exercise-3

Matthew McAvoy

August 30, 2016

Exercise 3

We begin by changing to the working directory and reading in the Diabetes data.

```r
setwd("C:/Users/homur/OneDrive/New College/EDA/Week 2/dataset_diabetes/dataset_diabetes/")
DiabetesData <- read.csv(file="diabetic_data.csv",head=TRUE,sep=",")
```

2.

The following commands converts our file into a data frame, then looks for all the empty values in the data.

```r
DiabetesDataFrame <- data.frame(DiabetesData)
#str(DiabetesDataFrame)
emptyvals <- sapply(DiabetesDataFrame, is.null)
emptyvals
```

```
##             encounter_id              patient_nbr                     race
##                    FALSE                    FALSE                    FALSE
##                   gender                      age                   weight
##                    FALSE                    FALSE                    FALSE
##        admission_type_id discharge_disposition_id      admission_source_id
##                    FALSE                    FALSE                    FALSE
##         time_in_hospital               payer_code        medical_specialty
##                    FALSE                    FALSE                    FALSE
##       num_lab_procedures           num_procedures          num_medications
##                    FALSE                    FALSE                    FALSE
##        number_outpatient         number_emergency         number_inpatient
##                    FALSE                    FALSE                    FALSE
##                   diag_1                   diag_2                   diag_3
##                    FALSE                    FALSE                    FALSE
##         number_diagnoses            max_glu_serum                A1Cresult
##                    FALSE                    FALSE                    FALSE
##                metformin              repaglinide              nateglinide
##                    FALSE                    FALSE                    FALSE
##           chlorpropamide              glimepiride            acetohexamide
##                    FALSE                    FALSE                    FALSE
##                glipizide                glyburide              tolbutamide
##                    FALSE                    FALSE                    FALSE
##             pioglitazone            rosiglitazone                 acarbose
##                    FALSE                    FALSE                    FALSE
##                 miglitol             troglitazone               tolazamide
##                    FALSE                    FALSE                    FALSE
##                  examide              citoglipton                  insulin
##                    FALSE                    FALSE                    FALSE
##      glyburide.metformin      glipizide.metformin glimepiride.pioglitazone
##                    FALSE                    FALSE                    FALSE
##  metformin.rosiglitazone   metformin.pioglitazone                   change
##                    FALSE                    FALSE                    FALSE
##              diabetesMed               readmitted
##                    FALSE                    FALSE
```

3.

We need to subset on admission_type_id=1 and discharge_disposition_id. This gives the number of people who were admitted to the emergency room. We can compare this to the total number of patients to find the percentage admitted to the emergency room. Then we subset again to find the the number of people with discharged status of expired using discharge_id=11. The ID's ere identified by ID_mapping.csv file. We then find the ratio between the two.

```
emergencyAdmissions <- DiabetesDataFrame[,c("admission_type_id","discharge_disposition_id")]
numPatients <- nrow(emergencyAdmissions)
numEmergencyAdmissions <- nrow(subset(emergencyAdmissions, admission_type_id == 1))

percentAdmissions <- numEmergencyAdmissions/numPatients

numExpiredAdmissions <- nrow(subset(emergencyAdmissions, admission_type_id == 1 & discharge_disposition_
percentExpired <- numExpiredAdmissions/numEmergencyAdmissions
```

```
cat('The percentage of patients admitted from the emergency room is', percentAdmissions)
```

```
## The percentage of patients admitted from the emergency room is 0.5305308
```

```
cat('The percentage of admitted patients from the emergency room is', percentExpired)
```

```
## The percentage of admitted patients from the emergency room is 0.02041119
```

4.

To find the most frequent admission status, it would be nice if we could fine the mode. To do this, we need to use a function. Found online at 'http://www.tutorialspoint.com/r/r_mean_median_mode.htm'

```
# Create the function.
getmode <- function(v) {
   uniqv <- unique(v)
   uniqv[which.max(tabulate(match(v, uniqv)))]
}
```

Then vectorizing and applying the function, we can get the most frequent status for admissions and discharges.

```
modein <- getmode(DiabetesDataFrame$"admission_type_id")
modeout <- getmode(DiabetesDataFrame$"discharge_disposition_id")
```

```
cat('We see the most common admission-id is', modein, 'and the most common discharge-id is', modeout)
```

```
## We see the most common admission-id is 1 and the most common discharge-id is 1
```

Next we want to make a new data frame with admission and discharge ids so that we can select on admission id's. Then we subset on discharge id and find it's mode.

```
newFrame <- DiabetesDataFrame[,c("admission_type_id","discharge_disposition_id")]
newSubset <- subset(newFrame, admission_type_id == modein)
mode3 <- getmode(newFrame$"discharge_disposition_id")
```

```
cat('We find the mode for this is', mode3, 'again.')
```

```
## We find the mode for this is 1 again.
```

5. To characterize the distribution, we can vectorize on admission source and run boxplot(). It looks like most of admission's are between 1-7, with a few outlier id's outside this range.

```
boxplot(DiabetesDataFrame$admission_source_id)
```