])){
    #...define followup using the earlier of the two dates
    minimum <- min(haart[i,'date.death'],haart[i,'last.visit'])
    followup[i] <- difftime(minimum, haart[i,'init.date'], units='days')
  #If patient is not known to be dead...
  } else if(haart[i, 'death'] == 0){
    #...define followup using last.visit date
    followup[i] <- difftime(haart[i,'last.visit'], haart[i,'init.date'], units='days')
  }
  #Censor any followup times beyond one year
  if(followup[i] > 365) followup[i] <- 365
}
#Add new variable to the data
haart[,'followup.days'] <- followup

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?

```
#Initialize with 1's, then correct records that have known outcomes (i.e. not lost to followup)
lossToFollowup <- rep(1, length=nrow(haart))

#Run for each patient record
for(i in 1:nrow(haart)){
  #If patient is dead, they are not lost to followup
```

```
  if(haart[i,'death'] == 1) lossToFollowup[i] <- 0
  #If patient's last.visit date is not missing
  else if(!is.na(haart[i,'last.visit'])){
    #Determine days between init.date and last.visit
    x <- difftime(haart[i,'last.visit'], haart[i,'init.date'], units = 'days')
    #If last.visit is later than one year of init.date, patient is not lost to followup
    if(x > 365) lossToFollowup[i] <- 0
  }
}
#Add to data
haart[,'loss.to.followup'] <- lossToFollowup

sum(lossToFollowup)
```

```
## [1] 173
```

So 173 records were lost to followup.

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 regimen are found over 100 times?

```
#Code from class
#List drug regimen for each patient
reg_list <- strsplit(as.character(haart[,'init.reg']), ',')
#Create unique list of all drugs used
all_drugs <- unique(unlist(reg_list))
#Initialize empty matrix for each patient and drug
reg_drugs <- matrix(nrow=nrow(haart), ncol=length(all_drugs))

#Fill in matrix with a 1 if that drug was used for that patient, and a 0 otherwise
for(i in seq_along(all_drugs)){
  #'+' makes this list 1/0 instead of T/F
  reg_drugs[,i] <- +sapply(reg_list, function(x) all_drugs[i] %in% x)
}
colnames(reg_drugs) <- all_drugs
#Add to data
haart <- cbind(haart, reg_drugs)
```

```
drug.totals <- numeric(length=length(colnames(reg_drugs)))
names(drug.totals) <- colnames(reg_drugs)

#For each of the drugs, sum to get total number of patients prescribed that medicine
for(i in 1:ncol(reg_drugs)){
  drug.totals[i] <- sum(reg_drugs[,i])
  #If drug was used more than 100 times, print the drug name
  if(drug.totals[i] > 100) print(names(drug.totals[i]))
}
```

```
## [1] "3TC"
## [1] "AZT"
## [1] "EFV"
```

```
## [1] "NVP"
## [1] "D4T"
```

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.

To add these to the existing haart data, we clean up haart2 by repeating the steps taken above for questions 1.1-1.5.

```
haart2 <- read.csv("haart2.csv", header=TRUE)

#Fix dates
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")

#Deaths in a year
deathOneYear <- numeric(length=nrow(haart2))
for(i in 1:nrow(haart2)){
  x <- difftime(haart2[i,'date.death'], haart2[i,'init.date'], units='days')
  ifelse(x <= 365, deathOneYear[i] <- 1, deathOneYear[i] <- 0)
}
haart2[,'deathInOneYear'] <- deathOneYear

#Days to follow up
followup <- numeric(length=nrow(haart2))
for(i in 1:nrow(haart2)){
  if(haart2[i,'death'] == 1 & is.na(haart2[i,'last.visit'])){
    followup[i] <- difftime(haart2[i,'date.death'], haart2[i,'init.date'], units='days')
  } else if(haart2[i,'death'] == 1 & !is.na(haart2[i,'last.visit'])){
    minimum <- min(haart2[i,'date.death'],haart2[i,'last.visit'])
    followup[i] <- difftime(minimum, haart2[i,'init.date'], units='days')
  } else if(haart2[i, 'death'] == 0){
    followup[i] <- difftime(haart2[i,'last.visit'], haart2[i,'init.date'], units='days')
  }
  if(followup[i] > 365) followup[i] <- 365
}
haart2[,'followup.days'] <- followup

#Determine records lost to follow up
lossToFollowup <- rep(1, length=nrow(haart2))
for(i in 1:nrow(haart2)){
  if(haart2[i,'death'] == 1) lossToFollowup[i] <- 0
  else if(is.na(haart2[i,'last.visit'])) lossToFollowup[i] <- 0
  else if(!is.na(haart2[i,'last.visit'])){
    x <- difftime(haart2[i,'last.visit'], haart2[i,'init.date'], units = 'days')
    if(x > 365) lossToFollowup[i] <- 0
  }
}
haart2[,'loss.to.followup'] <- lossToFollowup
```

```r
#Determine drugs used for each patient
reg_list <- strsplit(as.character(haart2[,'init.reg']), ',')
reg_drugs <- matrix(nrow=nrow(haart2), ncol=length(all_drugs))
for(i in seq_along(all_drugs)){
  #'+' makes this list 1/0 instead of T/F
  reg_drugs[,i] <- +sapply(reg_list, function(x) all_drugs[i] %in% x)
}
colnames(reg_drugs) <- all_drugs
haart2 <- cbind(haart2, reg_drugs)
```

```r
#Combine two haart datasets
haartMaster <- rbind(haart, haart2)
#First five records
head(haartMaster, 5)
```

```
##   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
##    init.date last.visit death date.death deathInOneYear followup.days
## 1 2003-07-01 2007-02-26     0       <NA>              0           365
## 2 2004-11-23 2008-02-22     0       <NA>              0           365
## 3 2003-04-30 2005-11-21     1 2006-01-11              0           365
## 4 2006-03-25 2006-05-05     1 2006-05-07              1            41
## 5 2004-09-01 2007-11-13     0       <NA>              0           365
##   loss.to.followup 3TC AZT EFV NVP D4T ABC DDI IDV LPV RTV SQV FTC TDF DDC
## 1                0   1   1   1   0   0   0   0   0   0   0   0   0   0   0
## 2                0   1   1   1   0   0   0   0   0   0   0   0   0   0   0
## 3                0   1   1   1   0   0   0   0   0   0   0   0   0   0   0
## 4                0   1   1   0   1   0   0   0   0   0   0   0   0   0   0
## 5                0   1   0   1   0   1   0   0   0   0   0   0   0   0   0
##   NFV T20 ATV FPV
## 1   0   0   0   0
## 2   0   0   0   0
## 3   0   0   0   0
## 4   0   0   0   0
## 5   0   0   0   0
```

```r
#Last five records
tail(haartMaster, 5)
```

```
##      male      age aids cd4baseline    logvl  weight hemoglobin
## 1000    0 40.00000    1         131       NA 46.2672          8
## 1001    0 27.00000    0         232       NA      NA         NA
## 1002    1 38.72142    0         170       NA 84.0000         NA
## 1003    1 23.00000   NA         154 3.995635 65.5000         14
## 1004    0 31.00000    0         236       NA 45.8136         NA
##         init.reg  init.date last.visit death date.death deathInOneYear
## 1000 3TC,D4T,NVP 2003-07-03 2008-02-29     0       <NA>              0
## 1001 3TC,AZT,NVP 2003-12-01 2004-01-05     0       <NA>              0
```

```
## 1002 3TC,AZT,NVP 2002-09-26 2004-03-29     0       <NA>              0
## 1003 3TC,DDI,EFV 2007-01-31 2007-04-16     0       <NA>              0
## 1004 3TC,D4T,NVP 2003-12-03 2007-10-11     0       <NA>              0
##      followup.days loss.to.followup 3TC AZT EFV NVP D4T ABC DDI IDV LPV
## 1000           365                0   1   0   0   1   1   0   0   0   0
## 1001            35                1   1   1   0   1   0   0   0   0   0
## 1002           365                0   1   1   0   1   0   0   0   0   0
## 1003            75                1   1   0   1   0   0   0   1   0   0
## 1004           365                0   1   0   0   1   1   0   0   0   0
##      RTV SQV FTC TDF DDC NFV T20 ATV FPV
## 1000   0   0   0   0   0   0   0   0   0
## 1001   0   0   0   0   0   0   0   0   0
## 1002   0   0   0   0   0   0   0   0   0
## 1003   0   0   0   0   0   0   0   0   0
## 1004   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:

```
logistic <- function(x) 1 / (1 + exp(-x))
```

Using the modified logistic file:

```
haart <- read.csv("haart.csv", header=TRUE)

haart <- haart[,c('death', 'weight', 'hemoglobin', 'cd4baseline')]
haart <- haart[complete.cases(haart),]

haartDeath <- haart[,'death']
haartFactors <- haart[,c('weight', 'hemoglobin', 'cd4baseline')]

# Logistic function
logistic <- function(x) 1 / (1 + exp(-x))

x <- haartFactors
y <- haartDeath


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(x, y)
```

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

(Note that `rep(1,n)` represents the intercept).

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 middleinitials or abbreviated names in the firstname column. Print out the entire data.frame.

```
#Read in data
data <- readLines("addr.txt")

#Make each line a list
all.data <- character(length=length(data))
for(i in 1:length(data)){
  #Data fields in file are split by two or more spaces
```

```
  all.data[i] <- strsplit(data[i], split = "  +")
}

#Row-bind each line of data
info <- do.call(rbind, all.data)


#Split street address column into street number and street name
#Append the split columns to the data frame
library(stringr)
info <- cbind(info, str_split_fixed(info[,3], " ", 2))

#Drop the column containing the combined street information
info <- info[,-3]
#Reorder columns to match address format
info <- info[,c(1,2,6,7,3,4,5)]
#Label columns
colnames(info) <- c("last.name", "first.name", "street.no", "street.name", "city", "state", "zip")

info
```

```
##       last.name     first.name   street.no street.name
##  [1,] "Bania"       "Thomas M."  "725"     "Commonwealth Ave."
##  [2,] "Barnaby"     "David"      "373"     "W. Geneva St."
##  [3,] "Bausch"      "Judy"       "373"     "W. Geneva St."
##  [4,] "Bolatto"     "Alberto"    "725"     "Commonwealth Ave."
##  [5,] "Carlstrom"   "John"       "933"     "E. 56th St."
##  [6,] "Chamberlin"  "Richard A." "111"     "Nowelo St."
##  [7,] "Chuss"       "Dave"       "2145"    "Sheridan Rd"
##  [8,] "Davis"       "E. J."      "933"     "E. 56th St."
##  [9,] "Depoy"       "Darren"     "174"     "W. 18th Ave."
## [10,] "Griffin"     "Greg"       "5000"    "Forbes Ave."
## [11,] "Halvorsen"   "Nils"       "933"     "E. 56th St."
## [12,] "Harper"      "Al"         "373"     "W. Geneva St."
## [13,] "Huang"       "Maohai"     "725"     "W. Commonwealth Ave."
## [14,] "Ingalls"     "James G."   "725"     "W. Commonwealth Ave."
## [15,] "Jackson"     "James M."   "725"     "W. Commonwealth Ave."
## [16,] "Knudsen"     "Scott"      "373"     "W. Geneva St."
## [17,] "Kovac"       "John"       "5640"    "S. Ellis Ave."
## [18,] "Landsberg"   "Randy"      "5640"    "S. Ellis Ave."
## [19,] "Lo"          "Kwok-Yung"  "1002"    "W. Green St."
## [20,] "Loewenstein" "Robert F."  "373"     "W. Geneva St."
## [21,] "Lynch"       "John"       "4201"    "Wilson Blvd"
## [22,] "Martini"     "Paul"       "174"     "W. 18th Ave."
## [23,] "Meyer"       "Stephan"    "933"     "E. 56th St."
## [24,] "Mrozek"      "Fred"       "373"     "W. Geneva St."
## [25,] "Newcomb"     "Matt"       "5000"    "Forbes Ave."
## [26,] "Novak"       "Giles"      "2145"    "Sheridan Rd"
## [27,] "Odalen"      "Nancy"      "373"     "W. Geneva St."
## [28,] "Pernic"      "Dave"       "373"     "W. Geneva St."
## [29,] "Pernic"      "Bob"        "373"     "W. Geneva St."
## [30,] "Peterson"    "Jeffrey"    "5000"    "Forbes Ave."
## [31,] "Pryke"       "Clem"       "933"     "E. 56th St."
```

```
## [32,] "Rebull"      "Luisa"      "5640"    "S. Ellis Ave."
## [33,] "Renbarger"   "Thomas"     "2145"    "Sheridan Rd"
## [34,] "Rottman"     "Joe"        "8730"    "W. Mountain View Ln"
## [35,] "Schartman"   "Ethan"      "933"     "E. 56th St."
## [36,] "Spotz"       "Bob"        "373"     "W. Geneva St."
## [37,] "Thoma"       "Mark"       "373"     "W. Geneva St."
## [38,] "Walker"      "Chris"      "933"     "N. Cherry St."
## [39,] "Wehrer"      "Cheryl"     "5000"    "Forbes Ave."
## [40,] "Wirth"       "Jesse"      "373"     "W. Geneva St."
## [41,] "Wright"      "Greg"       "791"     "Holmdel-Keyport Rd."
## [42,] "Zingale"     "Michael"    "5640"    "S. Ellis Ave."
##       city          state zip
##  [1,] "Boston"      "MA"  "02215 "
##  [2,] "Wms. Bay"    "WI"  "53191"
##  [3,] "Wms. Bay"    "WI"  "53191"
##  [4,] "Boston"      "MA"  "02215 "
##  [5,] "Chicago"     "IL"  "60637"
##  [6,] "Hilo"        "HI"  "96720"
##  [7,] "Evanston"    "IL"  "60208-3112 "
##  [8,] "Chicago"     "IL"  "60637"
##  [9,] "Columbus"    "OH"  "43210"
## [10,] "Pittsburgh"  "PA"  "15213"
## [11,] "Chicago"     "IL"  "60637"
## [12,] "Wms. Bay"    "WI"  "53191"
## [13,] "Boston"      "MA"  "02215 "
## [14,] "Boston"      "MA"  "02215 "
## [15,] "Boston"      "MA"  "02215 "
## [16,] "Wms. Bay"    "WI"  "53191"
## [17,] "Chicago"     "IL"  "60637"
## [18,] "Chicago"     "IL"  "60637"
## [19,] "Urbana"      "IL"  "61801"
## [20,] "Wms. Bay"    "WI"  "53191"
## [21,] "Arlington"   "VA"  "22230"
## [22,] "Columbus"    "OH"  "43210"
## [23,] "Chicago"     "IL"  "60637"
## [24,] "Wms. Bay"    "WI"  "53191"
## [25,] "Pittsburgh"  "PA"  "15213"
## [26,] "Evanston"    "IL"  "60208-3112 "
## [27,] "Wms. Bay"    "WI"  "53191"
## [28,] "Wms. Bay"    "WI"  "53191"
## [29,] "Wms. Bay"    "WI"  "53191"
## [30,] "Pittsburgh"  "PA"  "15213"
## [31,] "Chicago"     "IL"  "60637"
## [32,] "Chicago"     "IL"  "60637"
## [33,] "Evanston"    "IL"  "60208-3112 "
## [34,] "Littleton"   "CO"  "80125"
## [35,] "Chicago"     "IL"  "60637"
## [36,] "Wms. Bay"    "WI"  "53191"
## [37,] "Wms. Bay"    "WI"  "53191"
## [38,] "Tucson"      "AZ"  "85721"
## [39,] "Pittsburgh"  "PA"  "15213"
## [40,] "Wms. Bay"    "WI"  "53191"
## [41,] "Holmdel"     "NY"  "07733-1988 "
## [42,] "Chicago"     "IL"  "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(form, 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.

I think that the way `myfun` is written, it's not able to figure out how to interpret the `death` variable. Typically, when being passed to a model, the dataset from which the variable is drawn must be specified. In `myfun`, `death` is brought into the function a line before `haart` is. Since the function doesn't know what `death` corresponds to, it fails to evaluate `form` and subsequently `glm` correctly.

5 bonus points

Create a working function.