EDA Exercise 3

Aaron Niskin

August 30, 2016

University of California Irvine maintains an interesting collection of data sets for those interested in machine learning

https://archive.ics.uci.edu/ml/datasets.html

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

and download the zipped diabetes data set.

Prepare an R Markdown document which documents the use of R tools/code to address the following

1. Read the data into memory.

```
df <- read.csv("dataset_diabetes/diabetic_data.csv")
```

2. Document any missing values in the data.

Sorry... I think I'm enjoying this a bit too much. I wanted to try functional programming, so this is me doing that. So, I'm sorry for nerding out super hard on this. But look how cool that is! I can completely reuse that function.

```
hasInvalidDataNames <- function(dats, nullValues) {
  hasInvalidData <- function(col_name) {
    occurrences <- sapply(dats[,col_name], function(x) {
      is.null(x) || is.na(x) || x %in% nullValues
      })
    length(which(occurrences))
  }
  sapply(names(dats), hasInvalidData)
}
invalidRows_tmp <- hasInvalidDataNames(df, c("", "None", "?", "Unknown/Invalid"))
invalidRows     <- invalidRows_tmp[invalidRows_tmp > 0]
```

```
invalidRows
```

```
##              race            gender            weight        payer_code
##              2273                 3             98569             40256
## medical_specialty            diag_1            diag_2            diag_3
##             49949                21               358              1423
##     max_glu_serum        A1Cresult
##             96420             84748
```

So, the rows that have missing values (to include any empty strings, question marks, "None"'s, or "Unknown/Invalid"). If one were to decide to not consider one of those values as missing, just recompute with a different second argument vector.

3. What percentage of patients are admitted from the emergency room? Given a patient is admitted from the emergency room, what is the probability that their discharge status will be "expired"?

So first let's store the subset of those who were admitted from the emergency room, get the count and the total count of all patients admitted into these 150 hospitals.

```
emergency_admits <- df[df$admission_source_id == 7,]
num_emergency <- dim(emergency_admits)[1]
num_patients <- dim(df)[1]
```

Now, we can divide the number of emergency room admitees by the total number of patients (then multiply by 100) to get the percentage.

```
answer3a <- num_emergency / num_patients
```

The percentage of people admitted to the hospital from the emergency room is approximately, 56.5%.

Now we can further subset the emergencyroom admitees and redo the same.

```
num_exp <- dim(emergency_admits[emergency_admits$discharge_disposition_id == 11,])[1]
answer3b <- num_exp / num_emergency
```

So, the total probability that a person who was admitted from the emergency room will also have an expired discharge status is approximately, 2.0%.

4. What is the most frequent admission status? What is the most frequent discharge status? For the most frequent admission status, what is the most frequent discharge status?

Let's check out our data a bit just to see what we're working with.

```
table(df$admission_type_id)
```

```
##
##     1     2     3     4     5     6     7     8
## 53990 18480 18869    10  4785  5291    21   320
```

We can see that it's 1 (corresponding to Emergencyroom admittance), but let's do this programmatically:

```
admTab <- table(df$admission_type_id)
mostFreqAdm <- names(which(admTab == max(admTab)))
mostFreqAdm
```

```
## [1] "1"
```

And again for disposition id:

```
table(df$discharge_disposition_id)
```

```
##
##     1     2     3     4     5     6     7     8     9    10    11    12
## 60234  2128 13954   815  1184 12902   623   108    21     6  1642     3
##    13    14    15    16    17    18    19    20    22    23    24    25
##   399   372    63    11    14  3691     8     2  1993   412    48   989
##    27    28
##     5   139
```

So we should get 1 again (corresponding to "discharged to home").

```
disTabs <- table(df$discharge_disposition_id)
mostFreqDis <- names(which(disTabs == max(disTabs)))
mostFreqDis
```

```
## [1] "1"
```

Just to make sure we're not getting 1 due to some anomaly or misunderstanding of R somewhere, let's try to get a value whose name is not the same as the array index, like 22,1993 by computing:

```
names(which(disTabs == 1993))
```

```
## [1] "22"
```

```
table(df[df$admission_type_id == mostFreqAdm, "discharge_disposition_id"])
```

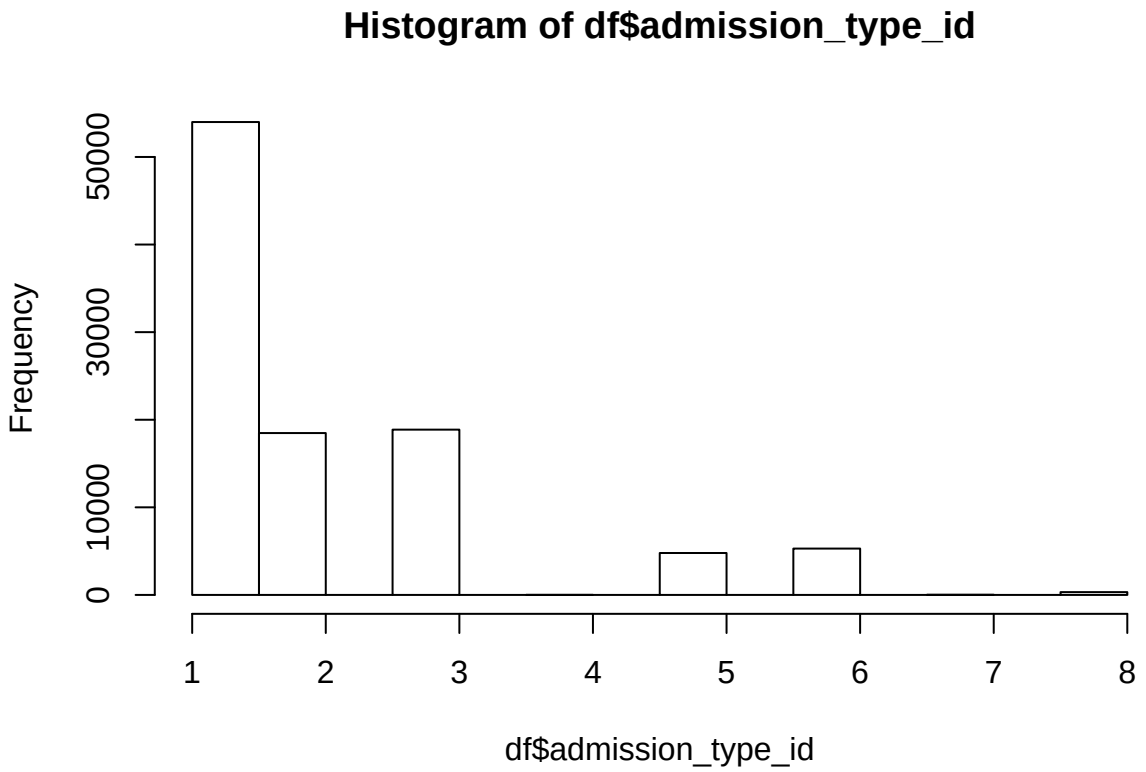
```
##
##     1     2     3     4     5     6     7     8     9    11    12    13
## 31695  1189  7813   512   587  6572   417    43    10  1102     1   278
##    14    15    17    18    19    20    22    23    24    25    27    28
##   235    38     2  2142     3     1   960   257    37    12     5    79
```

```
MFATab <- table(df[df$admission_type_id == mostFreqAdm, "discharge_disposition_id"])
names(which(MFATab == max(MFATab)))
```

```
## [1] "1"
```

5. Characterize the distribution of admission type.

```
hist(df$admission_type_id)
```



It looks roughly like a geometric distribution to me, but I will investigate tools to identify the distribution a bit more so as to get a better estimation.