EDA Exercise 4

Aaron Niskin

September 2, 2016

Read the following paper:

http://web.mit.edu/tdqm/www/tdqmpub/PipinoLeeWangCACMApr02.pdf

Consider the data from Exercise 3,

https://archive.ics.uci.edu/ml/machine-learning-databases/00296/

Consider the following prompt: To what extent is it possible to predict discharge status?

Prepare an R Markdown document which documents the use of R tools/code to give a preliminary assessment of the quality of the data for a colleague asked to work on the same prompt.

Let's begin by trying to get a good predictor of when a patient will be discharged to home (so as to get some experience with a simpler problem). To do this, we'll start by exploring the data a little bit. First, let's import the data along with some key tables (that store the values associated to each key of their respective fields). The latter are really just to save us some time. Hopefully it will actually do so.

```r
diab_data <- read.csv("../a3/dataset_diabetes/diabetic_data.csv")
dischargeDisp_keys <- read.csv("../a3/dataset_diabetes/dischargeDisposition.csv")
admissionType_keys <- read.csv("../a3/dataset_diabetes/admissionType.csv")
admissionSource_keys <- read.csv("../a3/dataset_diabetes/admissionSource.csv")
```

Just to show what I mean by the key tables,

```r
dischargeDisp_keys[dischargeDisp_keys$discharge_disposition_id == 1,]
```

```
##   discharge_disposition_id        description
## 1                        1 Discharged to home
```

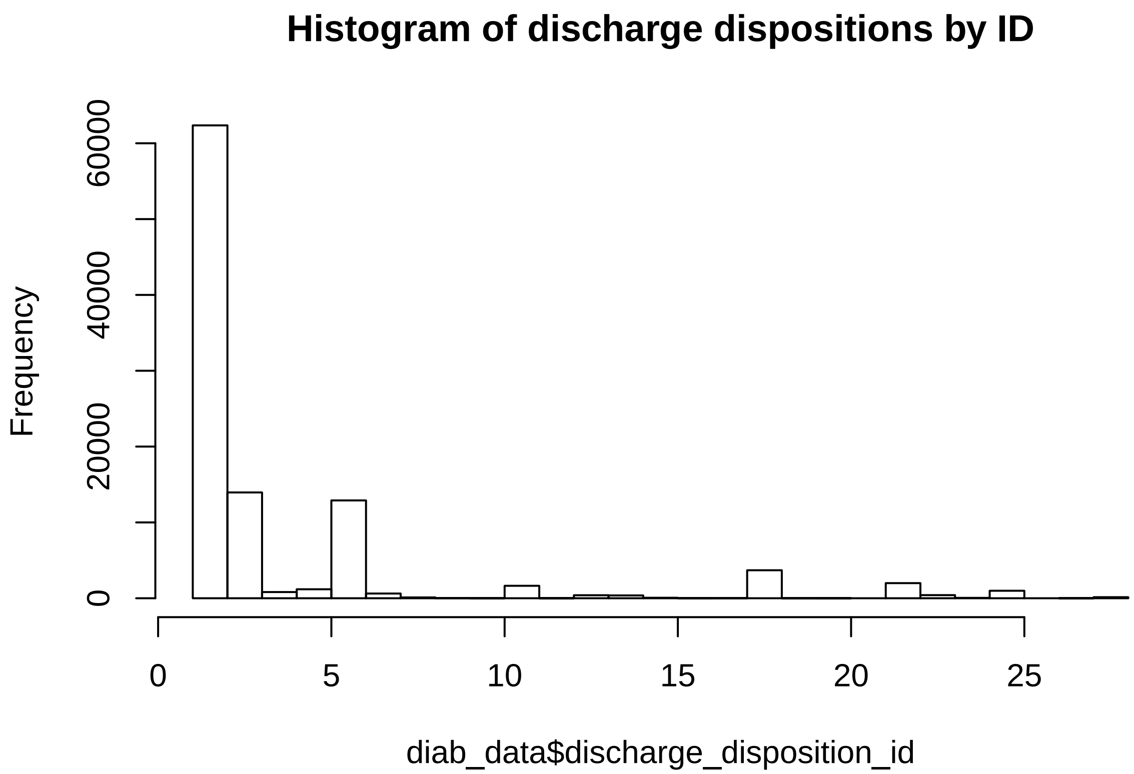
and

```r
dischargeDisp_keys[dischargeDisp_keys$discharge_disposition_id == 3,]
```

```
##   discharge_disposition_id                   description
## 3                        3 Discharged/transferred to SNF
```

The last three CSV files were created by writing the different sections of the "IDs_mapping.csv" file included with the data. They were not transcribed, but copied directly (so there should not be any transcription errors in the indices nor in the field names).

```r
hist(diab_data$discharge_disposition_id,
     main="Histogram of discharge dispositions by ID",
     breaks = 30)
```



```
num_discharge_1 <- length(diab_data[diab_data$discharge_disposition_id == 1, "discharge_disposition_id"
num_discharge_total <- length(diab_data[,"discharge_disposition_id"])
prob_discharge_1 <- num_discharge_1 / num_discharge_total
```

So right off the bat, we can estimate that there is about a 59% probability that someone coming in to the hospital with any sort of diabetes diagnosis will be discharged to home. First, let's make a general purpose function that will take a dataset, a desired attribute value, a dependent attribute and an independent attribute.

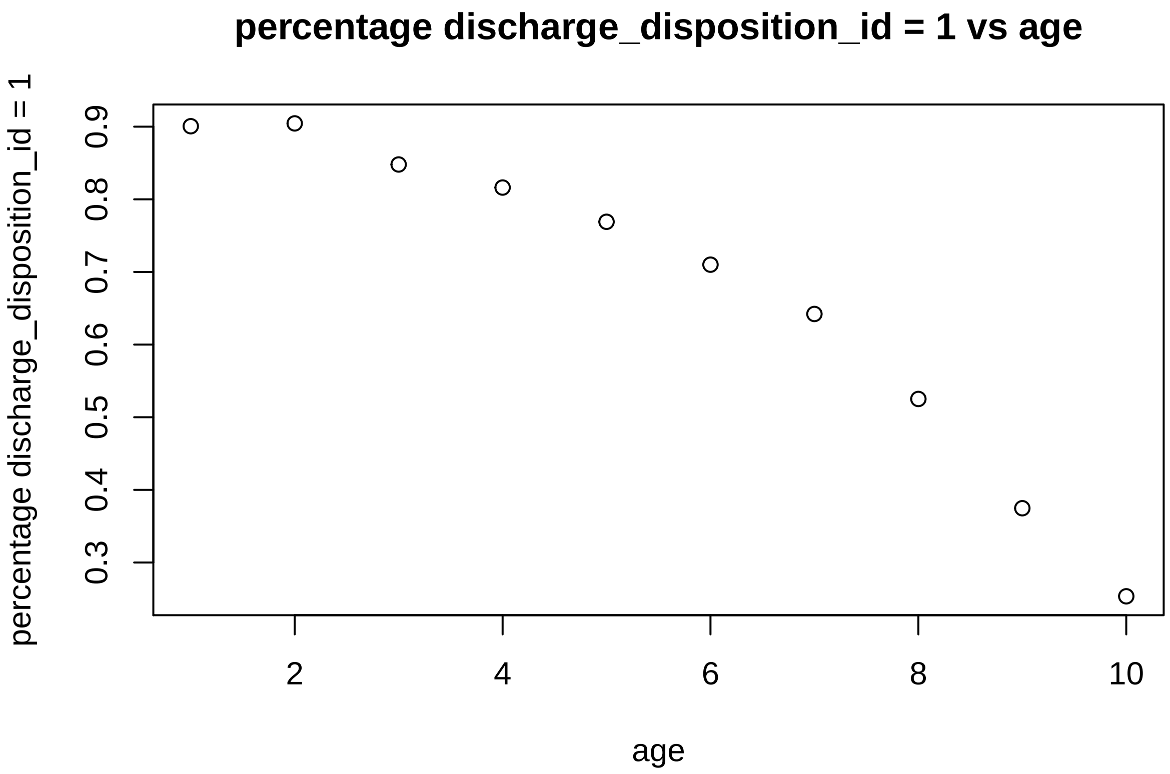
```
getProbsAttr1_givenAttr2 <- function(dats, id, dependent, independent) {
  indsVec <- unique(dats[,independent])
  retVal <- sapply(indsVec, function(t) {
    tmp <- dats[dats[,independent] == t, dependent]
    c(independent=t, dependent=as.numeric(length(tmp[which(tmp == id)]) / length(tmp)))
    })
  retVal <- as.data.frame(t(retVal))
  plot(retVal$independent, retVal$dependent, xlab=independent,
       ylab=paste("percentage", dependent, "=", id),
       main=paste("percentage", dependent, "=", id, "vs", independent))
  retVal
}
```

We can get a quick picture of how the probability one will be discharged home depends on each individual variable by executing the command below: (note, I did not execute it here because I don't want to include 20 pages of mostly nonsensical graphs)

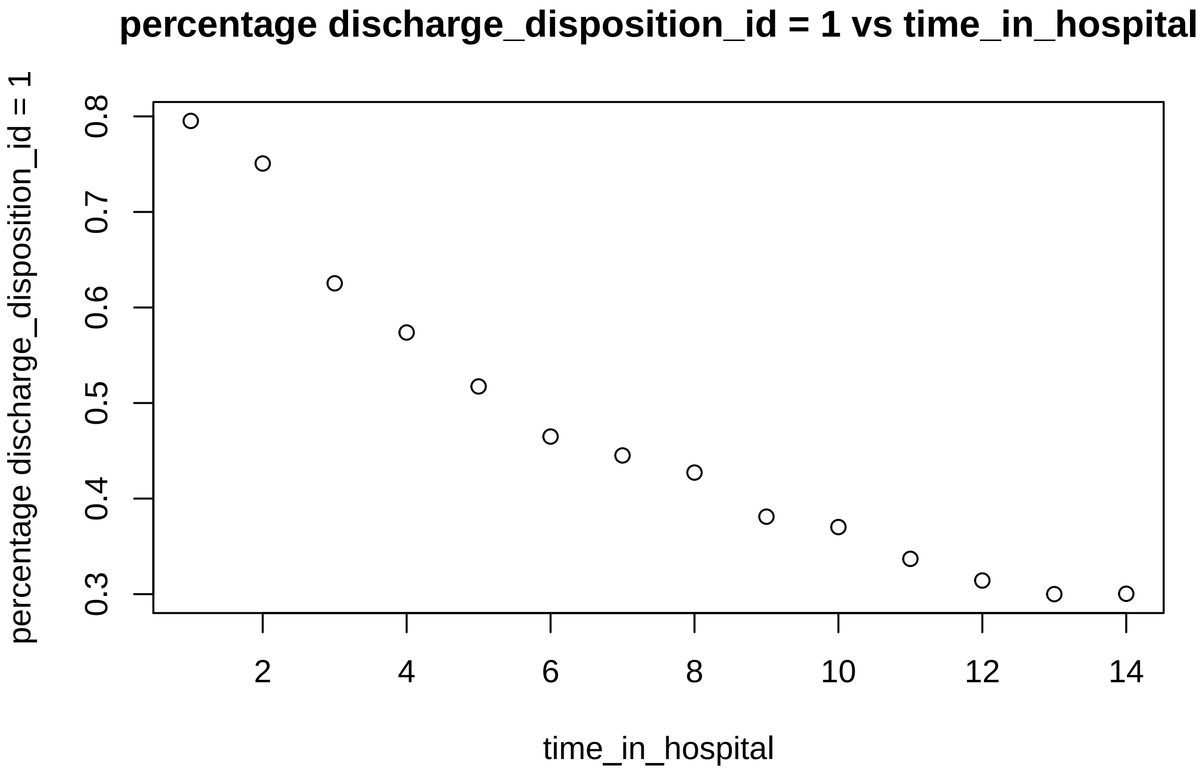
```
sapply(names(diab_data), function(n) {
  getProbsAttr1_givenAttr2(diab_data, 1, "discharge_disposition_id", n)
})
```

But I've already executed that and noticed that there seem to be two important factors involved: age, and time spent in the hospital.

```
tmp  <- getProbsAttr1_givenAttr2(diab_data, 1,
                                 "discharge_disposition_id",
                                 "age")
```



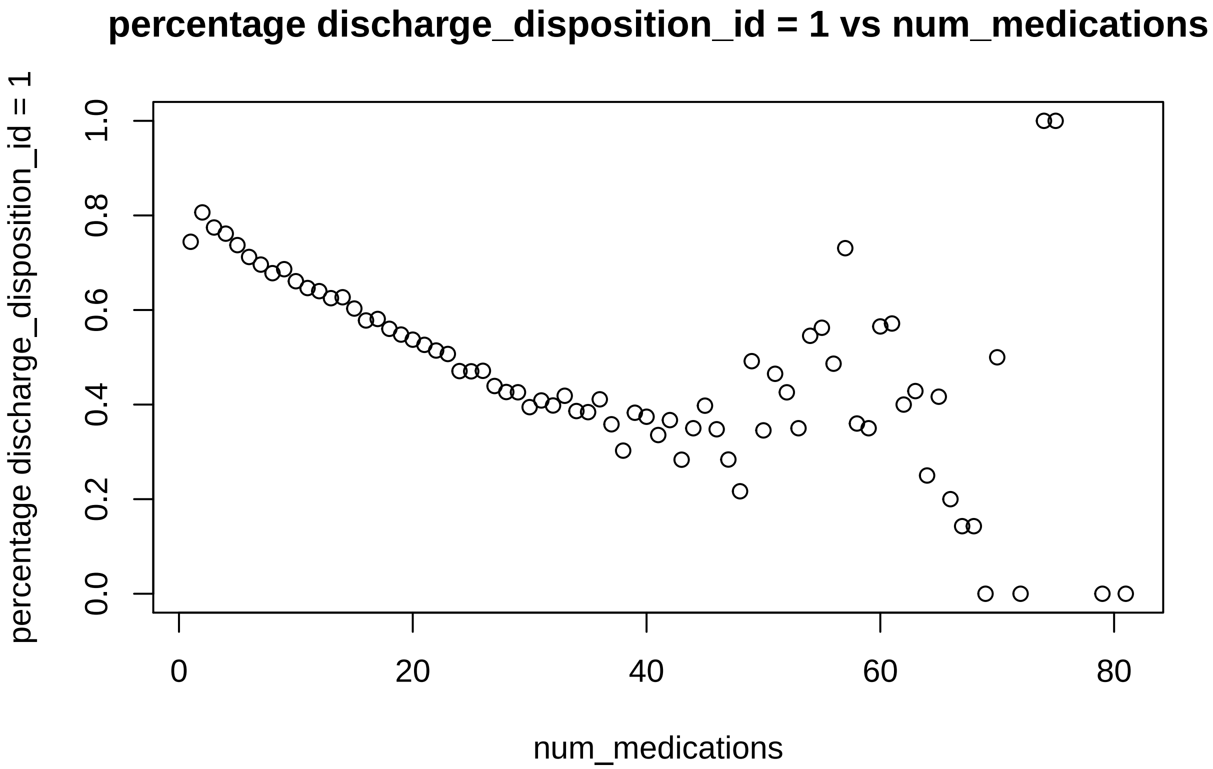
```
tmp <- getProbsAttr1_givenAttr2(diab_data, 1,
                                "discharge_disposition_id",
                                "time_in_hospital")
```



So, for instance, we can see that if a person is discharged within a day there is about an 79.5% probability of being released to home. Let's check out some more. By executing the command below you can see graphs for the probability that the patient will be sent home against all the other parameters independently.

It is also useful to note that there is a pretty clear trend apparent with the number of medications:

```
tmp <- getProbsAttr1_givenAttr2(diab_data, 1,
                         "discharge_disposition_id",
                         "num_medications")
```



What I find really interesting about this plot is that you can somewhat see the law of large numbers in effect. There is a considerable amount of data for patients on fewer than 20 medications, but whne you get past 60 medications there's so little data that the outliers are far more pronounced. I'm assuming that's why we see a more erratic tail end of that graph.

Predicting any other discharge disposition seems like it would be very difficult with this dataset due to insufficient data with any discharge disposition other than 1 (which we will discuss on the next page). One can somewhat test this theory by checking to see if any good predictors arise when executing the code below:

```
sapply(names(diab_data), function(n) {
  getProbsAttr1_givenAttr2(diab_data, 2, "discharge_disposition_id", n)
})
```

To whom it may concern:

On the Diabetes Dataset

Now that we've made some preliminary predictors (albeit very rudamentary), let's begin analyzing the quality of the data used to determine these predictors.

In order to do this, I will attempt to construct a 16 element vector of scores (each referring to one of the keywords in the paper).

Let's start with some of the obvious ones:

Accessibility: To assess the accessibility of this data, import the data and use the `View` function to inspect it. You'll note that all of the data is very neatly organized and imports very easily. I don't suspect that data can get much more accessible than that. 1.0

Appropriate Amount of Data: Although there are about 100k data points in this data set, the data is not nearly evenly distributed among certain variables (such as admission type and admission source, etc). So it seems as though in order to do this correctly, one would need to subset our data further to obtain an even distribution of each of the observed variables. If you want to remove some of the variance, you could then redo that "experiment" several times, sampling the same number of points from the total dataset each time (while ensuring a more even distribution).

Believability:

Completeness:

Concise representation: The data could be more concise by doing things like using integers for values across the board. But that wouldn't really make too much of a difference and it's really not worth coming up with a metric and then taking a ratio. We can just default this measure to 1.0. You can see how concise it is by using the `View` function on the imported dataframe.

Consistent representation: The data seems very consistent within each column (maybe 1.0 as a score), but between columns different values are used to signify null values, for instance. In one column they might use a questionmark, while in another "Unknown/Invalid" and in another "None" (maybe a 0.7 score between columns). So, overall, since column consistency is far more important than consistency between columns, I'm going to use something like a weighted average of the two scores instead of the suggested min operator and call it 0.9 in consistency.

Ease of manipulation: Due to the extensive data-munging already done on the files, the data is extremely easy to manipulate. Although this is somewhat a function of the representation. I give it a score of 1.0.

Free-of-Error:

Interpretability:

Objectivity:

Relevancy:

Reputation:

Security:

Timeliness:

Understandability:

Value-Added:

So, it seems as though the fact that the data was collected from 130 different hospitals over the course of 10 years leads me to believe that a lot of data processing happened between the actual data input and what we got. After all, most of the hospitals were probably working on different systems with different schemata for storing data. And over the course of 10 years, one would imagine that many of these hospitals would have updated their systems, causing even more inconsistencies in data storage schemata.

With all of that been said, I believe that the people at UCI probably did a good job at not messing up the data they were given. By that I mean that I estimate the probability of an error in any one particular value being due to a data-munging error introduced by UCI at something less than 0.1% (that still seems far too high, but let's be conservative). The fact that UCI got this the data from another intermediate database, however, raises the probabilities of transcription error (hence raising the probability of the data being free-of-error and consequently lowering the believability).

Now let's discuss the possible data quality issues associated from the data collection. For the sake of discussion, let's assume that the admitting official is diligent and