Bios 6301: Assignment 5

Nick Strayer

Question 1

24 points

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

```
setwd("/Users/Nick/Dropbox/vandy/computing/Bios6301/datasets")
h <- read.csv("haart.csv", stringsAsFactors = F)
library(lubridate)
```

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

```
fix_dates = function(h){
  fix1900s <- function(x, year=16){ #Make it 1900s if the 10s digits are above 16.
    m <- year(x) %% 100
    year(x) <- ifelse(m > year, 1900+m, 2000+m)
    x
  }
  h[ , "init.date"]  <- fix1900s(mdy(h[, "init.date"]))
  h[ , "last.visit"] <- fix1900s(mdy(h[, "last.visit"]))
  h[ , "date.death"] <- fix1900s(mdy(h[, "date.death"]))
  h
}
h = fix_dates(h)

table(format(h[ , "init.date"], "%Y"))
```

```
##
## 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?

We check to see if the `date.death` column is a real date, then if it is we assign `1` if they died within a year and `0` if they either died more than a year later or there was no death reported.

```
death_in_year = function(h){
  h[,"death.in.year"] <- ifelse(!is.na(h[ , "date.death"]), + h[ , "date.death"] - h[ , "init.date"] < 3
  h
}
h = death_in_year(h)

sum(h[,"death.in.year"])
```

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

So we see that there were 92 deaths within a 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.*

```
followup_time = function(h){
  for(i in 1:dim(h)[1]){
  dif <- NULL #initialize difference
  #if last visit is not na calc the difference
  if(!is.na(h[i, "last.visit"])) dif <- difftime(h[i, "last.visit"], h[i,"init.date"], units = "days")
  #if the date death is not na make the difference the minimum between the old dif and the new one.
  if(!is.na(h[i, "date.death"])) dif <- min(dif, difftime(h[i, "date.death"], h[i,"init.date"], units =

  h[i,"followup.time"] <- min(365, dif)
  }
  h
}
h = followup_time(h)

quantile(h[,"followup.time"])
```

```
##     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?

```
loss_to_followup = function(h){
  h[,"loss.to.followup"] <- ifelse(is.na(h[,"date.death"]) & h[,"followup.time"] < 365, 1,0)
  h
}
h = loss_to_followup(h)
```

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?*

```
drug_counts = function(h){
  #grab unique drug names
  drugs <- unique(unlist(sapply(h[,"init.reg"], function(d) unlist(strsplit(d, ",") ) ))) #this is ugly

  for(drug in drugs) h[,drug] = 0 #add empty columns

  for(i in 1:dim(h)[1]){ #for each row in the dataframe
    for(drug in drugs){
      if(drug %in% unlist(strsplit(h[i,"init.reg"], ",") )) h[i, drug] = 1 #if the drug is there add to
    }
  }
  for(drug in drugs) if(sum(h[, drug]) > 100) print(drug) #Print the drugs that are prescribed more tha
  h
}
h = drug_counts(h)
```

```
## [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.*

```
setwd("/Users/Nick/Dropbox/vandy/computing/Bios6301/datasets")
h0 <- read.csv("haart.csv", stringsAsFactors = F) #Read in the two datasets
h1 <- read.csv("haart2.csv", stringsAsFactors = F)
h2  <- rbind(h0,h1) #Merge them

h2 = fix_dates(h2) #Run all the previously written functions on the new data.
h2 = death_in_year(h2)
h2 = followup_time(h2)
h2 = loss_to_followup(h2)
h2 = drug_counts(h2)
```

```
## [1] "3TC"
## [1] "AZT"
## [1] "EFV"
## [1] "NVP"
## [1] "D4T"
```

```
h2[1:5,]
```

```
##   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 death.in.year followup.time
## 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
```

```
rows = dim(h2)[1]
h[(rows-5):rows,]
```

```
##      male age aids cd4baseline logvl  weight hemoglobin    init.reg
## 999     0  31    0         102    NA 61.6896         11 3TC,AZT,NVP
## 1000    0  40    1         131    NA 46.2672          8 3TC,D4T,NVP
## NA     NA  NA   NA          NA    NA      NA         NA        <NA>
## NA.1   NA  NA   NA          NA    NA      NA         NA        <NA>
## NA.2   NA  NA   NA          NA    NA      NA         NA        <NA>
## NA.3   NA  NA   NA          NA    NA      NA         NA        <NA>
##       init.date last.visit death date.death death.in.year followup.time
## 999  2003-05-22 2008-03-07     0       <NA>             0           365
## 1000 2003-07-03 2008-02-29     0       <NA>             0           365
## NA         <NA>       <NA>    NA       <NA>            NA            NA
## NA.1       <NA>       <NA>    NA       <NA>            NA            NA
## NA.2       <NA>       <NA>    NA       <NA>            NA            NA
## NA.3       <NA>       <NA>    NA       <NA>            NA            NA
##      loss.to.followup 3TC AZT EFV NVP D4T ABC DDI IDV LPV RTV SQV FTC TDF
## 999                 0   1   1   0   1   0   0   0   0   0   0   0   0   0
## 1000                0   1   0   0   1   1   0   0   0   0   0   0   0   0
## NA                 NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA
## NA.1               NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA
## NA.2               NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA
## NA.3               NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA
##      DDC NFV T20 ATV FPV
## 999    0   0   0   0   0
## 1000   0   0   0   0   0
## NA    NA  NA  NA  NA  NA
## NA.1  NA  NA  NA  NA  NA
## NA.2  NA  NA  NA  NA  NA
## NA.3  NA  NA  NA  NA  NA
```

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))
```

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.

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))))
```

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 is.data.frame(data): object 'haart_df' not found>
```

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

5 bonus points

Create a working function.