ETM538 HW 4

Jordan Hilton
Mengyu Li
Peter Boss
Andey Nunes, MS

February 13, 2019

Initial Setup

Here's a modified version of the provided data load:

```
mem_claims              <- read.csv("Claims_Y1.csv")
mem_days                <- read.csv("DayInHospital_Y2.csv")
mem_info                <- read.csv("Members_Y1.csv")
risk_model              <- read.csv("risk_model_1.csv")
```

Here's the processing code:

```
colnames(mem_days) <- c("MemberID", "Days")
mem_to_risk <- merge(mem_days, risk_model, by = "Days")
claims_to_risk <- merge(mem_claims, mem_to_risk, by = "MemberID")
```

Here's the code to calculate the a priori probabilities:

```
n_claims <- length(claims_to_risk[,1])     # note that we have to pick a column.

risks <- as.data.frame(as.character(claims_to_risk$RiskLevel))

riskl <- as.list(risks)

risk_count <- aggregate(risks, riskl, FUN = length)

colnames(risk_count) <- c("RiskLevel", "RiskCount")

a_priori <- risk_count

a_priori$Total <- n_claims

a_priori$Prob <- a_priori$RiskCount / n_claims

colnames(a_priori) <- c("RiskLevel", "RiskCount","Total", "Prob") ##Modified this from original code

write.csv(a_priori, file = "out_a_priori.csv", row.names = FALSE)
```

The calculation for the condition on Charlson index:

```
on_charlson <- data.frame(as.character(claims_to_risk$RiskLevel),
                          as.character(claims_to_risk$CharlsonIndex))

colnames(on_charlson) <- c("RiskLevel", "Charlson")

df_char <- as.data.frame(as.character(on_charlson$Charlson))
```

```
colnames(df_char) <- c("Charlson")

l_char <- on_charlson$Charlson

l_risk <- on_charlson$RiskLevel

count_char <- aggregate(df_char, by = list(l_char, l_risk), FUN=length)

colnames(count_char) <- c("Charlson", "RiskLevel", "Count")

# check the total to make sure everything is present and accounted for.

n_chars <- sum(count_char$Count)

n_missing <- n_claims - n_chars

print(paste("A Posteriori Charlson -- ", toString(n_missing), " are missing."))
```

```
## [1] "A Posteriori Charlson --  0  are missing."
```

```
post_char <- merge(count_char, risk_count, by = "RiskLevel")

post_char$Prob <- post_char$Count / post_char$RiskCount

post_char$Label <- paste(post_char$Charlson, post_char$RiskLevel, sep = "|")

# reorder the columns

post_char <- post_char[c("Label", "Charlson", "RiskLevel", "Count", "RiskCount", "Prob")]
```

The calculation for the condition on length of (first) stay:

```
# extract length of stay as a vector

new_stay <- as.character(claims_to_risk$LengthOfStay)

# assign default value to missing columns

new_stay[new_stay == ''] <- '0 days'

on_stay <- data.frame(as.character(claims_to_risk$RiskLevel),
                      as.character(new_stay))

colnames(on_stay) <- c("RiskLevel", "Stay")

###for question 4, we're going to add in 1 count for every combination of stay and risk level. this firs
addon <- expand.grid(levels(on_stay$RiskLevel),levels(on_stay$Stay))
colnames(addon) <- c("RiskLevel", "Stay")

##and this second line adds that to the bottom of df_stay
on_stay <- rbind(on_stay, addon)

df_stay <- as.data.frame(as.character(on_stay$Stay))

colnames(df_stay) <- c("Stay")
```

```
l_stay <- on_stay$Stay

l_risk <- on_stay$RiskLevel

count_stay <- aggregate(df_stay, by = list(l_stay, l_risk), FUN = length)

colnames(count_stay) <- c("Stay", "RiskLevel", "Count")

# check the total to make sure everything is present and accounted for.

n_stays <- sum(count_stay$Count)

n_missing <- n_claims - n_stays

print(paste("A Posteriori stay -- ", toString(n_missing), " are missing."))
```

```
## [1] "A Posteriori stay --  -65  are missing."
```

```
post_stay <- merge(count_stay, risk_count, by = "RiskLevel")

post_stay$Prob <- post_stay$Count / post_stay$RiskCount

post_stay$Label <- paste(post_stay$Stay, post_stay$RiskLevel, sep = "|")

# reorder the columns

post_stay <- post_stay[c("Label", "Stay", "RiskLevel", "Count", "RiskCount", "Prob")]
```

The calculation for the condition on primary condition group:

```
on_pcg <- data.frame(as.character(claims_to_risk$RiskLevel),
                     as.character(claims_to_risk$PrimaryConditionGroup))

colnames(on_pcg) <- c("RiskLevel", "pcg")

df_pcg <- as.data.frame(as.character(on_pcg$pcg))

colnames(df_pcg) <- c("pcg")

l_pcg <- on_pcg$pcg

l_risk <- on_pcg$RiskLevel

count_pcg <- aggregate(df_pcg, by = list(l_pcg, l_risk), FUN = length)

colnames(count_pcg) <- c("pcg", "RiskLevel", "Count")

# check the total to make sure everything is present and accounted for.

n_pcgs <- sum(count_pcg$Count)

n_missing <- n_claims - n_pcgs

print(paste("A Posteriori pcg -- ", toString(n_missing), " are missing."))
```

```
## [1] "A Posteriori pcg --  0  are missing."
```

```
post_pcg <- merge(count_pcg, risk_count, by = "RiskLevel")

post_pcg$Prob <- post_pcg$Count / post_pcg$RiskCount

post_pcg$Label <- paste(post_pcg$pcg, post_pcg$RiskLevel, sep = "|")

# reorder the columns

post_pcg <- post_pcg[c("Label", "pcg", "RiskLevel", "Count", "RiskCount", "Prob")]
```

Writing out to csvs:

```
write.csv(post_char, file = "out_charlson.csv")
write.csv(post_stay, file = "out_stay.csv")
write.csv(post_pcg, file = "out_pcg.csv")
```

Question responses:

1. Dataset explanation

Explain how the R data pipeline works, by describing the role of each of the following R data sets:

- mem_to_risk - Combines the "Days in Hospital Y2" and "Risk Level" tables to provide information about the risk level of each member ID.
- claims_to_risk - Combines the "Claims Y1" table with the mem-to-risk table, so that now we have a single table with information about the claims in Y1 and the risk level in Y2 for each member.
- risk_count - A simple count of how many rows there are in claims-to-risk for each different risk level.
- a_priori - Contains the a priori probability for each risk level calculated from claims-to-risk.
- on_charlson - A subset of the claims-to-risk table containing only the Charlson index and risk level fields
- count_char - Counts the number of claims for each possible combination of risk level and Charlson index in claims-to-risk.
- post_char - Gets the total number of risk counts for each risk level, then uses the count from count_char to calculate the a posteriori probability

2. Spreadsheet operation

Charlson index seems to be the most influential variable- when you hold other variables constant and change Charlson index, you get the biggest change in predicted outcome.

3. Label column

The label column is produced in the R code by this line, which pastes together the risk level and the independent variable we're looking at:

```
post_pcg$Label <- paste(post_pcg$pcg, post_pcg$RiskLevel, sep = "|")
```

It's used by the Excel sheet as the key for the vlookup function to reference the chosen combination of independent variable and risk level for the naive Bayes calculation.

4. Missing values for long hospital stays

This happens because there are 0 occurences in the data of some combinations of length of stay and risk level. To fix this, we'll add a little seed to the data in the middle of the probability calculation for stay that adds 1 row for every possible combination. We did this in the code above at line 112 as follows:

```
###for question 4, we're going to add in 1 count for every combination of stay and risk level. this firs
addon <- expand.grid(levels(on_stay$RiskLevel),levels(on_stay$Stay))
colnames(addon) <- c("RiskLevel", "Stay")

##and this second line adds that to the bottom of df_stay
on_stay <- rbind(on_stay, addon)
```

As a minor note, we also had to adjust the formula in the excel model to include the additional rows this generated

5. Unlikely B

First, here are some example combinations that predict high risk levels for A, C, and D. We couldn't find any combinations where A or C had the highest probability, for reasons similar to B, but here are some with high probabilities for A and C.

- A Charlson:5+ Stay:4-8 weeks PCG: COPD predicts for A at 9.9%
- C Charlson: 3-4 Stay:12-26 weeks PCG: GIOBSENT predicts for C at 14.8%
- D Charlson:3-4 Stay:4- 8 weeks PCG:SEPSIS predicts for D at 67.2%

Looking at the probability tables, what's happening is just that because that the counts for risk level B are lower than for other categories- presumably because the odds of spending exactly 2 days in the hospital are lower than the odds of spending either 1 day or between 3 and 5 days.

6 Changing Risk Buckets

After combing risk level B & C, under the condition Charlson:3-4 Stay:4- 8 weeks PCG:SEPSIS prediction, the result of prediction of B increased from 2% to 8%. We're including a modifed excel model showing the changed risk buckets.

```
# these first 3 are already loaded and do not change, no need to recall them
#mem_claims              <- read.csv("Claims_Y1.csv")
#mem_days                <- read.csv("DayInHospital_Y2.csv")
#mem_info                <- read.csv("Members_Y1.csv")

# to implement the balanced bucket option, comment out reading in the
# new csv risk model (line 267) and uncomment the last 2 lines in this
# code chunk that reassigns the bucket count (line 272), to go back
# reverse that... comment out the last 2 lines and uncomment out reading in
# "risk_model_2.csv", but make sure it is in the working directory

# this risk model combines B & C buckets into a single bucket leaving
# 4 buckets instead of 5
# risk_model              <- read.csv("risk_model_2.csv") # added for Question 6


# use this a different interpretation of the model where the same
# number of buckets is retained, but buckets "B" and "C" are balanced
risk_model[4,2] <- "B"
write.csv(risk_model, "alt_risk_model.csv")
```

```
#colnames(mem_days) <- c("MemberID", "Days")
Q6_mem_to_risk <- merge(mem_days, risk_model, by = "Days")
Q6_claims_to_risk <- merge(mem_claims, mem_to_risk, by = "MemberID")
```

```r
# again nothing changes in this code chunk

Q6_n_claims <- length(Q6_claims_to_risk[,1])     # note that we have to pick a column.

Q6_risks <- as.data.frame(as.character(Q6_claims_to_risk$RiskLevel))

Q6_riskl <- as.list(Q6_risks)

Q6_risk_count <- aggregate(Q6_risks, Q6_riskl, FUN = length)

colnames(Q6_risk_count) <- c("RiskLevel", "RiskCount")

Q6_a_priori <- Q6_risk_count

Q6_a_priori$Total <- Q6_n_claims

Q6_a_priori$Prob <- Q6_a_priori$RiskCount / Q6_n_claims

colnames(Q6_a_priori) <- c("RiskLevel", "RiskCount","Total", "Prob") ##Modified this from original code

write.csv(Q6_a_priori, file = "out_Q6_a_priori.csv", row.names = FALSE)
```

```r
# now things are different because the

Q6_on_charlson <- data.frame(
  as.character(Q6_claims_to_risk$RiskLevel),
  as.character(Q6_claims_to_risk$CharlsonIndex))

colnames(Q6_on_charlson) <- c("RiskLevel", "Charlson")

Q6_df_char <- as.data.frame(as.character(Q6_on_charlson$Charlson))

colnames(Q6_df_char) <- c("Charlson")

Q6_l_char <- Q6_on_charlson$Charlson

Q6_l_risk <- Q6_on_charlson$RiskLevel

Q6_count_char <- aggregate(Q6_df_char, by = list(Q6_l_char, Q6_l_risk), FUN = length)

colnames(Q6_count_char) <- c("Charlson", "RiskLevel", "Count")

# check the total to make sure everything is present and accounted for.

Q6_n_chars <- sum(Q6_count_char$Count)

Q6_n_missing <- Q6_n_claims - Q6_n_chars

print(paste("A Posteriori Charlson -- ", toString(Q6_n_missing), " are missing."))
```

```
## [1] "A Posteriori Charlson --  0  are missing."
```

```r
Q6_post_char <- merge(Q6_count_char, Q6_risk_count, by = "RiskLevel")
```

```r
Q6_post_char$Prob <- Q6_post_char$Count / Q6_post_char$RiskCount

Q6_post_char$Label <- paste(Q6_post_char$Charlson, Q6_post_char$RiskLevel, sep = "|")

# reorder the columns

Q6_post_char <- Q6_post_char[c("Label", "Charlson", "RiskLevel", "Count", "RiskCount", "Prob")]
```

```r
# extract length of stay as a vector
Q6_new_stay <- as.character(Q6_claims_to_risk$LengthOfStay)

# assign default value to missing columns
Q6_new_stay[Q6_new_stay == ''] <- '0 days'
Q6_on_stay <- data.frame(as.character(Q6_claims_to_risk$RiskLevel),
                         as.character(Q6_new_stay))
colnames(Q6_on_stay) <- c("RiskLevel", "Stay")

Q6_addon <- expand.grid(levels(Q6_on_stay$RiskLevel),levels(Q6_on_stay$Stay))
colnames(Q6_addon) <- c("RiskLevel", "Stay")

Q6_on_stay <- rbind(Q6_on_stay, Q6_addon)

Q6_df_stay <- as.data.frame(as.character(Q6_on_stay$Stay))

colnames(Q6_df_stay) <- c("Stay")

Q6_l_stay <- Q6_on_stay$Stay

Q6_l_risk <- Q6_on_stay$RiskLevel

Q6_count_stay <- aggregate(Q6_df_stay, by = list(Q6_l_stay, Q6_l_risk), FUN = length)

colnames(Q6_count_stay) <- c("Stay", "RiskLevel", "Count")

# check the total to make sure everything is present and accounted for.

Q6_n_stays <- sum(Q6_count_stay$Count)

Q6_n_missing <- Q6_n_claims - Q6_n_stays

print(paste("A Posteriori stay -- ", toString(Q6_n_missing), " are missing."))
```

```
## [1] "A Posteriori stay --  -65  are missing."
```

```r
Q6_post_stay <- merge(Q6_count_stay, Q6_risk_count, by = "RiskLevel")

Q6_post_stay$Prob <- Q6_post_stay$Count / Q6_post_stay$RiskCount

Q6_post_stay$Label <- paste(Q6_post_stay$Stay, Q6_post_stay$RiskLevel, sep = "|")

# reorder the columns

Q6_post_stay <- Q6_post_stay[c("Label", "Stay", "RiskLevel", "Count", "RiskCount", "Prob")]
```

```r
Q6_on_pcg <- data.frame(
  as.character(Q6_claims_to_risk$RiskLevel),
  as.character(Q6_claims_to_risk$PrimaryConditionGroup))

colnames(Q6_on_pcg) <- c("RiskLevel", "pcg")

Q6_df_pcg <- as.data.frame(as.character(Q6_on_pcg$pcg))

colnames(Q6_df_pcg) <- c("pcg")

Q6_l_pcg <- Q6_on_pcg$pcg

Q6_l_risk <- Q6_on_pcg$RiskLevel

Q6_count_pcg <- aggregate(Q6_df_pcg, by = list(Q6_l_pcg, Q6_l_risk), FUN = length)

colnames(Q6_count_pcg) <- c("pcg", "RiskLevel", "Count")

# check the total to make sure everything is present and accounted for.

Q6_n_pcgs <- sum(Q6_count_pcg$Count)

Q6_n_missing <- Q6_n_claims - Q6_n_pcgs

print(paste("A Posteriori pcg -- ", toString(Q6_n_missing), " are missing."))
```

```
## [1] "A Posteriori pcg --  0  are missing."
```

```r
Q6_post_pcg <- merge(Q6_count_pcg, Q6_risk_count, by = "RiskLevel")

Q6_post_pcg$Prob <- Q6_post_pcg$Count / Q6_post_pcg$RiskCount

Q6_post_pcg$Label <- paste(Q6_post_pcg$pcg, Q6_post_pcg$RiskLevel, sep = "|")

# reorder the columns

Q6_post_pcg <- Q6_post_pcg[c("Label", "pcg", "RiskLevel", "Count", "RiskCount", "Prob")]
```

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
val_char <- '3-4'
val_stay <- '4- 8 weeks'
val_pcg <- 'SEPSIS'

prob_a <- Q6_a_priori %>%
```

```
  select(RiskLevel, prob.a = Prob)

prob_c <- Q6_post_char %>%
  filter(Charlson == val_char) %>%
  select(RiskLevel, prob.c = Prob)

prob_s <- Q6_post_stay %>%
  filter(Stay == val_stay) %>%
  select(RiskLevel, prob.s = Prob)

prob_p <- Q6_post_pcg %>%
  filter(pcg == val_pcg) %>%
  select(RiskLevel, prob.p = Prob)

prob <- inner_join(prob_c, prob_s, by = 'RiskLevel') %>%
  inner_join(prob_p, by = 'RiskLevel') %>%
  inner_join(prob_a, by = 'RiskLevel') %>%
  mutate(prob = prob.c * prob.s * prob.p * prob.a * 100)

prob$pred <- prob$prob/sum(prob$prob)
prob
```

```
##   RiskLevel     prob.c       prob.s       prob.p     prob.a         prob
## 1         A 0.01706620 0.0004540896 0.0002270448 0.08198000 1.442441e-08
## 2         B 0.01802083 0.0010600486 0.0001247116 0.02487490 5.926097e-09
## 3         C 0.02163655 0.0001150880 0.0003740361 0.05390984 5.021112e-09
## 4         D 0.03673945 0.0014903130 0.0003671786 0.07181413 1.443768e-07
## 5         O 0.01442313 0.0002748802 0.0001536095 0.76742112 4.673634e-08
##         pred
## 1 0.06663014
## 2 0.02737420
## 3 0.02319384
## 4 0.66691437
## 5 0.21588745
```

```
write.csv(Q6_post_char, file = "Q6_out_charlson.csv")
write.csv(Q6_post_stay, file = "Q6_out_stay.csv")
write.csv(Q6_post_pcg, file = "Q6_out_pcg.csv")
```

7. Adding a variable

To add another variable to the original risk model problem, we can recycle the original 3 csv files generated in the first part of the homework, and create a new csv for the new variable. Then we will add this as another sheet in the Excel tool and modify the front sheet to calculate the probability and likelihood based on the four variables.

Here is an additional file generated on the `dsfs` days since first stay variable. We picked `dsfs` since it shouldn't correlate too strongly with any of our 3 current variables, and the values looked nice.

We're including in the submission the excel model for this added variable, and another excel model for adding the variable "specialty".

```
on_dsfs <- data.frame(as.character(claims_to_risk$RiskLevel),
                      as.character(claims_to_risk$dsfs))

colnames(on_dsfs) <- c("RiskLevel", "dsfs")
```

```r
df_dsfs <- as.data.frame(as.character(on_dsfs$dsfs))

colnames(df_dsfs) <- c("dsfs")

l_dsfs <- on_dsfs$dsfs

l_risk <- on_dsfs$RiskLevel

count_dsfs <- aggregate(df_dsfs, by = list(l_dsfs, l_risk), FUN = length)

colnames(count_dsfs) <- c("dsfs", "RiskLevel", "Count")

# check the total to make sure everything is present and accounted for.

n_dsfs <- sum(count_dsfs$Count)

n_missing <- n_claims - n_dsfs

print(paste("A Posteriori dsfs -- ", toString(n_missing), " are missing."))
```

```
## [1] "A Posteriori dsfs --  0  are missing."
```

```r
post_dsfs <- merge(count_dsfs, risk_count, by = "RiskLevel")

post_dsfs$Prob <- post_dsfs$Count / post_dsfs$RiskCount

post_dsfs$Label <- paste(post_dsfs$dsfs, post_char$RiskLevel, sep = "|")

# reorder the columns

post_dsfs <- post_dsfs[c("Label", "dsfs", "RiskLevel", "Count", "RiskCount", "Prob")]

write.csv(post_dsfs, file = "out_dsfs.csv")
```