Bios 6301: Assignment 5

Andrea Perreault

Due Tuesday, 15 November, 1:00 PM

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

50 points total.

Grade: 47/50 Check out how Cole's solution question 2 with lapply and tapply.

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)

```
haart <- "https://raw.githubusercontent.com/fonnesbeck/Bios6301/master/datasets/haart.csv"
haart_df <- read.csv(haart)
```

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_df['init.date'] <- as.Date(haart_df$init.date, format='%m/%d/%y')
haart_df['last.visit'] <- as.Date(haart_df$last.visit, format='%m/%d/%y')
haart_df['date.death'] <- as.Date(haart_df$date.death, format='%m/%d/%y')

table(format(haart_df$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?

```
haart_df$death.1yr <- ifelse((haart_df$date.death - haart_df$init.date > 365 | is.na(haart_df$date.deatl
sum(haart_df$death.1yr==1)
```

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

92 patients died 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.

```
haart_df$follow.up <- ifelse(is.na(haart_df$last.visit), haart_df$date.death - haart_df$init.date, haar
haart_df$follow.up[haart_df$follow.up > 365] <- 365
quantile(haart_df$follow.up)
```

```
##     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?

```
haart_df$loss <- ifelse((is.na(haart_df$date.death) & haart_df$follow.up < 365 | haart_df$follow.up > 3
table(haart_df$loss)
```

```
##
##   0   1
## 173 827
```

173 patients were lost to follow up.

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?

```
haart_df$init.reg <- as.character(haart_df$init.reg)
all.reg <- strsplit(haart_df$init.reg, ',')
all.reg <- unlist(all.reg)
all.reg <- unique(all.reg)
row.reg <- strsplit(haart_df$init.reg, ',')
patient.reg <- sapply(all.reg, function(j) sapply(row.reg, function(i) j %in% i))
patient.reg <- as.data.frame(+patient.reg)
haart_df <- cbind(haart_df, patient.reg)

colSums(patient.reg)
```

```
## 3TC AZT EFV NVP D4T ABC DDI IDV LPV RTV SQV FTC TDF DDC NFV T20 ATV FPV
## 973 794 516 358 146  56  38  27  31  79  29   8  10   1   8   1   2   2
```

5 drugs were found over 100 times each.

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.

```
haart1 <- "https://raw.githubusercontent.com/fonnesbeck/Bios6301/master/datasets/haart.csv"
haart1_df <- read.csv(haart1)
haart2 <- "https://raw.githubusercontent.com/fonnesbeck/Bios6301/master/datasets/haart2.csv"
haart2_df <- read.csv(haart2)
haart_comb <- rbind(haart1_df, haart2_df)

cleanData <- function(data) {
  data$init.date <- as.Date(data$init.date, format='%m/%d/%y')
  data$last.visit <- as.Date(data$last.visit, format='%m/%d/%y')
  data$date.death <- as.Date(data$date.death, format='%m/%d/%y')

  data$death.1yr <- ifelse((data$date.death - data$init.date > 365 | is.na(data$date.death)), 0, 1)
```

```
  data$follow.up <- ifelse(is.na(data$last.visit), data$date.death - data$init.date, data$last.visit - d
  data$follow.up[data$follow.up > 365] <- 365

  data$loss <- ifelse((is.na(data$date.death) & data$follow.up < 365 | data$follow.up > 365), 0, 1)

  data$init.reg <- as.character(data$init.reg)
  all.reg <- strsplit(data$init.reg, ',')
  all.reg <- unlist(all.reg)
  all.reg <- unique(all.reg)
  row.reg <- strsplit(data$init.reg, ',')
  patient.reg <- sapply(all.reg, function(j) sapply(row.reg, function(i) j %in% i))
  patient.reg <- as.data.frame(+patient.reg)
  data <- cbind(data, patient.reg)

  print(head(data))
  print(tail(data))
}

cleanData(haart_comb)
```

```
##   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 death.1yr follow.up loss 3TC AZT
## 1 2003-07-01 2007-02-26     0       <NA>         0       365    1   1   1
## 2 2004-11-23 2008-02-22     0       <NA>         0       365    1   1   1
## 3 2003-04-30 2005-11-21     1 2006-01-11         0       365    1   1   1
## 4 2006-03-25 2006-05-05     1 2006-05-07         1        41    1   1   1
## 5 2004-09-01 2007-11-13     0       <NA>         0       365    1   1   0
## 6 2003-12-02 2008-02-28     0       <NA>         0       365    1   1   1
##   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
## 6   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##      male      age aids cd4baseline    logvl  weight hemoglobin
## 999     0 31.00000    0         102       NA 61.6896         11
## 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 death.1yr
## 999  3TC,AZT,NVP 2003-05-22 2008-03-07     0       <NA>         0
## 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
## 999        365    1   1   1   0   1   0   0   0   0   0   0   0   0   0
## 1000       365    1   1   0   0   1   1   0   0   0   0   0   0   0   0
## 1001        35    0   1   1   0   1   0   0   0   0   0   0   0   0   0
## 1002       365    1   1   1   0   1   0   0   0   0   0   0   0   0   0
## 1003        75    0   1   0   1   0   0   0   1   0   0   0   0   0   0
## 1004       365    1   1   0   0   1   1   0   0   0   0   0   0   0   0
##      DDC NFV T20 ATV FPV
## 999    0   0   0   0   0
## 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

14 points

Use the following code to generate data for patients with repeated measures of A1C (a test for levels of blood glucose).

```
genData <- function(n) {
    if(exists(".Random.seed", envir = .GlobalEnv)) {
        save.seed <- get(".Random.seed", envir= .GlobalEnv)
        on.exit(assign(".Random.seed", save.seed, envir = .GlobalEnv))
    } else {
        on.exit(rm(".Random.seed", envir = .GlobalEnv))
    }
    set.seed(n)
    subj <- ceiling(n / 10)
    id <- sample(subj, n, replace=TRUE)
    times <- as.integer(difftime(as.POSIXct("2005-01-01"), as.POSIXct("2000-01-01"), units='secs'))
    dt <- as.POSIXct(sample(times, n), origin='2000-01-01')
    mu <- runif(subj, 4, 10)
    a1c <- unsplit(mapply(rnorm, tabulate(id), mu, SIMPLIFY=FALSE), id)
    data.frame(id, dt, a1c)
}
x <- genData(500)
```

Perform the following manipulations: (2 points each)

1. Order the data set by `id` and `dt`.

```
patient <- as.data.frame(x)
patient_sort <- patient[order(patient[,'id'], patient[,'dt']),]
```

2. For each `id`, determine if there is more than a one year gap in between observations. Add a new row at the one year mark, with the `a1c` value set to missing. A two year gap would require two new rows, and so forth.

```
y <- data.frame()
for (i in unique(patient_sort$id)) {
  temp <- patient_sort[patient_sort[,1] == i, c(1,2,3)]
  for (j in seq(nrow(temp))) {
```

```
    temprow <- matrix(c(NA, NA, ""), nrow=1, ncol=length(patient_sort))
    newrow <- data.frame(temprow)
    colnames(newrow) <- colnames(patient_sort)
    if (is.na(temp$dt[j+1] - temp$dt[j])) {
      temp = temp
    } else if (temp$dt[j+1] - temp$dt[j] >= 365) {
      temp[seq(j+1, nrow(temp)+1),] <- temp[seq(j, nrow(temp)),]
      temp[j+1,] <- newrow
    } else {
      temp = temp
    }
  }

  y <- rbind.data.frame(y, temp)
  y
}

for (k in 1:553) {
  if (is.na(y$id[k] == y$id[k+2])) {
    y$id[k+1] = y$id[k+1]
  } else if (y$id[k] == y$id[k+2]) {
    y$id[k+1] = y$id[k]
  } else {
    y$id[k+1] = y$id[k+1]
  }
}
```

3. Create a new column `visit`. For each `id`, add the visit number. This should be 1 to `n` where `n` is the number of observations for an individual. This should include the observations created with missing a1c values.

```
y$visit = rep(0, length(nrow(y)))
y1 <- data.frame()
for (i in unique(y$id)) {
  temp <- y[y[,1] == i, c(1,2,3,4)]
  for (j in seq(nrow(temp))) {
    temp$visit[j] = j
  }
  y1 <- rbind.data.frame(y1, temp)
}
```

4. For each `id`, replace missing values with the mean `a1c` value for that individual.

```
y1$a1c[y1$a1c == 1] <- NA
y2 <- data.frame()
for (i in unique(y1$id)) {
  temp <- y1[y1[,1] == i, c(1,2,3,4)]
  for (j in seq(nrow(temp))) {
    if (is.na(temp$a1c[j])) {
      temp$a1c[j] = mean(temp$a1c, na.rm = TRUE)
    } else {
      temp$a1c[j] = temp$a1c[j]
    }
  }
  y2 <- rbind.data.frame(y2, temp)
```

```
}
```

5. Print mean `a1c` for each `id`.

```
for (i in unique(y2$id)) {
  temp <- y2[y2[,1] == i, c(1,2,3,4)]
  print(paste("The mean a1c for ID", i, "is", mean(temp$a1c, na.rm = TRUE)))
}
```

```
## [1] "The mean a1c for ID 1 is 4.0633722154334"
## [1] "The mean a1c for ID 2 is 7.54464252139816"
## [1] "The mean a1c for ID 3 is 6.75763969016784"
## [1] "The mean a1c for ID 4 is 3.89212739139609"
## [1] "The mean a1c for ID 5 is 9.51231068111361"
## [1] "The mean a1c for ID 6 is 7.55596508893044"
## [1] "The mean a1c for ID 7 is 9.16168557475124"
## [1] "The mean a1c for ID 8 is 7.18906443342637"
## [1] "The mean a1c for ID 9 is 9.28387318771948"
## [1] "The mean a1c for ID 10 is 7.97521696324126"
## [1] "The mean a1c for ID 11 is 6.91756203273274"
## [1] "The mean a1c for ID 12 is 7.0340208877463"
## [1] "The mean a1c for ID 13 is 9.14528157392063"
## [1] "The mean a1c for ID 14 is 6.62375644624112"
## [1] "The mean a1c for ID 15 is 8.01240569465381"
## [1] "The mean a1c for ID 16 is 4.22215766516924"
## [1] "The mean a1c for ID 17 is 3.99603367716249"
## [1] "The mean a1c for ID 18 is 9.16487326421613"
## [1] "The mean a1c for ID 19 is 5.50721007909014"
## [1] "The mean a1c for ID 20 is 3.72667487583177"
## [1] "The mean a1c for ID 21 is 8.14093868907524"
## [1] "The mean a1c for ID 22 is 5.63750143653249"
## [1] "The mean a1c for ID 23 is 7.36688886897263"
## [1] "The mean a1c for ID 24 is 7.43931572282878"
## [1] "The mean a1c for ID 25 is 6.87713482527697"
## [1] "The mean a1c for ID 26 is 6.55675863918465"
## [1] "The mean a1c for ID 27 is 4.92645727067693"
## [1] "The mean a1c for ID 28 is 7.43391725083977"
## [1] "The mean a1c for ID 29 is 4.50808596846108"
## [1] "The mean a1c for ID 30 is 6.04557747595203"
## [1] "The mean a1c for ID 31 is 7.11658561119926"
## [1] "The mean a1c for ID 32 is 6.56879125101402"
## [1] "The mean a1c for ID 33 is 6.49406946855475"
## [1] "The mean a1c for ID 34 is 6.76861498300695"
## [1] "The mean a1c for ID 35 is 8.47669959224807"
## [1] "The mean a1c for ID 36 is 9.60440982770843"
## [1] "The mean a1c for ID 37 is 9.60625272268161"
## [1] "The mean a1c for ID 38 is 5.35597946973571"
## [1] "The mean a1c for ID 39 is 6.9170128007528"
## [1] "The mean a1c for ID 40 is 9.53013621503612"
## [1] "The mean a1c for ID 41 is 9.80242433699468"
## [1] "The mean a1c for ID 42 is 3.89176951483657"
## [1] "The mean a1c for ID 43 is 6.09584879350571"
## [1] "The mean a1c for ID 44 is 9.09166981531122"
## [1] "The mean a1c for ID 45 is 6.73720440708592"
## [1] "The mean a1c for ID 46 is 9.62176304469632"
```

```
## [1] "The mean a1c for ID 47 is 9.23148863726925"
## [1] "The mean a1c for ID 48 is 6.40459996804318"
## [1] "The mean a1c for ID 49 is 6.09607633914946"
## [1] "The mean a1c for ID 50 is 8.96231874982749"
```

6. Print total number of visits for each `id`.

```
for (i in unique(y2$id)) {
  temp <- y2[y2[,1] == i, c(1,2,3,4)]
  print(paste("The total number of visits for ID", i, "is", nrow(temp)))
}
```

```
## [1] "The total number of visits for ID 1 is 11"
## [1] "The total number of visits for ID 2 is 20"
## [1] "The total number of visits for ID 3 is 14"
## [1] "The total number of visits for ID 4 is 12"
## [1] "The total number of visits for ID 5 is 14"
## [1] "The total number of visits for ID 6 is 10"
## [1] "The total number of visits for ID 7 is 9"
## [1] "The total number of visits for ID 8 is 12"
## [1] "The total number of visits for ID 9 is 11"
## [1] "The total number of visits for ID 10 is 12"
## [1] "The total number of visits for ID 11 is 10"
## [1] "The total number of visits for ID 12 is 10"
## [1] "The total number of visits for ID 13 is 8"
## [1] "The total number of visits for ID 14 is 12"
## [1] "The total number of visits for ID 15 is 7"
## [1] "The total number of visits for ID 16 is 8"
## [1] "The total number of visits for ID 17 is 12"
## [1] "The total number of visits for ID 18 is 10"
## [1] "The total number of visits for ID 19 is 10"
## [1] "The total number of visits for ID 20 is 9"
## [1] "The total number of visits for ID 21 is 10"
## [1] "The total number of visits for ID 22 is 8"
## [1] "The total number of visits for ID 23 is 8"
## [1] "The total number of visits for ID 24 is 15"
## [1] "The total number of visits for ID 25 is 12"
## [1] "The total number of visits for ID 26 is 14"
## [1] "The total number of visits for ID 27 is 11"
## [1] "The total number of visits for ID 28 is 14"
## [1] "The total number of visits for ID 29 is 10"
## [1] "The total number of visits for ID 30 is 7"
## [1] "The total number of visits for ID 31 is 11"
## [1] "The total number of visits for ID 32 is 5"
## [1] "The total number of visits for ID 33 is 8"
## [1] "The total number of visits for ID 34 is 12"
## [1] "The total number of visits for ID 35 is 11"
## [1] "The total number of visits for ID 36 is 9"
## [1] "The total number of visits for ID 37 is 17"
## [1] "The total number of visits for ID 38 is 15"
## [1] "The total number of visits for ID 39 is 8"
## [1] "The total number of visits for ID 40 is 7"
## [1] "The total number of visits for ID 41 is 17"
## [1] "The total number of visits for ID 42 is 14"
## [1] "The total number of visits for ID 43 is 11"
```

```
## [1] "The total number of visits for ID 44 is 11"
## [1] "The total number of visits for ID 45 is 14"
## [1] "The total number of visits for ID 46 is 9"
## [1] "The total number of visits for ID 47 is 12"
## [1] "The total number of visits for ID 48 is 11"
## [1] "The total number of visits for ID 49 is 12"
## [1] "The total number of visits for ID 50 is 10"
```

7. Print the observations for `id = 15`.

```
i = 15
temp <- y2[y2[,1] == i, c(1,2,3,4)]
print(temp)
```

```
##     id                  dt      a1c visit
## 111 15 2000-04-30 00:34:50 7.527105     1
## 406 15 2001-01-17 21:11:02 5.898371     2
## 306 15 2001-04-25 06:23:05 8.566593     3
## 484 15                <NA> 8.012406     4
## 263 15 2003-06-06 14:06:00 9.133769     5
## 62  15                <NA> 8.012406     6
## 71  15 2004-08-20 17:47:11 8.936190     7
```

JC Grading -3 Missing an imputed year. There should be 8 rows. I'd be happy to work through this together during office hours if you'd like.

QUESTION 3

10 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 <- "https://raw.githubusercontent.com/fonnesbeck/Bios6301/master/datasets/addr.txt"
addr <- readLines(addr)
addr_line <- lapply(addr, function(a) {unlist(strsplit(a, split = "[ ]{2,}"))})
addr_df <- do.call(rbind.data.frame, addr_line)
colnames(addr_df) <- c("Last", "First", "Address", "City", "State", "ZipCode")
addr_df[] <- lapply(addr_df, as.character)

addr_df$StreetNo <- sapply(addr_df$Address, function(n) return(strsplit(n, " ")[[1]][1]))
addr_df$StreetName <- gsub("[0-9]{1,} ", "", addr_df$Address)
addr_df$Address <- NULL
addr_df <- addr_df[,c("Last", "First", "StreetNo", "StreetName", "City", "State", "ZipCode")]
print(addr_df)
```

```
##           Last      First 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
##        ZipCode
## 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(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.

```
There's a problem with 'death' in the function. The error says it cannot be found when using the tryCat
```

5 bonus points

Create a working function.

```
myfun_AP <- function(dat, response) {
  dat$resp = dat[,response]
  coef(summary(glm(resp ~ ., data=dat, family=binomial(logit))))
}

myfun_AP(haart_df, "death")
```

```
## Warning: glm.fit: algorithm did not converge
```

```
##                   Estimate  Std. Error       z value  Pr(>|z|)
## (Intercept) -2.656607e+01 115935.1524 -2.291459e-04 0.9998172
## death        5.313213e+01  69028.2910  7.697153e-04 0.9993859
## weight      -1.610484e-15   1939.0567 -8.305501e-19 1.0000000
## hemoglobin   1.697890e-14   9774.8170  1.737004e-18 1.0000000
## cd4baseline  4.076548e-17    184.0846  2.214497e-19 1.0000000
```

JC Grading +0

Coefficients table should match to output from start of question.