

Bios 6301: Assignment 5

Lingjun Fu

09 November, 2015

Due Tuesday, 10 November, 1:00 PM

$5^{n=\text{day}}$ 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)

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

```
### task 1
library(RCurl)
```

```
## Loading required package: bitops
```

```
download.file("https://raw.githubusercontent.com/fonnesbeck/Bios6301/master/datasets/haart.csv", destfile="haart.csv")
haart <- read.csv("haart.csv", header=TRUE, stringsAsFactors=FALSE)
haart$init.date <- as.Date(haart$init.date, "%m/%d/%y")
```

```

haart$last.visit <- as.Date(haart$last.visit,"%m/%d/%y")
haart$date.death <- as.Date(haart$date.death,"%m/%d/%y")
table(format(haart$init.date, "%Y"))

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

### task 2
haart$indicator <- ifelse((haart$date.death - haart$init.date > 365 | is.na(haart$date.death)),0,1)
sum(haart$indicator==1)

## [1] 92

### task 3
haart$follow.up <- ifelse(is.na(haart$last.visit), haart$date.death - haart$init.date,
                        haart$last.visit - haart$init.date)
haart$follow.up[haart$follow.up > 365] <- 365
quantile(haart$follow.up)

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

### task 4
haart$loss <- 0
for(i in seq(nrow(haart))){
  if((haart$death[i] == 0) && (haart$last.visit[i] - haart$init.date[i] <= 365)){
    haart$loss[i] = 1
  }
}
sum(haart$loss==1)

## [1] 173

### task 5
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) # append each unique drug to the database as new columns
drug <- as.data.frame(reg_drugs)
drug_sum <- sapply(drug, sum)
drug_sum[drug_sum>100] # show drug regimen found over 100 times

## 3TC AZT EFV NVP D4T
## 973 794 516 358 146

```

```

### task 6
download.file("https://raw.githubusercontent.com/fonnesbeck/Bios6301/master/datasets/haart2.csv", destf
t1 <- read.csv("haart.csv", header=TRUE, stringsAsFactors=FALSE)
t2 <- read.csv("haart2.csv", header=TRUE, stringsAsFactors=FALSE)
t <- rbind(t1, t2)
# then just repeat all what we did before
t$init.date <- as.Date(t$init.date, "%m/%d/%y")
t$last.visit <- as.Date(t$last.visit, "%m/%d/%y")
t$date.death <- as.Date(t$date.death, "%m/%d/%y")

t$indicator <- ifelse((t$date.death - t$init.date > 365 | is.na(t$date.death)), 0, 1)
t$follow.up <- ifelse(is.na(t$last.visit), t$date.death - t$init.date,
                     t$last.visit - t$init.date)
t$follow.up[t$follow.up > 365] <- 365

t$loss <- 0
for(i in seq(nrow(t))){
  if((t$death[i] == 0) && (t$last.visit[i] - t$init.date[i] <= 365)){
    t$loss[i] = 1
  }
}

reg_list <- strsplit(as.character(t[, 'init.reg']), ',')
all_drugs <- unique(unlist(reg_list))
reg_drugs <- matrix(nrow=nrow(t), 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
t <- cbind(t, reg_drugs)
head(t, 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 indicator follow.up loss 3TC AZT
## 1 2003-07-01 2007-02-26     0      <NA>         0      365    0  1  1
## 2 2004-11-23 2008-02-22     0      <NA>         0      365    0  1  1
## 3 2003-04-30 2005-11-21     1 2006-01-11         0      365    0  1  1
## 4 2006-03-25 2006-05-05     1 2006-05-07         1       41    0  1  1
## 5 2004-09-01 2007-11-13     0      <NA>         0      365    0  1  0
##   EFV NVP D4T ABC DDI IDV LPV RTV SQV FTC TDF DDC NFV T20 ATV FPV
## 1    1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 2    1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 3    1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 4    0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 5    1  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0

```

```
tail(t, 5)
```

```
##      male      age aids cd4baseline      logv1  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 indicator
## 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
##      follow.up loss 3TC AZT EFV NVP D4T ABC DDI IDV LPV RTV SQV FTC TDF
## 1000        365    0  1  0  0  1  1  0  0  0  0  0  0  0  0
## 1001         35    1  1  1  0  1  0  0  0  0  0  0  0  0  0
## 1002        365    0  1  1  0  1  0  0  0  0  0  0  0  0  0
## 1003         75    1  1  0  1  0  0  0  1  0  0  0  0  0  0
## 1004        365    0  1  0  0  1  1  0  0  0  0  0  0  0  0
##      DDC NFV T20 ATV FPV
## 1000    0  0  0  0  0
## 1001    0  0  0  0  0
## 1002    0  0  0  0  0
## 1003    0  0  0  0  0
## 1004    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:

```
data <- read.csv("haart.csv", header=TRUE, stringsAsFactors=FALSE)
# modify the logistic function
logistic <- function(x) 1 / (1 + exp(-x))

data <- data[complete.cases(data[,c("weight", "hemoglobin", "cd4baseline", "death")]),]
x <- data[,c("weight", "hemoglobin", "cd4baseline")]
y <- data[,c("death")]

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

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

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

```

We see that our estimate_logistic function has the same result as the R's built-in linear regression.

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.

```

library(RCurl)
library(stringr)
download.file("https://raw.githubusercontent.com/fonnesbeck/Bios6301/master/datasets/addr.txt", destfile="addr.txt")
tt<-read.table("addr.txt",header=F,sep="\t",colClasses=c("character"))
temp<-unlist(strsplit(tt[,1], " "))
trim <- function (x) gsub("^\\s+|\\s+$", "", x)
temp<-trim(temp)
temp<-temp[temp!=""]
mt<-matrix(temp,ncol=6,byrow=T)
rexp <- "^((\\w+)\\s?(\\s+))$"
y <- data.frame(streetno=sub(rexp,"\\1",mt[,3]), streetname=sub(rexp,"\\2",mt[,3]))
mt<-cbind(y,mt)
df<-as.data.frame(mt[,-5])
colnames(df)<-c("streetno", "streetname", "lastname", "firstname", "city", "state", "zip")
df<-df[,c(3,4,1,2,5,6,7)]
print(df)

```

##	lastname	firstname	streetno	streetname	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.

```
haart <- read.csv("haart.csv", header=TRUE, stringsAsFactors=FALSE)
haart_df <- haart[,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.

The problem is that `as.formula` function needs a character object variable and one needs to "paste" the entire formula together.

5 bonus points

Create a working function.

```
myfun_1 <- function(dat, response) {
  form <- as.formula(paste(response, "~."))
  coef(summary(glm(form, data=dat, family=binomial(logit))))
}
myfun_1(haart_df, 'death')
```

```
##              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
```