CKME 136 Readmission Data Analysis

# uncomment the next lines to install required packages and load the libraries  
#install.packages("Hmisc")  
#install.packages("summarytools")  
#install.packages("pastecs")  
#install.packages("corrplot")  
#install.packages("moments")  
#install.packages("rcompanion") # required for transformTukey function  
#install.packages("dplyr")  
#install.packages("e1071")  
#install.packages("FSelector") # required for FSelector and RWeka libraries  
#install.packages("mlbench")  
#install.packages("tictoc")  
#install.packages("DMwR")  
#install.packages("RWeka")  
#install.packages("randomForest")  
#install.packages("mlbench")  
#install.packages("DescTools") # g-test goodness of fit  
#install.packages("arules")  
#install.packages("fastAdaboost")  
  
library("pastecs") # required for stat.desc function for summary statistics

## Warning: package 'pastecs' was built under R version 3.4.4

# library("summarytools")  
library("Hmisc") # describe function for summary statistics

## Warning: package 'Hmisc' was built under R version 3.4.4

## Loading required package: lattice

## Loading required package: survival

## Loading required package: Formula

## Loading required package: ggplot2

##   
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:base':  
##   
## format.pval, units

library("corrplot") # corrplot function to plot correlation graphics

## Warning: package 'corrplot' was built under R version 3.4.4

## corrplot 0.84 loaded

# library("moments")  
library("rcompanion") # required for transformTukey function

## Warning: package 'rcompanion' was built under R version 3.4.4

library("plyr")

## Warning: package 'plyr' was built under R version 3.4.4

##   
## Attaching package: 'plyr'

## The following objects are masked from 'package:Hmisc':  
##   
## is.discrete, summarize

library("dplyr") # required for %>% pipe

## Warning: package 'dplyr' was built under R version 3.4.4

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:plyr':  
##   
## arrange, count, desc, failwith, id, mutate, rename, summarise,  
## summarize

## The following objects are masked from 'package:Hmisc':  
##   
## src, summarize

## The following objects are masked from 'package:pastecs':  
##   
## first, last

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

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

library("FSelector") # feature selection algorithm

## Warning: package 'FSelector' was built under R version 3.4.4

library("mlbench")

## Warning: package 'mlbench' was built under R version 3.4.4

library("caret") # used for createPartition,

## Warning: package 'caret' was built under R version 3.4.4

##   
## Attaching package: 'caret'

## The following object is masked from 'package:survival':  
##   
## cluster

# library("Rfast")  
library("arules") # association rules ... apriori

## Warning: package 'arules' was built under R version 3.4.4

## Loading required package: Matrix

##   
## Attaching package: 'arules'

## The following object is masked from 'package:dplyr':  
##   
## recode

## The following objects are masked from 'package:base':  
##   
## abbreviate, write

library("e1071") # required for naivebayes algorithm

## Warning: package 'e1071' was built under R version 3.4.4

##   
## Attaching package: 'e1071'

## The following object is masked from 'package:Hmisc':  
##   
## impute

library("RWeka") # required for J48 decision tree algorithm

## Warning: package 'RWeka' was built under R version 3.4.4

library("randomForest") # required for randomForest algorithm

## Warning: package 'randomForest' was built under R version 3.4.4

## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##   
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':  
##   
## combine

## The following object is masked from 'package:ggplot2':  
##   
## margin

library("tictoc") # used to measure runtime of code

## Warning: package 'tictoc' was built under R version 3.4.4

library("DMwR") # required for SMOTE

## Warning: package 'DMwR' was built under R version 3.4.4

## Loading required package: grid

##   
## Attaching package: 'DMwR'

## The following object is masked from 'package:plyr':  
##   
## join

library("DescTools") # required for G-test ... goodness of fit

## Warning: package 'DescTools' was built under R version 3.4.4

##   
## Attaching package: 'DescTools'

## The following objects are masked from 'package:caret':  
##   
## MAE, RMSE

## The following objects are masked from 'package:Hmisc':  
##   
## %nin%, Label, Mean, Quantile

library("fastAdaboost") # required for Adaboost ensemble package

## Warning: package 'fastAdaboost' was built under R version 3.4.4

readmission\_data\_original <- read.csv("C:/Data/Education/Ryerson/CKME136/dataset\_diabetes/diabetic\_data.csv")  
# using a working copy of the data set to preserve original data for reference  
readmission\_data <-readmission\_data\_original  
## view the first few rows of the data  
head(readmission\_data)

## encounter\_id patient\_nbr race gender age weight  
## 1 2278392 8222157 Caucasian Female [0-10) ?  
## 2 149190 55629189 Caucasian Female [10-20) ?  
## 3 64410 86047875 AfricanAmerican Female [20-30) ?  
## 4 500364 82442376 Caucasian Male [30-40) ?  
## 5 16680 42519267 Caucasian Male [40-50) ?  
## 6 35754 82637451 Caucasian Male [50-60) ?  
## admission\_type\_id discharge\_disposition\_id admission\_source\_id  
## 1 6 25 1  
## 2 1 1 7  
## 3 1 1 7  
## 4 1 1 7  
## 5 1 1 7  
## 6 2 1 2  
## time\_in\_hospital payer\_code medical\_specialty num\_lab\_procedures  
## 1 1 ? Pediatrics-Endocrinology 41  
## 2 3 ? ? 59  
## 3 2 ? ? 11  
## 4 2 ? ? 44  
## 5 1 ? ? 51  
## 6 3 ? ? 31  
## num\_procedures num\_medications number\_outpatient number\_emergency  
## 1 0 1 0 0  
## 2 0 18 0 0  
## 3 5 13 2 0  
## 4 1 16 0 0  
## 5 0 8 0 0  
## 6 6 16 0 0  
## number\_inpatient diag\_1 diag\_2 diag\_3 number\_diagnoses max\_glu\_serum  
## 1 0 250.83 ? ? 1 None  
## 2 0 276 250.01 255 9 None  
## 3 1 648 250 V27 6 None  
## 4 0 8 250.43 403 7 None  
## 5 0 197 157 250 5 None  
## 6 0 414 411 250 9 None  
## A1Cresult metformin repaglinide nateglinide chlorpropamide glimepiride  
## 1 None No No No No No  
## 2 None No No No No No  
## 3 None No No No No No  
## 4 None No No No No No  
## 5 None No No No No No  
## 6 None No No No No No  
## acetohexamide glipizide glyburide tolbutamide pioglitazone rosiglitazone  
## 1 No No No No No No  
## 2 No No No No No No  
## 3 No Steady No No No No  
## 4 No No No No No No  
## 5 No Steady No No No No  
## 6 No No No No No No  
## acarbose miglitol troglitazone tolazamide examide citoglipton insulin  
## 1 No No No No No No No  
## 2 No No No No No No Up  
## 3 No No No No No No No  
## 4 No No No No No No Up  
## 5 No No No No No No Steady  
## 6 No No No No No No Steady  
## glyburide.metformin glipizide.metformin glimepiride.pioglitazone  
## 1 No No No  
## 2 No No No  
## 3 No No No  
## 4 No No No  
## 5 No No No  
## 6 No No No  
## metformin.rosiglitazone metformin.pioglitazone change diabetesMed  
## 1 No No No No  
## 2 No No Ch Yes  
## 3 No No No Yes  
## 4 No No Ch Yes  
## 5 No No Ch Yes  
## 6 No No No Yes  
## readmitted  
## 1 NO  
## 2 >30  
## 3 NO  
## 4 NO  
## 5 NO  
## 6 >30

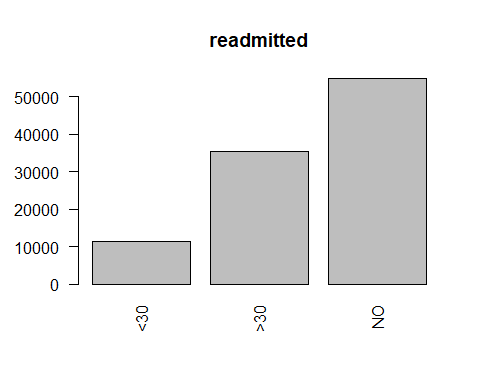
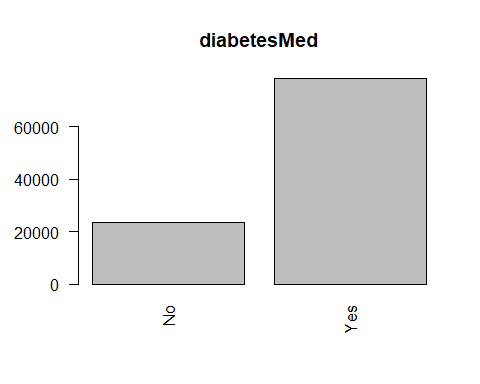
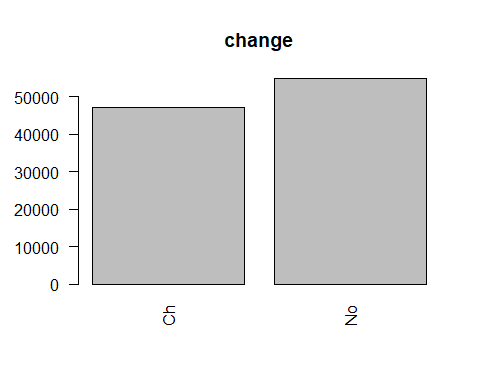
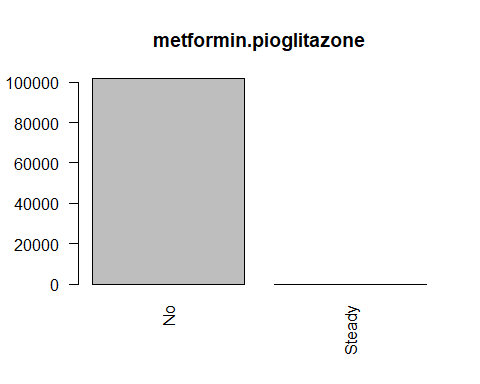
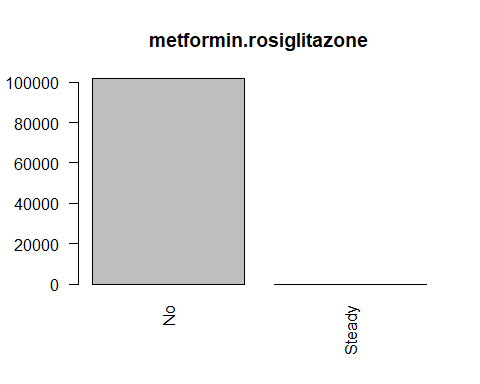
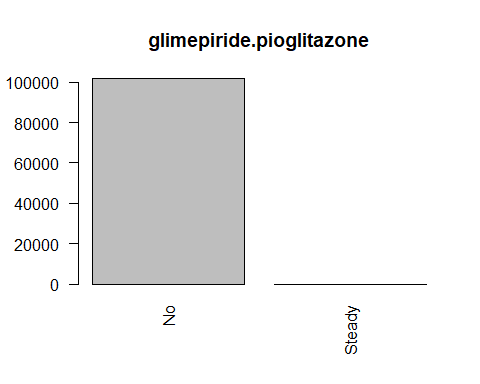
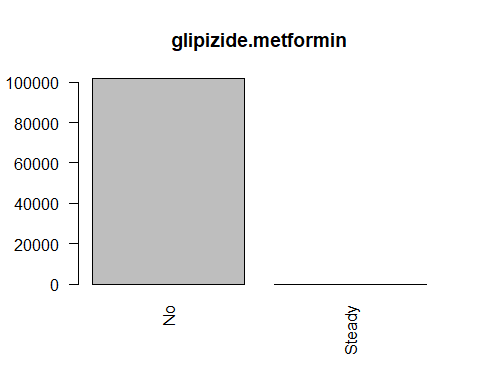
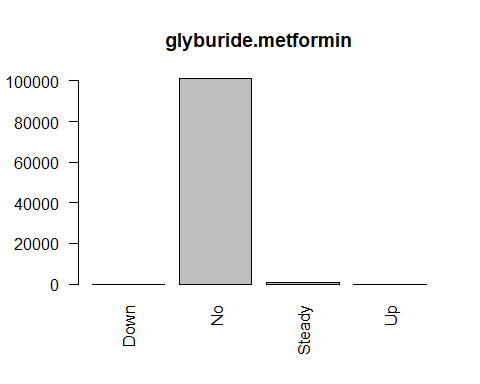
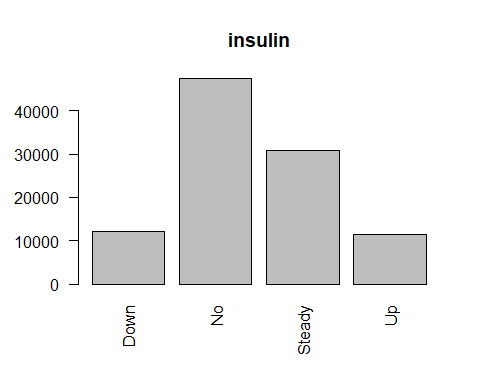
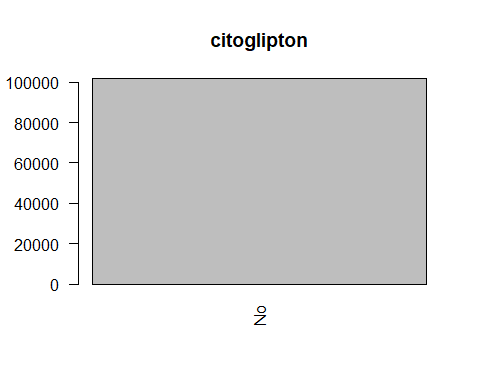
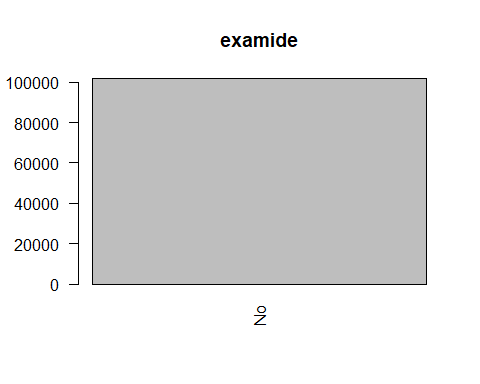
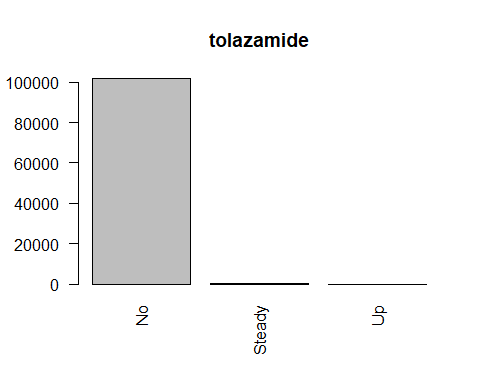
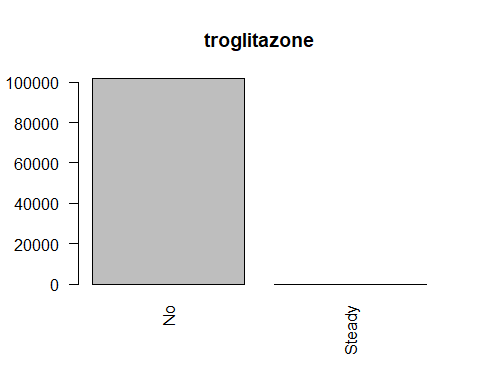
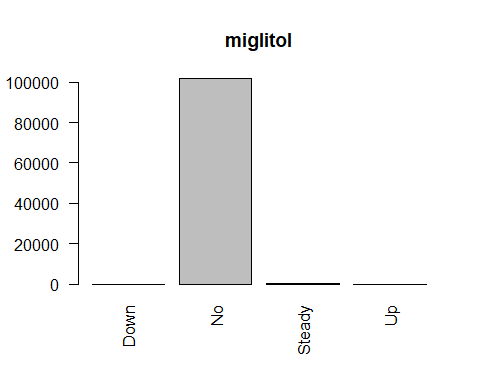
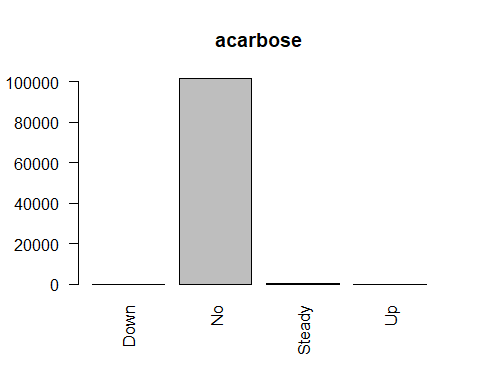
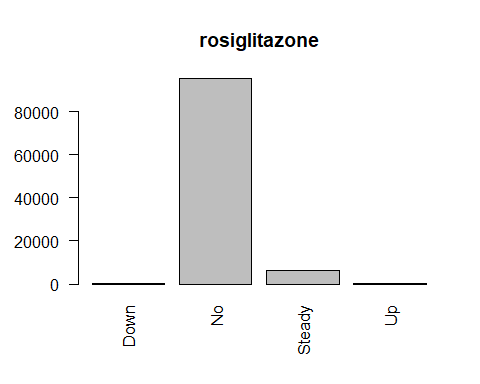
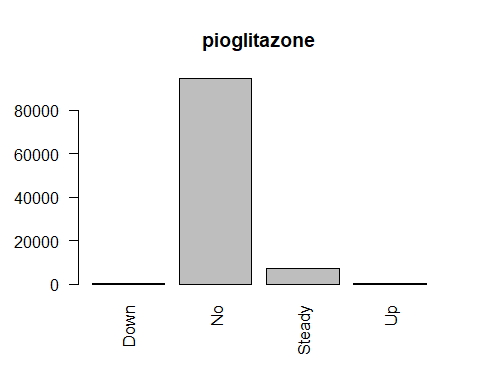
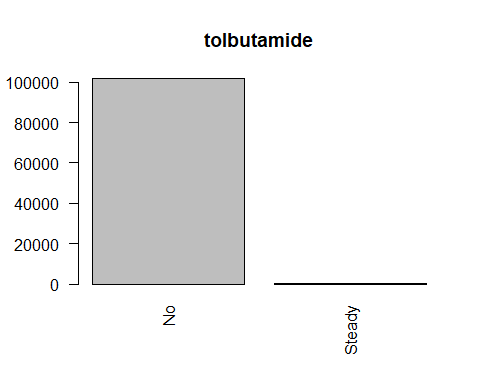
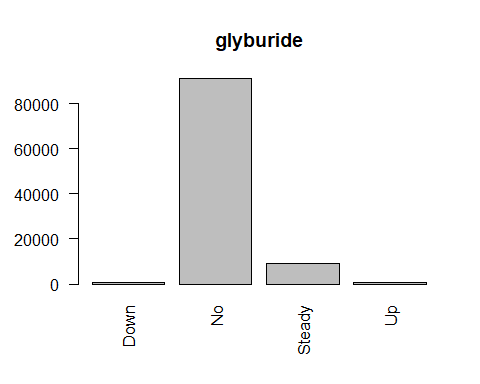
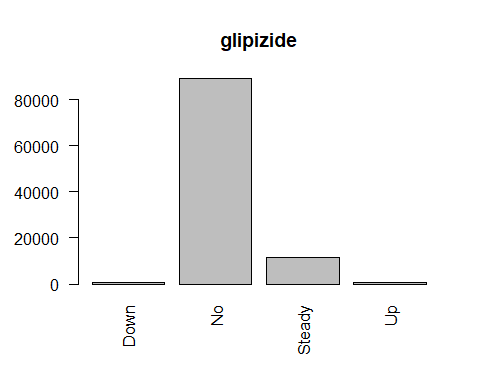
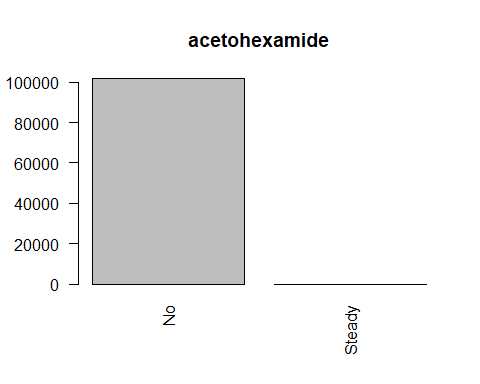
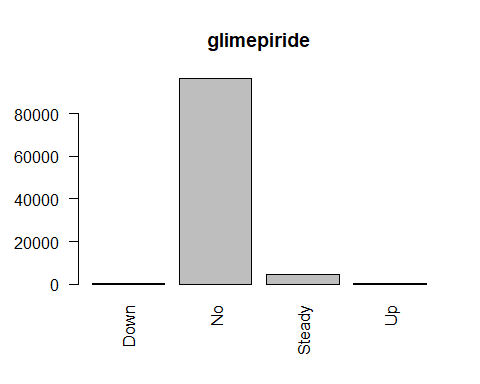
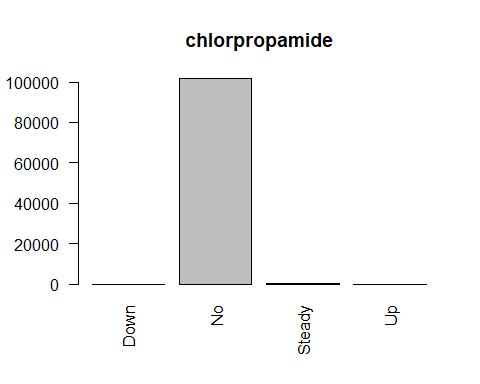
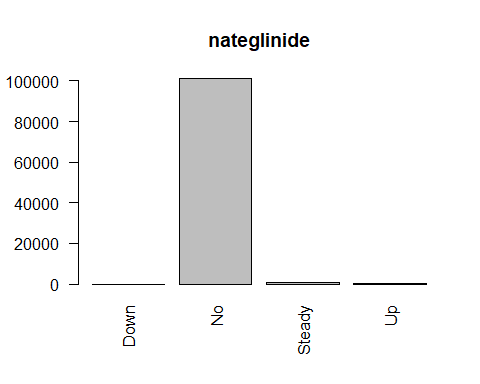
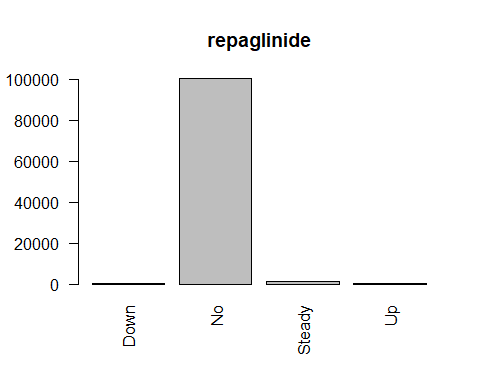
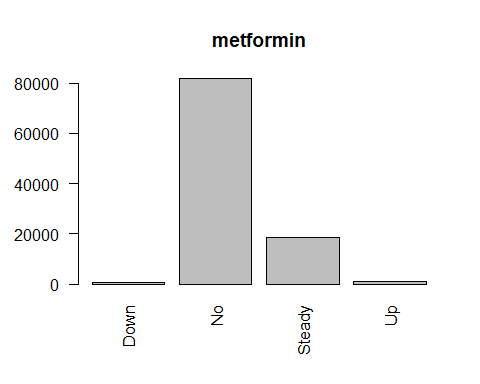
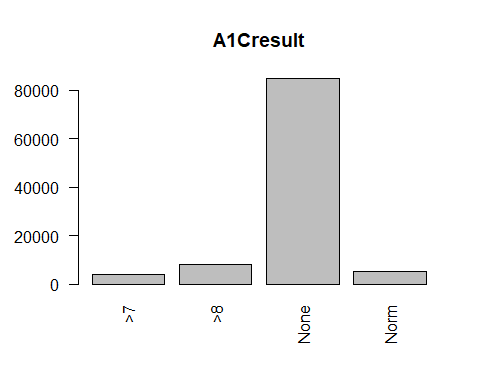
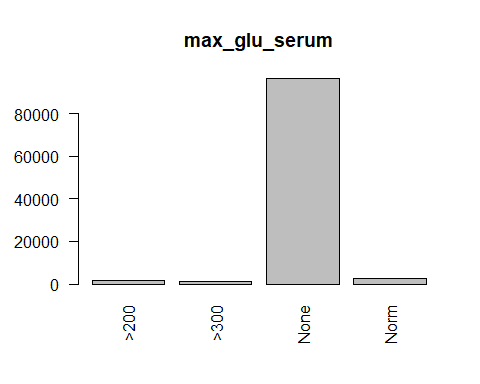
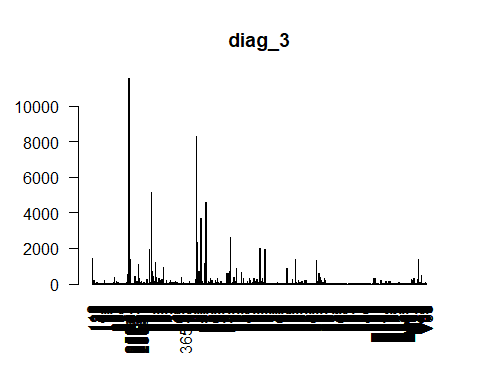
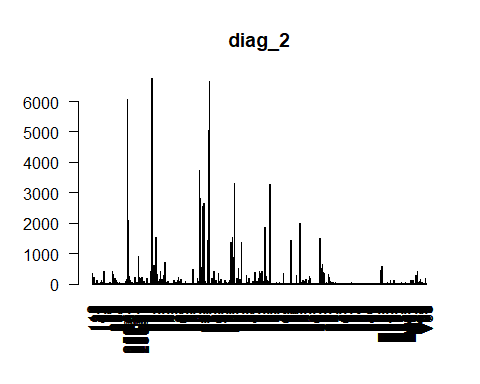
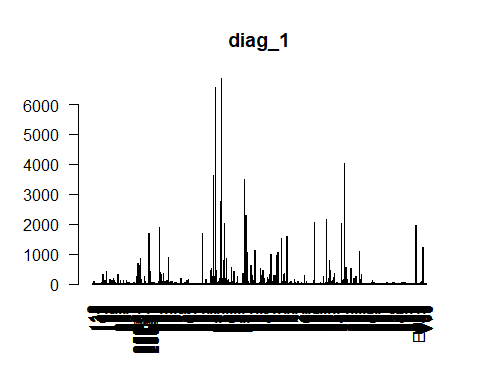
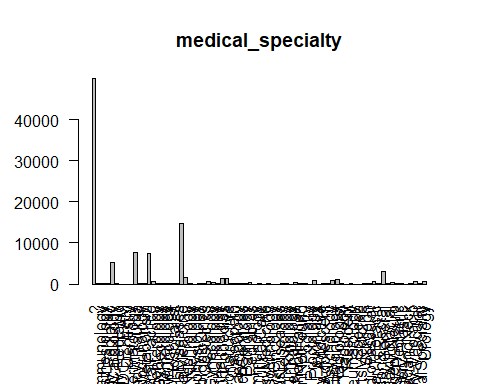
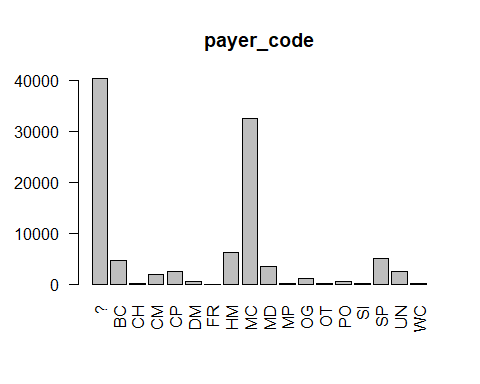
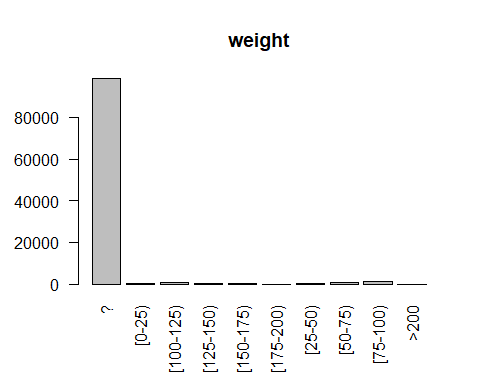
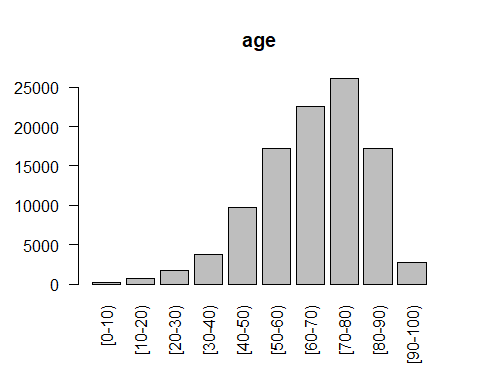
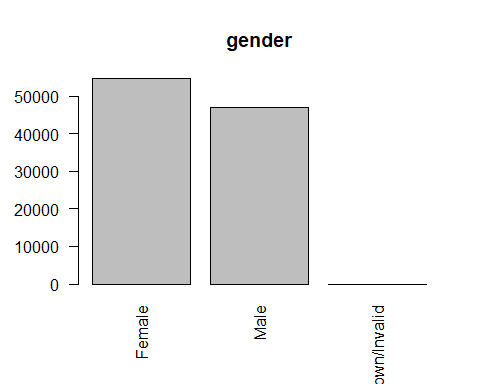
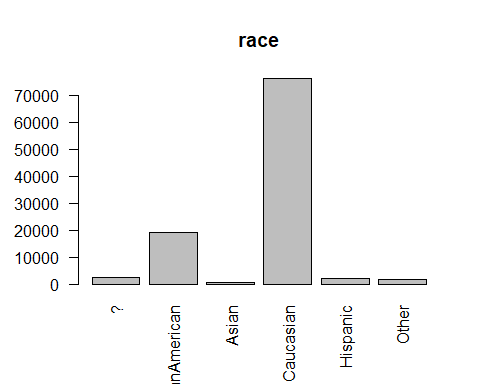
data\_desc <- read.csv("C:/Data/Education/Ryerson/CKME136/dataset\_diabetes/IDs\_mapping.csv")  
# view the first few rows of the data description  
data\_desc

## admission\_type\_id  
## 1 1  
## 2 2  
## 3 3  
## 4 4  
## 5 5  
## 6 6  
## 7 7  
## 8 8  
## 9   
## 10 discharge\_disposition\_id  
## 11 1  
## 12 2  
## 13 3  
## 14 4  
## 15 5  
## 16 6  
## 17 7  
## 18 8  
## 19 9  
## 20 10  
## 21 11  
## 22 12  
## 23 13  
## 24 14  
## 25 15  
## 26 16  
## 27 17  
## 28 18  
## 29 19  
## 30 20  
## 31 21  
## 32 22  
## 33 23  
## 34 24  
## 35 25  
## 36 26  
## 37 30  
## 38 27  
## 39 28  
## 40 29  
## 41   
## 42 admission\_source\_id  
## 43 1  
## 44 2  
## 45 3  
## 46 4  
## 47 5  
## 48 6  
## 49 7  
## 50 8  
## 51 9  
## 52 10  
## 53 11  
## 54 12  
## 55 13  
## 56 14  
## 57 15  
## 58 17  
## 59 18  
## 60 19  
## 61 20  
## 62 21  
## 63 22  
## 64 23  
## 65 24  
## 66 25  
## 67 26  
## description  
## 1 Emergency  
## 2 Urgent  
## 3 Elective  
## 4 Newborn  
## 5 Not Available  
## 6 NULL  
## 7 Trauma Center  
## 8 Not Mapped  
## 9   
## 10 description  
## 11 Discharged to home  
## 12 Discharged/transferred to another short term hospital  
## 13 Discharged/transferred to SNF  
## 14 Discharged/transferred to ICF  
## 15 Discharged/transferred to another type of inpatient care institution  
## 16 Discharged/transferred to home with home health service  
## 17 Left AMA  
## 18 Discharged/transferred to home under care of Home IV provider  
## 19 Admitted as an inpatient to this hospital  
## 20 Neonate discharged to another hospital for neonatal aftercare  
## 21 Expired  
## 22 Still patient or expected to return for outpatient services  
## 23 Hospice / home  
## 24 Hospice / medical facility  
## 25 Discharged/transferred within this institution to Medicare approved swing bed  
## 26 Discharged/transferred/referred another institution for outpatient services  
## 27 Discharged/transferred/referred to this institution for outpatient services  
## 28 NULL  
## 29 Expired at home. Medicaid only, hospice.  
## 30 Expired in a medical facility. Medicaid only, hospice.  
## 31 Expired, place unknown. Medicaid only, hospice.  
## 32 Discharged/transferred to another rehab fac including rehab units of a hospital .  
## 33 Discharged/transferred to a long term care hospital.  
## 34 Discharged/transferred to a nursing facility certified under Medicaid but not certified under Medicare.  
## 35 Not Mapped  
## 36 Unknown/Invalid  
## 37 Discharged/transferred to another Type of Health Care Institution not Defined Elsewhere  
## 38 Discharged/transferred to a federal health care facility.  
## 39 Discharged/transferred/referred to a psychiatric hospital of psychiatric distinct part unit of a hospital  
## 40 Discharged/transferred to a Critical Access Hospital (CAH).  
## 41   
## 42 description  
## 43 Physician Referral  
## 44 Clinic Referral  
## 45 HMO Referral  
## 46 Transfer from a hospital  
## 47 Transfer from a Skilled Nursing Facility (SNF)  
## 48 Transfer from another health care facility  
## 49 Emergency Room  
## 50 Court/Law Enforcement  
## 51 Not Available  
## 52 Transfer from critial access hospital  
## 53 Normal Delivery  
## 54 Premature Delivery  
## 55 Sick Baby  
## 56 Extramural Birth  
## 57 Not Available  
## 58 NULL  
## 59 Transfer From Another Home Health Agency  
## 60 Readmission to Same Home Health Agency  
## 61 Not Mapped  
## 62 Unknown/Invalid  
## 63 Transfer from hospital inpt/same fac reslt in a sep claim  
## 64 Born inside this hospital  
## 65 Born outside this hospital  
## 66 Transfer from Ambulatory Surgery Center  
## 67 Transfer from Hospice

# Basic summary statistics  
# Make a vector of categorical (factor) variables  
a <- (sapply(readmission\_data,class) == "factor")  
  
# Categorical variables (use describe function of Hmisc package to display summary)  
cat\_vars <- readmission\_data[, a]  
Hmisc::describe(cat\_vars)

## cat\_vars   
##   
## 37 Variables 101766 Observations  
## ---------------------------------------------------------------------------  
## race   
## n missing distinct   
## 101766 0 6   
##   
## Value ? AfricanAmerican Asian Caucasian  
## Frequency 2273 19210 641 76099  
## Proportion 0.022 0.189 0.006 0.748  
##   
## Value Hispanic Other  
## Frequency 2037 1506  
## Proportion 0.020 0.015  
## ---------------------------------------------------------------------------  
## gender   
## n missing distinct   
## 101766 0 3   
##   
## Value Female Male Unknown/Invalid  
## Frequency 54708 47055 3  
## Proportion 0.538 0.462 0.000  
## ---------------------------------------------------------------------------  
## age   
## n missing distinct   
## 101766 0 10   
##   
## Value [0-10) [10-20) [20-30) [30-40) [40-50) [50-60) [60-70)  
## Frequency 161 691 1657 3775 9685 17256 22483  
## Proportion 0.002 0.007 0.016 0.037 0.095 0.170 0.221  
##   
## Value [70-80) [80-90) [90-100)  
## Frequency 26068 17197 2793  
## Proportion 0.256 0.169 0.027  
## ---------------------------------------------------------------------------  
## weight   
## n missing distinct   
## 101766 0 10   
##   
## Value ? [0-25) [100-125) [125-150) [150-175) [175-200)  
## Frequency 98569 48 625 145 35 11  
## Proportion 0.969 0.000 0.006 0.001 0.000 0.000  
##   
## Value [25-50) [50-75) [75-100) >200  
## Frequency 97 897 1336 3  
## Proportion 0.001 0.009 0.013 0.000  
## ---------------------------------------------------------------------------  
## payer\_code   
## n missing distinct   
## 101766 0 18   
##   
## Value ? BC CH CM CP DM FR HM MC MD  
## Frequency 40256 4655 146 1937 2533 549 1 6274 32439 3532  
## Proportion 0.396 0.046 0.001 0.019 0.025 0.005 0.000 0.062 0.319 0.035  
##   
## Value MP OG OT PO SI SP UN WC  
## Frequency 79 1033 95 592 55 5007 2448 135  
## Proportion 0.001 0.010 0.001 0.006 0.001 0.049 0.024 0.001  
## ---------------------------------------------------------------------------  
## medical\_specialty   
## n missing distinct   
## 101766 0 73   
##   
## lowest : ? AllergyandImmunology Anesthesiology Anesthesiology-Pediatric Cardiology   
## highest: Surgery-PlasticwithinHeadandNeck Surgery-Thoracic Surgery-Vascular SurgicalSpecialty Urology   
## ---------------------------------------------------------------------------  
## diag\_1   
## n missing distinct   
## 101766 0 717   
##   
## lowest : ? 10 11 110 112, highest: V63 V66 V67 V70 V71  
## ---------------------------------------------------------------------------  
## diag\_2   
## n missing distinct   
## 101766 0 749   
##   
## lowest : ? 11 110 111 112, highest: V69 V70 V72 V85 V86  
## ---------------------------------------------------------------------------  
## diag\_3   
## n missing distinct   
## 101766 0 790   
##   
## lowest : ? 11 110 111 112, highest: V66 V70 V72 V85 V86  
## ---------------------------------------------------------------------------  
## max\_glu\_serum   
## n missing distinct   
## 101766 0 4   
##   
## Value >200 >300 None Norm  
## Frequency 1485 1264 96420 2597  
## Proportion 0.015 0.012 0.947 0.026  
## ---------------------------------------------------------------------------  
## A1Cresult   
## n missing distinct   
## 101766 0 4   
##   
## Value >7 >8 None Norm  
## Frequency 3812 8216 84748 4990  
## Proportion 0.037 0.081 0.833 0.049  
## ---------------------------------------------------------------------------  
## metformin   
## n missing distinct   
## 101766 0 4   
##   
## Value Down No Steady Up  
## Frequency 575 81778 18346 1067  
## Proportion 0.006 0.804 0.180 0.010  
## ---------------------------------------------------------------------------  
## repaglinide   
## n missing distinct   
## 101766 0 4   
##   
## Value Down No Steady Up  
## Frequency 45 100227 1384 110  
## Proportion 0.000 0.985 0.014 0.001  
## ---------------------------------------------------------------------------  
## nateglinide   
## n missing distinct   
## 101766 0 4   
##   
## Value Down No Steady Up  
## Frequency 11 101063 668 24  
## Proportion 0.000 0.993 0.007 0.000  
## ---------------------------------------------------------------------------  
## chlorpropamide   
## n missing distinct   
## 101766 0 4   
##   
## Value Down No Steady Up  
## Frequency 1 101680 79 6  
## Proportion 0.000 0.999 0.001 0.000  
## ---------------------------------------------------------------------------  
## glimepiride   
## n missing distinct   
## 101766 0 4   
##   
## Value Down No Steady Up  
## Frequency 194 96575 4670 327  
## Proportion 0.002 0.949 0.046 0.003  
## ---------------------------------------------------------------------------  
## acetohexamide   
## n missing distinct   
## 101766 0 2   
##   
## Value No Steady  
## Frequency 101765 1  
## Proportion 1 0  
## ---------------------------------------------------------------------------  
## glipizide   
## n missing distinct   
## 101766 0 4   
##   
## Value Down No Steady Up  
## Frequency 560 89080 11356 770  
## Proportion 0.006 0.875 0.112 0.008  
## ---------------------------------------------------------------------------  
## glyburide   
## n missing distinct   
## 101766 0 4   
##   
## Value Down No Steady Up  
## Frequency 564 91116 9274 812  
## Proportion 0.006 0.895 0.091 0.008  
## ---------------------------------------------------------------------------  
## tolbutamide   
## n missing distinct   
## 101766 0 2   
##   
## Value No Steady  
## Frequency 101743 23  
## Proportion 1 0  
## ---------------------------------------------------------------------------  
## pioglitazone   
## n missing distinct   
## 101766 0 4   
##   
## Value Down No Steady Up  
## Frequency 118 94438 6976 234  
## Proportion 0.001 0.928 0.069 0.002  
## ---------------------------------------------------------------------------  
## rosiglitazone   
## n missing distinct   
## 101766 0 4   
##   
## Value Down No Steady Up  
## Frequency 87 95401 6100 178  
## Proportion 0.001 0.937 0.060 0.002  
## ---------------------------------------------------------------------------  
## acarbose   
## n missing distinct   
## 101766 0 4   
##   
## Value Down No Steady Up  
## Frequency 3 101458 295 10  
## Proportion 0.000 0.997 0.003 0.000  
## ---------------------------------------------------------------------------  
## miglitol   
## n missing distinct   
## 101766 0 4   
##   
## Value Down No Steady Up  
## Frequency 5 101728 31 2  
## Proportion 0 1 0 0  
## ---------------------------------------------------------------------------  
## troglitazone   
## n missing distinct   
## 101766 0 2   
##   
## Value No Steady  
## Frequency 101763 3  
## Proportion 1 0  
## ---------------------------------------------------------------------------  
## tolazamide   
## n missing distinct   
## 101766 0 3   
##   
## Value No Steady Up  
## Frequency 101727 38 1  
## Proportion 1 0 0  
## ---------------------------------------------------------------------------  
## examide   
## n missing distinct value   
## 101766 0 1 No   
##   
## Value No  
## Frequency 101766  
## Proportion 1  
## ---------------------------------------------------------------------------  
## citoglipton   
## n missing distinct value   
## 101766 0 1 No   
##   
## Value No  
## Frequency 101766  
## Proportion 1  
## ---------------------------------------------------------------------------  
## insulin   
## n missing distinct   
## 101766 0 4   
##   
## Value Down No Steady Up  
## Frequency 12218 47383 30849 11316  
## Proportion 0.120 0.466 0.303 0.111  
## ---------------------------------------------------------------------------  
## glyburide.metformin   
## n missing distinct   
## 101766 0 4   
##   
## Value Down No Steady Up  
## Frequency 6 101060 692 8  
## Proportion 0.000 0.993 0.007 0.000  
## ---------------------------------------------------------------------------  
## glipizide.metformin   
## n missing distinct   
## 101766 0 2   
##   
## Value No Steady  
## Frequency 101753 13  
## Proportion 1 0  
## ---------------------------------------------------------------------------  
## glimepiride.pioglitazone   
## n missing distinct   
## 101766 0 2   
##   
## Value No Steady  
## Frequency 101765 1  
## Proportion 1 0  
## ---------------------------------------------------------------------------  
## metformin.rosiglitazone   
## n missing distinct   
## 101766 0 2   
##   
## Value No Steady  
## Frequency 101764 2  
## Proportion 1 0  
## ---------------------------------------------------------------------------  
## metformin.pioglitazone   
## n missing distinct   
## 101766 0 2   
##   
## Value No Steady  
## Frequency 101765 1  
## Proportion 1 0  
## ---------------------------------------------------------------------------  
## change   
## n missing distinct   
## 101766 0 2   
##   
## Value Ch No  
## Frequency 47011 54755  
## Proportion 0.462 0.538  
## ---------------------------------------------------------------------------  
## diabetesMed   
## n missing distinct   
## 101766 0 2   
##   
## Value No Yes  
## Frequency 23403 78363  
## Proportion 0.23 0.77  
## ---------------------------------------------------------------------------  
## readmitted   
## n missing distinct   
## 101766 0 3   
##   
## Value <30 >30 NO  
## Frequency 11357 35545 54864  
## Proportion 0.112 0.349 0.539  
## ---------------------------------------------------------------------------

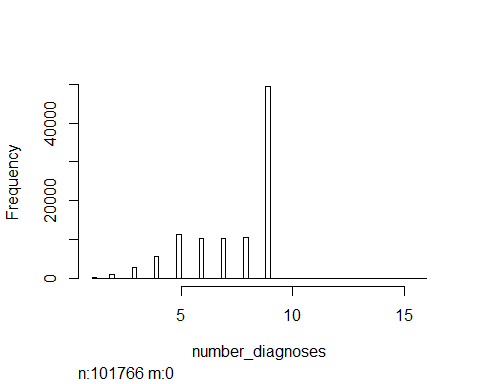
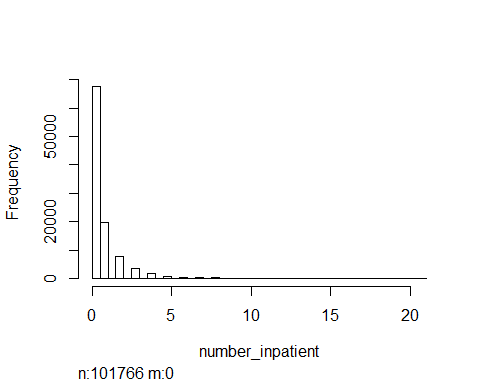
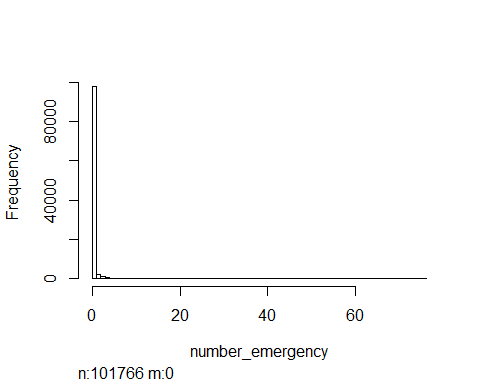
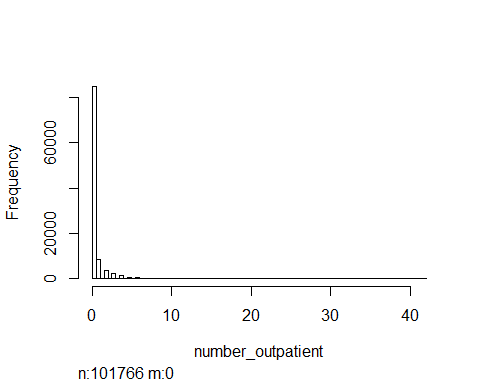
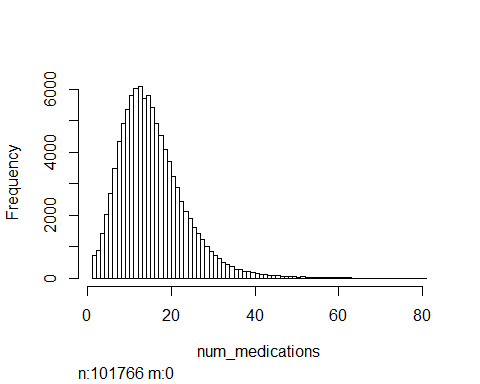
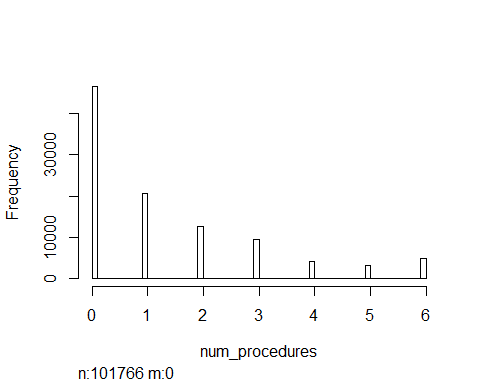
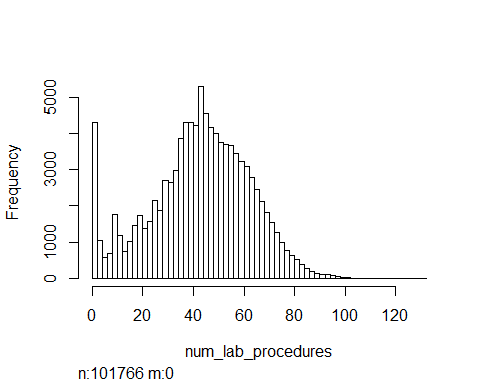
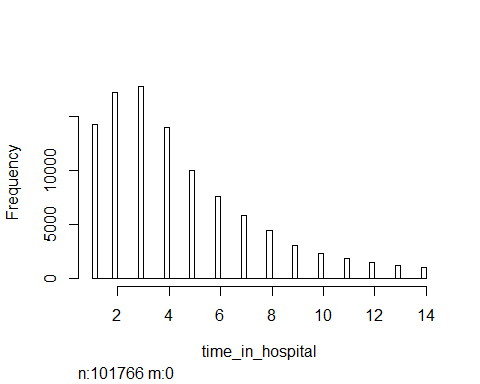
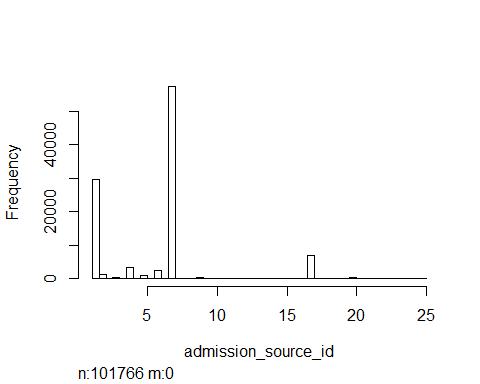
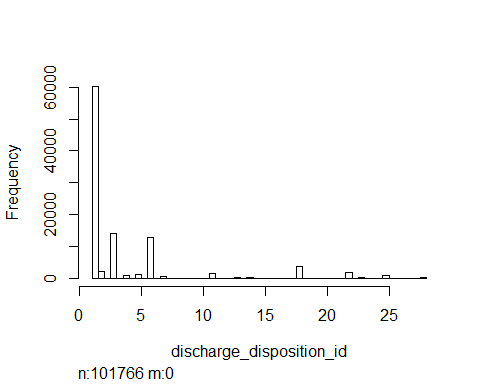
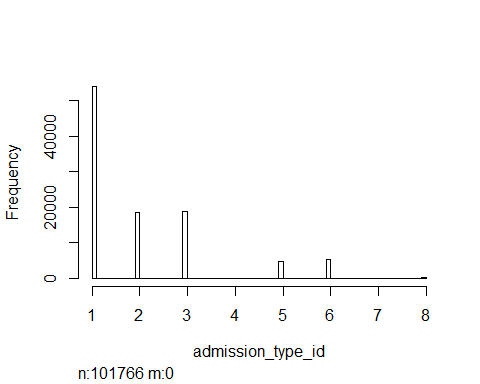
for (cat\_var in (1:ncol(cat\_vars))){  
 barplot(table(cat\_vars[, cat\_var]), main = names(cat\_vars[cat\_var]), las=2)  
}



# Numerical variables (use stat.desc function of pastecs package to display summary)  
num\_vars <- readmission\_data[, !a]  
# Do not include the encounter\_id or the patient\_nbr  
num\_vars <- subset(num\_vars, select = -c(1,2))  
options(scipen=100)  
options(digits=3)  
stat.desc(num\_vars)

## admission\_type\_id discharge\_disposition\_id  
## nbr.val 101766.00000 101766.0000  
## nbr.null 0.00000 0.0000  
## nbr.na 0.00000 0.0000  
## min 1.00000 1.0000  
## max 8.00000 28.0000  
## range 7.00000 27.0000  
## sum 205975.00000 378126.0000  
## median 1.00000 1.0000  
## mean 2.02401 3.7156  
## SE.mean 0.00453 0.0166  
## CI.mean.0.95 0.00888 0.0324  
## var 2.08919 27.8801  
## std.dev 1.44540 5.2802  
## coef.var 0.71413 1.4211  
## admission\_source\_id time\_in\_hospital num\_lab\_procedures  
## nbr.val 101766.0000 101766.00000 101766.0000  
## nbr.null 0.0000 0.00000 0.0000  
## nbr.na 0.0000 0.00000 0.0000  
## min 1.0000 1.00000 1.0000  
## max 25.0000 14.00000 132.0000  
## range 24.0000 13.00000 131.0000  
## sum 585606.0000 447362.00000 4385671.0000  
## median 7.0000 4.00000 44.0000  
## mean 5.7544 4.39599 43.0956  
## SE.mean 0.0127 0.00936 0.0617  
## CI.mean.0.95 0.0250 0.01834 0.1209  
## var 16.5168 8.91087 387.0805  
## std.dev 4.0641 2.98511 19.6744  
## coef.var 0.7063 0.67905 0.4565  
## num\_procedures num\_medications number\_outpatient  
## nbr.val 101766.00000 101766.0000 101766.00000  
## nbr.null 46652.00000 0.0000 85027.00000  
## nbr.na 0.00000 0.0000 0.00000  
## min 0.00000 1.0000 0.00000  
## max 6.00000 81.0000 42.00000  
## range 6.00000 80.0000 42.00000  
## sum 136339.00000 1630479.0000 37588.00000  
## median 1.00000 15.0000 0.00000  
## mean 1.33973 16.0218 0.36936  
## SE.mean 0.00535 0.0255 0.00397  
## CI.mean.0.95 0.01048 0.0499 0.00779  
## var 2.90978 66.0573 1.60596  
## std.dev 1.70581 8.1276 1.26727  
## coef.var 1.27325 0.5073 3.43100  
## number\_emergency number\_inpatient number\_diagnoses  
## nbr.val 101766.00000 101766.00000 101766.00000  
## nbr.null 90383.00000 67630.00000 0.00000  
## nbr.na 0.00000 0.00000 0.00000  
## min 0.00000 0.00000 1.00000  
## max 76.00000 21.00000 16.00000  
## range 76.00000 21.00000 15.00000  
## sum 20133.00000 64679.00000 755369.00000  
## median 0.00000 0.00000 8.00000  
## mean 0.19784 0.63557 7.42261  
## SE.mean 0.00292 0.00396 0.00606  
## CI.mean.0.95 0.00572 0.00776 0.01188  
## var 0.86578 1.59482 3.73881  
## std.dev 0.93047 1.26286 1.93360  
## coef.var 4.70325 1.98699 0.26050

for (num\_var in (1:ncol(num\_vars))){  
 hist(num\_vars[num\_var])  
}



# Removal of Records  
# Remove if gender is "Unknown/Invalid" since there are so few records (3) in this category  
readmission\_data <- readmission\_data[!readmission\_data$gender == "Unknown/Invalid",]  
# Remove unused "Unknown/Invalid" level in gender factor attribute  
readmission\_data$gender <- factor(readmission\_data$gender)  
  
# Remove if all three diagnoses are missing (at least one of these should be diabetes related)  
readmission\_data <- readmission\_data[!(readmission\_data$diag\_1 == "?" & readmission\_data$diag\_2 == '?' & readmission\_data$diag\_3 == '?'),]  
  
# Remove if patient died or discharged to hospice since they can not be readmitted  
readmission\_data <- readmission\_data[!(readmission\_data$discharge\_disposition\_id %in% c(11, 13, 14, 19, 20, 21)),]  
  
# Retain only records for the first encounter for each patient to avoid biasing toward patients with multiple encounters  
readmission\_data <- readmission\_data[order(readmission\_data$patient\_nbr, readmission\_data$encounter\_id),]  
readmission\_data <- readmission\_data[!duplicated(readmission\_data$patient\_nbr),]  
  
# Remove "weight" attribute due to 98% missing values  
readmission\_data <- subset(readmission\_data, select = -c(weight))

# Categorical Variables  
#Recategorize diagnoses ICD-9 codes to 9 disease categories including "Other" category for infrequent codes  
# \*\*\*  
# Attribute diag\_1  
# \*\*\*  
#create a working copy of diagnosis 1 attribute  
readmission\_data["copy\_diag\_1"] <- readmission\_data$diag\_1  
#convert ICD9 codes to numeric to facilitate searching  
# set non-numeric codes to zero (these will ultimately be reclassified as "Other")  
levels(readmission\_data$copy\_diag\_1)[levels(readmission\_data$copy\_diag\_1) == "?"] <- '0'  
levels(readmission\_data$copy\_diag\_1)[substr(levels(readmission\_data$copy\_diag\_1), 1, 1) == "V"] <- '0'  
levels(readmission\_data$copy\_diag\_1)[substr(levels(readmission\_data$copy\_diag\_1), 1, 1) == "E"] <- '0'  
  
#diabetes codes begin with 250 but may have decimal code after 250, so include up to 5 digits  
options(digits=5)  
  
# change diagnosis date type from factor to numeric (convert to character first, otherwise get levels of factor not the actual values)  
readmission\_data$copy\_diag\_1 <- as.numeric(as.character(readmission\_data$copy\_diag\_1))  
  
#add levels for 9 disease categories  
levels(readmission\_data$diag\_1) <- c(levels(readmission\_data$diag\_1),"Circulatory", "Respiratory", "Digestive", "Diabetes", "Injury", "Musculoskeletal", "Genitourinary", "Neoplasms", "Other")  
  
#set all entries for first primary diagnosis to "Other" as the default and only reassign those entries that belong to one of the other eight diagnosis categories  
readmission\_data$diag\_1 <- "Other"  
  
readmission\_data$diag\_1[((readmission\_data$copy\_diag\_1 >= 390) & (readmission\_data$copy\_diag\_1 <= 459)) | (readmission\_data$copy\_diag\_1 == 785)] <- "Circulatory"  
  
readmission\_data$diag\_1[((readmission\_data$copy\_diag\_1 >= 460) & (readmission\_data$copy\_diag\_1 <= 519)) | (readmission\_data$copy\_diag\_1 == 786)] <- 'Respiratory'  
  
readmission\_data$diag\_1[((readmission\_data$copy\_diag\_1 >= 520) & (readmission\_data$copy\_diag\_1 <= 579)) | (readmission\_data$copy\_diag\_1 == 787)] <- 'Digestive'  
  
readmission\_data$diag\_1[((readmission\_data$copy\_diag\_1 >= 250) & (readmission\_data$copy\_diag\_1 < 251))] <- 'Diabetes'  
  
readmission\_data$diag\_1[((readmission\_data$copy\_diag\_1 >= 800) & (readmission\_data$copy\_diag\_1 <= 999))] <- 'Injury'  
  
readmission\_data$diag\_1[((readmission\_data$copy\_diag\_1 >= 710) & (readmission\_data$copy\_diag\_1 <= 739))] <- 'Musculoskeletal'  
  
readmission\_data$diag\_1[((readmission\_data$copy\_diag\_1 >= 580) & (readmission\_data$copy\_diag\_1 <= 629)) | (readmission\_data$copy\_diag\_1 == 788)] <- 'Genitourinary'  
  
readmission\_data$diag\_1[((readmission\_data$copy\_diag\_1 >= 140) & (readmission\_data$copy\_diag\_1 <= 239))] <- 'Neoplasm'  
  
  
# \*\*\*  
# Attribute diag\_2  
# \*\*\*  
#create a working copy of diagnosis 2 attribute  
readmission\_data["copy\_diag\_2"] <- readmission\_data$diag\_2  
#convert ICD9 codes to numeric to facilitate searching  
# set non-numeric codes to zero (these will ultimately be reclassified as "Other")  
levels(readmission\_data$copy\_diag\_2)[levels(readmission\_data$copy\_diag\_2) == "?"] <- '0'  
levels(readmission\_data$copy\_diag\_2)[substr(levels(readmission\_data$copy\_diag\_2), 1, 1) == "V"] <- '0'  
levels(readmission\_data$copy\_diag\_2)[substr(levels(readmission\_data$copy\_diag\_2), 1, 1) == "E"] <- '0'  
  
#diabetes codes begin with 250 but may have decimal code after 250, so include up to 5 digits  
options(digits=5)  
  
# change diagnosis date type from character to numeric  
readmission\_data$copy\_diag\_2 <- as.numeric(as.character(readmission\_data$copy\_diag\_2))  
  
#add levels for 9 disease categories  
levels(readmission\_data$diag\_2) <- c(levels(readmission\_data$diag\_2),"Circulatory", "Respiratory", "Digestive", "Diabetes", "Injury", "Musculoskeletal", "Genitourinary", "Neoplasms", "Other")  
  
#set all entries for second primary diagnosis to "Other" as the default and only reassign those entries that belong to one of the other eight diagnosis categories  
readmission\_data$diag\_2 <- "Other"  
  
readmission\_data$diag\_2[((readmission\_data$copy\_diag\_2 >= 390) & (readmission\_data$copy\_diag\_2 <= 459)) | (readmission\_data$copy\_diag\_2 == 785)] <- "Circulatory"  
  
readmission\_data$diag\_2[((readmission\_data$copy\_diag\_2 >= 460) & (readmission\_data$copy\_diag\_2 <= 519)) | (readmission\_data$copy\_diag\_2 == 786)] <- 'Respiratory'  
  
readmission\_data$diag\_2[((readmission\_data$copy\_diag\_2 >= 520) & (readmission\_data$copy\_diag\_2 <= 579)) | (readmission\_data$copy\_diag\_2 == 787)] <- 'Digestive'  
  
readmission\_data$diag\_2[((readmission\_data$copy\_diag\_2 >= 250) & (readmission\_data$copy\_diag\_2 < 251))] <- 'Diabetes'  
  
readmission\_data$diag\_2[((readmission\_data$copy\_diag\_2 >= 800) & (readmission\_data$copy\_diag\_2 <= 999))] <- 'Injury'  
  
readmission\_data$diag\_2[((readmission\_data$copy\_diag\_2 >= 710) & (readmission\_data$copy\_diag\_2 <= 739))] <- 'Musculoskeletal'  
  
readmission\_data$diag\_2[((readmission\_data$copy\_diag\_2 >= 580) & (readmission\_data$copy\_diag\_2 <= 629)) | (readmission\_data$copy\_diag\_2 == 788)] <- 'Genitourinary'  
  
readmission\_data$diag\_2[((readmission\_data$copy\_diag\_2 >= 140) & (readmission\_data$copy\_diag\_2 <= 239))] <- 'Neoplasm'  
  
# \*\*\*  
# Attribute diag\_3  
# \*\*\*  
#create a working copy of diagnosis 3 attribute  
readmission\_data["copy\_diag\_3"] <- readmission\_data$diag\_3  
  
#convert ICD9 codes to numeric to facilitate searching  
# set non-numeric codes to zero (these will ultimately be reclassified as "Other")  
levels(readmission\_data$copy\_diag\_3)[levels(readmission\_data$copy\_diag\_3) == "?"] <- '0'  
levels(readmission\_data$copy\_diag\_3)[substr(levels(readmission\_data$copy\_diag\_3), 1, 1) == "V"] <- '0'  
levels(readmission\_data$copy\_diag\_3)[substr(levels(readmission\_data$copy\_diag\_3), 1, 1) == "E"] <- '0'  
  
#diabetes codes begin with 250 but may have decimal code after 250, so include up to 5 digits  
options(digits=5)  
  
# change diagnosis date type from character to numeric  
readmission\_data$copy\_diag\_3 <- as.numeric(as.character(readmission\_data$copy\_diag\_3))  
  
#add levels for 9 disease categories  
levels(readmission\_data$diag\_3) <- c(levels(readmission\_data$diag\_3),"Circulatory", "Respiratory", "Digestive", "Diabetes", "Injury", "Musculoskeletal", "Genitourinary", "Neoplasms", "Other")  
  
#set all entries for third primary diagnosis to "Other" as the default and only reassign those entries that belong to one of the other eight diagnosis categories  
readmission\_data$diag\_3 <- "Other"  
  
readmission\_data$diag\_3[((readmission\_data$copy\_diag\_3 >= 390) & (readmission\_data$copy\_diag\_3 <= 459)) | (readmission\_data$copy\_diag\_3 == 785)] <- "Circulatory"  
  
readmission\_data$diag\_3[((readmission\_data$copy\_diag\_3 >= 460) & (readmission\_data$copy\_diag\_3 <= 519)) | (readmission\_data$copy\_diag\_3 == 786)] <- 'Respiratory'  
  
readmission\_data$diag\_3[((readmission\_data$copy\_diag\_3 >= 520) & (readmission\_data$copy\_diag\_3 <= 579)) | (readmission\_data$copy\_diag\_3 == 787)] <- 'Digestive'  
  
readmission\_data$diag\_3[((readmission\_data$copy\_diag\_3 >= 250) & (readmission\_data$copy\_diag\_3 < 251))] <- 'Diabetes'  
  
readmission\_data$diag\_3[((readmission\_data$copy\_diag\_3 >= 800) & (readmission\_data$copy\_diag\_3 <= 999))] <- 'Injury'  
  
readmission\_data$diag\_3[((readmission\_data$copy\_diag\_3 >= 710) & (readmission\_data$copy\_diag\_3 <= 739))] <- 'Musculoskeletal'  
  
readmission\_data$diag\_3[((readmission\_data$copy\_diag\_3 >= 580) & (readmission\_data$copy\_diag\_3 <= 629)) | (readmission\_data$copy\_diag\_3 == 788)] <- 'Genitourinary'  
  
readmission\_data$diag\_3[((readmission\_data$copy\_diag\_3 >= 140) & (readmission\_data$copy\_diag\_3 <= 239))] <- 'Neoplasm'  
  
# drop temporary columns for diagnoses  
readmission\_data <- subset(readmission\_data, select = -c(copy\_diag\_1, copy\_diag\_2, copy\_diag\_3))  
  
# change diag attributes back into factors  
readmission\_data$diag\_1 <- as.factor(readmission\_data$diag\_1)  
readmission\_data$diag\_2 <- as.factor(readmission\_data$diag\_2)  
readmission\_data$diag\_3 <- as.factor(readmission\_data$diag\_3)  
  
# Rename values for unknown entries to 'Unknown' if applicable & create summaries for categorical variables  
  
# race ... not well balanced (e.g. code Asian represents less than 1% of data) create levels: Caucasian and NotCaucasian  
levels(readmission\_data$race)[!(levels(readmission\_data$race) == "Caucasian")] <- 'NotCaucasian'  
summary (readmission\_data$race)

## NotCaucasian Caucasian   
## 17682 52305

# admission\_type\_id ... unbalanced data, so retain codes for 1 (Emergency), 2 (Urgent), and 3 (Elective) and relabel the rest as Uncommon (note: codes 5, 6, and 8 represent missing or unknown data)  
readmission\_data$admission\_type\_id <- as.factor(readmission\_data$admission\_type\_id)  
levels(readmission\_data$admission\_type\_id)[levels(readmission\_data$admission\_type\_id) %in% c('4', '5', '6', '7', '8')] <- 'Uncommon'  
levels(readmission\_data$admission\_type\_id)[levels(readmission\_data$admission\_type\_id) == '1'] <- 'Emergency'  
levels(readmission\_data$admission\_type\_id)[levels(readmission\_data$admission\_type\_id) == '2'] <- 'Urgent'  
levels(readmission\_data$admission\_type\_id)[levels(readmission\_data$admission\_type\_id) == '3'] <- 'Elective'  
summary (readmission\_data$admission\_type\_id)

## Emergency Urgent Elective Uncommon   
## 35478 12803 13786 7920

# discharge disposition id ... many levels and unbalanced ... combine all levels that represent less than 2% of the observations into an "Uncommon" category  
readmission\_data$discharge\_disposition\_id <- as.factor(readmission\_data$discharge\_disposition\_id)  
# discharge\_disposition\_id ... codes 18, 25, and 26 represent missing or unknown data  
levels(readmission\_data$discharge\_disposition\_id)[levels(readmission\_data$discharge\_disposition\_id) %in% c('18', '25', '26')] <- 'Unknown'  
# discharge\_disposition\_id ... codes 12, 16, and 17 represent outpatient transfer / discharge /admission   
levels(readmission\_data$discharge\_disposition\_id)[levels(readmission\_data$discharge\_disposition\_id) %in% c('12', '16', '17')] <- 'Outpatient'  
levels(readmission\_data$discharge\_disposition\_id)[levels(readmission\_data$discharge\_disposition\_id) == '1'] <- 'Home'  
levels(readmission\_data$discharge\_disposition\_id)[levels(readmission\_data$discharge\_disposition\_id) == '2'] <- 'Short term hospital'  
levels(readmission\_data$discharge\_disposition\_id)[levels(readmission\_data$discharge\_disposition\_id) == '3'] <- 'SNF'  
levels(readmission\_data$discharge\_disposition\_id)[levels(readmission\_data$discharge\_disposition\_id) == '4'] <- 'ICF'  
levels(readmission\_data$discharge\_disposition\_id)[levels(readmission\_data$discharge\_disposition\_id) == '5'] <- 'Inpatient care institution'  
levels(readmission\_data$discharge\_disposition\_id)[levels(readmission\_data$discharge\_disposition\_id) == '6'] <- 'Home health service'  
levels(readmission\_data$discharge\_disposition\_id)[levels(readmission\_data$discharge\_disposition\_id) == '7'] <- 'Left AMA'  
levels(readmission\_data$discharge\_disposition\_id)[levels(readmission\_data$discharge\_disposition\_id) == '8'] <- 'Home IV provider'  
levels(readmission\_data$discharge\_disposition\_id)[levels(readmission\_data$discharge\_disposition\_id) == '9'] <- 'Inpatient to this hospital'  
levels(readmission\_data$discharge\_disposition\_id)[levels(readmission\_data$discharge\_disposition\_id) == '10'] <- 'Neonatal aftercare'  
levels(readmission\_data$discharge\_disposition\_id)[levels(readmission\_data$discharge\_disposition\_id) == '15'] <- 'Medicare swing bed'  
levels(readmission\_data$discharge\_disposition\_id)[levels(readmission\_data$discharge\_disposition\_id) == '22'] <- 'Rehab'  
levels(readmission\_data$discharge\_disposition\_id)[levels(readmission\_data$discharge\_disposition\_id) == '23'] <- 'Long term care'  
levels(readmission\_data$discharge\_disposition\_id)[levels(readmission\_data$discharge\_disposition\_id) == '24'] <- 'Nursing facility'  
levels(readmission\_data$discharge\_disposition\_id)[levels(readmission\_data$discharge\_disposition\_id) == '27'] <- 'Federal HCF'  
levels(readmission\_data$discharge\_disposition\_id)[levels(readmission\_data$discharge\_disposition\_id) == '28'] <- 'Psychiatric hospital'  
levels(readmission\_data$discharge\_disposition\_id)[levels(readmission\_data$discharge\_disposition\_id) == '29'] <- 'Critical Access Hospital'  
levels(readmission\_data$discharge\_disposition\_id)[levels(readmission\_data$discharge\_disposition\_id) == '30'] <- 'Another HCI'  
# reassign any levels with less than 2% of the data to "Uncommon"  
# calculate the proportions for each level  
props <- table(readmission\_data$discharge\_disposition\_id) / length (readmission\_data$discharge\_disposition\_id)  
levels(readmission\_data$discharge\_disposition\_id)[props < 0.02] <- 'Uncommon'  
# remove any unused levels  
readmission\_data$discharge\_disposition\_id <- factor(readmission\_data$discharge\_disposition\_id)  
summary (readmission\_data$discharge\_disposition\_id)

## Home Short term hospital SNF   
## 44320 1541 8790   
## Uncommon Home health service Unknown   
## 2383 8291 3252   
## Rehab   
## 1410

# admission source id ... many levels and unbalanced ... combine all levels that represent less than 2% of the observations into an "Uncommon" category  
readmission\_data$admission\_source\_id <- as.factor(readmission\_data$admission\_source\_id)  
# admission\_source\_id ... codes 9, 15, 17, 20, and 21 represent missing or unknown data  
levels(readmission\_data$admission\_source\_id)[levels(readmission\_data$admission\_source\_id) %in% c('9', '15', '17', '20', '21')] <- 'Unknown'  
  
# codes 11, 12, 13, 14, 23, 24 refer to babies, childbirth, infants and will be releveled as "Babies"  
levels(readmission\_data$admission\_source\_id)[levels(readmission\_data$admission\_source\_id) %in% c('11', '12', '13', '14', '23', '24')] <- 'Babies'  
  
# codes 4, 26 both mean transfer from hospital ... relevel as "Transfer from hospital"  
levels(readmission\_data$admission\_source\_id)[levels(readmission\_data$admission\_source\_id) %in% c('4', '26')] <- 'Transfer from hospital'  
  
levels(readmission\_data$admission\_source\_id)[levels(readmission\_data$admission\_source\_id) == '1'] <- 'Physician Referral'  
levels(readmission\_data$admission\_source\_id)[levels(readmission\_data$admission\_source\_id) == '2'] <- 'Clinic Referral'  
levels(readmission\_data$admission\_source\_id)[levels(readmission\_data$admission\_source\_id) == '3'] <- 'HMO Referral'  
levels(readmission\_data$admission\_source\_id)[levels(readmission\_data$admission\_source\_id) == '5'] <- 'Transfer from SNF'  
levels(readmission\_data$admission\_source\_id)[levels(readmission\_data$admission\_source\_id) == '6'] <- 'Transfer from HCF'  
levels(readmission\_data$admission\_source\_id)[levels(readmission\_data$admission\_source\_id) == '7'] <- 'ER'  
levels(readmission\_data$admission\_source\_id)[levels(readmission\_data$admission\_source\_id) == '8'] <- 'Court'  
levels(readmission\_data$admission\_source\_id)[levels(readmission\_data$admission\_source\_id) == '10'] <- 'Transfer from CAH'  
levels(readmission\_data$admission\_source\_id)[levels(readmission\_data$admission\_source\_id) == '18'] <- 'Transfer From HHA'  
levels(readmission\_data$admission\_source\_id)[levels(readmission\_data$admission\_source\_id) == '19'] <- 'Readmission to HHA'  
levels(readmission\_data$admission\_source\_id)[levels(readmission\_data$admission\_source\_id) == '22'] <- 'Transfer from inpatients'  
levels(readmission\_data$admission\_source\_id)[levels(readmission\_data$admission\_source\_id) == '25'] <- 'Transfer from ASC'  
# reassign any levels with less than 2% of the data to "Uncommon"  
# calculate the proportions for each level  
props <- table(readmission\_data$admission\_source\_id) / length (readmission\_data$admission\_source\_id)  
levels(readmission\_data$admission\_source\_id)[props < 0.25] <- 'Uncommon'  
# remove unused factors  
readmission\_data$admission\_source\_id <- factor(readmission\_data$admission\_source\_id)  
summary (readmission\_data$admission\_source\_id)

## Physician Referral Uncommon ER   
## 21748 10968 37271

# payer\_code ... many levels and not well balanced (e.g. code FR only has 1 observation)  
# create three levels: Unkown, MC (Medicaid), and Uncommon  
levels(readmission\_data$payer\_code)[levels(readmission\_data$payer\_code) == "?"] <- 'Unknown'  
levels(readmission\_data$payer\_code)[!(levels(readmission\_data$payer\_code) == "Unknown") & !(levels(readmission\_data$payer\_code) == "MC")] <- 'Uncommon'  
summary (readmission\_data$payer\_code)

## Unknown Uncommon MC   
## 30415 19777 19795

# medical specialty ... reassign any levels with less than 5% of the data to "Uncommon"  
# calculate the proportions for each level  
levels(readmission\_data$medical\_specialty)[levels(readmission\_data$medical\_specialty) == "?"] <- 'Unknown'  
props <- table(readmission\_data$medical\_specialty) / length (readmission\_data$medical\_specialty)  
levels(readmission\_data$medical\_specialty)[props < 0.05] <- 'Uncommon'  
summary (readmission\_data$medical\_specialty)

## Unknown Uncommon Cardiology   
## 33652 12116 4207   
## Emergency/Trauma Family/GeneralPractice InternalMedicine   
## 4393 4978 10641

# drop all medication attributes (due to data sparseness) except insulin  
readmission\_data <- subset(readmission\_data, select = -c(24:40,42:46))  
  
# drop patient\_nbr and encounter\_id since these are unique for each observation  
readmission\_data <- subset(readmission\_data, select = -c(encounter\_id, patient\_nbr))  
  
# gender  
summary (readmission\_data$gender)

## Female Male   
## 37239 32748

# age ... choose the midpoint of the age range given and convert to a numeric variables  
for (i in (0:9))   
 {  
 x <- paste("[", i\*10, "-", (i+1)\*10, ")", sep = "")  
 mid\_x <- as.character(5 + 10\*i)  
 levels(readmission\_data$age)[levels(readmission\_data$age) == x] <- mid\_x  
 }  
readmission\_data$age <- as.numeric(levels(readmission\_data$age))[readmission\_data$age]  
summary (readmission\_data$age)

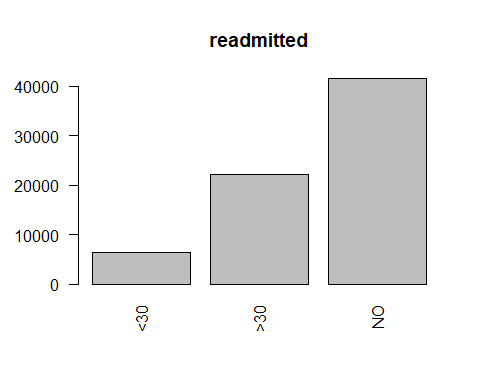
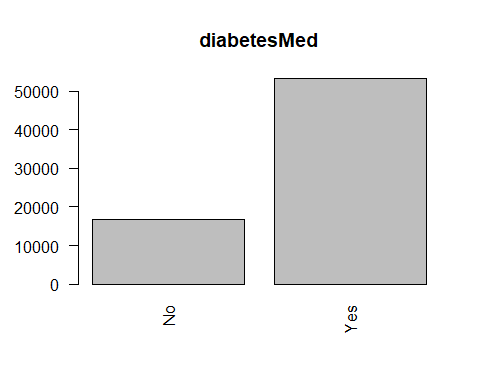
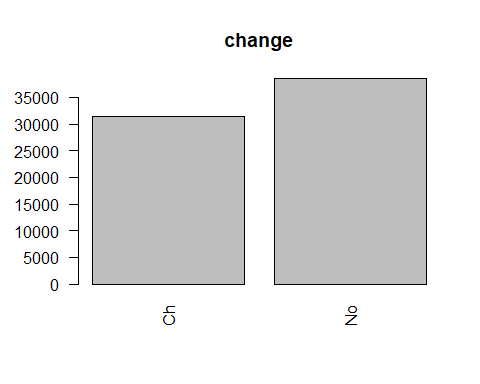
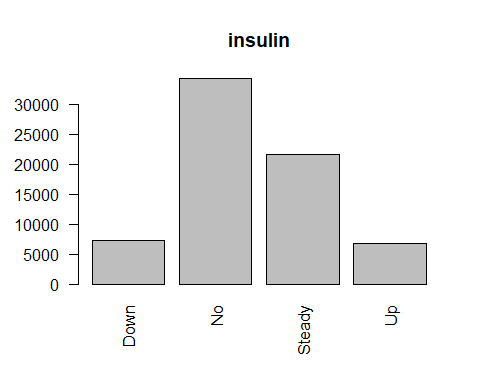
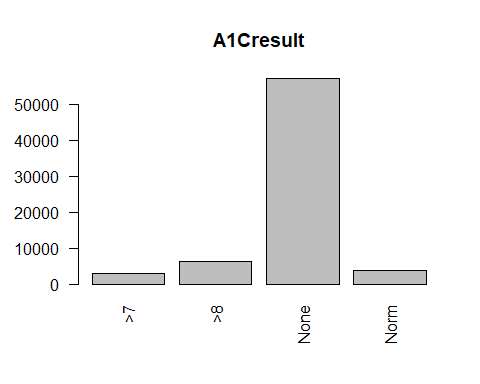
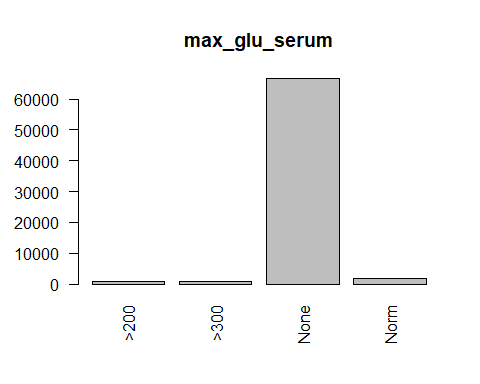
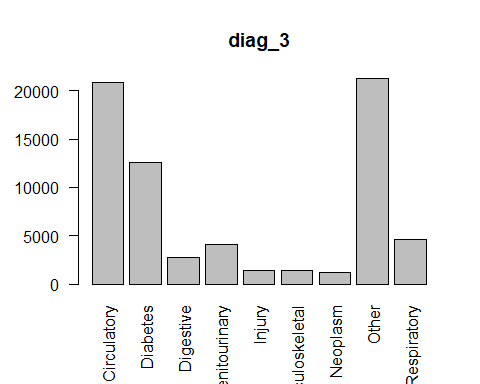
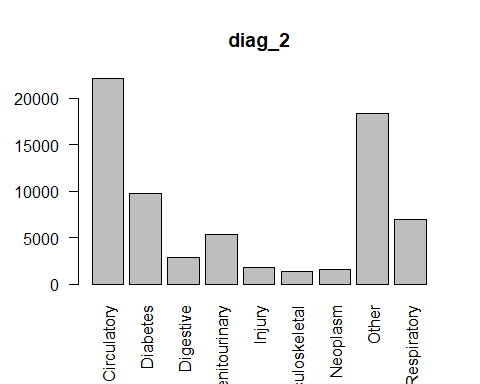
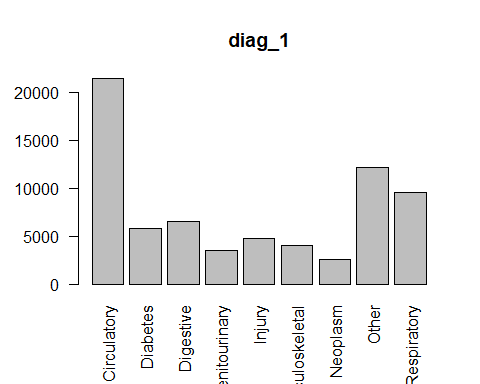
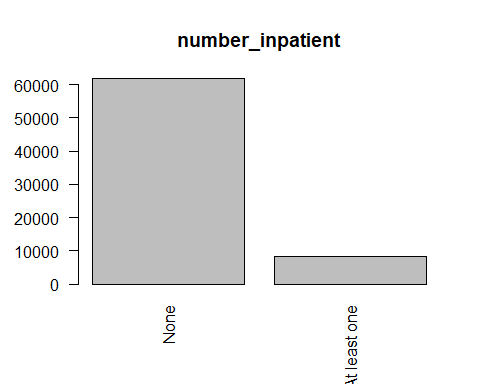
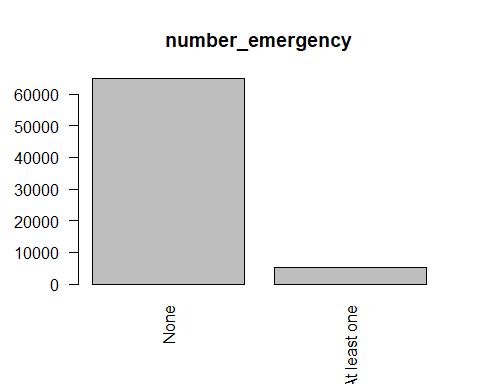
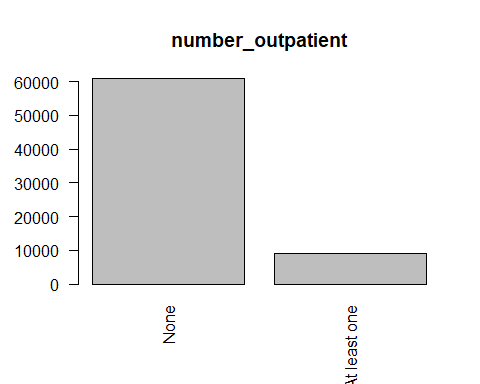
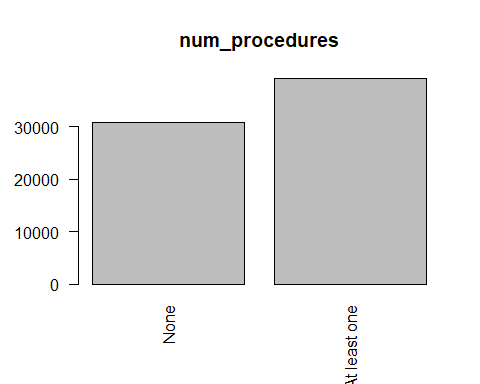
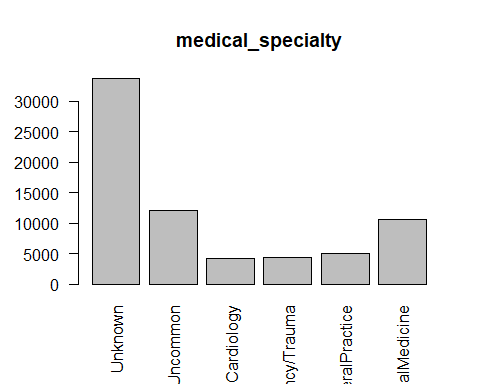
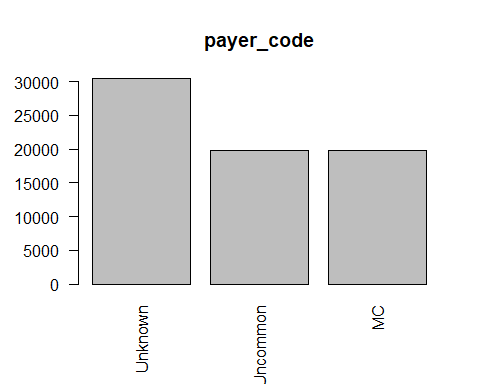
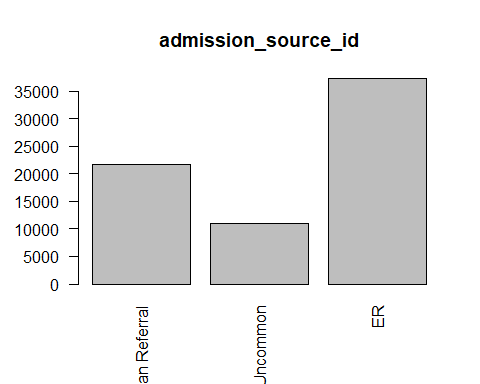
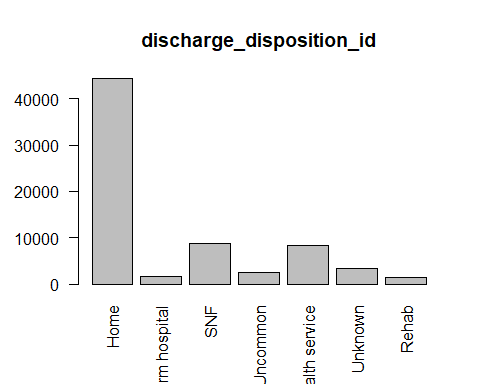
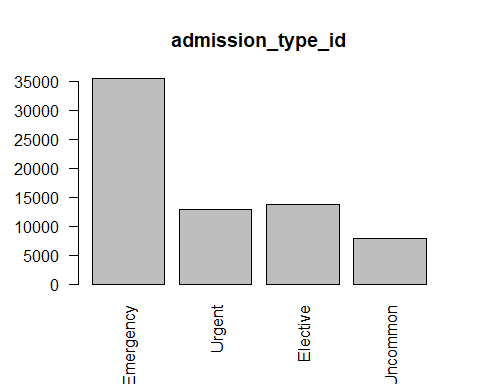
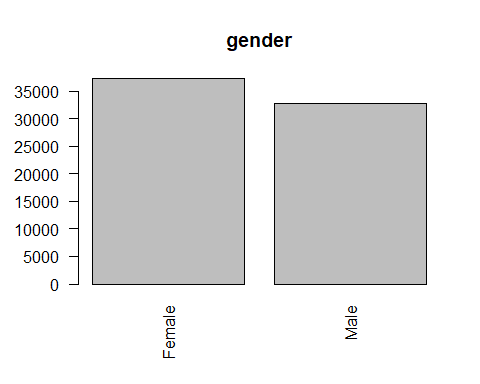
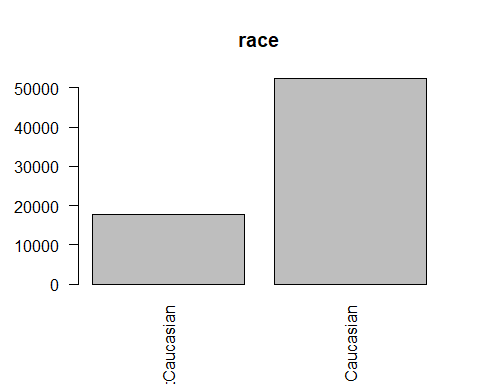
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 5.0 55.0 65.0 65.4 75.0 95.0

#convert num\_procedures, number\_outpatient, number\_emergency, and number\_inpatient to categorical variables with values "None" and "At least one" since most are zeroes ... not normal  
convert\_function <- function(x){  
 x[x > 0] <- 1  
 x <- as.factor(x)  
 levels(x)[levels(x)=="0"] <- "None"  
 levels(x)[levels(x)=="1"] <- "At least one"  
 return(x)  
}  
  
readmission\_data$num\_procedures <- convert\_function(readmission\_data$num\_procedures)  
readmission\_data$number\_outpatient <- convert\_function(readmission\_data$number\_outpatient)  
readmission\_data$number\_emergency <- convert\_function(readmission\_data$number\_emergency)  
readmission\_data$number\_inpatient <- convert\_function(readmission\_data$number\_inpatient)

# Basic summary statistics revisited after initial data cleaning and attribute selection  
# Make a vector of categorical (factor) variables  
a<- (sapply(readmission\_data,class) == "factor")  
  
# Categorical variables (use describe function of Hmisc package to display summary)  
cat\_vars <- readmission\_data[, a]  
Hmisc::describe(cat\_vars)

## cat\_vars   
##   
## 20 Variables 69987 Observations  
## ---------------------------------------------------------------------------  
## race   
## n missing distinct   
## 69987 0 2   
##   
## Value NotCaucasian Caucasian  
## Frequency 17682 52305  
## Proportion 0.253 0.747  
## ---------------------------------------------------------------------------  
## gender   
## n missing distinct   
## 69987 0 2   
##   
## Value Female Male  
## Frequency 37239 32748  
## Proportion 0.532 0.468  
## ---------------------------------------------------------------------------  
## admission\_type\_id   
## n missing distinct   
## 69987 0 4   
##   
## Value Emergency Urgent Elective Uncommon  
## Frequency 35478 12803 13786 7920  
## Proportion 0.507 0.183 0.197 0.113  
## ---------------------------------------------------------------------------  
## discharge\_disposition\_id   
## n missing distinct   
## 69987 0 7   
##   
## Home (44320, 0.633), Short term hospital (1541, 0.022), SNF (8790, 0.126),  
## Uncommon (2383, 0.034), Home health service (8291, 0.118), Unknown (3252,  
## 0.046), Rehab (1410, 0.020)  
## ---------------------------------------------------------------------------  
## admission\_source\_id   
## n missing distinct   
## 69987 0 3   
##   
## Value Physician Referral Uncommon ER  
## Frequency 21748 10968 37271  
## Proportion 0.311 0.157 0.533  
## ---------------------------------------------------------------------------  
## payer\_code   
## n missing distinct   
## 69987 0 3   
##   
## Value Unknown Uncommon MC  
## Frequency 30415 19777 19795  
## Proportion 0.435 0.283 0.283  
## ---------------------------------------------------------------------------  
## medical\_specialty   
## n missing distinct   
## 69987 0 6   
##   
## Unknown (33652, 0.481), Uncommon (12116, 0.173), Cardiology (4207, 0.060),  
## Emergency/Trauma (4393, 0.063), Family/GeneralPractice (4978, 0.071),  
## InternalMedicine (10641, 0.152)  
## ---------------------------------------------------------------------------  
## num\_procedures   
## n missing distinct   
## 69987 0 2   
##   
## Value None At least one  
## Frequency 30841 39146  
## Proportion 0.441 0.559  
## ---------------------------------------------------------------------------  
## number\_outpatient   
## n missing distinct   
## 69987 0 2   
##   
## Value None At least one  
## Frequency 60865 9122  
## Proportion 0.87 0.13  
## ---------------------------------------------------------------------------  
## number\_emergency   
## n missing distinct   
## 69987 0 2   
##   
## Value None At least one  
## Frequency 64886 5101  
## Proportion 0.927 0.073  
## ---------------------------------------------------------------------------  
## number\_inpatient   
## n missing distinct   
## 69987 0 2   
##   
## Value None At least one  
## Frequency 61792 8195  
## Proportion 0.883 0.117  
## ---------------------------------------------------------------------------  
## diag\_1   
## n missing distinct   
## 69987 0 9   
##   
## Circulatory (21389, 0.306), Diabetes (5748, 0.082), Digestive (6488,  
## 0.093), Genitourinary (3441, 0.049), Injury (4694, 0.067), Musculoskeletal  
## (4064, 0.058), Neoplasm (2538, 0.036), Other (12134, 0.173), Respiratory  
## (9491, 0.136)  
## ---------------------------------------------------------------------------  
## diag\_2   
## n missing distinct   
## 69987 0 9   
##   
## Circulatory (22083, 0.316), Diabetes (9700, 0.139), Digestive (2856,  
## 0.041), Genitourinary (5330, 0.076), Injury (1822, 0.026), Musculoskeletal  
## (1295, 0.019), Neoplasm (1600, 0.023), Other (18373, 0.263), Respiratory  
## (6928, 0.099)  
## ---------------------------------------------------------------------------  
## diag\_3   
## n missing distinct   
## 69987 0 9   
##   
## Circulatory (20866, 0.298), Diabetes (12547, 0.179), Digestive (2700,  
## 0.039), Genitourinary (4049, 0.058), Injury (1409, 0.020), Musculoskeletal  
## (1368, 0.020), Neoplasm (1146, 0.016), Other (21250, 0.304), Respiratory  
## (4652, 0.066)  
## ---------------------------------------------------------------------------  
## max\_glu\_serum   
## n missing distinct   
## 69987 0 4   
##   
## Value >200 >300 None Norm  
## Frequency 936 712 66638 1701  
## Proportion 0.013 0.010 0.952 0.024  
## ---------------------------------------------------------------------------  
## A1Cresult   
## n missing distinct   
## 69987 0 4   
##   
## Value >7 >8 None Norm  
## Frequency 2866 6239 57141 3741  
## Proportion 0.041 0.089 0.816 0.053  
## ---------------------------------------------------------------------------  
## insulin   
## n missing distinct   
## 69987 0 4   
##   
## Value Down No Steady Up  
## Frequency 7324 34265 21621 6777  
## Proportion 0.105 0.490 0.309 0.097  
## ---------------------------------------------------------------------------  
## change   
## n missing distinct   
## 69987 0 2   
##   
## Value Ch No  
## Frequency 31495 38492  
## Proportion 0.45 0.55  
## ---------------------------------------------------------------------------  
## diabetesMed   
## n missing distinct   
## 69987 0 2   
##   
## Value No Yes  
## Frequency 16685 53302  
## Proportion 0.238 0.762  
## ---------------------------------------------------------------------------  
## readmitted   
## n missing distinct   
## 69987 0 3   
##   
## Value <30 >30 NO  
## Frequency 6285 22226 41476  
## Proportion 0.090 0.318 0.593  
## ---------------------------------------------------------------------------

for (cat\_var in (1:ncol(cat\_vars))){  
 barplot(table(cat\_vars[, cat\_var]), main = names(cat\_vars[cat\_var]), las=2)  
}



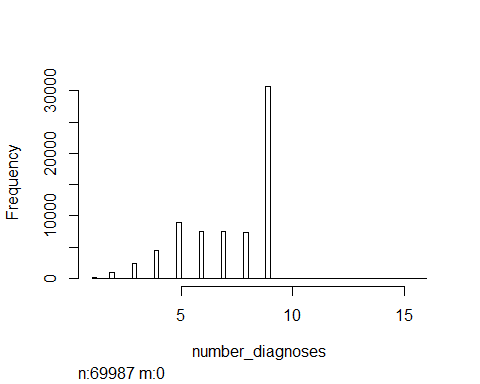
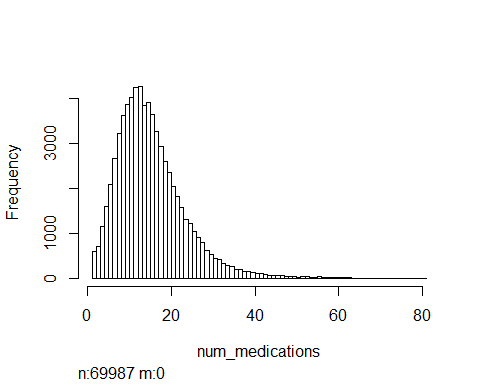
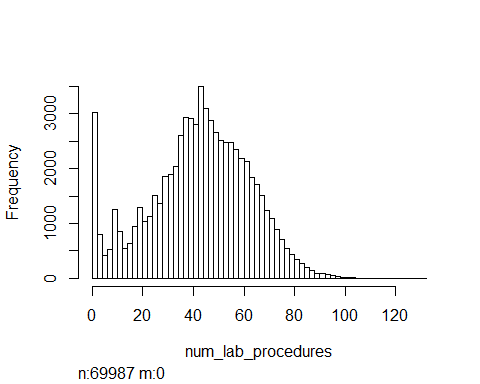
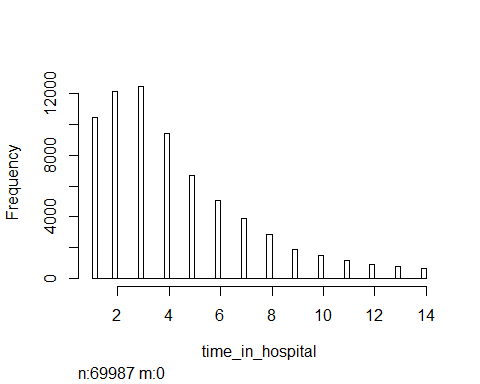
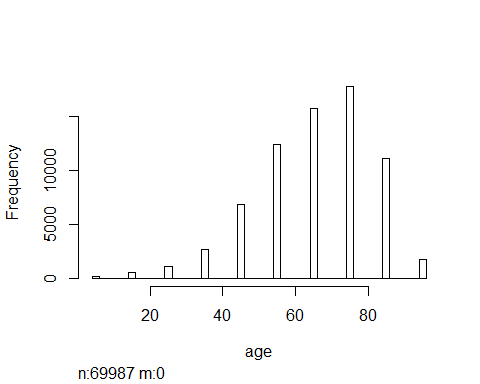
# Numerical variables (use stat.desc function of pastecs package to display summary)  
num\_vars <- readmission\_data[, !a]  
options(scipen=100)  
options(digits=1)  
stat.desc(num\_vars)

## age time\_in\_hospital num\_lab\_procedures  
## nbr.val 69987.00 69987.00 69987.00  
## nbr.null 0.00 0.00 0.00  
## nbr.na 0.00 0.00 0.00  
## min 5.00 1.00 1.00  
## max 95.00 14.00 132.00  
## range 90.00 13.00 131.00  
## sum 4580125.00 299078.00 3000753.00  
## median 65.00 3.00 44.00  
## mean 65.44 4.27 42.88  
## SE.mean 0.06 0.01 0.08  
## CI.mean.0.95 0.12 0.02 0.15  
## var 255.17 8.61 395.78  
## std.dev 15.97 2.93 19.89  
## coef.var 0.24 0.69 0.46  
## num\_medications number\_diagnoses  
## nbr.val 69987.00 69987.000  
## nbr.null 0.00 0.000  
## nbr.na 0.00 0.000  
## min 1.00 1.000  
## max 81.00 16.000  
## range 80.00 15.000  
## sum 1096366.00 505598.000  
## median 14.00 8.000  
## mean 15.67 7.224  
## SE.mean 0.03 0.008  
## CI.mean.0.95 0.06 0.015  
## var 68.68 4.005  
## std.dev 8.29 2.001  
## coef.var 0.53 0.277

summary(num\_vars)

## age time\_in\_hospital num\_lab\_procedures num\_medications  
## Min. : 5 Min. : 1 Min. : 1 Min. : 1   
## 1st Qu.:55 1st Qu.: 2 1st Qu.: 31 1st Qu.:10   
## Median :65 Median : 3 Median : 44 Median :14   
## Mean :65 Mean : 4 Mean : 43 Mean :16   
## 3rd Qu.:75 3rd Qu.: 6 3rd Qu.: 57 3rd Qu.:20   
## Max. :95 Max. :14 Max. :132 Max. :81   
## number\_diagnoses  
## Min. : 1   
## 1st Qu.: 6   
## Median : 8   
## Mean : 7   
## 3rd Qu.: 9   
## Max. :16

for (num\_var in (1:ncol(num\_vars))){  
 hist(num\_vars[num\_var])  
}



# Check skew and kurtosis for age, time\_in\_hospital, num\_lab\_procedures, and num\_medications to determine normality  
kurt <- kurtosis(num\_vars$age)  
skew <- skewness(num\_vars$age)  
print (paste("The distribution of the age attribute has skewness ", format(skew, digits = 2), " and kurtosis ", format(kurt, digits = 2, trim = FALSE, justify = "right"), "." , sep = ""))

## [1] "The distribution of the age attribute has skewness -0.63 and kurtosis 0.35."

kurt <- kurtosis(num\_vars$time\_in\_hospital)  
skew <- skewness(num\_vars$time\_in\_hospital)  
print (paste("The distribution of the time\_in\_hospital attribute has skewness ", format(skew, digits = 2), " and kurtosis ", format(kurt, digits = 2), "." , sep = ""))

## [1] "The distribution of the time\_in\_hospital attribute has skewness 1.2 and kurtosis 1."

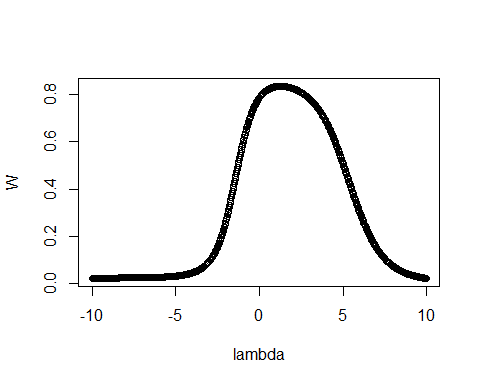
kurt <- kurtosis(num\_vars$num\_lab\_procedures)  
skew <- skewness(num\_vars$num\_lab\_procedures)  
print (paste("The distribution of the num\_lab\_procedures attribute has skewness ", format(skew, digits = 2), " and kurtosis ", format(kurt, digits = 2), "." , sep = ""))

## [1] "The distribution of the num\_lab\_procedures attribute has skewness -0.22 and kurtosis -0.3."

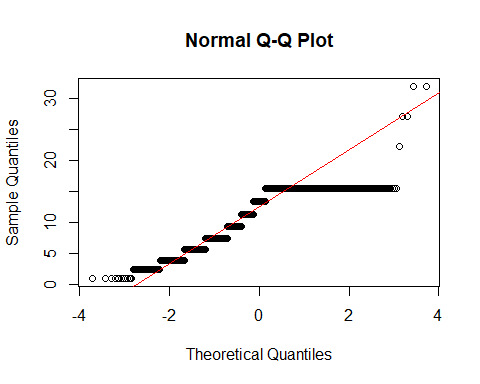
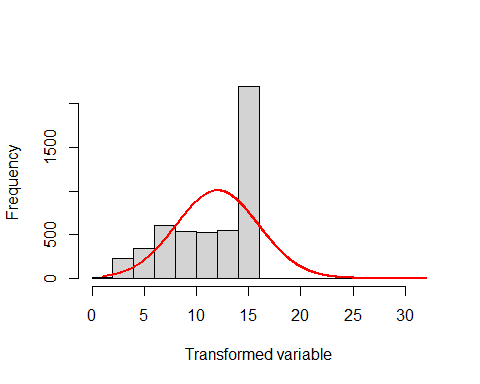
kurt <- kurtosis(num\_vars$num\_medications)  
skew <- skewness(num\_vars$num\_medications)  
print (paste("The distribution of the num\_medications attribute has skewness ", format(skew, digits = 2), " and kurtosis ", format(kurt, digits = 2), "." , sep = ""))

## [1] "The distribution of the num\_medications attribute has skewness 1.4 and kurtosis 3.9."

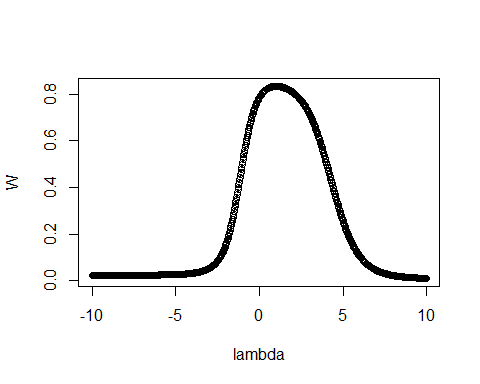
# Attempt to transform number of diagnoses attribute for normality  
# Since dataset is too large to run Tukey's Ladder of Powers, choose a random sample from the dataset, perform the Tukey transformation and then apply the result to the entire dataset.  
num\_var\_sample <- sample\_n (num\_vars, 5000)  
tuk <- transformTukey(num\_var\_sample$number\_diagnoses)



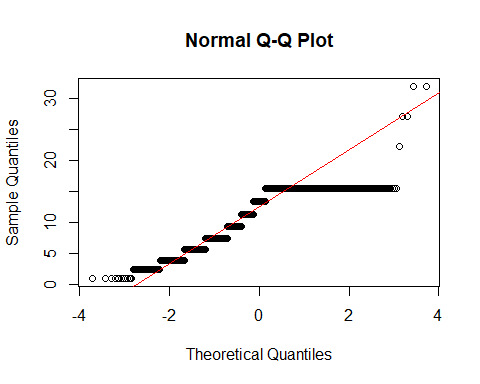
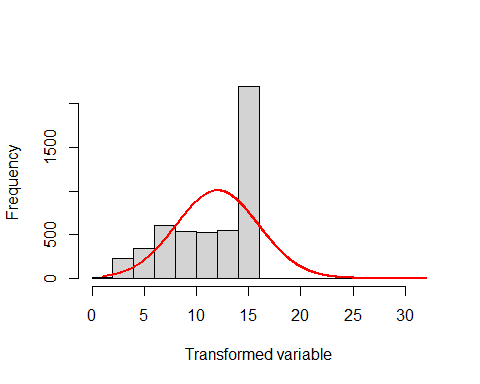
##   
## lambda W  
## 451 1 0.8  
## Shapiro.p.value  
## 451 0.0000000000000000000000000000000000000000000000000000000004  
##   
## if (lambda > 0){TRANS = x ^ lambda}   
## if (lambda == 0){TRANS = log(x)}   
## if (lambda < 0){TRANS = -1 \* x ^ lambda}



tuk2 <- transformTukey(tuk)



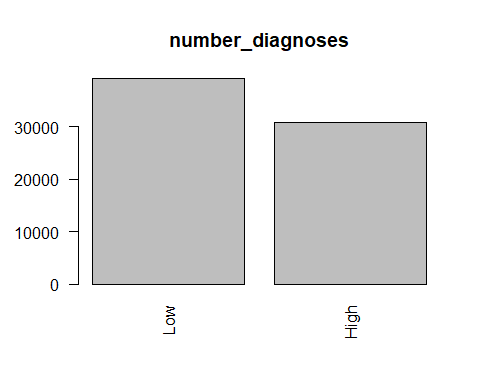
##   
## lambda W  
## 441 1 0.8  
## Shapiro.p.value  
## 441 0.0000000000000000000000000000000000000000000000000000000004  
##   
## if (lambda > 0){TRANS = x ^ lambda}   
## if (lambda == 0){TRANS = log(x)}   
## if (lambda < 0){TRANS = -1 \* x ^ lambda}



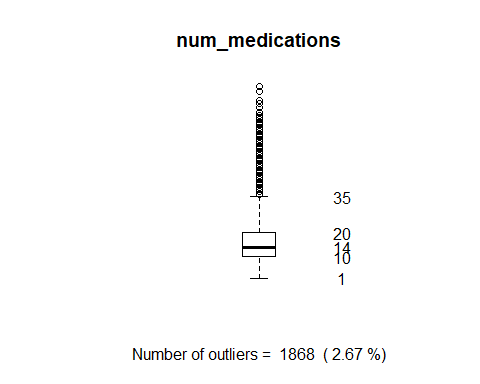
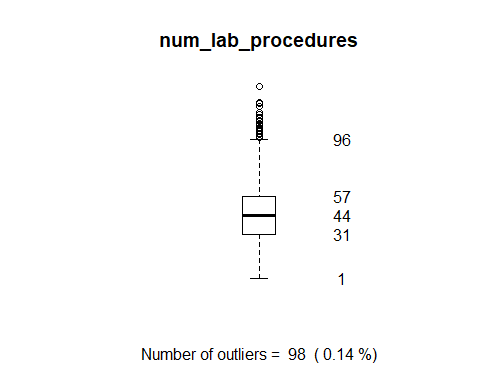
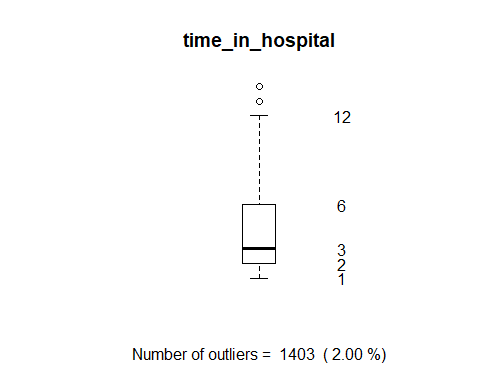
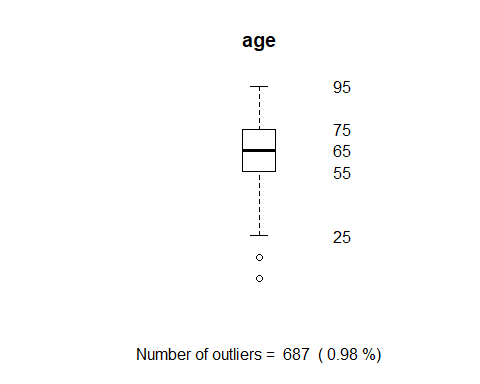
# strong evidence of non-normality even after Tukey transformation ... using binning instead (convert to categorical)  
readmission\_data$number\_diagnoses[readmission\_data$number\_diagnoses <= 8] <- 0  
readmission\_data$number\_diagnoses[readmission\_data$number\_diagnoses >= 9] <- 1  
readmission\_data$number\_diagnoses <- as.factor(readmission\_data$number\_diagnoses)  
levels(readmission\_data$number\_diagnoses)[levels(readmission\_data$number\_diagnoses)=="0"] <- "Low"  
levels(readmission\_data$number\_diagnoses)[levels(readmission\_data$number\_diagnoses)=="1"] <- "High"  
  
#include number\_diagnoses in dataframe for categorical variables and remove from numerical variables dataframe  
a<- (sapply(readmission\_data,class) == "factor")  
cat\_vars <- readmission\_data[, a]  
num\_vars <- readmission\_data[, !a]  
Hmisc::describe(cat\_vars[15])

## cat\_vars[15]   
##   
## 1 Variables 69987 Observations  
## ---------------------------------------------------------------------------  
## number\_diagnoses   
## n missing distinct   
## 69987 0 2   
##   
## Value Low High  
## Frequency 39196 30791  
## Proportion 0.6 0.4  
## ---------------------------------------------------------------------------

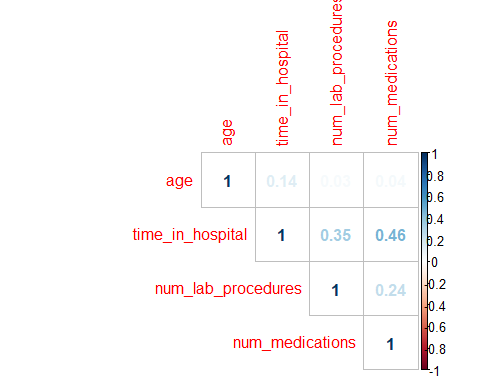
barplot(table(cat\_vars[15]), main = names(cat\_vars[15]), las=2)



# Treatment of Outliers  
# Boxplots of numerical attributes  
  
num\_obs <- nrow(num\_vars) # total number of records  
  
# loop through all numeric variables to create a boxplot for each one  
for (num\_var in (1:ncol(num\_vars))) {  
 current\_num\_var <- num\_vars[[num\_var]]  
 num\_outliers <- length(boxplot.stats(current\_num\_var)$out)  
 percent\_outliers <- format(round(100 \* num\_outliers / num\_obs, 2), nsmall = 2)  
 boxplot(current\_num\_var, axes = FALSE, main=names(num\_vars)[num\_var], boxwex=0.2, xlab = paste("Number of outliers = ", num\_outliers, " (", percent\_outliers, "%)"))  
 text(y = boxplot.stats(current\_num\_var)$stats, labels = boxplot.stats(current\_num\_var)$stats, x = 1.25)  
 }



#  
# Check correlation between pairs of numeric attributes (use Spearman due to non-normality in some of attributes)  
num\_cor <- cor(num\_vars, method = "spearman")  
corrplot(num\_cor, method="number", type="upper")



# no significant correlation between pairs of numeric attributes  
  
# loop through pairs of categorical variables checking G-test for correlation significance (use 0.01 significance level)  
num\_cat\_vars <- ncol(cat\_vars)  
num\_num\_vars <- ncol(num\_vars)  
  
for (i in 1:(num\_cat\_vars - 1)){  
 for (j in (i + 1) : num\_cat\_vars) {  
 gtest <- GTest(cat\_vars[,i], cat\_vars[,j])  
 if (gtest$p.value <= 0.05) {  
 print (paste(names(cat\_vars[i]), " and ", names(cat\_vars[j]), " are not independent.", format(gtest$p.value, digits = 4)))  
 } else {  
 print (paste(names(cat\_vars[i]), " and ", names(cat\_vars[j]), " are independent.", format(gtest$p.value, digits = 4)))  
 }  
 }   
}

## [1] "race and gender are not independent. 0"  
## [1] "race and admission\_type\_id are not independent. 0"  
## [1] "race and discharge\_disposition\_id are not independent. 0"  
## [1] "race and admission\_source\_id are not independent. 0"  
## [1] "race and payer\_code are not independent. 0"  
## [1] "race and medical\_specialty are not independent. 0"  
## [1] "race and num\_procedures are not independent. 0.000000000000001887"  
## [1] "race and number\_outpatient are not independent. 0"  
## [1] "race and number\_emergency are not independent. 0.000001292"  
## [1] "race and number\_inpatient are not independent. 0.000000000000001554"  
## [1] "race and diag\_1 are not independent. 0"  
## [1] "race and diag\_2 are not independent. 0"  
## [1] "race and diag\_3 are not independent. 0.00000000000000111"  
## [1] "race and number\_diagnoses are not independent. 0"  
## [1] "race and max\_glu\_serum are not independent. 0"  
## [1] "race and A1Cresult are not independent. 0"  
## [1] "race and insulin are not independent. 0"  
## [1] "race and change are not independent. 0.009256"  
## [1] "race and diabetesMed are not independent. 0.006732"  
## [1] "race and readmitted are not independent. 0"  
## [1] "gender and admission\_type\_id are not independent. 0.0006538"  
## [1] "gender and discharge\_disposition\_id are not independent. 0"  
## [1] "gender and admission\_source\_id are independent. 0.3954"  
## [1] "gender and payer\_code are not independent. 0.0000000000000191"  
## [1] "gender and medical\_specialty are not independent. 0"  
## [1] "gender and num\_procedures are not independent. 0"  
## [1] "gender and number\_outpatient are not independent. 0.0003337"  
## [1] "gender and number\_emergency are not independent. 0.00000000006211"  
## [1] "gender and number\_inpatient are independent. 0.6813"  
## [1] "gender and diag\_1 are not independent. 0"  
## [1] "gender and diag\_2 are not independent. 0"  
## [1] "gender and diag\_3 are not independent. 0"  
## [1] "gender and number\_diagnoses are independent. 0.6208"  
## [1] "gender and max\_glu\_serum are independent. 0.6583"  
## [1] "gender and A1Cresult are not independent. 0.0000000000006762"  
## [1] "gender and insulin are independent. 0.1411"  
## [1] "gender and change are not independent. 0.0000005038"  
## [1] "gender and diabetesMed are not independent. 0.0000003247"  
## [1] "gender and readmitted are not independent. 0.00005725"  
## [1] "admission\_type\_id and discharge\_disposition\_id are not independent. 0"  
## [1] "admission\_type\_id and admission\_source\_id are not independent. 0"  
## [1] "admission\_type\_id and payer\_code are not independent. 0"  
## [1] "admission\_type\_id and medical\_specialty are not independent. 0"  
## [1] "admission\_type\_id and num\_procedures are not independent. 0"  
## [1] "admission\_type\_id and number\_outpatient are not independent. 0"  
## [1] "admission\_type\_id and number\_emergency are not independent. 0"  
## [1] "admission\_type\_id and number\_inpatient are not independent. 0"  
## [1] "admission\_type\_id and diag\_1 are not independent. 0"  
## [1] "admission\_type\_id and diag\_2 are not independent. 0"  
## [1] "admission\_type\_id and diag\_3 are not independent. 0"  
## [1] "admission\_type\_id and number\_diagnoses are not independent. 0"  
## [1] "admission\_type\_id and max\_glu\_serum are not independent. 0"  
## [1] "admission\_type\_id and A1Cresult are not independent. 0"  
## [1] "admission\_type\_id and insulin are not independent. 0"  
## [1] "admission\_type\_id and change are not independent. 0.0001168"  
## [1] "admission\_type\_id and diabetesMed are not independent. 0.00002823"  
## [1] "admission\_type\_id and readmitted are not independent. 0"  
## [1] "discharge\_disposition\_id and admission\_source\_id are not independent. 0"  
## [1] "discharge\_disposition\_id and payer\_code are not independent. 0"  
## [1] "discharge\_disposition\_id and medical\_specialty are not independent. 0"  
## [1] "discharge\_disposition\_id and num\_procedures are not independent. 0"  
## [1] "discharge\_disposition\_id and number\_outpatient are not independent. 0"  
## [1] "discharge\_disposition\_id and number\_emergency are not independent. 0"  
## [1] "discharge\_disposition\_id and number\_inpatient are not independent. 0"  
## [1] "discharge\_disposition\_id and diag\_1 are not independent. 0"  
## [1] "discharge\_disposition\_id and diag\_2 are not independent. 0"  
## [1] "discharge\_disposition\_id and diag\_3 are not independent. 0"  
## [1] "discharge\_disposition\_id and number\_diagnoses are not independent. 0"  
## [1] "discharge\_disposition\_id and max\_glu\_serum are not independent. 0"  
## [1] "discharge\_disposition\_id and A1Cresult are not independent. 0"  
## [1] "discharge\_disposition\_id and insulin are not independent. 0"  
## [1] "discharge\_disposition\_id and change are not independent. 0"  
## [1] "discharge\_disposition\_id and diabetesMed are not independent. 0"  
## [1] "discharge\_disposition\_id and readmitted are not independent. 0"  
## [1] "admission\_source\_id and payer\_code are not independent. 0"  
## [1] "admission\_source\_id and medical\_specialty are not independent. 0"  
## [1] "admission\_source\_id and num\_procedures are not independent. 0"  
## [1] "admission\_source\_id and number\_outpatient are not independent. 0"  
## [1] "admission\_source\_id and number\_emergency are not independent. 0"  
## [1] "admission\_source\_id and number\_inpatient are not independent. 0"  
## [1] "admission\_source\_id and diag\_1 are not independent. 0"  
## [1] "admission\_source\_id and diag\_2 are not independent. 0"  
## [1] "admission\_source\_id and diag\_3 are not independent. 0"  
## [1] "admission\_source\_id and number\_diagnoses are not independent. 0"  
## [1] "admission\_source\_id and max\_glu\_serum are not independent. 0"  
## [1] "admission\_source\_id and A1Cresult are not independent. 0"  
## [1] "admission\_source\_id and insulin are not independent. 0"  
## [1] "admission\_source\_id and change are not independent. 0.00000000001768"  
## [1] "admission\_source\_id and diabetesMed are independent. 0.8534"  
## [1] "admission\_source\_id and readmitted are not independent. 0"  
## [1] "payer\_code and medical\_specialty are not independent. 0"  
## [1] "payer\_code and num\_procedures are not independent. 0"  
## [1] "payer\_code and number\_outpatient are not independent. 0"  
## [1] "payer\_code and number\_emergency are not independent. 0"  
## [1] "payer\_code and number\_inpatient are not independent. 0"  
## [1] "payer\_code and diag\_1 are not independent. 0"  
## [1] "payer\_code and diag\_2 are not independent. 0"  
## [1] "payer\_code and diag\_3 are not independent. 0"  
## [1] "payer\_code and number\_diagnoses are not independent. 0"  
## [1] "payer\_code and max\_glu\_serum are not independent. 0"  
## [1] "payer\_code and A1Cresult are not independent. 0"  
## [1] "payer\_code and insulin are not independent. 0"  
## [1] "payer\_code and change are not independent. 0"  
## [1] "payer\_code and diabetesMed are not independent. 0"  
## [1] "payer\_code and readmitted are not independent. 0"  
## [1] "medical\_specialty and num\_procedures are not independent. 0"  
## [1] "medical\_specialty and number\_outpatient are not independent. 0"  
## [1] "medical\_specialty and number\_emergency are not independent. 0"  
## [1] "medical\_specialty and number\_inpatient are not independent. 0"  
## [1] "medical\_specialty and diag\_1 are not independent. 0"  
## [1] "medical\_specialty and diag\_2 are not independent. 0"  
## [1] "medical\_specialty and diag\_3 are not independent. 0"  
## [1] "medical\_specialty and number\_diagnoses are not independent. 0"  
## [1] "medical\_specialty and max\_glu\_serum are not independent. 0"  
## [1] "medical\_specialty and A1Cresult are not independent. 0"  
## [1] "medical\_specialty and insulin are not independent. 0"  
## [1] "medical\_specialty and change are not independent. 0"  
## [1] "medical\_specialty and diabetesMed are not independent. 0.00000000001467"  
## [1] "medical\_specialty and readmitted are not independent. 0"  
## [1] "num\_procedures and number\_outpatient are independent. 0.8701"  
## [1] "num\_procedures and number\_emergency are not independent. 0.000000000001229"  
## [1] "num\_procedures and number\_inpatient are independent. 0.4589"  
## [1] "num\_procedures and diag\_1 are not independent. 0"  
## [1] "num\_procedures and diag\_2 are not independent. 0"  
## [1] "num\_procedures and diag\_3 are not independent. 0"  
## [1] "num\_procedures and number\_diagnoses are not independent. 0"  
## [1] "num\_procedures and max\_glu\_serum are not independent. 0"  
## [1] "num\_procedures and A1Cresult are not independent. 0"  
## [1] "num\_procedures and insulin are independent. 0.1381"  
## [1] "num\_procedures and change are not independent. 0.04982"  
## [1] "num\_procedures and diabetesMed are not independent. 0.000000000225"  
## [1] "num\_procedures and readmitted are not independent. 0"  
## [1] "number\_outpatient and number\_emergency are not independent. 0"  
## [1] "number\_outpatient and number\_inpatient are not independent. 0"  
## [1] "number\_outpatient and diag\_1 are not independent. 0"  
## [1] "number\_outpatient and diag\_2 are not independent. 0.000001026"  
## [1] "number\_outpatient and diag\_3 are not independent. 0.0000002413"  
## [1] "number\_outpatient and number\_diagnoses are not independent. 0"  
## [1] "number\_outpatient and max\_glu\_serum are not independent. 0"  
## [1] "number\_outpatient and A1Cresult are not independent. 0"  
## [1] "number\_outpatient and insulin are not independent. 0"  
## [1] "number\_outpatient and change are not independent. 0.0000000000002063"  
## [1] "number\_outpatient and diabetesMed are not independent. 0.000000000000001776"  
## [1] "number\_outpatient and readmitted are not independent. 0"  
## [1] "number\_emergency and number\_inpatient are not independent. 0"  
## [1] "number\_emergency and diag\_1 are not independent. 0"  
## [1] "number\_emergency and diag\_2 are not independent. 0.00000000000001066"  
## [1] "number\_emergency and diag\_3 are not independent. 0.000000000001447"  
## [1] "number\_emergency and number\_diagnoses are not independent. 0"  
## [1] "number\_emergency and max\_glu\_serum are not independent. 0"  
## [1] "number\_emergency and A1Cresult are not independent. 0.00004262"  
## [1] "number\_emergency and insulin are not independent. 0"  
## [1] "number\_emergency and change are not independent. 0"  
## [1] "number\_emergency and diabetesMed are not independent. 0"  
## [1] "number\_emergency and readmitted are not independent. 0"  
## [1] "number\_inpatient and diag\_1 are not independent. 0"  
## [1] "number\_inpatient and diag\_2 are not independent. 0"  
## [1] "number\_inpatient and diag\_3 are not independent. 0.00000000000019"  
## [1] "number\_inpatient and number\_diagnoses are not independent. 0"  
## [1] "number\_inpatient and max\_glu\_serum are independent. 0.317"  
## [1] "number\_inpatient and A1Cresult are not independent. 0"  
## [1] "number\_inpatient and insulin are not independent. 0"  
## [1] "number\_inpatient and change are independent. 0.1696"  
## [1] "number\_inpatient and diabetesMed are not independent. 0.00000000003077"  
## [1] "number\_inpatient and readmitted are not independent. 0"  
## [1] "diag\_1 and diag\_2 are not independent. 0"  
## [1] "diag\_1 and diag\_3 are not independent. 0"  
## [1] "diag\_1 and number\_diagnoses are not independent. 0"  
## [1] "diag\_1 and max\_glu\_serum are not independent. 0"  
## [1] "diag\_1 and A1Cresult are not independent. 0"  
## [1] "diag\_1 and insulin are not independent. 0"  
## [1] "diag\_1 and change are not independent. 0"  
## [1] "diag\_1 and diabetesMed are not independent. 0"  
## [1] "diag\_1 and readmitted are not independent. 0"  
## [1] "diag\_2 and diag\_3 are not independent. 0"  
## [1] "diag\_2 and number\_diagnoses are not independent. 0"  
## [1] "diag\_2 and max\_glu\_serum are not independent. 0"  
## [1] "diag\_2 and A1Cresult are not independent. 0"  
## [1] "diag\_2 and insulin are not independent. 0"  
## [1] "diag\_2 and change are not independent. 0"  
## [1] "diag\_2 and diabetesMed are not independent. 0"  
## [1] "diag\_2 and readmitted are not independent. 0"  
## [1] "diag\_3 and number\_diagnoses are not independent. 0"  
## [1] "diag\_3 and max\_glu\_serum are not independent. 0.000000000003274"  
## [1] "diag\_3 and A1Cresult are not independent. 0"  
## [1] "diag\_3 and insulin are not independent. 0"  
## [1] "diag\_3 and change are not independent. 0"  
## [1] "diag\_3 and diabetesMed are not independent. 0.0000000000001038"  
## [1] "diag\_3 and readmitted are not independent. 0"  
## [1] "number\_diagnoses and max\_glu\_serum are not independent. 0"  
## [1] "number\_diagnoses and A1Cresult are not independent. 0"  
## [1] "number\_diagnoses and insulin are not independent. 0"  
## [1] "number\_diagnoses and change are not independent. 0"  
## [1] "number\_diagnoses and diabetesMed are not independent. 0.000000000005147"  
## [1] "number\_diagnoses and readmitted are not independent. 0"  
## [1] "max\_glu\_serum and A1Cresult are not independent. 0"  
## [1] "max\_glu\_serum and insulin are not independent. 0"  
## [1] "max\_glu\_serum and change are not independent. 0"  
## [1] "max\_glu\_serum and diabetesMed are not independent. 0"  
## [1] "max\_glu\_serum and readmitted are not independent. 0.0000007331"  
## [1] "A1Cresult and insulin are not independent. 0"  
## [1] "A1Cresult and change are not independent. 0"  
## [1] "A1Cresult and diabetesMed are not independent. 0"  
## [1] "A1Cresult and readmitted are not independent. 0.000000006354"  
## [1] "insulin and change are not independent. 0"  
## [1] "insulin and diabetesMed are not independent. 0"  
## [1] "insulin and readmitted are not independent. 0"  
## [1] "change and diabetesMed are not independent. 0"  
## [1] "change and readmitted are not independent. 0"  
## [1] "diabetesMed and readmitted are not independent. 0"

# ANOVA to check independence of categorical and numerical attributes  
for (i in (1 : num\_num\_vars)) {  
 for (j in (1 : num\_cat\_vars)) {  
 aov\_result <- aov(num\_vars[[i]] ~ cat\_vars[[j]])  
 aov\_sum <- unlist(summary(aov\_result))  
 if (aov\_sum["Pr(>F)1"] <= 0.05) {  
 print (paste(names(num\_vars[i]), " and ", names(cat\_vars[j]), " are not independent. ", aov\_sum["Pr(>F)1"]))  
 } else {  
 print (paste(names(num\_vars[i]), " and ", names(cat\_vars[j]), " are independent. ", aov\_sum["Pr(>F)1"]))  
 }  
 }  
}

## [1] "age and race are not independent. 0"  
## [1] "age and gender are not independent. 0.00000000000000000000000000000000000000000000741796731809181"  
## [1] "age and admission\_type\_id are not independent. 0.000000000000442845271342735"  
## [1] "age and discharge\_disposition\_id are not independent. 0"  
## [1] "age and admission\_source\_id are not independent. 0.000000000000000000000000000000147064515223353"  
## [1] "age and payer\_code are not independent. 0"  
## [1] "age and medical\_specialty are not independent. 2.93919795718214e-172"  
## [1] "age and num\_procedures are not independent. 0.0000549060425928571"  
## [1] "age and number\_outpatient are not independent. 0.00000000000000241041429087982"  
## [1] "age and number\_emergency are not independent. 0.0000000000000000136214120555395"  
## [1] "age and number\_inpatient are not independent. 0.0000000000000000000420544538539246"  
## [1] "age and diag\_1 are not independent. 0"  
## [1] "age and diag\_2 are not independent. 0"  
## [1] "age and diag\_3 are not independent. 0"  
## [1] "age and number\_diagnoses are not independent. 0"  
## [1] "age and max\_glu\_serum are not independent. 0.0000000000000000000000000000000000000000396861852625204"  
## [1] "age and A1Cresult are not independent. 0"  
## [1] "age and insulin are not independent. 4.67507537709423e-173"  
## [1] "age and change are not independent. 0.000000000000000000117209125716034"  
## [1] "age and diabetesMed are not independent. 0.0000000211795605980208"  
## [1] "age and readmitted are not independent. 0.000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000318989384154782"  
## [1] "time\_in\_hospital and race are independent. 0.589188422143392"  
## [1] "time\_in\_hospital and gender are not independent. 0.0000000000292598052662173"  
## [1] "time\_in\_hospital and admission\_type\_id are not independent. 0.0000000000000000000000108710580113847"  
## [1] "time\_in\_hospital and discharge\_disposition\_id are not independent. 0"  
## [1] "time\_in\_hospital and admission\_source\_id are not independent. 0.000000000000000000000000000000000000000000000000000000641601709688819"  
## [1] "time\_in\_hospital and payer\_code are not independent. 8.29060111900671e-113"  
## [1] "time\_in\_hospital and medical\_specialty are not independent. 0.00000000000000000000000000000000000000000000000000000000000000000000000000000000259489955018829"  
## [1] "time\_in\_hospital and num\_procedures are not independent. 0"  
## [1] "time\_in\_hospital and number\_outpatient are not independent. 0.0000000000215240996064405"  
## [1] "time\_in\_hospital and number\_emergency are independent. 0.139308126705471"  
## [1] "time\_in\_hospital and number\_inpatient are not independent. 0.00000000000000000000000000000000000000000000000000000000000000000000000000000173941380428506"  
## [1] "time\_in\_hospital and diag\_1 are not independent. 2.32611737036527e-130"  
## [1] "time\_in\_hospital and diag\_2 are not independent. 1.24458950266259e-242"  
## [1] "time\_in\_hospital and diag\_3 are not independent. 1.7599179329145e-266"  
## [1] "time\_in\_hospital and number\_diagnoses are not independent. 0"  
## [1] "time\_in\_hospital and max\_glu\_serum are not independent. 0.00000000000000000000000018623421606505"  
## [1] "time\_in\_hospital and A1Cresult are not independent. 0.0000000000000000000000000000000000000000000000000000000000000000000000000000000000000926513914319535"  
## [1] "time\_in\_hospital and insulin are not independent. 3.91948972178203e-295"  
## [1] "time\_in\_hospital and change are not independent. 2.97866721276935e-193"  
## [1] "time\_in\_hospital and diabetesMed are not independent. 0.0000000000000000000000000000000000000000000000000000000000000000000000000710111650826825"  
## [1] "time\_in\_hospital and readmitted are not independent. 0.000000000000000000000000000000000000000000000000000000000000000000000000000000000512672007735593"  
## [1] "num\_lab\_procedures and race are not independent. 0.0000000215783226191629"  
## [1] "num\_lab\_procedures and gender are independent. 0.208033951439136"  
## [1] "num\_lab\_procedures and admission\_type\_id are not independent. 0"  
## [1] "num\_lab\_procedures and discharge\_disposition\_id are not independent. 0.00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000354753738247493"  
## [1] "num\_lab\_procedures and admission\_source\_id are not independent. 0"  
## [1] "num\_lab\_procedures and payer\_code are not independent. 0.000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000114424730936202"  
## [1] "num\_lab\_procedures and medical\_specialty are not independent. 0"  
## [1] "num\_lab\_procedures and num\_procedures are not independent. 0.00000313817519016568"  
## [1] "num\_lab\_procedures and number\_outpatient are not independent. 0.00000000000000157718112687631"  
## [1] "num\_lab\_procedures and number\_emergency are not independent. 0.00000408916076332311"  
## [1] "num\_lab\_procedures and number\_inpatient are not independent. 0.0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000544246906322565"  
## [1] "num\_lab\_procedures and diag\_1 are not independent. 0"  
## [1] "num\_lab\_procedures and diag\_2 are not independent. 2.51185972885371e-142"  
## [1] "num\_lab\_procedures and diag\_3 are not independent. 7.676637435032e-138"  
## [1] "num\_lab\_procedures and number\_diagnoses are not independent. 0"  
## [1] "num\_lab\_procedures and max\_glu\_serum are not independent. 3.44350593969052e-289"  
## [1] "num\_lab\_procedures and A1Cresult are not independent. 0"  
## [1] "num\_lab\_procedures and insulin are not independent. 1.54480292829679e-241"  
## [1] "num\_lab\_procedures and change are not independent. 0.0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000356704982625781"  
## [1] "num\_lab\_procedures and diabetesMed are not independent. 0.00000000000000000000000000000000475488565772446"  
## [1] "num\_lab\_procedures and readmitted are not independent. 0.000000000000000000000000000000000000000000000000701503978176949"  
## [1] "num\_medications and race are not independent. 0.000000000000000000000000000000000000000000000000000000000788469894152961"  
## [1] "num\_medications and gender are not independent. 0.000580638036770454"  
## [1] "num\_medications and admission\_type\_id are not independent. 0"  
## [1] "num\_medications and discharge\_disposition\_id are not independent. 0"  
## [1] "num\_medications and admission\_source\_id are not independent. 9.36025627305454e-269"  
## [1] "num\_medications and payer\_code are not independent. 0.0000000000000000000000000000000000000000000000531768799127784"  
## [1] "num\_medications and medical\_specialty are not independent. 1.47869436611537e-294"  
## [1] "num\_medications and num\_procedures are not independent. 0"  
## [1] "num\_medications and number\_outpatient are not independent. 0.000000000000000000000578887236158152"  
## [1] "num\_medications and number\_emergency are not independent. 0.00319232095384498"  
## [1] "num\_medications and number\_inpatient are not independent. 0.000000000000000000000000000000916927815301032"  
## [1] "num\_medications and diag\_1 are not independent. 0"  
## [1] "num\_medications and diag\_2 are not independent. 0"  
## [1] "num\_medications and diag\_3 are not independent. 1.60416230107751e-307"  
## [1] "num\_medications and number\_diagnoses are not independent. 0"  
## [1] "num\_medications and max\_glu\_serum are not independent. 0.0000000000000000000000000242387140626336"  
## [1] "num\_medications and A1Cresult are not independent. 0.000000000369376585823622"  
## [1] "num\_medications and insulin are not independent. 0"  
## [1] "num\_medications and change are not independent. 0"  
## [1] "num\_medications and diabetesMed are not independent. 0"  
## [1] "num\_medications and readmitted are not independent. 0.0000000000000000000000000000000767466460995823"

# Normalize the numerical attributes to assist support vector machine algorithm (since it uses a distance measure)  
n <- !(sapply(readmission\_data,class) == "factor") # numeric attributes  
min\_values <- sapply(readmission\_data[, n], min) # store minimum values for future "unscaling"  
max\_values <- sapply(readmission\_data[, n], max) # store maxmimum values for future "unscaling"  
readmission\_data[, n] <- sapply(readmission\_data[, n], function(x) {(x - min(x)) / (max(x) - min(x))})

# Create a categorical only dataset. Discretize continuous variables using interval = square root of the range and convert to factors. This is required to convert dataset to transactions for apriori arules algorithm  
num\_vars\_disc <- num\_vars  
discretize <- function (df\_column) {  
  
 interval <- round (sqrt (max(df\_column) - min(df\_column)))  
 df\_column <- cut\_interval(df\_column, n = interval)  
   
 return (df\_column)  
}  
  
for (i in 1:ncol(num\_vars)) {  
 num\_vars\_disc[,i] <- as.factor(discretize (num\_vars[,i]))  
}  
  
# create dataframe with only categorical data for entire readmission dataset  
readmission\_all\_cat <- cbind(num\_vars\_disc, cat\_vars)

# use association rules (apriori algorithm) to induce knowledge about relationships between independent variables and class variable  
  
#convert dataframe to transactions  
readmission\_trans <- as(readmission\_all\_cat, "transactions")  
summary(readmission\_trans)

## transactions as itemMatrix in sparse format with  
## 69987 rows (elements/itemsets/transactions) and  
## 116 columns (items) and a density of 0.2   
##   
## most frequent items:  
## max\_glu\_serum=None number\_emergency=None number\_inpatient=None   
## 66638 64886 61792   
## number\_outpatient=None A1Cresult=None (Other)   
## 60865 57141 1438353   
##   
## element (itemset/transaction) length distribution:  
## sizes  
## 25   
## 69987   
##   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 25 25 25 25 25 25   
##   
## includes extended item information - examples:  
## labels variables levels  
## 1 age=[5,15] age [5,15]  
## 2 age=(15,25] age (15,25]  
## 3 age=(25,35] age (25,35]  
##   
## includes extended transaction information - examples:  
## transactionID  
## 1 4268  
## 2 5828  
## 3 67609

itemLabels(readmission\_trans)

## [1] "age=[5,15]"   
## [2] "age=(15,25]"   
## [3] "age=(25,35]"   
## [4] "age=(35,45]"   
## [5] "age=(45,55]"   
## [6] "age=(55,65]"   
## [7] "age=(65,75]"   
## [8] "age=(75,85]"   
## [9] "age=(85,95]"   
## [10] "time\_in\_hospital=[1,4.25]"   
## [11] "time\_in\_hospital=(4.25,7.5]"   
## [12] "time\_in\_hospital=(7.5,10.8]"   
## [13] "time\_in\_hospital=(10.8,14]"   
## [14] "num\_lab\_procedures=[1,12.9]"   
## [15] "num\_lab\_procedures=(12.9,24.8]"   
## [16] "num\_lab\_procedures=(24.8,36.7]"   
## [17] "num\_lab\_procedures=(36.7,48.6]"   
## [18] "num\_lab\_procedures=(48.6,60.5]"   
## [19] "num\_lab\_procedures=(60.5,72.5]"   
## [20] "num\_lab\_procedures=(72.5,84.4]"   
## [21] "num\_lab\_procedures=(84.4,96.3]"   
## [22] "num\_lab\_procedures=(96.3,108]"   
## [23] "num\_lab\_procedures=(108,120]"   
## [24] "num\_lab\_procedures=(120,132]"   
## [25] "num\_medications=[1,9.89]"   
## [26] "num\_medications=(9.89,18.8]"   
## [27] "num\_medications=(18.8,27.7]"   
## [28] "num\_medications=(27.7,36.6]"   
## [29] "num\_medications=(36.6,45.4]"   
## [30] "num\_medications=(45.4,54.3]"   
## [31] "num\_medications=(54.3,63.2]"   
## [32] "num\_medications=(63.2,72.1]"   
## [33] "num\_medications=(72.1,81]"   
## [34] "race=NotCaucasian"   
## [35] "race=Caucasian"   
## [36] "gender=Female"   
## [37] "gender=Male"   
## [38] "admission\_type\_id=Emergency"   
## [39] "admission\_type\_id=Urgent"   
## [40] "admission\_type\_id=Elective"   
## [41] "admission\_type\_id=Uncommon"   
## [42] "discharge\_disposition\_id=Home"   
## [43] "discharge\_disposition\_id=Short term hospital"  
## [44] "discharge\_disposition\_id=SNF"   
## [45] "discharge\_disposition\_id=Uncommon"   
## [46] "discharge\_disposition\_id=Home health service"  
## [47] "discharge\_disposition\_id=Unknown"   
## [48] "discharge\_disposition\_id=Rehab"   
## [49] "admission\_source\_id=Physician Referral"   
## [50] "admission\_source\_id=Uncommon"   
## [51] "admission\_source\_id=ER"   
## [52] "payer\_code=Unknown"   
## [53] "payer\_code=Uncommon"   
## [54] "payer\_code=MC"   
## [55] "medical\_specialty=Unknown"   
## [56] "medical\_specialty=Uncommon"   
## [57] "medical\_specialty=Cardiology"   
## [58] "medical\_specialty=Emergency/Trauma"   
## [59] "medical\_specialty=Family/GeneralPractice"   
## [60] "medical\_specialty=InternalMedicine"   
## [61] "num\_procedures=None"   
## [62] "num\_procedures=At least one"   
## [63] "number\_outpatient=None"   
## [64] "number\_outpatient=At least one"   
## [65] "number\_emergency=None"   
## [66] "number\_emergency=At least one"   
## [67] "number\_inpatient=None"   
## [68] "number\_inpatient=At least one"   
## [69] "diag\_1=Circulatory"   
## [70] "diag\_1=Diabetes"   
## [71] "diag\_1=Digestive"   
## [72] "diag\_1=Genitourinary"   
## [73] "diag\_1=Injury"   
## [74] "diag\_1=Musculoskeletal"   
## [75] "diag\_1=Neoplasm"   
## [76] "diag\_1=Other"   
## [77] "diag\_1=Respiratory"   
## [78] "diag\_2=Circulatory"   
## [79] "diag\_2=Diabetes"   
## [80] "diag\_2=Digestive"   
## [81] "diag\_2=Genitourinary"   
## [82] "diag\_2=Injury"   
## [83] "diag\_2=Musculoskeletal"   
## [84] "diag\_2=Neoplasm"   
## [85] "diag\_2=Other"   
## [86] "diag\_2=Respiratory"   
## [87] "diag\_3=Circulatory"   
## [88] "diag\_3=Diabetes"   
## [89] "diag\_3=Digestive"   
## [90] "diag\_3=Genitourinary"   
## [91] "diag\_3=Injury"   
## [92] "diag\_3=Musculoskeletal"   
## [93] "diag\_3=Neoplasm"   
## [94] "diag\_3=Other"   
## [95] "diag\_3=Respiratory"   
## [96] "number\_diagnoses=Low"   
## [97] "number\_diagnoses=High"   
## [98] "max\_glu\_serum=>200"   
## [99] "max\_glu\_serum=>300"   
## [100] "max\_glu\_serum=None"   
## [101] "max\_glu\_serum=Norm"   
## [102] "A1Cresult=>7"   
## [103] "A1Cresult=>8"   
## [104] "A1Cresult=None"   
## [105] "A1Cresult=Norm"   
## [106] "insulin=Down"   
## [107] "insulin=No"   
## [108] "insulin=Steady"   
## [109] "insulin=Up"   
## [110] "change=Ch"   
## [111] "change=No"   
## [112] "diabetesMed=No"   
## [113] "diabetesMed=Yes"   
## [114] "readmitted=<30"   
## [115] "readmitted=>30"   
## [116] "readmitted=NO"

# use lower support and confidence for readmitted<=30 because very few rules generated  
rules\_under30 <- apriori(readmission\_trans, parameter = list(supp = 0.001, conf = 0.5, target = "rules"), appearance = list(rhs = "readmitted=<30"))

## Apriori  
##   
## Parameter specification:  
## confidence minval smax arem aval originalSupport maxtime support minlen  
## 0.5 0.1 1 none FALSE TRUE 5 0.001 1  
## maxlen target ext  
## 10 rules FALSE  
##   
## Algorithmic control:  
## filter tree heap memopt load sort verbose  
## 0.1 TRUE TRUE FALSE TRUE 2 TRUE  
##   
## Absolute minimum support count: 69   
##   
## set item appearances ...[1 item(s)] done [0.00s].  
## set transactions ...[116 item(s), 69987 transaction(s)] done [0.08s].  
## sorting and recoding items ... [112 item(s)] done [0.03s].  
## creating transaction tree ... done [0.04s].  
## checking subsets of size 1 2 3 4 5

## Warning in apriori(readmission\_trans, parameter = list(supp = 0.001, conf =  
## 0.5, : Mining stopped (time limit reached). Only patterns up to a length of  
## 5 returned!

## done [28.20s].  
## writing ... [2 rule(s)] done [0.23s].  
## creating S4 object ... done [0.09s].

# use slightly higher support for readmitted>=30 to generate a reasonable number of rules  
rules\_over30 <- apriori(readmission\_trans, parameter = list(supp = 0.015, conf = 0.5, target = "rules"), appearance = list(rhs = "readmitted=>30"))

## Apriori  
##   
## Parameter specification:  
## confidence minval smax arem aval originalSupport maxtime support minlen  
## 0.5 0.1 1 none FALSE TRUE 5 0.01 1  
## maxlen target ext  
## 10 rules FALSE  
##   
## Algorithmic control:  
## filter tree heap memopt load sort verbose  
## 0.1 TRUE TRUE FALSE TRUE 2 TRUE  
##   
## Absolute minimum support count: 1049   
##   
## set item appearances ...[1 item(s)] done [0.00s].  
## set transactions ...[116 item(s), 69987 transaction(s)] done [0.08s].  
## sorting and recoding items ... [105 item(s)] done [0.03s].  
## creating transaction tree ... done [0.05s].  
## checking subsets of size 1 2 3 4 5

## Warning in apriori(readmission\_trans, parameter = list(supp = 0.015, conf =  
## 0.5, : Mining stopped (time limit reached). Only patterns up to a length of  
## 5 returned!

## done [17.36s].  
## writing ... [14 rule(s)] done [0.02s].  
## creating S4 object ... done [0.02s].

# use higher confidence for readmitted=NO to generate a reasonable number of rules  
rules\_NOT <- apriori(readmission\_trans, parameter = list(supp = 0.002, conf = 0.89, target = "rules"), appearance = list(rhs = "readmitted=NO"))

## Apriori  
##   
## Parameter specification:  
## confidence minval smax arem aval originalSupport maxtime support minlen  
## 0.9 0.1 1 none FALSE TRUE 5 0.002 1  
## maxlen target ext  
## 10 rules FALSE  
##   
## Algorithmic control:  
## filter tree heap memopt load sort verbose  
## 0.1 TRUE TRUE FALSE TRUE 2 TRUE  
##   
## Absolute minimum support count: 139   
##   
## set item appearances ...[1 item(s)] done [0.00s].  
## set transactions ...[116 item(s), 69987 transaction(s)] done [0.08s].  
## sorting and recoding items ... [111 item(s)] done [0.03s].  
## creating transaction tree ... done [0.05s].  
## checking subsets of size 1 2 3 4 5

## Warning in apriori(readmission\_trans, parameter = list(supp = 0.002, conf  
## = 0.89, : Mining stopped (time limit reached). Only patterns up to a length  
## of 5 returned!

## done [36.49s].  
## writing ... [6 rule(s)] done [0.16s].  
## creating S4 object ... done [0.06s].

inspect(head(rules\_under30, by = "lift")) # view rules for patients readmitted in under 30 days

## lhs rhs support confidence lift count  
## [1] {discharge\_disposition\_id=Rehab,   
## admission\_source\_id=ER,   
## medical\_specialty=Unknown,   
## number\_diagnoses=Low} => {readmitted=<30} 0.001 0.5 6 82  
## [2] {admission\_type\_id=Emergency,   
## discharge\_disposition\_id=Rehab,   
## medical\_specialty=Unknown,   
## number\_diagnoses=Low} => {readmitted=<30} 0.001 0.5 6 83

inspect(head(rules\_over30, by = "lift")) # view rules for patients readmitted in over 30 days

## lhs rhs support confidence lift count  
## [1] {discharge\_disposition\_id=Home,   
## payer\_code=Unknown,   
## number\_inpatient=At least one,   
## diabetesMed=Yes} => {readmitted=>30} 0.02 0.5 2 1157  
## [2] {race=Caucasian,   
## discharge\_disposition\_id=Home,   
## payer\_code=Unknown,   
## number\_inpatient=At least one} => {readmitted=>30} 0.02 0.5 2 1111  
## [3] {discharge\_disposition\_id=Home,   
## payer\_code=Unknown,   
## number\_inpatient=At least one} => {readmitted=>30} 0.02 0.5 2 1458  
## [4] {discharge\_disposition\_id=Home,   
## num\_procedures=None,   
## number\_inpatient=At least one} => {readmitted=>30} 0.02 0.5 2 1089  
## [5] {discharge\_disposition\_id=Home,   
## payer\_code=Unknown,   
## number\_inpatient=At least one,   
## max\_glu\_serum=None} => {readmitted=>30} 0.02 0.5 2 1326  
## [6] {discharge\_disposition\_id=Home,   
## payer\_code=Unknown,   
## number\_outpatient=None,   
## number\_inpatient=At least one} => {readmitted=>30} 0.02 0.5 2 1151

inspect(head(rules\_NOT, by = "lift")) # view rules for patients not readmitted

## lhs rhs support confidence lift count  
## [1] {age=(25,35],   
## admission\_type\_id=Elective,   
## diag\_1=Other,   
## change=No} => {readmitted=NO} 0.002 0.9 2 160  
## [2] {age=(25,35],   
## admission\_type\_id=Elective,   
## admission\_source\_id=Physician Referral,   
## diabetesMed=No} => {readmitted=NO} 0.002 0.9 2 165  
## [3] {time\_in\_hospital=[1,4.25],   
## admission\_type\_id=Uncommon,   
## payer\_code=Uncommon,   
## diabetesMed=No} => {readmitted=NO} 0.002 0.9 2 147  
## [4] {age=(25,35],   
## gender=Female,   
## admission\_type\_id=Elective,   
## diabetesMed=No} => {readmitted=NO} 0.002 0.9 2 145  
## [5] {admission\_type\_id=Uncommon,   
## payer\_code=Uncommon,   
## number\_emergency=None,   
## diabetesMed=No} => {readmitted=NO} 0.002 0.9 2 151  
## [6] {age=(25,35],   
## admission\_type\_id=Elective,   
## diag\_1=Other,   
## diag\_2=Other} => {readmitted=NO} 0.002 0.9 2 141

# combine readmitted<=30 and readmitted>=30 to create a binary class categorical dataset  
readmission\_binary\_cat <- readmission\_all\_cat  
readmission\_binary\_cat$readmitted <- factor(ifelse(readmission\_binary\_cat$readmitted == "NO","NO","YES"))

# Using the binary class dataset, determine association rules for simply readmitted or not  
readmission\_trans\_binary <- as(readmission\_binary\_cat, "transactions")  
summary(readmission\_trans\_binary)

## transactions as itemMatrix in sparse format with  
## 69987 rows (elements/itemsets/transactions) and  
## 115 columns (items) and a density of 0.2   
##   
## most frequent items:  
## max\_glu\_serum=None number\_emergency=None number\_inpatient=None   
## 66638 64886 61792   
## number\_outpatient=None A1Cresult=None (Other)   
## 60865 57141 1438353   
##   
## element (itemset/transaction) length distribution:  
## sizes  
## 25   
## 69987   
##   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 25 25 25 25 25 25   
##   
## includes extended item information - examples:  
## labels variables levels  
## 1 age=[5,15] age [5,15]  
## 2 age=(15,25] age (15,25]  
## 3 age=(25,35] age (25,35]  
##   
## includes extended transaction information - examples:  
## transactionID  
## 1 4268  
## 2 5828  
## 3 67609

itemLabels(readmission\_trans\_binary)

## [1] "age=[5,15]"   
## [2] "age=(15,25]"   
## [3] "age=(25,35]"   
## [4] "age=(35,45]"   
## [5] "age=(45,55]"   
## [6] "age=(55,65]"   
## [7] "age=(65,75]"   
## [8] "age=(75,85]"   
## [9] "age=(85,95]"   
## [10] "time\_in\_hospital=[1,4.25]"   
## [11] "time\_in\_hospital=(4.25,7.5]"   
## [12] "time\_in\_hospital=(7.5,10.8]"   
## [13] "time\_in\_hospital=(10.8,14]"   
## [14] "num\_lab\_procedures=[1,12.9]"   
## [15] "num\_lab\_procedures=(12.9,24.8]"   
## [16] "num\_lab\_procedures=(24.8,36.7]"   
## [17] "num\_lab\_procedures=(36.7,48.6]"   
## [18] "num\_lab\_procedures=(48.6,60.5]"   
## [19] "num\_lab\_procedures=(60.5,72.5]"   
## [20] "num\_lab\_procedures=(72.5,84.4]"   
## [21] "num\_lab\_procedures=(84.4,96.3]"   
## [22] "num\_lab\_procedures=(96.3,108]"   
## [23] "num\_lab\_procedures=(108,120]"   
## [24] "num\_lab\_procedures=(120,132]"   
## [25] "num\_medications=[1,9.89]"   
## [26] "num\_medications=(9.89,18.8]"   
## [27] "num\_medications=(18.8,27.7]"   
## [28] "num\_medications=(27.7,36.6]"   
## [29] "num\_medications=(36.6,45.4]"   
## [30] "num\_medications=(45.4,54.3]"   
## [31] "num\_medications=(54.3,63.2]"   
## [32] "num\_medications=(63.2,72.1]"   
## [33] "num\_medications=(72.1,81]"   
## [34] "race=NotCaucasian"   
## [35] "race=Caucasian"   
## [36] "gender=Female"   
## [37] "gender=Male"   
## [38] "admission\_type\_id=Emergency"   
## [39] "admission\_type\_id=Urgent"   
## [40] "admission\_type\_id=Elective"   
## [41] "admission\_type\_id=Uncommon"   
## [42] "discharge\_disposition\_id=Home"   
## [43] "discharge\_disposition\_id=Short term hospital"  
## [44] "discharge\_disposition\_id=SNF"   
## [45] "discharge\_disposition\_id=Uncommon"   
## [46] "discharge\_disposition\_id=Home health service"  
## [47] "discharge\_disposition\_id=Unknown"   
## [48] "discharge\_disposition\_id=Rehab"   
## [49] "admission\_source\_id=Physician Referral"   
## [50] "admission\_source\_id=Uncommon"   
## [51] "admission\_source\_id=ER"   
## [52] "payer\_code=Unknown"   
## [53] "payer\_code=Uncommon"   
## [54] "payer\_code=MC"   
## [55] "medical\_specialty=Unknown"   
## [56] "medical\_specialty=Uncommon"   
## [57] "medical\_specialty=Cardiology"   
## [58] "medical\_specialty=Emergency/Trauma"   
## [59] "medical\_specialty=Family/GeneralPractice"   
## [60] "medical\_specialty=InternalMedicine"   
## [61] "num\_procedures=None"   
## [62] "num\_procedures=At least one"   
## [63] "number\_outpatient=None"   
## [64] "number\_outpatient=At least one"   
## [65] "number\_emergency=None"   
## [66] "number\_emergency=At least one"   
## [67] "number\_inpatient=None"   
## [68] "number\_inpatient=At least one"   
## [69] "diag\_1=Circulatory"   
## [70] "diag\_1=Diabetes"   
## [71] "diag\_1=Digestive"   
## [72] "diag\_1=Genitourinary"   
## [73] "diag\_1=Injury"   
## [74] "diag\_1=Musculoskeletal"   
## [75] "diag\_1=Neoplasm"   
## [76] "diag\_1=Other"   
## [77] "diag\_1=Respiratory"   
## [78] "diag\_2=Circulatory"   
## [79] "diag\_2=Diabetes"   
## [80] "diag\_2=Digestive"   
## [81] "diag\_2=Genitourinary"   
## [82] "diag\_2=Injury"   
## [83] "diag\_2=Musculoskeletal"   
## [84] "diag\_2=Neoplasm"   
## [85] "diag\_2=Other"   
## [86] "diag\_2=Respiratory"   
## [87] "diag\_3=Circulatory"   
## [88] "diag\_3=Diabetes"   
## [89] "diag\_3=Digestive"   
## [90] "diag\_3=Genitourinary"   
## [91] "diag\_3=Injury"   
## [92] "diag\_3=Musculoskeletal"   
## [93] "diag\_3=Neoplasm"   
## [94] "diag\_3=Other"   
## [95] "diag\_3=Respiratory"   
## [96] "number\_diagnoses=Low"   
## [97] "number\_diagnoses=High"   
## [98] "max\_glu\_serum=>200"   
## [99] "max\_glu\_serum=>300"   
## [100] "max\_glu\_serum=None"   
## [101] "max\_glu\_serum=Norm"   
## [102] "A1Cresult=>7"   
## [103] "A1Cresult=>8"   
## [104] "A1Cresult=None"   
## [105] "A1Cresult=Norm"   
## [106] "insulin=Down"   
## [107] "insulin=No"   
## [108] "insulin=Steady"   
## [109] "insulin=Up"   
## [110] "change=Ch"   
## [111] "change=No"   
## [112] "diabetesMed=No"   
## [113] "diabetesMed=Yes"   
## [114] "readmitted=NO"   
## [115] "readmitted=YES"

# generate rules for "readmitted=YES"  
rules\_\_binary\_YES <- apriori(readmission\_trans\_binary, parameter = list(supp = 0.04, conf = 0.6, target = "rules"), appearance = list(rhs = "readmitted=YES"))

## Apriori  
##   
## Parameter specification:  
## confidence minval smax arem aval originalSupport maxtime support minlen  
## 0.6 0.1 1 none FALSE TRUE 5 0.04 1  
## maxlen target ext  
## 10 rules FALSE  
##   
## Algorithmic control:  
## filter tree heap memopt load sort verbose  
## 0.1 TRUE TRUE FALSE TRUE 2 TRUE  
##   
## Absolute minimum support count: 2799   
##   
## set item appearances ...[1 item(s)] done [0.00s].  
## set transactions ...[115 item(s), 69987 transaction(s)] done [0.09s].  
## sorting and recoding items ... [88 item(s)] done [0.03s].  
## creating transaction tree ... done [0.05s].  
## checking subsets of size 1 2 3 4 5

## Warning in apriori(readmission\_trans\_binary, parameter = list(supp =  
## 0.04, : Mining stopped (time limit reached). Only patterns up to a length  
## of 5 returned!

## done [10.47s].  
## writing ... [13 rule(s)] done [0.00s].  
## creating S4 object ... done [0.02s].

# use higher confidence for readmitted=NO to generate a reasonable number of rules  
rules\_binary\_NO <- apriori(readmission\_trans\_binary, parameter = list(supp = 0.004, conf = 0.85, target = "rules"), appearance = list(rhs = "readmitted=NO"))

## Apriori  
##   
## Parameter specification:  
## confidence minval smax arem aval originalSupport maxtime support minlen  
## 0.8 0.1 1 none FALSE TRUE 5 0.004 1  
## maxlen target ext  
## 10 rules FALSE  
##   
## Algorithmic control:  
## filter tree heap memopt load sort verbose  
## 0.1 TRUE TRUE FALSE TRUE 2 TRUE  
##   
## Absolute minimum support count: 279   
##   
## set item appearances ...[1 item(s)] done [0.00s].  
## set transactions ...[115 item(s), 69987 transaction(s)] done [0.09s].  
## sorting and recoding items ... [109 item(s)] done [0.04s].  
## creating transaction tree ... done [0.11s].  
## checking subsets of size 1 2 3 4 5

## Warning in apriori(readmission\_trans\_binary, parameter = list(supp =  
## 0.004, : Mining stopped (time limit reached). Only patterns up to a length  
## of 5 returned!

## done [32.19s].  
## writing ... [9 rule(s)] done [0.08s].  
## creating S4 object ... done [0.05s].

inspect(head(rules\_\_binary\_YES, by = "lift")) # view rules for patients readmitted in under 30 days

## lhs rhs support confidence lift count  
## [1] {payer\_code=Unknown,   
## number\_inpatient=At least one} => {readmitted=YES} 0.04 0.6 2 2965  
## [2] {admission\_source\_id=ER,   
## number\_inpatient=At least one} => {readmitted=YES} 0.04 0.6 2 2828  
## [3] {discharge\_disposition\_id=Home,   
## number\_inpatient=At least one} => {readmitted=YES} 0.04 0.6 2 2943  
## [4] {race=Caucasian,   
## number\_inpatient=At least one,   
## diabetesMed=Yes} => {readmitted=YES} 0.04 0.6 2 3138  
## [5] {race=Caucasian,   
## number\_inpatient=At least one} => {readmitted=YES} 0.06 0.6 1 3903  
## [6] {race=Caucasian,   
## number\_inpatient=At least one,   
## max\_glu\_serum=None,   
## diabetesMed=Yes} => {readmitted=YES} 0.04 0.6 1 3001

inspect(head(rules\_binary\_NO, by = "lift")) # view rules for patients readmitted in over 30 days

## lhs rhs support confidence lift count  
## [1] {age=(25,35],   
## admission\_source\_id=Physician Referral,   
## num\_procedures=At least one,   
## diag\_1=Other} => {readmitted=NO} 0.004 0.9 1 280  
## [2] {admission\_source\_id=Uncommon,   
## payer\_code=Uncommon,   
## number\_inpatient=None,   
## diabetesMed=No} => {readmitted=NO} 0.004 0.9 1 293  
## [3] {age=(25,35],   
## gender=Female,   
## num\_procedures=At least one,   
## diag\_1=Other} => {readmitted=NO} 0.005 0.9 1 352  
## [4] {age=(25,35],   
## num\_procedures=At least one,   
## diag\_1=Other,   
## change=No} => {readmitted=NO} 0.004 0.9 1 288  
## [5] {age=(25,35],   
## gender=Female,   
## admission\_source\_id=Physician Referral,   
## num\_procedures=At least one} => {readmitted=NO} 0.005 0.9 1 376  
## [6] {age=(25,35],   
## time\_in\_hospital=[1,4.25],   
## num\_procedures=At least one,   
## diag\_1=Other} => {readmitted=NO} 0.005 0.9 1 349

# Use FSelector and LVQ to identify and remove attributes that are not significantly contributing as predictors of the target variable "readmitted".  
  
#use sample of approximately 10% of data because algorithm takes too long with full dataset running LVQ  
set.seed(136)  
readmission\_sample <- readmission\_all\_cat[sample(1:nrow(readmission\_all\_cat), 7000), ]  
  
# use FSelector information gain to determine most important attributes based on information gain (print those values that are higher than the median)  
att.scores <- information.gain(readmitted ~ ., readmission\_sample)  
att.scores.sig <- att.scores[att.scores$attr\_importance > median(att.scores$attr\_importance),,drop = FALSE]  
top\_F <- att.scores.sig[order(-att.scores.sig),,drop = FALSE]  
  
# Use caret package and learning vector quantization (LVQ) to determine the most important attributes  
# prepare training scheme  
control <- trainControl(method="repeatedcv", number=3, repeats=2)  
# train the model  
lvq\_model <- train(readmitted ~ ., data=readmission\_sample, method="lvq", preProcess="scale", trControl=control)

## Warning in preProcess.default(method = "scale", x = structure(c(0, 0,  
## 0, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(72.1,81]

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(72.1,81]  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(72.1,81]  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(72.1,81]  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(72.1,81]  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(72.1,81]  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(72.1,81]  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(72.1,81]  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(72.1,81]  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(72.1,81]

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(72.1,81]  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(72.1,81]  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(72.1,81]  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(72.1,81]  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(72.1,81]  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(72.1,81]  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(72.1,81]  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(72.1,81]  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(72.1,81]

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(72.1,81]  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(72.1,81]  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(72.1,81]  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(72.1,81]  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(72.1,81]  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(72.1,81]  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(72.1,81]  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(72.1,81]  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(72.1,81]

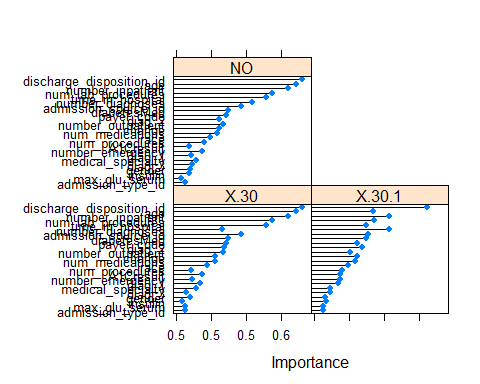
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(72.1,81]  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(72.1,81]  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(72.1,81]  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(72.1,81]  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(72.1,81]  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(72.1,81]  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(72.1,81]  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(72.1,81]  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(72.1,81]

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(63.2,72.1], num\_medications(72.1,81]  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(63.2,72.1], num\_medications(72.1,81]  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(63.2,72.1], num\_medications(72.1,81]  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(63.2,72.1], num\_medications(72.1,81]  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(63.2,72.1], num\_medications(72.1,81]  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(63.2,72.1], num\_medications(72.1,81]  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(63.2,72.1], num\_medications(72.1,81]  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(63.2,72.1], num\_medications(72.1,81]  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(63.2,72.1], num\_medications(72.1,81]

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(72.1,81]  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(72.1,81]  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(72.1,81]  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(72.1,81]  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(72.1,81]  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(72.1,81]  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(72.1,81]  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(72.1,81]  
  
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(72.1,81]

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut  
## = 10, : These variables have zero variances: num\_lab\_procedures(120,132],  
## num\_medications(72.1,81]

# estimate variable importance  
importance <- varImp(lvq\_model, scale=FALSE)  
plot(importance)



# create dataframe with the averages of importance for all three classes and keep only those above median  
top\_L <- importance$importance  
top\_L$average <- rowMeans(importance$importance)  
top\_L <- subset(top\_L, select = c("average"))  
importance.median <- median(top\_L$average)  
top\_L <- top\_L[top\_L$average > importance.median,, drop = FALSE]  
top\_L <- top\_L[order(-top\_L),, drop = FALSE]  
  
# create list of attributes above median for both FSelector and LVQ  
top\_F.names <- rownames(top\_F)  
top\_L.names <- rownames(top\_L)  
for (i in 1:length(top\_F.names)) {  
 if (top\_F.names[i] %in% top\_L.names) {print(top\_F.names[i])}  
}

## [1] "number\_inpatient"  
## [1] "discharge\_disposition\_id"  
## [1] "age"  
## [1] "num\_lab\_procedures"  
## [1] "number\_diagnoses"  
## [1] "time\_in\_hospital"  
## [1] "diag\_2"  
## [1] "diabetesMed"  
## [1] "payer\_code"  
## [1] "number\_outpatient"

#Split the data into training (80%) and test set (20%). Consider possibly treatment for imbalance in the target variable, readmitted.  
  
#Partition the dataset into training and test sets (use the discretized dataframe)  
set.seed(136)  
trainIndex <- createDataPartition(readmission\_data$readmitted, p = .8,   
 list = FALSE,   
 times = 1)  
  
readmitted\_train <- readmission\_data[trainIndex,]   
readmitted\_test <- readmission\_data[-trainIndex,]

# Apply various classification techniques (Naïve Bayes, decision tree, random forest, and support vector machines) to the data. First, use all attributes. Second, use only common attributes found in FSelector information gain and LVQ importance feature selection techniques from above.  
  
# \*\*\*\*\*  
#   
# Function to generate model using cross validation and create confusion matrix.  
# Parameters passed to function are ml\_type (NB for Naive Bayes, DT for decision tree, RF for random forest, and SV for support vector machines), num\_folds (number of folds in cross validation), and readmitted\_df (readmission data dataframe)  
# The function will create a total confusion matrix for the particular model being implemented by summing the confusion matrices of each of fold in the cross validation  
#  
generate\_conf\_matrix <- function (ml\_type, num\_folds, readmitted\_df) {  
  
#Partition the dataset into training and test sets for cross-validation  
 set.seed(136)  
 trainIndex <- createFolds(readmitted\_df$readmitted, k = num\_folds, list = TRUE)  
 conf\_matrix <- matrix(data = 0, nrow = nlevels(readmitted\_df$readmitted), ncol = nlevels(readmitted\_df$readmitted)) #initialize confusion matrix  
  
#Loop through all folds of cross validation, compute confusion matrix, keep running total  
 for (i in 1:num\_folds) {  
   
 r\_train <- readmitted\_df[-trainIndex[[i]],] # use all folds except current index for training  
 r\_test <- readmitted\_df[trainIndex[[i]],] # use current index fold for test  
  
 if (ml\_type == "NB") {model1 <- naiveBayes(readmitted ~ ., data = r\_train, laplace = 1)}   
 else if (ml\_type == "DT") {model1 <- J48(readmitted ~ ., data = r\_train)}  
 else if (ml\_type == "RF") {model1 <- randomForest(readmitted ~ ., data=r\_train, ntree=100)}  
 else if (ml\_type == "SV") {model1 <- svm(readmitted ~ ., data = r\_train, kernel = 'linear', gamma = 0.05, cost = 5)}  
   
 # predict classes based on trained model for test data   
 preds <- predict(model1, newdata = r\_test)  
 # generate confusion matrix   
 new\_conf\_matrix <- table(preds, r\_test$readmitted)  
 # keep running total of conf\_matrix to facilitate calculation of mean after loop finishes  
 conf\_matrix <- conf\_matrix + new\_conf\_matrix  
 }  
  
 # label and return confusion matrix  
 names(dimnames(conf\_matrix)) <- c("Predicted Class", "Actual Class")  
 return(conf\_matrix)  
 }

#calculate the accuracy and F1 score for a confusion matrix (cm). If binary class then use standard method for calculating accuracy and F1 score. If multiclass then use micro (one vs all) approach since class distribution is not even (note that for multiclass confusion matrix, micro precision, recall, and F1 are equal)  
calc\_perf\_meas <- function (cm) {  
 n\_instances = sum(cm) # number of instances  
 n\_classes = ncol(cm) # number of classes  
 n\_correct = diag(cm) # number of correctly classified instances per class   
 rowsums = apply(cm, 1, sum) # number of predictions per class  
 colsums = apply(cm, 2, sum) # number of instances per class  
 tp <- cm[1,1]  
 fp <- cm[1,2]  
 fn <- cm[2,1]  
  
 if (n\_classes == 2) {  
 acc <- sum(n\_correct) / n\_instances # accuracy  
 prec <- tp / (tp + fp) # precision  
 recall <- tp / (tp + fn) # recall  
 f1 <- 2 \* prec \* recall / (prec + recall) # F1 score  
 }   
 else if (n\_classes > 2) {  
#one versus all matrices for each of the 3 class (<30, >30, NO) ... assumes each class in turn is positive  
 oneVsAll = lapply(1 : n\_classes,   
 function(i){  
 v = c(cm[i,i],  
 rowsums[i] - cm[i,i],  
 colsums[i] - cm[i,i],  
 n\_instances - rowsums[i] - colsums[i] + cm[i,i]);  
 return(matrix(v, nrow = 2, byrow = TRUE))})  
 sum\_cm = matrix(0, nrow = 2, ncol = 2)  
 for(i in 1 : n\_classes){sum\_cm = sum\_cm + oneVsAll[[i]]} # add up confusion matrices for classes  
   
 acc <- sum(diag(sum\_cm)) / sum(sum\_cm) # average micro accuracy  
 f1 <- sum\_cm[1,1] / sum(sum\_cm[1,]) #find the micro average F1 score = precision = recall   
 }  
 perf\_measures <- list ("Accuracy" = acc, "F1 Score" = f1)  
 return(perf\_measures)  
}

# Function to run models (uses cross-validation on the training data sent to the function). Parameters passed to function are the dataframe to be used and codes for which models to be run (NB, DT, RF, SV). Use tictoc function to record runtime.  
  
run\_models <- function (df, models\_to\_use) {  
  
 num\_folds <- 4 # set number of folds for cross-validation  
 perf\_measures\_NB <- list ("Accuracy" = 0, "F1 Score" = 0) # initialize performance measures lists  
 perf\_measures\_DT <- list ("Accuracy" = 0, "F1 Score" = 0)  
 perf\_measures\_RF <- list ("Accuracy" = 0, "F1 Score" = 0)  
 perf\_measures\_SV <- list ("Accuracy" = 0, "F1 Score" = 0)  
  
 if ("NB" %in% models\_to\_use) {  
#  
# Naive Bayes  
#  
 tic("Naive Bayes - All attributes")  
 conf\_matrix <- generate\_conf\_matrix ("NB", num\_folds, df)  
 toc()  
 conf\_matrix  
 perf\_measures\_NB <- format(calc\_perf\_meas (conf\_matrix), digits = 3)  
 perf\_measures\_NB  
 }  
  
 if ("DT" %in% models\_to\_use) {  
#  
# Decision Tree  
#  
 tic("Decision Tree - All attributes")  
 conf\_matrix <- generate\_conf\_matrix ("DT", num\_folds, df)  
 toc()  
 conf\_matrix  
 perf\_measures\_DT <- format(calc\_perf\_meas (conf\_matrix), digits = 3)  
 perf\_measures\_DT  
 }  
  
 if ("RF" %in% models\_to\_use) {  
#  
# Random forest  
#  
 tic("Random Forest - All attributes")  
 conf\_matrix <- generate\_conf\_matrix ("RF", num\_folds, df)  
 toc()  
 conf\_matrix  
 perf\_measures\_RF <- format(calc\_perf\_meas (conf\_matrix), digits = 3)  
 perf\_measures\_RF  
 }  
   
 if ("SV" %in% models\_to\_use) {   
#  
# Support Vector Machine  
#  
 tic("Support Vector Machine - All attributes")  
 conf\_matrix <- generate\_conf\_matrix ("SV", num\_folds, df)  
 toc()  
 conf\_matrix  
 perf\_measures\_SV <- format(calc\_perf\_meas (conf\_matrix), digits = 3)  
 perf\_measures\_SV  
 }  
 perf\_measures <- list (NB = perf\_measures\_NB, DT = perf\_measures\_DT, RF = perf\_measures\_RF, SV = perf\_measures\_SV)  
 return(perf\_measures)  
}

#  
# Run the models using all attributes.  
#  
perf\_measures <- run\_models(readmitted\_train, c("NB", "DT", "RF")) # not using SV because it takes too long

## Naive Bayes - All attributes: 52.1 sec elapsed  
## Decision Tree - All attributes: 22.59 sec elapsed  
## Random Forest - All attributes: 211.17 sec elapsed

perf\_measures

## $NB  
## Accuracy F1 Score   
## "0.725" "0.588"   
##   
## $DT  
## Accuracy F1 Score   
## "0.716" "0.574"   
##   
## $RF  
## Accuracy F1 Score   
## "0.735" "0.603"   
##   
## $SV  
## $SV$Accuracy  
## [1] 0  
##   
## $SV$`F1 Score`  
## [1] 0

#  
# Run the models using attributes determined from feature selection: number\_inpatient, discharge\_disposition\_id, age, number\_diagnoses, num\_lab\_procedures, time\_in\_hospital, diag\_2, diabetesMed   
reduced\_attrib\_vector <- c('number\_inpatient', 'discharge\_disposition\_id', 'age', 'number\_diagnoses', 'num\_lab\_procedures', 'time\_in\_hospital', 'diag\_2', 'diabetesMed', 'payer\_code', 'number\_outpatient', 'readmitted')  
reduced\_df <- readmitted\_train[, reduced\_attrib\_vector]  
  
perf\_measures <- run\_models(reduced\_df, c("NB", "DT", "RF", "SV"))

## Naive Bayes - All attributes: 24.21 sec elapsed  
## Decision Tree - All attributes: 24.29 sec elapsed  
## Random Forest - All attributes: 82.32 sec elapsed  
## Support Vector Machine - All attributes: 5239.71 sec elapsed

perf\_measures

## $NB  
## Accuracy F1 Score   
## "0.727" "0.591"   
##   
## $DT  
## Accuracy F1 Score   
## "0.73" "0.595"   
##   
## $RF  
## Accuracy F1 Score   
## "0.735" "0.602"   
##   
## $SV  
## Accuracy F1 Score   
## "0.733" "0.599"

# combine readmitted<=30 and readmitted>=30 to create a binary class dataset  
readmission\_binary\_class <- readmission\_data  
readmission\_binary\_class$readmitted <- factor(ifelse(readmission\_binary\_class$readmitted == "NO","NO","YES"))  
t<-table(readmission\_binary\_class$readmitted)  
p<- t/sum(t)  
t

##   
## NO YES   
## 41476 28511

p

##   
## NO YES   
## 0.6 0.4

#  
# Run the models using a binary class attribute for readmitted (readmitted or not)  
#  
# partition the readmission data with binary class attribute into training and test sets using same index created previously for multiclass data  
readmitted\_binary\_train <- readmission\_binary\_class[trainIndex,]   
readmitted\_binary\_test <- readmission\_binary\_class[-trainIndex,]   
#  
# Run the models on the full attribute dataset (uses cross validation on the training set):  
#  
perf\_measures <- run\_models(readmitted\_binary\_train, c("NB", "DT", "RF"))

## Naive Bayes - All attributes: 47.78 sec elapsed  
## Decision Tree - All attributes: 19.7 sec elapsed  
## Random Forest - All attributes: 153.86 sec elapsed

perf\_measures

## $NB  
## Accuracy F1 Score   
## "0.615" "0.698"   
##   
## $DT  
## Accuracy F1 Score   
## "0.605" "0.695"   
##   
## $RF  
## Accuracy F1 Score   
## "0.628" "0.725"   
##   
## $SV  
## $SV$Accuracy  
## [1] 0  
##   
## $SV$`F1 Score`  
## [1] 0

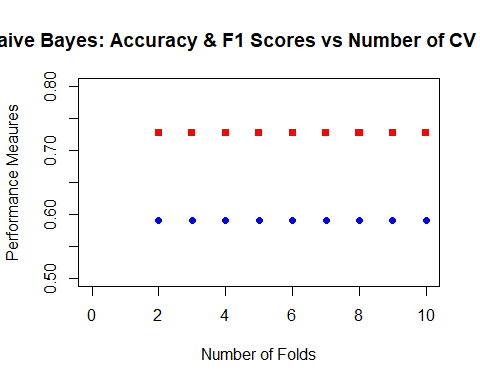
#  
# Run the models using attributes determined from feature selection (uses cross validation on the training set):  
#  
reduced\_binary\_df <- readmitted\_binary\_train[, reduced\_attrib\_vector]  
  
perf\_measures <- run\_models(reduced\_binary\_df, c("NB", "DT", "RF"))

## Naive Bayes - All attributes: 22.7 sec elapsed  
## Decision Tree - All attributes: 19.39 sec elapsed  
## Random Forest - All attributes: 63.47 sec elapsed

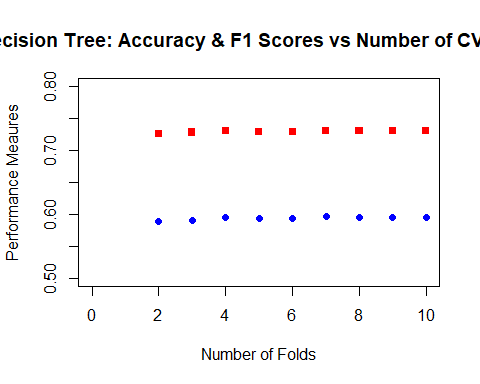
perf\_measures

## $NB  
## Accuracy F1 Score   
## "0.607" "0.704"   
##   
## $DT  
## Accuracy F1 Score   
## "0.617" "0.726"   
##   
## $RF  
## Accuracy F1 Score   
## "0.619" "0.732"   
##   
## $SV  
## $SV$Accuracy  
## [1] 0  
##   
## $SV$`F1 Score`  
## [1] 0

#  
# Experiment with various numbers of folds in cross-validation to find the optimal number for the training set (graph performance measures against number of cross-validation folds to determine where number of folds stabilizes)  
max\_folds <- 10  
# set up plot  
plot(1, type="n", xlab="Number of Folds", ylab="Performance Meaures", xlim=c(0, 10), ylim=c(0.5, 0.8))  
title(main = "Naive Bayes: Accuracy & F1 Scores vs Number of CV Folds")  
legend(0, 0.4, legend=c("Accuracy", "F1 Score"),col=c("red", "blue"))  
 #  
 # Naive Bayes on reduced set  
 #  
for (nf in 2:max\_folds) {  
 conf\_matrix <- generate\_conf\_matrix ("NB", nf, reduced\_df)  
 perf\_measures <- format(calc\_perf\_meas (conf\_matrix), digits = 3)  
 points(x = nf, y = perf\_measures[1], col = "red", pch = 15) # plot accuracy for current number of folds  
 points(x = nf, y = perf\_measures[2], col = "blue", pch = 19) # plot F1 Score for current number of folds  
}



#  
 # Decision tree on reduced set  
 #  
# set up plot  
plot(1, type="n", xlab="Number of Folds", ylab="Performance Meaures", xlim=c(0, 10), ylim=c(0.5, 0.8))  
title(main = "Decision Tree: Accuracy & F1 Scores vs Number of CV Folds")  
legend(0, 0.4, legend=c("Accuracy", "F1 Score"),col=c("red", "blue"))  
  
for (nf in 2:max\_folds) {  
 conf\_matrix <- generate\_conf\_matrix ("DT", nf, reduced\_df)  
 perf\_measures <- format(calc\_perf\_meas (conf\_matrix), digits = 3)  
 points(x = nf, y = perf\_measures[1], col = "red", pch = 15) # plot accuracy for current number of folds  
 points(x = nf, y = perf\_measures[2], col = "blue", pch = 19) # plot F1 Score for current number of folds  
}



# Balance the multiclass training dataset by undersampling the majority classes and rerun models to see if that improves accuracy and F1 scores.  
# Use the number of observations from the smallest class (readmitted <30) as number randomly selected from other classes  
n\_smallest\_class <- length(readmitted\_train$readmitted[readmitted\_train$readmitted=="<30"])  
readmitted\_balanced <- as.data.frame(readmitted\_train %>% group\_by(readmitted) %>% sample\_n(n\_smallest\_class))  
#  
# Run the models using the balanced dataset on the full attribute set (uses cross-validation)  
perf\_measures <- run\_models(readmitted\_balanced, c("NB", "DT", "RF"))

## Naive Bayes - All attributes: 16.02 sec elapsed  
## Decision Tree - All attributes: 4.64 sec elapsed  
## Random Forest - All attributes: 65.91 sec elapsed

perf\_measures

## $NB  
## Accuracy F1 Score   
## "0.621" "0.431"   
##   
## $DT  
## Accuracy F1 Score   
## "0.601" "0.401"   
##   
## $RF  
## Accuracy F1 Score   
## "0.619" "0.429"   
##   
## $SV  
## $SV$Accuracy  
## [1] 0  
##   
## $SV$`F1 Score`  
## [1] 0

#  
# Run the models using the balanced dataset on the reduced attribute set (uses cross-validation)  
reduced\_balanced <- readmitted\_balanced[, reduced\_attrib\_vector]  
perf\_measures <- run\_models(reduced\_balanced, c("NB", "DT", "RF"))

## Naive Bayes - All attributes: 8.42 sec elapsed  
## Decision Tree - All attributes: 5.77 sec elapsed  
## Random Forest - All attributes: 31.38 sec elapsed

perf\_measures

## $NB  
## Accuracy F1 Score   
## "0.618" "0.426"   
##   
## $DT  
## Accuracy F1 Score   
## "0.595" "0.392"   
##   
## $RF  
## Accuracy F1 Score   
## "0.605" "0.407"   
##   
## $SV  
## $SV$Accuracy  
## [1] 0  
##   
## $SV$`F1 Score`  
## [1] 0

# Convert to binary class (combine >30 and NOT classes ... i.e. either readmitted under 30 days or not at all).  
readmission\_binary\_class <- readmitted\_train  
readmission\_binary\_class$readmitted <- factor(ifelse(readmission\_binary\_class$readmitted == "<30", "YES", "NO"))  
readmission\_binary\_class$readmitted <- relevel(readmission\_binary\_class$readmitted, "YES")  
table(readmission\_binary\_class$readmitted)

##   
## YES NO   
## 5028 50962

# Balance the binary class training dataset by undersampling the majority class and then run models.  
# Use the number of observations from the smallest class (readmitted <30) as number randomly selected from other class  
n\_smallest\_class <- length(readmission\_binary\_class$readmitted[readmission\_binary\_class$readmitted == "YES"])  
readmitted\_balanced1 <- as.data.frame(readmission\_binary\_class %>% group\_by(readmitted) %>% sample\_n(n\_smallest\_class))  
#  
# Run the models using the balanced dataset on the full attribute set (uses cross-validation)  
perf\_measures <- run\_models(readmitted\_balanced1, c("NB", "DT", "RF"))

## Naive Bayes - All attributes: 8.52 sec elapsed  
## Decision Tree - All attributes: 1.81 sec elapsed  
## Random Forest - All attributes: 25.45 sec elapsed

perf\_measures

## $NB  
## Accuracy F1 Score   
## "0.603" "0.596"   
##   
## $DT  
## Accuracy F1 Score   
## "0.581" "0.572"   
##   
## $RF  
## Accuracy F1 Score   
## "0.608" "0.603"   
##   
## $SV  
## $SV$Accuracy  
## [1] 0  
##   
## $SV$`F1 Score`  
## [1] 0

#  
# Run the models using the balanced dataset on the reduced attribute set (uses cross-validation)  
reduced\_balanced1 <- readmitted\_balanced1[, reduced\_attrib\_vector]  
perf\_measures <- run\_models(reduced\_balanced1, c("NB", "DT", "RF"))

## Naive Bayes - All attributes: 4.09 sec elapsed  
## Decision Tree - All attributes: 1.56 sec elapsed  
## Random Forest - All attributes: 17.02 sec elapsed

perf\_measures

## $NB  
## Accuracy F1 Score   
## "0.608" "0.595"   
##   
## $DT  
## Accuracy F1 Score   
## "0.584" "0.567"   
##   
## $RF  
## Accuracy F1 Score   
## "0.585" "0.574"   
##   
## $SV  
## $SV$Accuracy  
## [1] 0  
##   
## $SV$`F1 Score`  
## [1] 0

# Balance the binary class training dataset using SMOTE (oversample minority class) and then run models.  
#  
# Use SMOTE to balance the classes  
t <- table(readmission\_binary\_class$readmitted) # table with number in each class  
o\_balance\_factor <- round(100 \* (max(t[1], t[2]) - min(t[1], t[2])) / (min(t[1], t[2]))) # factor for perc.over in smote algorithm to create enough observations to balance dataset  
u\_balance\_factor <- round( 100 \* max(t[1], t[2]) / (max(t[1], t[2]) - min(t[1], t[2]))) # factor for perc.under in smote algorithm to keep the majority class with the same number of observations as before  
readmitted\_balanced2 <- SMOTE(readmitted ~ ., readmission\_binary\_class, perc.over = o\_balance\_factor, perc.under = u\_balance\_factor)  
# Run models (Naive Bayes, decision tree, random forests) using cross-validation on balanced dataset with all attributes  
perf\_measures <- run\_models(readmitted\_balanced2, c("NB", "DT", "RF"))

## Naive Bayes - All attributes: 81.52 sec elapsed  
## Decision Tree - All attributes: 30.91 sec elapsed  
## Random Forest - All attributes: 202.33 sec elapsed

perf\_measures

## $NB  
## Accuracy F1 Score   
## "0.753" "0.751"   
##   
## $DT  
## Accuracy F1 Score   
## "0.902" "0.895"   
##   
## $RF  
## Accuracy F1 Score   
## "0.919" "0.917"   
##   
## $SV  
## $SV$Accuracy  
## [1] 0  
##   
## $SV$`F1 Score`  
## [1] 0

#  
# Run models (Naive Bayes, decision tree, random forests) using cross-validation on balanced dataset with reduced attribute set  
reduced\_balanced2 <- readmitted\_balanced2[, reduced\_attrib\_vector]  
perf\_measures <- run\_models(reduced\_balanced2, c("NB", "DT", "RF"))

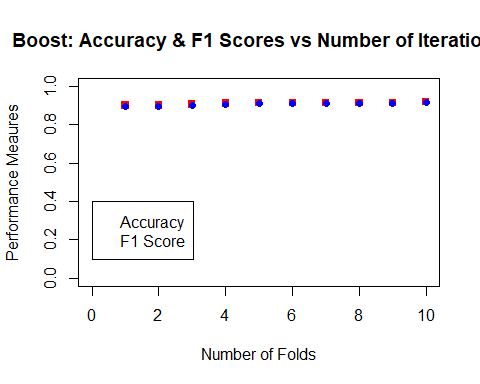
## Naive Bayes - All attributes: 42.2 sec elapsed  
## Decision Tree - All attributes: 27.72 sec elapsed  
## Random Forest - All attributes: 97.43 sec elapsed

perf\_measures

## $NB  
## Accuracy F1 Score   
## "0.699" "0.696"   
##   
## $DT  
## Accuracy F1 Score   
## "0.905" "0.897"   
##   
## $RF  
## Accuracy F1 Score   
## "0.903" "0.898"   
##   
## $SV  
## $SV$Accuracy  
## [1] 0  
##   
## $SV$`F1 Score`  
## [1] 0

# Generalize the results of the classification using boosting ensemble technique.  
#  
# function to create boost ensemble for given dataframe  
boost\_fun <- function (dframe, iters) {  
   
 # split dataframe into training and test set  
 set.seed(136)  
 ti <- createDataPartition(dframe$readmitted, p = .8, list = FALSE, times = 1)  
  
 btrain <- dframe[ti,] # training set  
 btest <- dframe[-ti,] # testing set  
  
 model\_adaboost <- adaboost(readmitted ~ ., btrain, iters) # run adaboost ensemble algorithm  
 pred <- predict(model\_adaboost, newdata=btest) # predict values for test set  
 pred$class <- relevel(pred$class, "YES") # make the "YES" class appear first  
 conf\_mat <- table(pred$class,btest$readmitted) # create confusion matrix  
 perf\_measures <- calc\_perf\_meas(conf\_mat)  
 return(perf\_measures)  
 }

# optimize iterations for boost ensemble (use reduced SMOTE dataset)  
max\_iters <- 10  
# set up plot  
plot(1, type="n", xlab="Number of Folds", ylab="Performance Meaures", xlim=c(0, 10), ylim=c(0, 1))  
title(main = "Boost: Accuracy & F1 Scores vs Number of Iterations")  
legend(0, 0.4, legend=c("Accuracy", "F1 Score"),col=c("red", "blue"))  
  
for (ni in 1:max\_iters) {  
 tic(paste("number of iterations", ni))  
 perf\_measures <- boost\_fun(reduced\_balanced2, ni)  
 toc()  
 points(x = ni, y = perf\_measures[1], col = "red", pch = 15) # plot acc. for current number of iterations  
 points(x = ni, y = perf\_measures[2], col = "blue", pch = 19) # plot F1 for current number of iterations  
}



## number of iterations 1: 7.68 sec elapsed  
## number of iterations 2: 16.34 sec elapsed  
## number of iterations 3: 24.31 sec elapsed  
## number of iterations 4: 33.61 sec elapsed  
## number of iterations 5: 43.58 sec elapsed  
## number of iterations 6: 57.91 sec elapsed  
## number of iterations 7: 63.55 sec elapsed  
## number of iterations 8: 67.23 sec elapsed  
## number of iterations 9: 77.69 sec elapsed  
## number of iterations 10: 87 sec elapsed

# boost ensemble for binary class data, unbalanced, full attribute set  
tic("Boost Ensemble - Unbalanced, binary, all attributes")  
perf\_measures <- boost\_fun(readmission\_binary\_class, 7)  
toc()

## Boost Ensemble - Unbalanced, binary, all attributes: 65.55 sec elapsed

format(perf\_measures, digits = 3)

## Accuracy F1 Score   
## "0.901" "0.0316"

# boost ensemble for binary class data, unbalanced, reduced attribute set  
tic("Boost Ensemble - Unbalanced, binary, reduced attribute set")  
perf\_measures <- boost\_fun(readmission\_binary\_class[, reduced\_attrib\_vector], 7)  
toc()

## Boost Ensemble - Unbalanced, binary, reduced attribute set: 29.94 sec elapsed

format(perf\_measures, digits = 3)

## Accuracy F1 Score   
## "0.899" "0.039"

# boost ensemble for balanced (undersampling majority) binary class data full attribute set  
tic("Boost Ensemble - Balanced (undersampling), binary, all attributes")  
perf\_measures <- boost\_fun(readmitted\_balanced1, 7)  
toc()

## Boost Ensemble - Balanced (undersampling), binary, all attributes: 6.92 sec elapsed

format(perf\_measures, digits = 3)

## Accuracy F1 Score   
## "0.566" "0.551"

# boost ensemble for balanced (undersampling majority) binary class data reduced attribute set  
tic("Boost Ensemble - Balanced (undersampling), binary, reduced attribute set")  
perf\_measures <- boost\_fun(reduced\_balanced1, 7)  
toc()

## Boost Ensemble - Balanced (undersampling), binary, reduced attribute set: 3.61 sec elapsed

format(perf\_measures, digits = 3)

## Accuracy F1 Score   
## "0.567" "0.446"

# boost ensemble for balanced (synthetic oversampling minority) binary class data full attribute set  
tic("Boost Ensemble - Balanced (SMOTE), binary, all attributes")  
perf\_measures <- boost\_fun(readmitted\_balanced2, 7)  
toc()

## Boost Ensemble - Balanced (SMOTE), binary, all attributes: 129.46 sec elapsed

format(perf\_measures, digits = 3)

## Accuracy F1 Score   
## "0.918" "0.915"

# boost ensemble for balanced (synthetic oversampling minority) binary class data reduced attribute set  
tic("Boost Ensemble - Balanced (SMOTE), binary, reduced attribute set")  
perf\_measures <- boost\_fun(reduced\_balanced2, 7)  
toc()

## Boost Ensemble - Balanced (SMOTE), binary, reduced attribute set: 58.66 sec elapsed

format(perf\_measures, digits = 3)

## Accuracy F1 Score   
## "0.915" "0.911"

# Use the test dataset to test the four best models  
#  
# Random forest on multiclass reduced attribute dataset  
reduced\_test <- readmitted\_test[, reduced\_attrib\_vector]  
tic("Random forest on multiclass full attribute dataset")  
model1 <- randomForest(readmitted ~ ., data=readmitted\_train[, reduced\_attrib\_vector], ntree=100)  
preds1 <- predict(model1, newdata = reduced\_test)  
toc()

## Random forest on multiclass full attribute dataset: 24.53 sec elapsed

conf\_matrix1 <- table(preds1, reduced\_test$readmitted)  
names(dimnames(conf\_matrix1)) <- c("Predicted Class", "Actual Class")  
perf\_measures <- calc\_perf\_meas(conf\_matrix1)  
format(perf\_measures, digits = 3)

## Accuracy F1 Score   
## "0.734" "0.602"

#  
# Create a binary class version of test dataset  
readmitted\_binary\_test <- readmitted\_test  
readmitted\_binary\_test$readmitted <- factor(ifelse(readmitted\_binary\_test$readmitted == "<30", "YES", "NO"))  
readmitted\_binary\_test$readmitted <- relevel(readmitted\_binary\_test$readmitted, "YES")  
#  
# Random forest on binary, SMOTE balanced, full attribute dataset  
tic("Random forest on binary, SMOTE balanced full attribute dataset")  
model2 <- randomForest(readmitted ~ ., data=readmitted\_balanced2, ntree=100)  
preds2 <- predict(model2, newdata = readmitted\_binary\_test)  
toc()

## Random forest on binary, SMOTE balanced full attribute dataset: 70.67 sec elapsed

conf\_matrix2 <- table(preds2, readmitted\_binary\_test$readmitted)  
names(dimnames(conf\_matrix2)) <- c("Predicted Class", "Actual Class")  
perf\_measures <- calc\_perf\_meas(conf\_matrix2)  
format(perf\_measures, digits = 3)

## Accuracy F1 Score   
## "0.817" "0.179"

#  
# Create a reduced attribute binary class version of test dataset  
reduced\_binary\_test <- readmitted\_binary\_test[, reduced\_attrib\_vector]  
#  
# Decision tree on binary, SMOTE balanced, reduced attribute dataset  
tic("Decision tree on binary, SMOTE balanced reduced attribute dataset")  
model3 <- J48(readmitted ~ ., data=readmitted\_balanced2[, reduced\_attrib\_vector])  
preds3 <- predict(model3, newdata = reduced\_binary\_test)  
toc()

## Decision tree on binary, SMOTE balanced reduced attribute dataset: 10.83 sec elapsed

conf\_matrix3 <- table(preds3, reduced\_binary\_test$readmitted)  
names(dimnames(conf\_matrix3)) <- c("Predicted Class", "Actual Class")  
perf\_measures <- calc\_perf\_meas(conf\_matrix3)  
format(perf\_measures, digits = 3)

## Accuracy F1 Score   
## "0.889" "0.091"

#  
# Adaboost on binary, SMOTE balanced, full attribute dataset  
tic("Adaboost on binary, SMOTE balanced full attribute dataset")  
model4 <- adaboost(readmitted ~ ., data=readmitted\_balanced2, 7)  
preds4 <- predict(model4, newdata = readmitted\_binary\_test)  
toc()

## Adaboost on binary, SMOTE balanced full attribute dataset: 170.67 sec elapsed

preds4$class <- relevel(preds4$class, "YES") # make the "YES" class appear first  
conf\_matrix4 <- table(preds4$class, readmitted\_binary\_test$readmitted)  
names(dimnames(conf\_matrix4)) <- c("Predicted Class", "Actual Class")  
perf\_measures <- calc\_perf\_meas(conf\_matrix4)  
format(perf\_measures, digits = 3)

## Accuracy F1 Score   
## "0.833" "0.151"

#  
#