# Which Diabetes Patients will be Hopsital Readmits?

*Aneesa Noorani*

*2019-November-21*

## Contents

**Executive Summary**

Increased healthcare costs are a high-priority concern for the nation. Every policy maker, healthcare administrator, healthcare provider, and patient is concerned with exponential costs. One method for achieving lower healthcare costs is to reduce hospital readmission rates. A logical approach to control costs is to use predictive analytic techniques to identify patients most likely to be readmitted within 30 days of being discharged from the hospital. This report uses a number of modeling techniques for the analyses. I conclude that the decision tree provides the best model for predicting whether a patient is likely to be readmitted in less than 30 days.

**Part A: Introduction**

Hospital readmission rates serve as a benchmark for healthcare administrators and the government in assessing a hospital's quality. This is because the cost burden of readmissions is high, and cost-reduction is an ever-elusive goal.

Amongst all-cause hospital readmissions, diabetes readmissions are more prevalent and also more costly as compared to other types of readmissions. At New York Presbyterian Hospital, diabetes patients have a 30-day readmission rate of 14.4 - 21%, as compared to the general inpatient population readmission rate of 8.5 - 13.5% (sinha-Gregory et al, 2015). In addition, according to a 2018 guide published by the Centers for Medicare and Medicaid Services (CMS), the annual cost associated with excess readmissions for diabetes patients, amongst Medicare beneficiaries, is $251 million. Thus, it is clear that diabetes readmissions deserve special attention from hospital administrators (CMS, 2018).

Some of the risk factors associated with a higher likelihood of readmission include a lower socioeconomic status, belonging to a racial minority, the type of admission and a recent histoty of prior hospitalization (Rubin, 2015). Hence, analyzing a dataset that includes some of these risk factors can be beneficial to a hospital in predicting which patients are more likely to be readmitted. The hospital staff can then take more precautionary measures such as infection-prevention, or educate patients more meticulously on discharge instructions. Simply put, readmission prediction for high-risk patients can lead to increased attention towards those predicted to be readmitted, and ideally lead to fewer unnecessary readmissions.

**Part B: Data Description**

This dataset was obtained from the University of California at Irvine's (UCI) Machine Learning Repository archives. It contains data collected from 1999 to 2008, from 130 US hospitals and integrated delivery networks. It consists of 101,767 observations on 71,518 unique patients, and has 50 explanatory variables. The encounter_id column, which is the first column in the data set, serves as an index column.

An observation was only included in the data set if it met the following criteria: 1) inpatient encounter (which means it was a hospital admission); 2) any kind of diabetes diagnosis was entered in the system; 3) length of stay was between 1-14 days; 4) laboratory tests were performed; 5) medeciations were administered.

Some of the key attributes in the data set include: race, gender, age, time in hospital, admission type, discharge type, number of medications, and number of inpatient, outpatient and emergency visits in the year before the documented hospitalization.

It is interesting to note that of the 50 explanatory variables, 23 indicate whether a patient's dosage for a given medication was increased, decreased, kept the same, or not prescribed. That is to say, there is an individual explanatory variable for 23 different medications.

Most importantly, the a-priori probability of the class of interest, which is patients who were admitted to the hospital in less than 30 days is 11.16%, or 11,357 rows.

One important weakness of the data set is that it does not include the date of the patient encounter. If it did, time series analyses would have been possible to see if there was a trend, seasonality, autocorrelations, etc. in the readmission rate.

Please see Supplement for full list of independent and target variables.

```
  #count number of unique patients
length(unique(diabetes.df$patient_nbr))
```

```
## [1] 71518
```

```
  #how rare is the class of interest?
Un1 <- prop.table(table(diabetes.df$readmitted))
Un1
```

```
##
##       <30       >30        NO
## 0.1115992 0.3492817 0.5391192
```

**Part C: Data Preprocessing**

Several columns in this data set had missing values, or incomplete information. Preprocessing the data occurred in ten steps, as outlined below.

*Step 1: eliminate variables with high proportion of missing values*

A) The weight variable had 97% missing values. A variable with such a high proportion of missing information is of little value. Thus, it was easy to decide to drop this variable from the analysis.

B) The payer_code variable had 52% missing values. If we simply used a 50% threshold as our benchmark for determining whether to eliminate this variable, it would have been eliminated. However, it is important to consider whether the presence or absence of this variable affects the outcome variable. To determine this, I conducted a t-test to compare the readmission rate for those observations with a value in the payer_code column, versus observations without a value.

```
##
##  One Sample t-test
##
## data:  payer_code_mean[2:18, 2]
## t = 1.9471, df = 16, p-value = 0.06929
## alternative hypothesis: true mean is not equal to 2.42662
## 95 percent confidence interval:
##  2.419604 2.591742
## sample estimates:
## mean of x
##  2.505673
```

Null hypothesis of the t-test is: true mean is equal to 2.4266. The t-test result is not significant at the 1%, or even 5%, significance level. This means we cannot reject the null hypothesis. In other words, the difference between the readmission means for those observations that do have a payer code associated with them vs. those observations that do not, is not significant. Hence, the variable can be removed from the analysis without having a significant impact on the final result.

C) I then repeated the process for the medical_specialty column, since it also has 53% missing values.

```
##
##  One Sample t-test
##
## data:  med_specialty_mean[2:73, 2]
## t = 2.2037, df = 71, p-value = 0.03079
## alternative hypothesis: true mean is not equal to 2.403832
## 95 percent confidence interval:
##  2.410972 2.546706
## sample estimates:
## mean of x
##  2.478839
```

Null hypothesis of the t-test is: true mean is equal to 2.4038. The t-test result is significant at the 5% significance level - the difference between the readmission means for those observations that do have a medical specialty associated with them vs. those observations that do not, IS significant. This means we can reject the null hypothesis. Hence, this variable cannot be removed.

*Step 2: Miscellaneous Column Elimination*

A) The max_glu_serum column does not have any missing values. However, the value of 'None' indicates that a measurement was not taken. A proportion table indicates that nearly 95% of the values are 'none.' Since a value of 'None' doesn't indicate anything of significance, this variable can be eliminated.

```
##
##       >200       >300       None       Norm
## 0.01459230 0.01242065 0.94746772 0.02551933
```

B) The A1Cresult column is similar – a value of 'None' indicates that a measurement was not taken. A proportion table indicates that 83% of observations did not have an A1c result taken. Since a value of 'None' doesn't indicate anything of significance, this variable can also be eliminated.

```
##
##         >7         >8       None       Norm
## 0.03745848 0.08073423 0.83277322 0.04903406
```

*Step 3: Replace ?s with NAs*
Since the race column has 2% missing values, and the diag_3 column has 1%, I replaced the question marks with NA's in those columns for ease of analysis.

```
## [1] 1423
```

*Step 4: Convert data types for some of the most important columns*
It is important to ensure the categorical variables are in proper format for analysis. Initially, many of the categorical variables were of the class "character." However, for the purpose of exploratory analysis and modeling, they should be factors, or categorical variables. The following variables were converted to factor variables: race, gender, admission_type_id, discharge_disposition_id, admission_source_id, insulin, change, and diabetesMed.

*Step 5: Drop Observation Rows*
Observations with a gender of "Unknown/Invalid" were dropped. Even though these accounted for only three rows in the entire data set, a gender of "unknown/invalid" does not contribute much to the dataset and only increases the number of categories that a potential model would have to discern through.

*Step 6: Reduce Number of Categories in 'Age' Variable*
It is likely that age will be a predictor in whether a diabetes patient is readmitted. In the original data set, this variable is divided into ten categories. For ease of analysis, the number of categories was reduced to five.

```
  #new age categories
age_factors <- c("0-19", "20-39", "40-59", "60-79", "80-99")
  #recoding
diabetes.df$age <- recode(diabetes.df$age, "[0-10)" = age_factors[1],
    "[10-20)" = age_factors[1], "[20-30)" = age_factors[2],
    "[30-40)" = age_factors[2], "[40-50)" = age_factors[3],
    "[50-60)" = age_factors[3], "[60-70)" = age_factors[4],
    "[70-80)" = age_factors[4], "[80-90)" = age_factors[5],
    "[90-100)" = age_factors[5]
  )
```

*Step 7: Reduce Number of Medications in Data Set*
The data set currently has 20+ medications, which is a bit excessive. I'd like to pick the top few medications that have the least proportion of 'No.'

```
  #creating new df with just the medications
meds.df <- diabetes.df[, 20:42]

  #counting proportion of 'No' in each column by calculating sum &
  #then dividing by 101763 (total # of observations)
count.No.per.column <- ldply(meds.df, function(c) sum(c=="No")/101763)
count.No.per.column <- count.No.per.column[order(count.No.per.column$V1),]
count.No.per.column
```

```
##                          .id        V1
## 18                   insulin 0.4655916
## 1                  metformin 0.8035927
## 7                  glipizide 0.8753476
## 8                  glyburide 0.8953451
## 10              pioglitazone 0.9279994
## 11             rosiglitazone 0.9374625
## 5                glimepiride 0.9489893
## 2                repaglinide 0.9848766
## 19       glyburide-metformin 0.9930623
## 3                nateglinide 0.9930918
## 12                   acarbose 0.9969734
## 4              chlorpropamide 0.9991549
## 15                tolazamide 0.9996168
## 13                  miglitol 0.9996266
## 9                 tolbutamide 0.9997740
## 20       glipizide-metformin 0.9998723
## 14               troglitazone 0.9999705
## 22  metformin-rosiglitazone 0.9999803
## 6              acetohexamide 0.9999902
## 21 glimepiride-pioglitazone 0.9999902
## 23   metformin-pioglitazone 0.9999902
## 16                   examide 1.0000000
## 17                citoglipton 1.0000000
```

There are only 4 medications that have less than 90% No's. With more than 90% No's, it becomes difficult to derive meaningful insights from the data. So, I only kept insulin, metformin, glipizide and glyburide in the modeling data set.

*Step 8: Group Medical Specialities*
For the medical specialities with a handful of observations, I decided to group them into 'Other.' This is because if there are only a handful of categories for a certain variable, and the training and test set each do not have enough, the model produces an error, stating that there weren't enough observations.

```
  #replacing question mark with NA
diabetes.df$medical_specialty[diabetes.df$medical_specialty == "?"] <- NA
```

```
spec_del <- count(diabetes.df, vars = "medical_specialty")
spec_del <- subset(spec_del, subset=(spec_del$freq) < 20)

diabetes.df$medical_specialty <- fct_collapse(diabetes.df$medical_specialty,
    'Other' = c(spec_del$medical_specialty))

diabetes.df$medical_specialty <- as.factor(diabetes.df$medical_specialty)
```

*Step 9: Group Admission Source IDs*
Similar to above, if an admission_source_id factor had fewer than 20 observations I decided to group it into 'Other.'

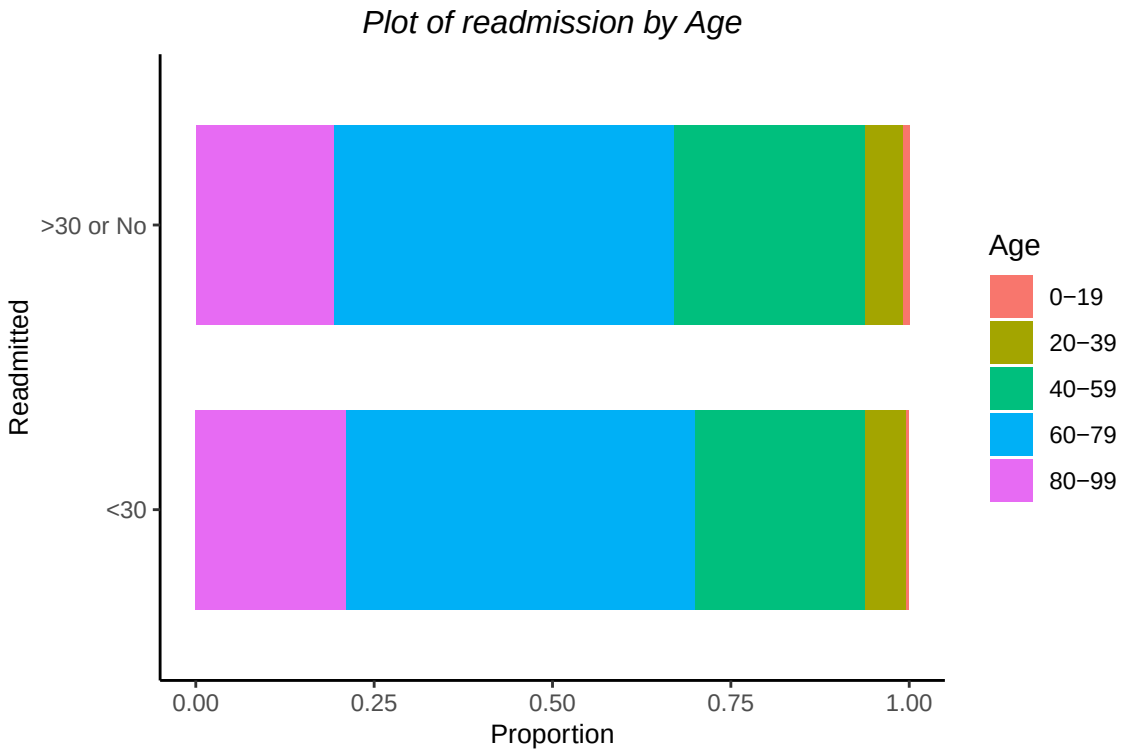
```
diabetes.df$admission_source_id <- as.factor(diabetes.df$admission_source_id)

adm_del <- count(diabetes.df, vars = "admission_source_id")
adm_del <- subset(adm_del, subset=(adm_del$freq) < 20)

diabetes.df$admission_source_id <- fct_collapse(diabetes.df$admission_source_id,
            'Other' = c("8", "10", "11", "13", "14", "22", "25"))
```

*Step 10: Convert Response Variable*
There are currently 3 outcome variables. I want to reduce to a binary outcome in the hopes that it will improve classification accuracy. In addition, I am mostly interested in whether or not a patient ends up readmitted in less than 30 days. For my analysis, if a patient is readmitted in more than 30 days, it is equivalent to the patient not being readmitted. This justifies the collapsing of the ">30" outcome possibility into the "No" category.

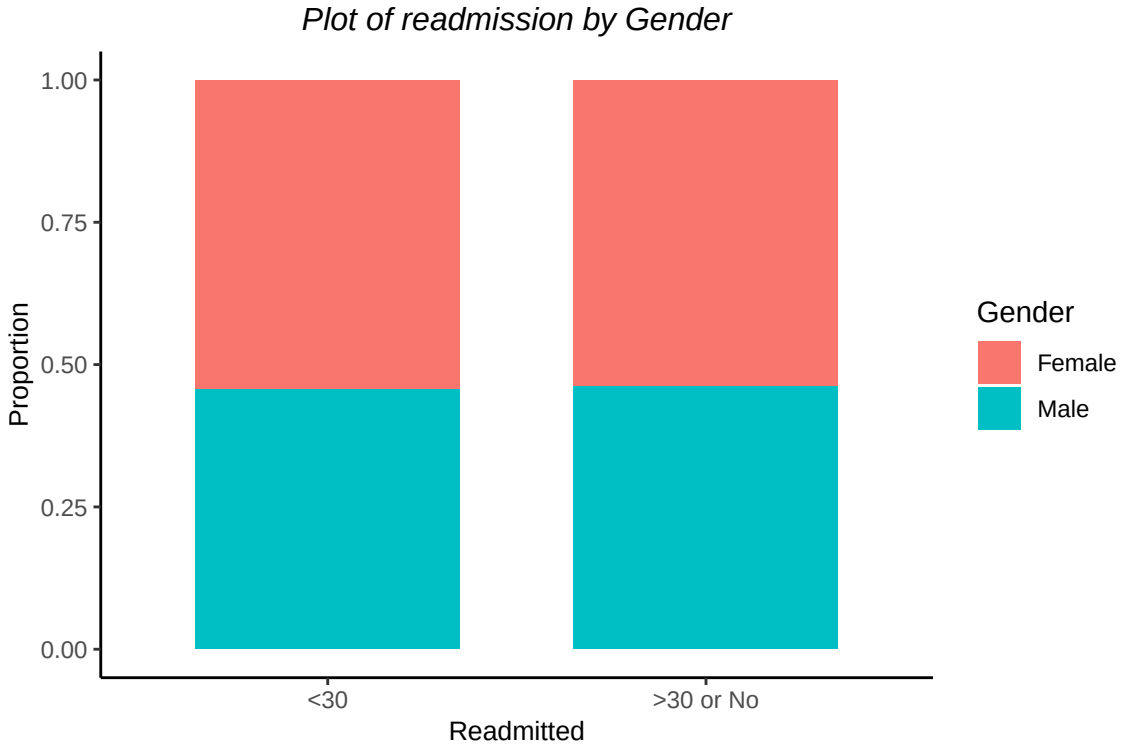
```
  #converting response variable so only 2 outcomes. But first, need to filter
  #out the 3 rows we deleted from original dataframe
original_df2 <- original_df %>% filter(gender %in% c("Female", "Male"))
diabetes.df$readmitted <- as.factor(original_df2$readmitted)
diabetes.df$readmitted <- fct_collapse(diabetes.df$readmitted,
                                       '>30 or No' = c(">30", "NO"))
```

### Part D: Exploratory Data Analysis

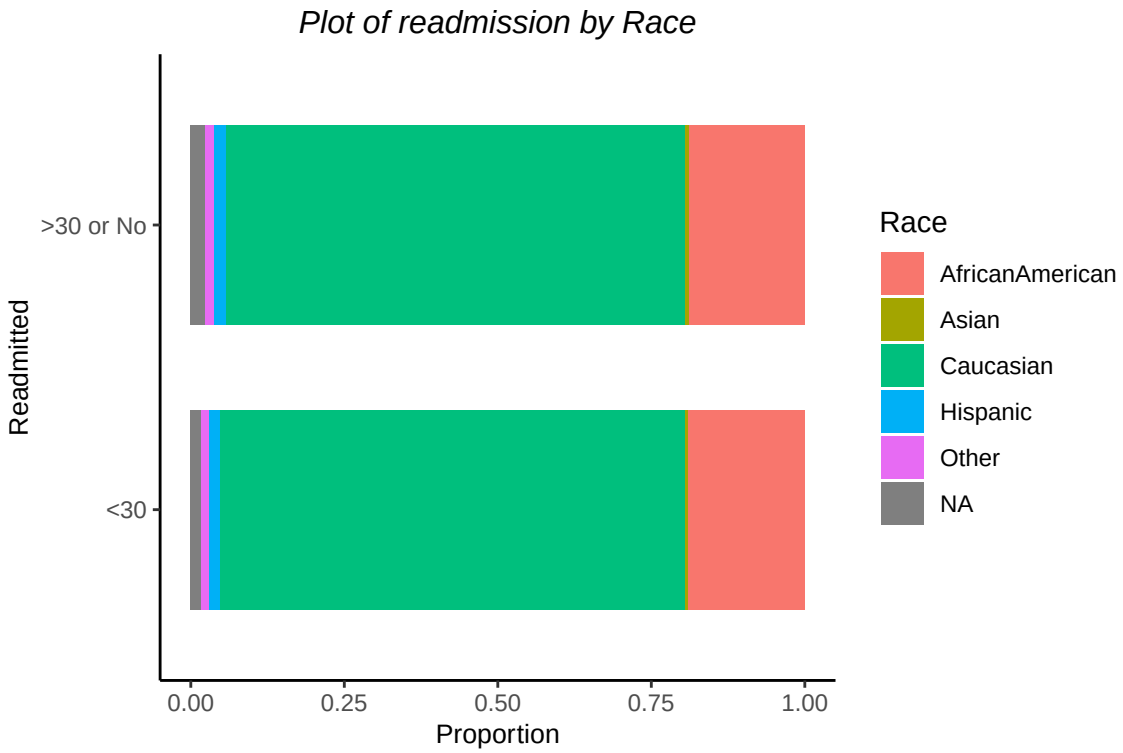
*Part I: Categorical Variables*
I first explored the categorical variables in this data set.
1) The below bar plot of the age variable shows slight variation in the age distribution between the readmission categories. But the variation is not as great as anticipated.



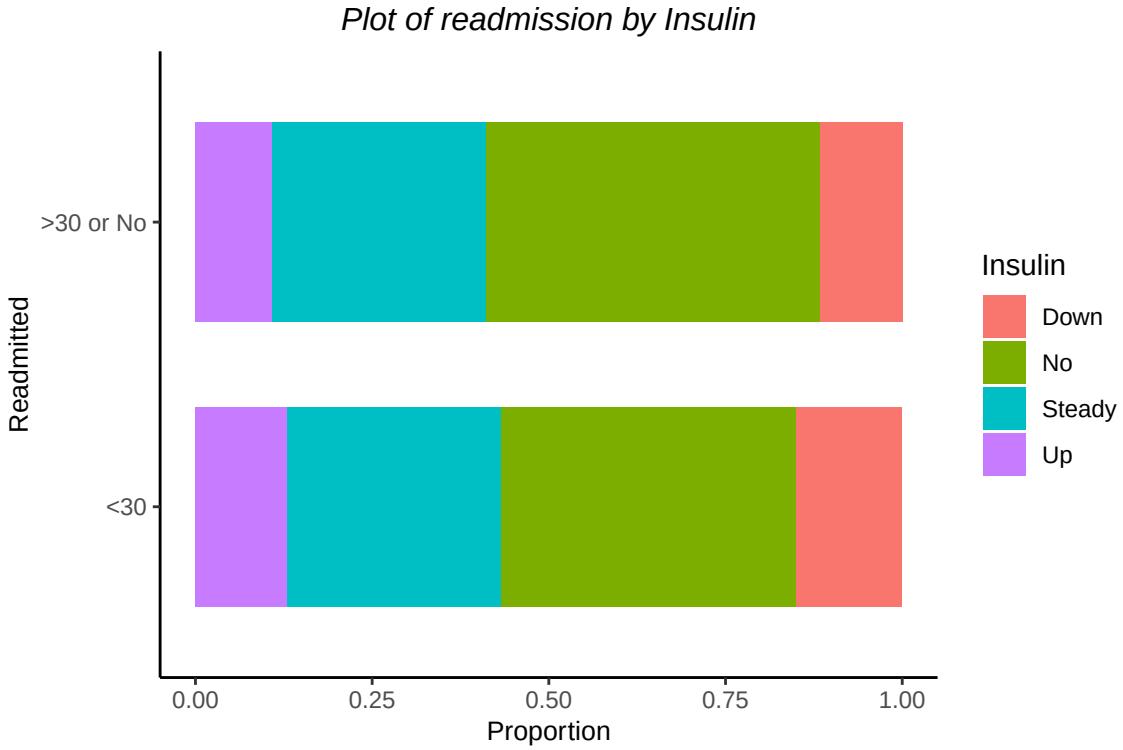

2) A bar plot of the gender variable shows nearly identical variation between the readmission categories. It doesn't look like this will be a strong predictor variable in any model.



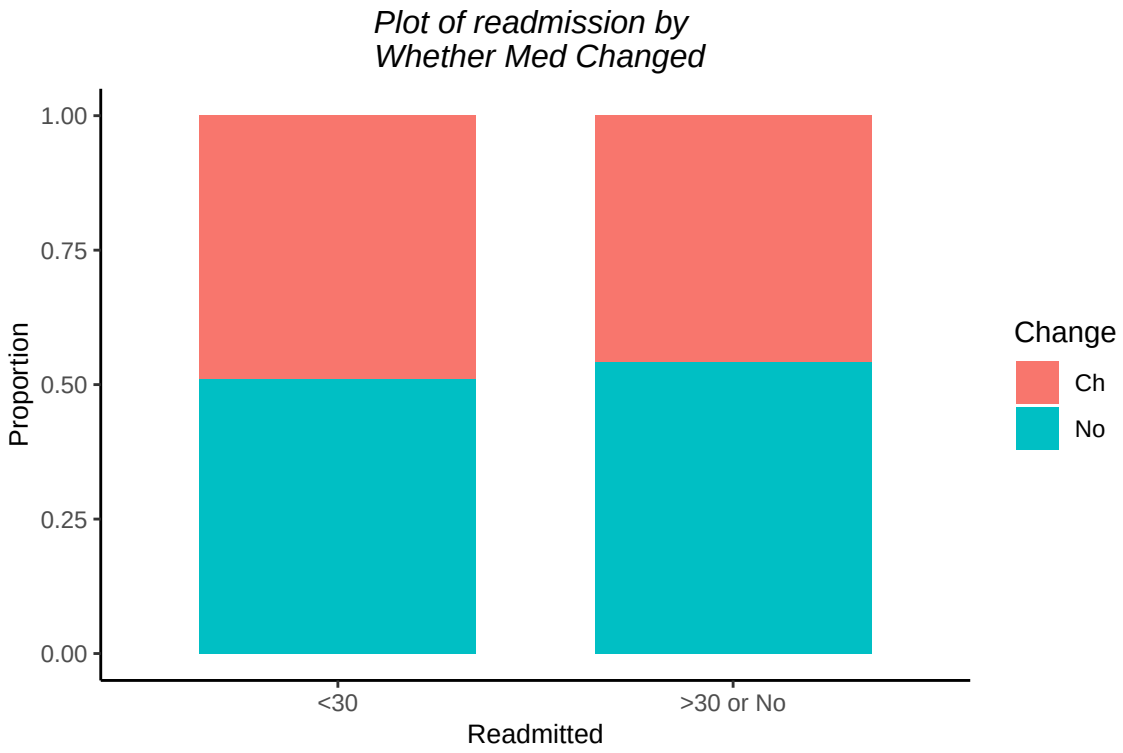

3) From a cursory inspection, a bar plot of the race variable shows that the two races with even slight variation are: Asian & Caucasian. Race will likely be included in the model.




4) A bar plot of the insulin variable shows that for patients who were readmitted within 30 days, there is a slightly higher proportion of patients for whom insulin dosage increased, as compared to patients who were not readmitted within 30 days.

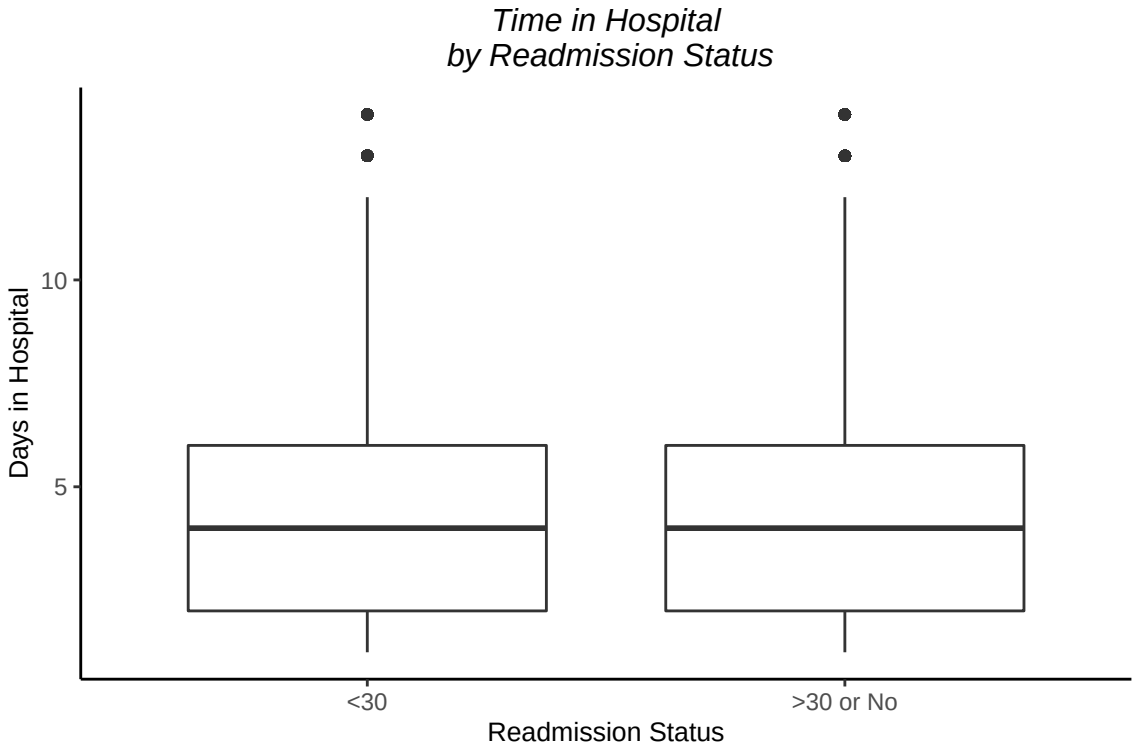



5) A bar plot of the med_changed variable indicates that for patients who were readmitted within 30 days, there was a slightly higher chance that they DID have their medication changed.

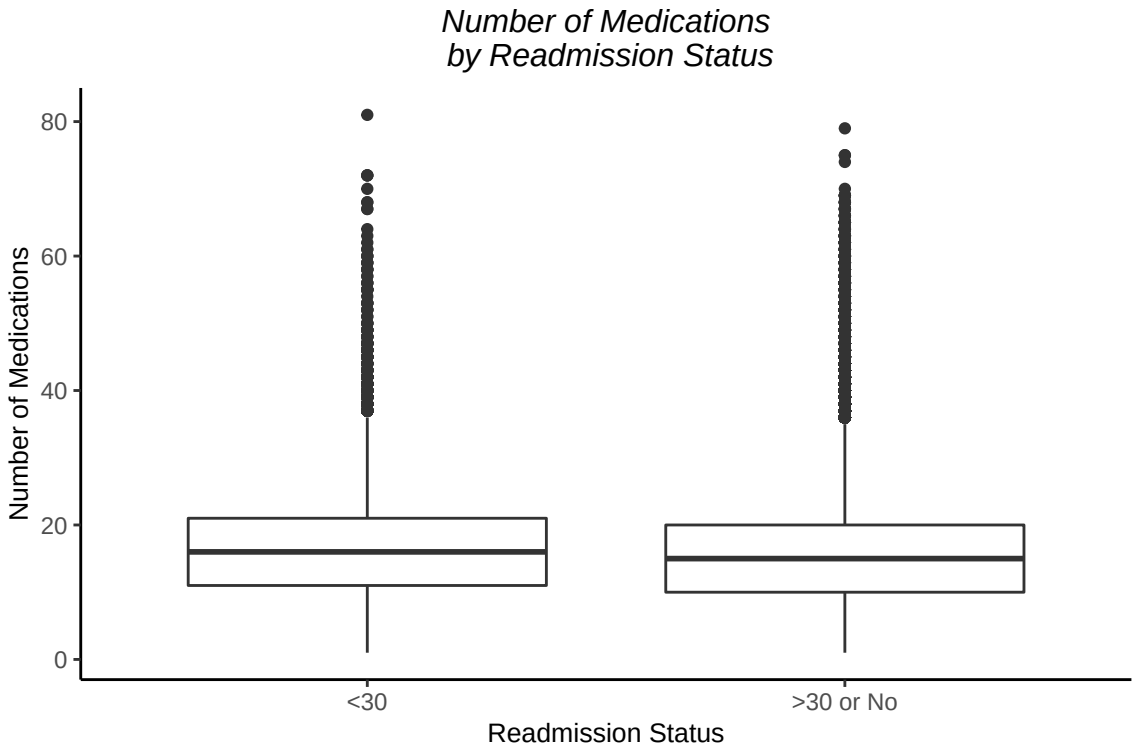



*Part II: Numerical Variables*

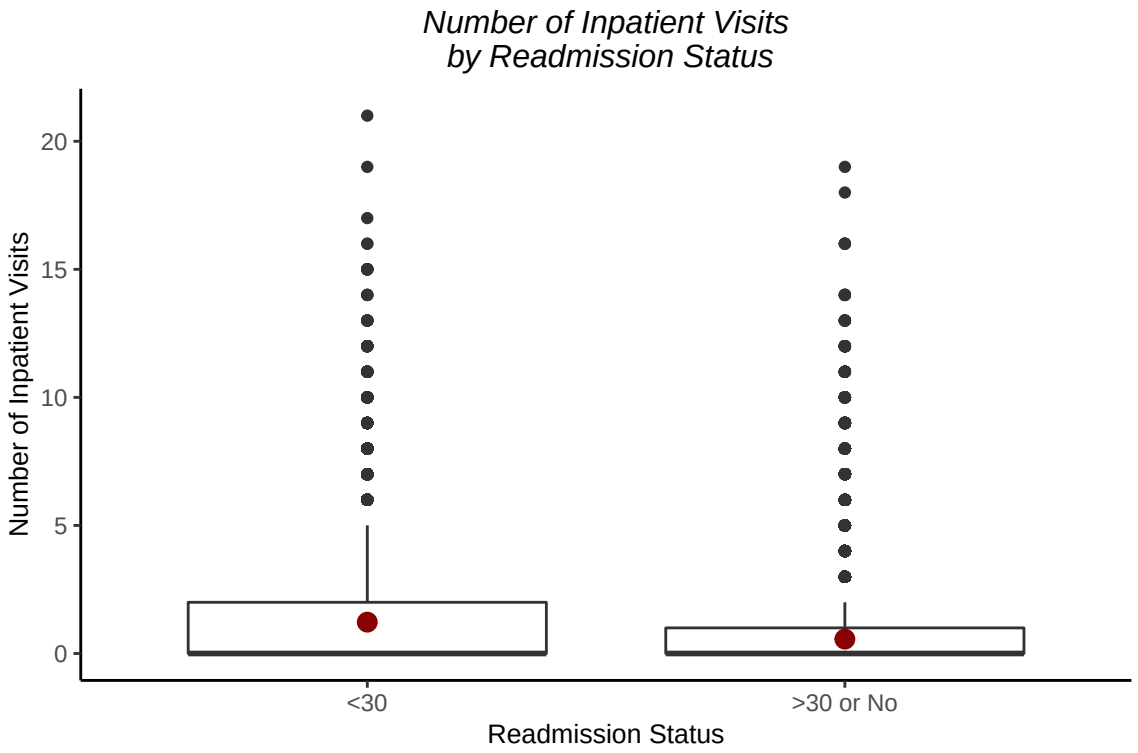
1) A box plot of the time_in_hospital variable shows that patients who were readmitted within 30 days did not have a significantly longer hospital stay as compared to patients who were not readmitted.



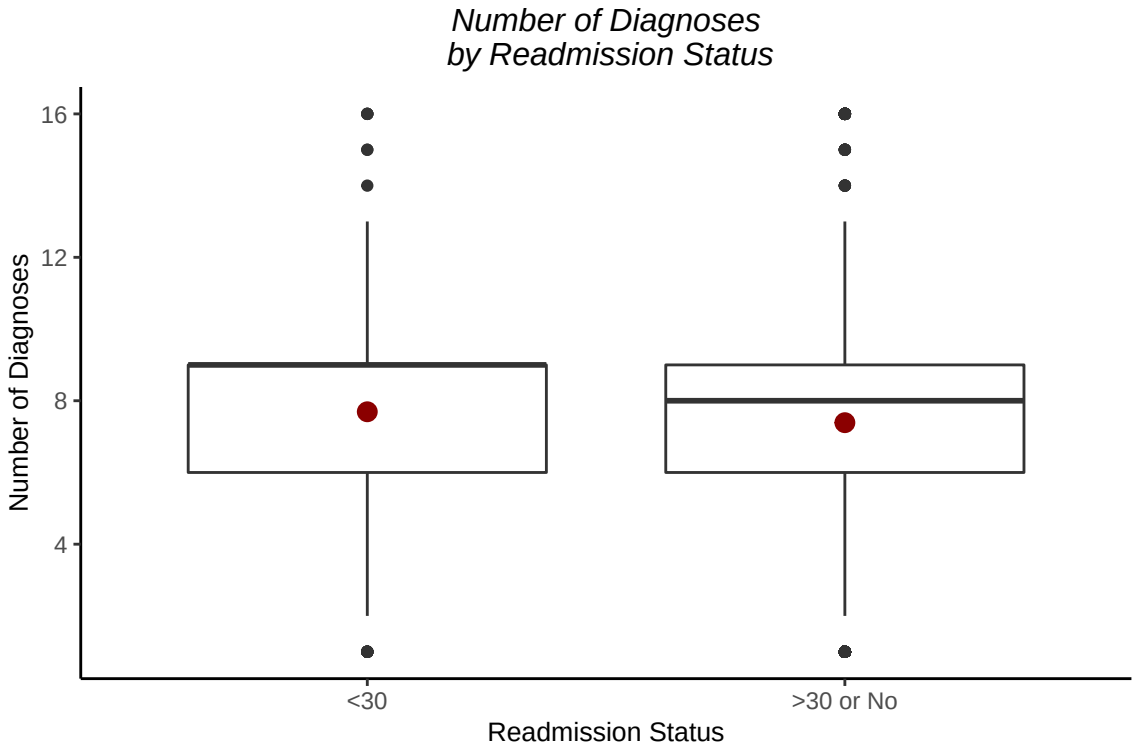
2) Box plot of number of medications doesn't indicate a noticeable difference between the categories.




3) Box plot of number of inpatient visits indicates that patients who were readmitted in less than 30 days have a higher mean as compared to the non-readmitted patients.



4) Box plot of number of diagnoses. Readmitted patients have a slightly higher number of diagnoses.




**Part E: Empirical Data Analysis**

***Part I: Pre-modeling Data Analysis***

***1. Readmission Frequency***

Based on the exploratory data analysis, I was a little concerned about whether any modeling technique would produce significant results. So, I split the data in different ways to see if there might be another hypothesis question worth exploring. For instance, one question that came to mind was: For all the patient encounters that resulted in a readmission within less than 30 days, were there duplicate patients? Put another way, were there patients who had hospital readmissions in less than 30 days, multiple times? If so, perhaps a question worth exploring could be: what are the explanatory variables that influence whether a patient has repeated hospital readmissions in less than 30 days?

```
  #distinct patient IDs
diabetes_pts <- diabetes.df %>% select(c(patient_nbr, readmitted))

  #only want pts who appear multiple times
duplicate_pts <- diabetes_pts[duplicated(diabetes_pts$patient_nbr)|duplicated(diabetes_pts$

  #then, filter to select those pts who were readmitted in less than 30 days
dup_pts_rRLT30 <- duplicate_pts %>% filter(readmitted %in% c("<30"))

num_readmits <- dup_pts_rRLT30 %>% group_by(patient_nbr, readmitted)
```

```
readmit_freq <- count(dup_pts_rRLT30)
#readmit_freq
sum(readmit_freq$freq > 1)
```

```
## [1] 1539
```

There were only 1,539 records with a frequency of more than 1. I decided that this was not enough data to produce a reliable model. Instead, I deemed it more statistically sound to continue modeling with a variation of the original data set.

***2: Minority Class Oversampling***

Since the cost of missing the minority class (which is the class of interest) is high, we want to correctly identify as many of the "less than 30 days" readmission cases as possible. This is where the hospital is penalized, and also incurs greater cost.

Based on the exploratory analysis, from cursory visual inspection, it doesn't look like there are any obvious indicator variables that would be strong candidates for a predictive model. Going back to the data, we see that this could be because we have an unbalanced classification problem. In other words, there is an uneven proportion of cases available for each class. Only ~11% of observations belong to the class of interest. Thus, I decided to oversample the minority class / undersample the majority class. I made sure that the modeling data set has a 50/50 split between patients readmitted in less than 30 days vs. all other patients. This technique also allows me to take into account the misclassification costs. The cost of misidentifying a patient likely to be readmitted is much greater than the cost of misidentifying a patient not likely to be readmitted.

***Part II: Modeling***

***Model 1: Principal Component Analysis***

At the moment, there are still 43 indicator variables. To produce a meaningful model, I should try to decrease the number of variables. Principal Component Analysis is a useful technique for determining which numerical variables are most important. In this technique, we aim for dimension reduction by trying to find a few variables that are weighted linear combinations of the original variables, yet retain the majority of the information from the dataset. This method is also useful since collinearity between some of the numerical variables is a strong possibility. For instance, there could be high correlation between number_medications and number_outpatient - the more medications a patient is on, the higher their number of outpatient visits likely are. PCA produces linear combinations of the independent variables that account for this.

One of the most important parts of running a PCA is scaling the variables, if necessary. If not scaled, one variable can have an outsized influence on the results simply due to the fact that it represents a measure that has high numerical values. From visual inspection of the data, it was clear that num_lab_procedures had very high numerical values. Hence, I made sure to scale the data before running the model.

```
  #selecting just the numeric variables from dataset
df.model.PCA <- df.modeling %>% select(c(time_in_hospital, num_lab_procedures, num_procedur
   num_medications, number_outpatient, number_emergency, number_inpatient, number_diagnoses
  #now run PCA. important to scale!
```

```
df.PCA.model.results <- prcomp(df.model.PCA, scale. = T)
df.PCA.model.results
```

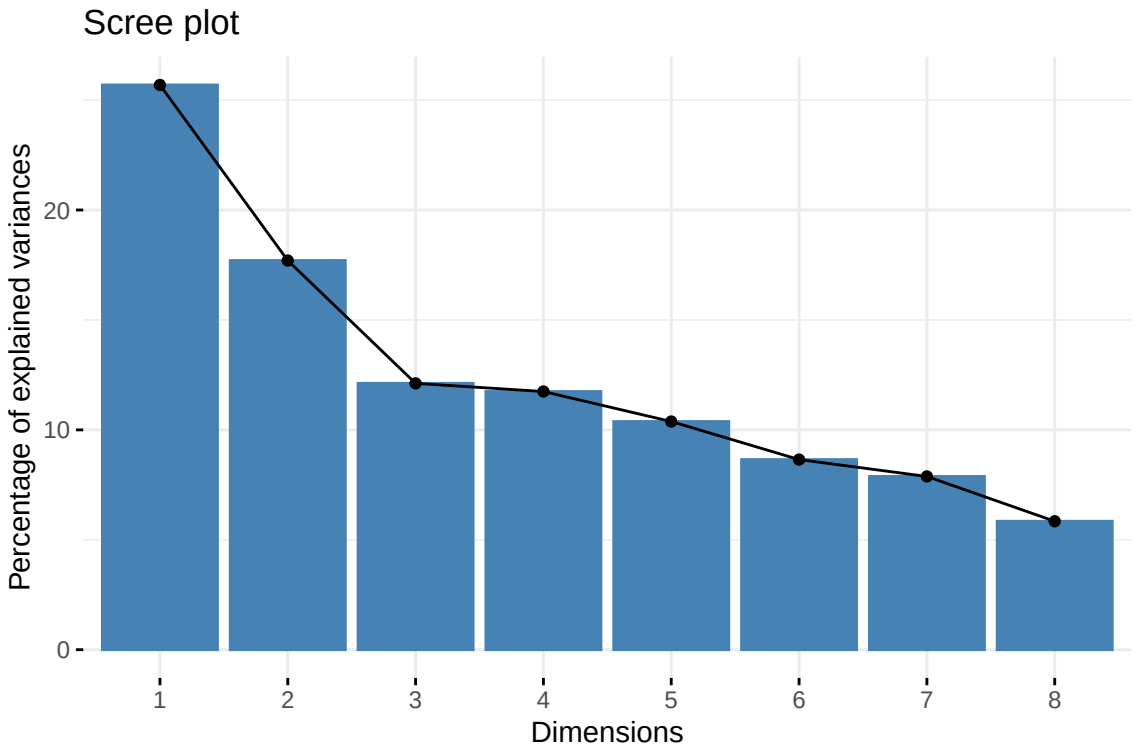
```
## Standard deviations (1, .., p=8):
## [1] 1.4335571 1.1900358 0.9844766 0.9693003 0.9112298 0.8317684 0.7940136
## [8] 0.6836343
##
## Rotation (n x k) = (8 x 8):
##                             PC1          PC2         PC3         PC4
## time_in_hospital     0.520276658 -0.039967924  0.09981164  0.17398605
## num_lab_procedures   0.391553983 -0.002729395 -0.00274347  0.61523960
## num_procedures       0.374115291 -0.208771756  0.20017899 -0.67646949
## num_medications      0.567928027 -0.010644823  0.04706516 -0.17298602
## number_outpatient    0.040419872  0.379578648 -0.75049682 -0.25952622
## number_emergency    -0.007798361  0.611807800  0.36179232 -0.15829597
## number_inpatient     0.048143314  0.630568089  0.30024590  0.07241225
## number_diagnoses     0.330878867  0.196701062 -0.40429766  0.07758185
##                             PC5          PC6         PC7          PC8
## time_in_hospital    -0.10260244 -0.272201489  0.61456641  0.474519198
## num_lab_procedures  -0.35078287  0.402565167 -0.42481766  0.050728320
## num_procedures      -0.07811376  0.126261170 -0.40751072  0.361470422
## num_medications     -0.04151035 -0.103647902  0.11476223 -0.787126919
## number_outpatient   -0.46473262 -0.001314789  0.05451954  0.068821759
## number_emergency     0.03804591  0.611891448  0.30626681  0.004594695
## number_inpatient    -0.04119922 -0.595817286 -0.38096590  0.053332407
## number_diagnoses     0.79967367  0.088044043 -0.14386094  0.120110885
```

```
summary(df.PCA.model.results)
```

```
## Importance of components:
##                            PC1    PC2    PC3    PC4    PC5     PC6     PC7
## Standard deviation     1.4336 1.1900 0.9845 0.9693 0.9112 0.83177 0.79401
## Proportion of Variance 0.2569 0.1770 0.1211 0.1174 0.1038 0.08648 0.07881
## Cumulative Proportion  0.2569 0.4339 0.5551 0.6725 0.7763 0.86277 0.94158
##                            PC8
## Standard deviation     0.68363
## Proportion of Variance 0.05842
## Cumulative Proportion  1.00000
```

Now let's visualize the influence each PC has.

```
fviz_eig(df.PCA.model.results)
```




We see that none of the eight principal components (PC) make an outweighted contribution towards the variability in the dataset. The first PC, which makes the most significant contribution and is the one that minimizes the sum-of-squared perpendicular distances from the line, only contributes ~23% of the variation. In this case, the first PC is most affected by the number of medications, the second PC by number_inpatient, the third PC by number_outpatient, the fourth by num_procedures, and so on. Thus, the top 4 most important numerical variables are the ones the PC emphasizes.

The visualization also indicates that besides the first two components, the other components each contribute ~6-12% towards the variation. Since the the first two components combined only account for 40% of the variation, a high proportion of information would be lost if we simply eliminated the other components.

The above results make it clear that 7 of the 8 PCs must be included to account for more than 90% of the variation in the model. Thus, the above results did not really help in reducing the number of numerical variables to ultiamtely include in the model.

***Model 2: Naive Bayes***

Since PCA looked just at the numerical variables, for the second model, I chose Naive Bayes, which is a good model for categorical variables. However, since the model is best suited for categorical variables, I binned the numerical variables and converted them into categorical variables.

One important model assumption that must be kept in mind is that Naive Bayes assumes that all the predictor variables are independent of each other. In reality, this is likely not the case. For instance, num_medications is likely related to time_in_hospital - the longer the hospital stay, the more medications that are likely to be administered.

```
  #new modeling data set for Naive Bayes
df.model.NB <- df.modeling
```

```r
#numerical variables must be converted to categorical
  #1) time_in_hospital. bins aren't exact, but it's the closest we can
    #get & also minimize # of bins
table(df.model.NB$time_in_hospital)
```

```
##
##    1    2    3    4    5    6    7    8    9   10   11   12   13   14
## 2742 3610 3931 3169 2323 1752 1412 1148  740  614  409  342  269  253
```

```r
break_pts <- c(-Inf, 3, 6, 9, Inf)
names <- c("1-3", "4-6", "7-9", "10-14")
df.model.NB$time_in_hospital.cat <- cut(df.model.NB$time_in_hospital,
      breaks = break_pts, labels = names)

  #2) number_outpatient. majority of patients did not have outpatient visit,
    #so only splitting into 2 categories
#table(df.model.NB$number_outpatient)
break_pts <- c(-Inf, 0, Inf)
names <- c("0", "1+")
df.model.NB$num_outpt.cat <- cut(df.model.NB$number_outpatient,
  breaks = break_pts, labels = names)

  #3) number_inpatient. again, majority of patients did not have inpatient
      #visit. but since there is slightly wider dispersion, dividing into
      #3 (unequal) bins.
#table(df.model.NB$number_inpatient)
break_pts <- c(-Inf, 0, 2, Inf)
names <- c("0", "1-2", "3+")
df.model.NB$num_inpt.cat <- cut(df.model.NB$number_inpatient,
    breaks = break_pts, labels = names)

  #4) number_diagnoses. ~50% of pts have 9+ diagnoses. splitting into 3 bins,
      #20%, 30%, 50%
#table(df.model.NB$number_diagnoses)
break_pts <- c(-Inf, 5, 8, Inf)
names <- c("0-5", "6-8", "9+")
df.model.NB$num_diag.cat <- cut(df.model.NB$number_diagnoses,
    breaks = break_pts, labels = names)

  #5) number_emergency. vast majority of pts didn't have emergency visit. so
      #only splitting into 2 categories.
#table(df.model.NB$number_emergency)
break_pts <- c(-Inf, 0, Inf)
names <- c("0", "1+")
df.model.NB$num_emer.cat <- cut(df.model.NB$number_emergency,
    breaks = break_pts, labels = names)
```

```
  #6) num_procedures. the range is 0 - 6, so let's just factor, & not bin.
break_pts <- c(-Inf, 1, 4, Inf)
names <- c("0-1", "2-4", "5-6")
df.model.NB$num_proc.cat <- cut(df.model.NB$num_procedures,
    breaks = break_pts, labels = names)

  #7) num_lab_procedures. since range is wide, I create 5 (rather than 4) bins
      #use rbin_quantiles to see where to split to get ~equal size bins
#table(df.model.NB$num_lab_procedures)
bins <- rbin_quantiles(df.model.NB, readmitted, num_lab_procedures, 5)
bins
```

```
## Binning Summary
## ----------------------------
## Method                Quantile
## Response              readmitted
## Predictor             num_lab_procedures
## Bins                  5
## Count                 22714
## Goods                 0
## Bads                  0
## Entropy               NA
## Information Value     NA
##
##
## # A tibble: 5 x 7
##   cut_point bin_count  good   bad   woe    iv entropy
##   <chr>         <int> <int> <int> <dbl> <dbl>   <dbl>
## 1 < 28           4273     0     0   NaN   NaN     NaN
## 2 < 40           4326     0     0   NaN   NaN     NaN
## 3 < 49           4662     0     0   NaN   NaN     NaN
## 4 < 60           4591     0     0   NaN   NaN     NaN
## 5 >= 60          4862     0     0   NaN   NaN     NaN
```

Bin cut points for num_lab_procedures should be: 28, 40, 49, 60.

```
break_pts <- c(-Inf, 28, 40, 49, 60, Inf)
names <- c("1-28", "29-40", "41-49", "50-60", "61+")
df.model.NB$num_lab_procs.cat <- cut(df.model.NB$num_lab_procedures,
    breaks = break_pts, labels = names)

  #8) num_medications. let's keep 4 bins the standard for this data set
bins <- rbin_quantiles(df.model.NB, readmitted, num_medications, 4)
bins
```

```
## Binning Summary
```

```
## ----------------------------
## Method                Quantile
## Response              readmitted
## Predictor             num_medications
## Bins                  4
## Count                 22714
## Goods                 0
## Bads                  0
## Entropy               NA
## Information Value     NA
##
##
## # A tibble: 4 x 7
##   cut_point bin_count  good   bad   woe    iv entropy
##   <chr>         <int> <int> <int> <dbl> <dbl>   <dbl>
## 1 < 11           5269     0     0   NaN   NaN     NaN
## 2 < 15           5165     0     0   NaN   NaN     NaN
## 3 < 21           6515     0     0   NaN   NaN     NaN
## 4 >= 21          5765     0     0   NaN   NaN     NaN
```

Bin cut points should be: 11, 15, 21

```
break_pts <- c(-Inf, 11, 15, 21, Inf)
names <- c("1-11", "12-15", "16-21", "22+")
df.model.NB$num_meds.cat <- cut(df.model.NB$num_medications,
    breaks = break_pts, labels = names)

  #also removing the numerical columns themselves, just for cleanliness' sake
df.model.NB <- df.model.NB %>% select(-c("time_in_hospital", "num_lab_procedures",
    "num_procedures", "num_medications", "number_outpatient",
    "number_emergency", "number_inpatient", "number_diagnoses"))
```

Now I run the Naive Bayes model itself, after splitting into train & test data sets.

```
set.seed(123)
diabetes.nb1 <- naiveBayes(readmitted ~ race + gender + age + admission_type_id +
      discharge_disposition_id + admission_source_id + metformin + glipizide +
      glyburide + insulin + change + diabetesMed + time_in_hospital.cat +
      num_outpt.cat + num_inpt.cat + num_diag.cat + num_emer.cat + num_proc.cat +
      num_lab_procs.cat,
    data = train.df.nb)
diabetes.nb1
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
```

```
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##       <30 >30 or No
## 0.5570719 0.4429281
##
## Conditional probabilities:
##            race
## Y           AfricanAmerican       Asian   Caucasian    Hispanic
##   <30           0.191504018 0.006084960 0.769230769 0.019862227
##   >30 or No     0.188791849 0.006404658 0.771324600 0.018777293
##            race
## Y                 Other
##   <30       0.013318025
##   >30 or No 0.014701601
##
##            gender
## Y              Female      Male Unknown/Invalid
##   <30       0.5406412 0.4593588       0.0000000
##   >30 or No 0.5376970 0.4623030       0.0000000
##
##            age
## Y                  0-19       20-39       40-59       60-79       80-99
##   <30       0.003386769 0.054865658 0.237412508 0.496726123 0.207608941
##   >30 or No 0.009938946 0.050546642 0.268919495 0.470396138 0.200198779
##
##            admission_type_id
## Y                          1           2           3           4
##   <30       0.5431248589 0.1807405735 0.1761119892 0.0001128923
##   >30 or No 0.5331534857 0.1830186000 0.1825926452 0.0001419849
##            admission_type_id
## Y                          5           6           7           8
##   <30       0.0430119666 0.0543011967 0.0000000000 0.0025965229
##   >30 or No 0.0462870936 0.0512565668 0.0004259548 0.0031236689
##
##            discharge_disposition_id
## Y                          1           2           3           4
##   <30       0.4954843080 0.0286746444 0.1746443893 0.0092571687
##   >30 or No 0.5949169388 0.0184580434 0.1331818827 0.0076671873
##            discharge_disposition_id
## Y                          5           6           7           8
##   <30       0.0220139986 0.1480018063 0.0073379995 0.0013547076
##   >30 or No 0.0105068863 0.1289223342 0.0056793980 0.0008519097
##            discharge_disposition_id
## Y                          9          10          11          12
```

```
##   <30       0.0006773538 0.0000000000 0.0000000000 0.0002257846
##   >30 or No 0.0000000000 0.0000000000 0.0186000284 0.0000000000
##             discharge_disposition_id
## Y                       13           14           15           16
##   <30       0.0015804922 0.0021449537 0.0024836306 0.0000000000
##   >30 or No 0.0036916087 0.0041175635 0.0004259548 0.0002839699
##             discharge_disposition_id
## Y                       17           18           19           20
##   <30       0.0000000000 0.0416572590 0.0000000000 0.0000000000
##   >30 or No 0.0002839699 0.0383359364 0.0002839699 0.0000000000
##             discharge_disposition_id
## Y                       22           23           24           25
##   <30       0.0483179047 0.0023707383 0.0005644615 0.0093700610
##   >30 or No 0.0168962090 0.0052534431 0.0008519097 0.0100809314
##             discharge_disposition_id
## Y                       27           28
##   <30       0.0000000000 0.0038383382
##   >30 or No 0.0000000000 0.0007099247
##
##             admission_source_id
## Y                        1            2            3            4
##   <30       0.2791826597 0.0101603071 0.0024836306 0.0287875367
##   >30 or No 0.2873775380 0.0106488712 0.0015618344 0.0318046287
##             admission_source_id
## Y                        5            6            7        Other
##   <30       0.0083540303 0.0170467374 0.5865883947 0.0004515692
##   >30 or No 0.0066732926 0.0258412608 0.5646741445 0.0005679398
##             admission_source_id
## Y                        9           17           20
##   <30       0.0011289230 0.0635583653 0.0022578460
##   >30 or No 0.0012778645 0.0681527758 0.0014198495
##
##             metformin
## Y                  Down           No       Steady           Up
##   <30       0.006434861 0.825355611 0.160645744 0.007563784
##   >30 or No 0.004827488 0.796677552 0.186568224 0.011926736
##
##             glipizide
## Y                  Down           No       Steady           Up
##   <30       0.008128246 0.875028223 0.109053963 0.007789569
##   >30 or No 0.004969473 0.876473094 0.110038336 0.008519097
##
##             glyburide
## Y                  Down           No       Steady           Up
##   <30       0.005080154 0.897380899 0.090088056 0.007450892
##   >30 or No 0.004827488 0.894789152 0.092574187 0.007809172
##
```

```
##            insulin
## Y               Down        No    Steady        Up
##   <30       0.1482276 0.4210883 0.3042448 0.1264394
##   >30 or No 0.1205452 0.4740877 0.3044157 0.1009513
##
##            change
## Y                 Ch        No
##   <30       0.4892752 0.5107248
##   >30 or No 0.4580434 0.5419566
##
##            diabetesMed
## Y                 No       Yes
##   <30       0.1974486 0.8025514
##   >30 or No 0.2347011 0.7652989
##
##            time_in_hospital.cat
## Y                 1-3        4-6        7-9      10-14
##   <30       0.41431474 0.33935426 0.16098442 0.08534658
##   >30 or No 0.48956411 0.30271191 0.12934829 0.07837569
##
##            num_outpt.cat
## Y                  0        1+
##   <30       0.7965681 0.2034319
##   >30 or No 0.8429646 0.1570354
##
##            num_inpt.cat
## Y                   0        1-2         3+
##   <30       0.50191917 0.34070896 0.15737187
##   >30 or No 0.69047281 0.25060344 0.05892375
##
##            num_diag.cat
## Y                 0-5       6-8        9+
##   <30       0.1556785 0.3037932 0.5405283
##   >30 or No 0.2151072 0.3000142 0.4848786
##
##            num_emer.cat
## Y                  0        1+
##   <30       0.8360804 0.1639196
##   >30 or No 0.8964930 0.1035070
##
##            num_proc.cat
## Y                 0-1        2-4        5-6
##   <30       0.67080605 0.26145857 0.06773538
##   >30 or No 0.65739032 0.26238819 0.08022150
##
##            num_lab_procs.cat
## Y                1-28     29-40     41-49     50-60       61+
```

```
##   <30       0.1889817 0.1967713 0.2085121 0.1991420 0.2065929
##   >30 or No 0.2109896 0.2016186 0.2003408 0.1972171 0.1898339
```

The interpretation of the Naive Bayes model is uniform. For instance, the num_emer.cat variable, amongst patients who were readmitted in less than 30 days, 83.3% of them did not have an emergency room visit in the past year, while the other 16.7% of patients did have at least one emergency room visit. Similarly, amongst patients who were readmitted in greater than 30 days or not readmitted at all, 89.4% of them did not have emergency room visit in the past year, while the other 10.6% did. The probabilities represent: P(X | Y). Given Y, what is the probability of X?

Now, I evaluate the accuracy of both the training and test data sets.

Training Data Confusion Matrix:

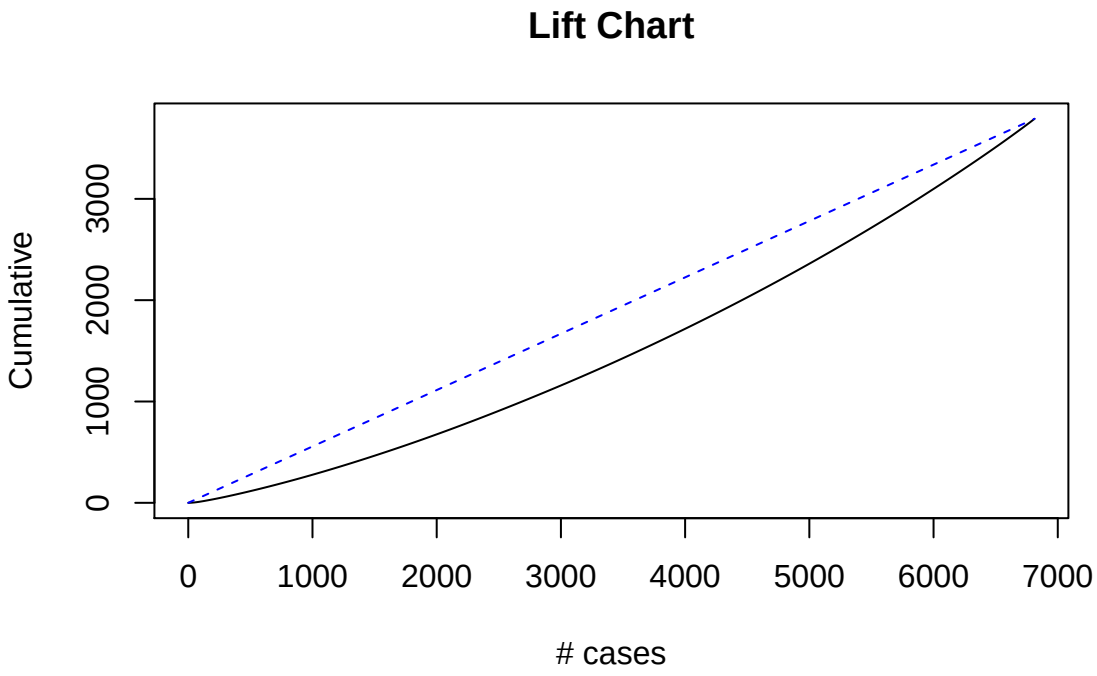
```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction   <30 >30 or No
##   <30       6125      3254
##   >30 or No 2733      3789
##
##                Accuracy : 0.6235
##                  95% CI : (0.6159, 0.631)
##     No Information Rate : 0.5571
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.2312
##
##  Mcnemar's Test P-Value : 1.812e-11
##
##             Sensitivity : 0.6915
##             Specificity : 0.5380
##          Pos Pred Value : 0.6531
##          Neg Pred Value : 0.5810
##              Prevalence : 0.5571
##          Detection Rate : 0.3852
##    Detection Prevalence : 0.5898
##       Balanced Accuracy : 0.6147
##
##        'Positive' Class : <30
##
```

Test Data Confusion Matrix:

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction   <30 >30 or No
```

```
##   <30       2624      1452
##   >30 or No 1171      1566
##
##                Accuracy : 0.615
##                  95% CI : (0.6033, 0.6266)
##     No Information Rate : 0.557
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.2123
##
##  Mcnemar's Test P-Value : 4.574e-08
##
##             Sensitivity : 0.6914
##             Specificity : 0.5189
##          Pos Pred Value : 0.6438
##          Neg Pred Value : 0.5722
##              Prevalence : 0.5570
##          Detection Rate : 0.3851
##    Detection Prevalence : 0.5983
##       Balanced Accuracy : 0.6052
##
##        'Positive' Class : <30
##
```

The above Naive Bayes model only resulted in a 62.35% accuracy rate for the training data, and 61.5% for the test set. In addition, the sensitivity rate for the test is a decent 69.14%. Below is a lift chart to visualize the performance of this model.




The lift chart indicates that the model does not perform much better than random. A 'good' model would have a lift chart with a line to the far left corner of the naive line.

I then ran another iteration of Naive Bayes, this time with fewer predictor variables, since 19 is quite a few. The only predictor variables I retaine are: race, age, adminission_type_id, metformin, time_in_hospital category, and num_inpatient category.

```
set.seed(123)
diabetes.nb2 <- naiveBayes(readmitted ~ race + age + admission_type_id +
    metformin + time_in_hospital.cat + num_inpt.cat, data = train.df.nb)
diabetes.nb2
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##       <30 >30 or No
## 0.5570719 0.4429281
##
## Conditional probabilities:
##             race
## Y            AfricanAmerican       Asian   Caucasian    Hispanic
```

```
##   <30           0.191504018 0.006084960 0.769230769 0.019862227
##   >30 or No     0.188791849 0.006404658 0.771324600 0.018777293
##            race
## Y                 Other
##   <30       0.013318025
##   >30 or No 0.014701601
##
##            age
## Y                  0-19       20-39       40-59       60-79       80-99
##   <30       0.003386769 0.054865658 0.237412508 0.496726123 0.207608941
##   >30 or No 0.009938946 0.050546642 0.268919495 0.470396138 0.200198779
##
##            admission_type_id
## Y                      1            2            3            4
##   <30       0.5431248589 0.1807405735 0.1761119892 0.0001128923
##   >30 or No 0.5331534857 0.1830186000 0.1825926452 0.0001419849
##            admission_type_id
## Y                      5            6            7            8
##   <30       0.0430119666 0.0543011967 0.0000000000 0.0025965229
##   >30 or No 0.0462870936 0.0512565668 0.0004259548 0.0031236689
##
##            metformin
## Y                  Down          No      Steady          Up
##   <30       0.006434861 0.825355611 0.160645744 0.007563784
##   >30 or No 0.004827488 0.796677552 0.186568224 0.011926736
##
##            time_in_hospital.cat
## Y                  1-3        4-6        7-9      10-14
##   <30       0.41431474 0.33935426 0.16098442 0.08534658
##   >30 or No 0.48956411 0.30271191 0.12934829 0.07837569
##
##            num_inpt.cat
## Y                     0        1-2         3+
##   <30       0.50191917 0.34070896 0.15737187
##   >30 or No 0.69047281 0.25060344 0.05892375
```

Training Data Confusion Matrix:

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction   <30 >30 or No
##   <30       6233      3714
##   >30 or No 2625      3329
##
##                Accuracy : 0.6013
##                  95% CI : (0.5937, 0.609)
```

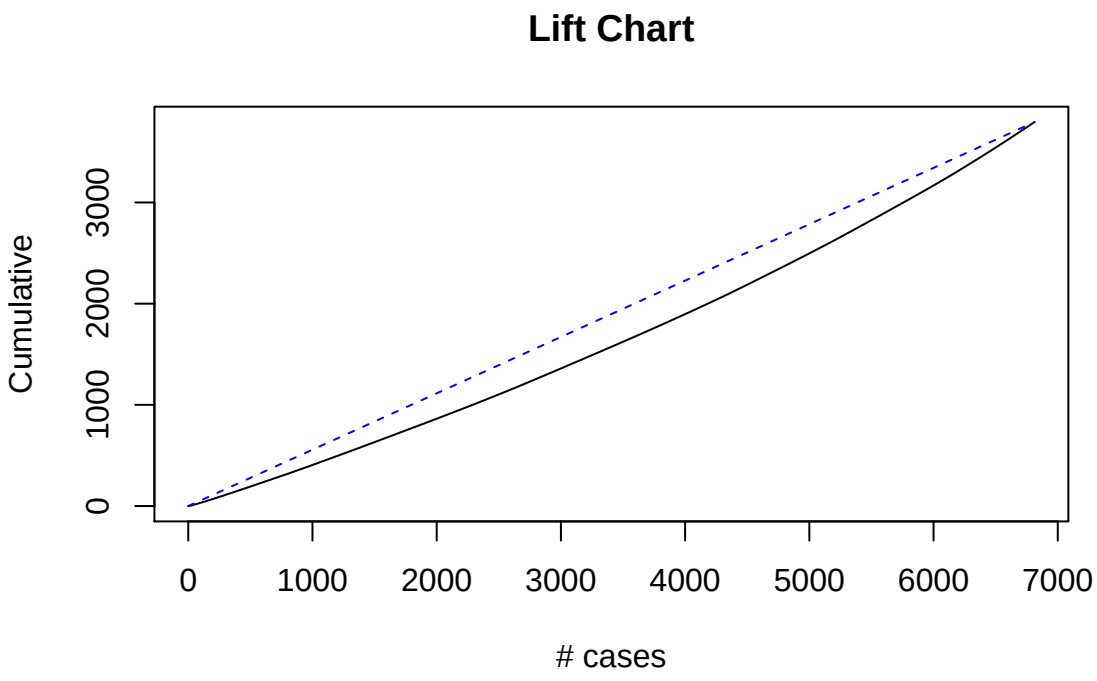
```
##     No Information Rate : 0.5571
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.1792
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.7037
##             Specificity : 0.4727
##          Pos Pred Value : 0.6266
##          Neg Pred Value : 0.5591
##              Prevalence : 0.5571
##          Detection Rate : 0.3920
##    Detection Prevalence : 0.6256
##       Balanced Accuracy : 0.5882
##
##        'Positive' Class : <30
##
```

Test Data Confusion Matrix:

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction   <30 >30 or No
##   <30       2611      1653
##   >30 or No 1184      1365
##
##                Accuracy : 0.5836
##                  95% CI : (0.5718, 0.5953)
##     No Information Rate : 0.557
##     P-Value [Acc > NIR] : 5.124e-06
##
##                   Kappa : 0.1426
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.6880
##             Specificity : 0.4523
##          Pos Pred Value : 0.6123
##          Neg Pred Value : 0.5355
##              Prevalence : 0.5570
##          Detection Rate : 0.3832
##    Detection Prevalence : 0.6259
##       Balanced Accuracy : 0.5701
##
##        'Positive' Class : <30
```

##

The absolute accuracies are lower than they were in the first iteration, as is the sensitivity, at 68.8% (versus the first iteration, where it was 69.14%).




Upon eliminating 13 predictors and only keeping 6, the accuracy on the test data set decreased to 58.36%, as did the sensitivity. Thus, between these two iterations of Naive Bayes, the first is the preferred option.

***Model 3: Support Vector Machine (SVM)***

The third model I explore is Support Vector Machines (SVM). In general, this is a powerful method for classification because it has the capacity to handle non-linear relationships between the predictor variables and the target variable. The model aims to create a linear, radial, polynomial or sigmoidal hyperplane that best disrciminates between the two target variables (i.e. create the largest distance between the support vectors and the hyperplane).

First, I recreating training & test data sets for the SVM model. For this model, there's no need to pre-process the dataset too much because SVM can handle both numerical & categorical variables. We simply have to omit the NA's, as SVM is unable to handle missing data efficiently.

```
  #new modeling data set for SVM. removing diabetesMed column,
  #since that info is already captured in the 4 medication columns.
df.model.SVM <- df.modeling %>% select(-c(diabetesMed))

  #also omitting NA's, because SVM can't handle NA's
df.model.SVM <- na.omit(df.model.SVM)
```

All iterations of SVM will include all 19 indicator variables. The first iteration is for a linear kernel. I use the SVM function, which automatically scales the variables.

```
svm.linear <- svm(readmitted ~ ., data = train.df.svm, kernel = "linear")
summary(svm.linear)
```

```
##
## Call:
## svm(formula = readmitted ~ ., data = train.df.svm, kernel = "linear")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  1
##
## Number of Support Vectors:  6438
##
##  ( 3215 3223 )
##
##
## Number of Classes:  2
##
## Levels:
##  <30 >30 or No
```

Test data accuracy:

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction   <30 >30 or No
##   <30       1156       554
##   >30 or No  672       972
##
##                Accuracy : 0.6345
##                  95% CI : (0.6179, 0.6508)
##     No Information Rate : 0.545
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.2676
##
##  Mcnemar's Test P-Value : 0.0008333
##
##             Sensitivity : 0.6324
##             Specificity : 0.6370
```

```
##          Pos Pred Value : 0.6760
##          Neg Pred Value : 0.5912
##              Prevalence : 0.5450
##          Detection Rate : 0.3447
##    Detection Prevalence : 0.5098
##       Balanced Accuracy : 0.6347
##
##        'Positive' Class : <30
##
```

This model results in a 63.45% accuracy, with a sensitivity of 63.24%. I next run more iterations, with different kernels, to see if we can get improved results.

The second iteration of the SVM model implements a radial kernel.

```
svm.radial <- svm(readmitted ~ ., data = train.df.svm, kernel = "radial")
summary(svm.radial)
```

```
##
## Call:
## svm(formula = readmitted ~ ., data = train.df.svm, kernel = "radial")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  1
##
## Number of Support Vectors:  6746
##
##  ( 3361 3385 )
##
##
## Number of Classes:  2
##
## Levels:
##  <30 >30 or No
```

Test data accuracy:

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction   <30 >30 or No
##   <30       1268       685
##   >30 or No  560       841
```

```
##
##                Accuracy : 0.6288
##                  95% CI : (0.6122, 0.6452)
##     No Information Rate : 0.545
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.2464
##
##  Mcnemar's Test P-Value : 0.0004409
##
##             Sensitivity : 0.6937
##             Specificity : 0.5511
##          Pos Pred Value : 0.6493
##          Neg Pred Value : 0.6003
##              Prevalence : 0.5450
##          Detection Rate : 0.3781
##    Detection Prevalence : 0.5823
##       Balanced Accuracy : 0.6224
##
##        'Positive' Class : <30
##
```

With the radial kernel, we get a slightly lower accuracy at 62.88%. But, more importantly, the range between the sensitivity and specificity increases significantly. While the sensitivity increased to 69.37%, the specificity deteriorated to 55.11%. If the goal is the have a model with a high sensitivity, since patients who are readmitted in less than 30 days are costly, then this is a reasonable model.

The third iteration of the SVM model implements a polynomial kernel.

```
svm.poly <- svm(readmitted ~ ., data = train.df.svm, kernel = "polynomial")
summary(svm.poly)
```

```
##
## Call:
## svm(formula = readmitted ~ ., data = train.df.svm, kernel = "polynomial")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  polynomial
##        cost:  1
##      degree:  3
##      coef.0:  0
##
## Number of Support Vectors:  7187
##
```

```
##  ( 3560 3627 )
##
##
## Number of Classes:  2
##
## Levels:
##  <30 >30 or No
```

Test data accuracy:

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction   <30 >30 or No
##   <30       1827      1525
##   >30 or No    1         1
##
##                Accuracy : 0.545
##                  95% CI : (0.528, 0.562)
##     No Information Rate : 0.545
##     P-Value [Acc > NIR] : 0.5071
##
##                   Kappa : 1e-04
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.9994530
##             Specificity : 0.0006553
##          Pos Pred Value : 0.5450477
##          Neg Pred Value : 0.5000000
##              Prevalence : 0.5450209
##          Detection Rate : 0.5447227
##    Detection Prevalence : 0.9994037
##       Balanced Accuracy : 0.5000541
##
##        'Positive' Class : <30
##
```

This kernel produces a model with a lower accuracy than both the first and second iterations. However, the sensitivity is strikingly high, at 99.94%. Unsurprisingly, the specificity is nearly 0%. Even if our goal is to have a model with a high sensitivity, we need for the model to also have some specificity power. Clearly, this model is unable to predict which patients will not be readmitted. Hence, this is not a viable model.

The last iteration of the SVM model implements a sigmoid kernel.

```r
svm.sig <- svm(readmitted ~ ., data = train.df.svm, kernel = "sigmoid")
summary(svm.sig)
```

```
##
## Call:
## svm(formula = readmitted ~ ., data = train.df.svm, kernel = "sigmoid")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  sigmoid
##        cost:  1
##      coef.0:  0
##
## Number of Support Vectors:  6809
##
##  ( 3402 3407 )
##
##
## Number of Classes:  2
##
## Levels:
##  <30 >30 or No
```

Test data accuracy:

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction   <30 >30 or No
##   <30       1262       685
##   >30 or No  566       841
##
##                Accuracy : 0.627
##                  95% CI : (0.6104, 0.6434)
##     No Information Rate : 0.545
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.2431
##
##  Mcnemar's Test P-Value : 0.0008493
##
##             Sensitivity : 0.6904
##             Specificity : 0.5511
##          Pos Pred Value : 0.6482
##          Neg Pred Value : 0.5977
```

```
##              Prevalence : 0.5450
##          Detection Rate : 0.3763
##    Detection Prevalence : 0.5805
##       Balanced Accuracy : 0.6207
##
##        'Positive' Class : <30
##
```

This sigmoid kernel model isn't strikingly different from the first or second iterations. The accuracy is still in the low 60% range, and the range between the sensitivity and specificity is about the same as it was with the radial kernel. If the hospital wishes to place nearly equal emphasis on readmitted vs. not readmitted patients, this model is a good candidate.

Due to computer memory limitations, a hyperparameter optimization model was not possible. But that is one area in which this model coule be improved, given the appropriate resources. If I had been able to optimize the model, I would have had the option to choose amongst cost values that represent the tradeoff between minimizing the training set error and maximizing the margins.

***Model 4: Logistic Regression***

For logistic regression, I revert back to using the entire data set. This is because with the smaller, oversampled data set, there were some categories missing for some columns. This led to errors when trying to predict the probability of a new observation falling into the 'less than 30 day readmission' category. Hence, I went back and created the logistic regression data set from the original (but processed) dataset.

```
  #new modeling data set for Logistic regression. removing diabetesMed column
df.model.log <- diabetes.df %>% select(-c(patient_nbr ,diabetesMed))

  #creating new variable that takes on value of 1 if readmitted in <30 days, and 0 otheru
df.model.log$isReadmit = 1 * (df.model.log$readmitted == "<30")

  #need to factorize age variables
df.model.log$age <- as.factor(df.model.log$age)

  #dropping 'readmitted' column, bc I created a new binary column to capture that
df.model.log <- df.model.log %>% select(-c(readmitted))

diabetes.df$race[diabetes.df$race == "?"] <- NA

  #disch to delete
dis_keep <- count(df.model.log, vars = "discharge_disposition_id")
dis_keep <- subset(dis_keep,subset=(dis_keep$freq>20))
df.model.log <- df.model.log %>% filter(discharge_disposition_id %in% dis_keep$discharge_di

df.model.log <- na.omit(df.model.log)
```

```r
logit.reg <- glm(isReadmit ~., data = train.df.log, family = "binomial")
#options(scipen = 999)
summary(logit.reg)
```

```
##
## Call:
## glm(formula = isReadmit ~ ., family = "binomial", data = train.df.log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2084  -0.4942  -0.4099  -0.3487   2.9984
##
## Coefficients:
##                                                    Estimate
## (Intercept)                                      -1.969e+00
## raceAsian                                         2.369e-01
## raceCaucasian                                     1.606e-02
## raceHispanic                                      1.064e-01
## raceOther                                        -4.314e-02
## genderMale                                        4.174e-02
## age20-39                                          4.298e-01
## age40-59                                          2.968e-01
## age60-79                                          4.028e-01
## age80-99                                          3.981e-01
## admission_type_id2                               -1.045e-01
## admission_type_id3                               -1.515e-01
## admission_type_id4                               -1.379e+01
## admission_type_id5                               -6.655e-02
## admission_type_id6                                2.246e-01
## admission_type_id8                               -1.781e-01
## discharge_disposition_id2                         6.537e-01
## discharge_disposition_id3                         5.371e-01
## discharge_disposition_id4                         1.578e-01
## discharge_disposition_id5                         9.857e-01
## discharge_disposition_id6                         1.926e-01
## discharge_disposition_id7                         8.197e-02
## discharge_disposition_id8                         5.230e-01
## discharge_disposition_id9                         1.766e+00
## discharge_disposition_id11                       -1.449e+01
## discharge_disposition_id13                       -1.357e+00
## discharge_disposition_id14                       -6.088e-01
## discharge_disposition_id15                        2.131e+00
## discharge_disposition_id18                        4.584e-01
## discharge_disposition_id22                        1.108e+00
## discharge_disposition_id23                       -3.996e-01
## discharge_disposition_id24                        8.941e-01
```

```
## discharge_disposition_id25                                  -2.253e-01
## discharge_disposition_id28                                   2.194e+00
## admission_source_id2                                        -5.317e-01
## admission_source_id3                                        -7.048e-02
## admission_source_id4                                        -3.166e-01
## admission_source_id5                                        -5.145e-01
## admission_source_id6                                        -1.977e-01
## admission_source_id7                                        -1.380e-01
## admission_source_idOther                                    -1.451e+01
## admission_source_id9                                         9.502e-01
## admission_source_id17                                       -2.453e-01
## time_in_hospital                                             2.843e-03
## medical_specialtyCardiology                                 -2.752e-01
## medical_specialtyEmergency/Trauma                           -2.591e-01
## medical_specialtyEndocrinology                              -9.736e-01
## medical_specialtyFamily/GeneralPractice                     -8.917e-02
## medical_specialtyGastroenterology                           -1.924e-01
## medical_specialtyGynecology                                 -1.419e+01
## medical_specialtyHematology                                  9.379e-01
## medical_specialtyHematology/Oncology                         7.321e-01
## medical_specialtyHospitalist                                -2.355e-01
## medical_specialtyInfectiousDiseases                          5.997e-01
## medical_specialtyInternalMedicine                           -8.731e-02
## medical_specialtyNephrology                                  2.039e-01
## medical_specialtyNeurology                                  -6.108e-01
## medical_specialtyObsterics&Gynecology-GynecologicOnco       -5.626e-02
## medical_specialtyObstetricsandGynecology                    -6.112e-01
## medical_specialtyOncology                                    5.345e-01
## medical_specialtyOphthalmology                              -1.123e+00
## medical_specialtyOrthopedics                                -3.512e-01
## medical_specialtyOrthopedics-Reconstructive                 -7.151e-01
## medical_specialtyOsteopath                                  -3.080e-01
## medical_specialtyOtolaryngology                             -2.176e+00
## medical_specialtyPediatrics                                 -3.925e-01
## medical_specialtyPediatrics-CriticalCare                    -1.216e+00
## medical_specialtyPediatrics-Endocrinology                   -1.793e+00
## medical_specialtyPediatrics-Pulmonology                     -1.482e-01
## medical_specialtyPhysicalMedicineandRehabilitation           3.776e-01
## medical_specialtyPodiatry                                   -4.563e-01
## medical_specialtyPsychiatry                                  4.750e-02
## medical_specialtyPsychology                                 -4.729e-01
## medical_specialtyPulmonology                                -2.712e-01
## medical_specialtyRadiologist                                -3.033e-01
## medical_specialtyRadiology                                   3.212e-01
## medical_specialtySurgeon                                    -3.078e-01
## medical_specialtySurgery-Cardiovascular                     -1.011e+00
## medical_specialtySurgery-Cardiovascular/Thoracic            -8.510e-01
```

```
## medical_specialtySurgery-General                -1.316e-01
## medical_specialtySurgery-Neuro                  -5.039e-01
## medical_specialtySurgery-Plastic                 3.724e-02
## medical_specialtySurgery-Thoracic               -5.724e-02
## medical_specialtySurgery-Vascular                7.749e-02
## medical_specialtySurgicalSpecialty              -6.680e-02
## medical_specialtyUrology                         4.533e-03
## num_lab_procedures                              -2.341e-03
## num_procedures                                  -6.348e-03
## num_medications                                  3.801e-03
## number_outpatient                               -8.976e-03
## number_emergency                                 1.884e-02
## number_inpatient                                 2.643e-01
## number_diagnoses                                 5.402e-02
## metforminNo                                     -6.180e-01
## metforminSteady                                 -6.185e-01
## metforminUp                                     -7.033e-01
## glipizideNo                                     -5.302e-01
## glipizideSteady                                 -4.785e-01
## glipizideUp                                     -4.799e-01
## glyburideNo                                      3.055e-01
## glyburideSteady                                  3.847e-01
## glyburideUp                                      5.233e-01
## insulinNo                                       -3.160e-01
## insulinSteady                                   -1.740e-01
## insulinUp                                       -1.871e-02
## changeNo                                         5.678e-02
##                                                 Std. Error z value
## (Intercept)                                      5.801e-01  -3.394
## raceAsian                                        1.866e-01   1.269
## raceCaucasian                                    4.602e-02   0.349
## raceHispanic                                     1.228e-01   0.867
## raceOther                                        1.537e-01  -0.281
## genderMale                                       3.623e-02   1.152
## age20-39                                         3.241e-01   1.326
## age40-59                                         3.189e-01   0.931
## age60-79                                         3.188e-01   1.263
## age80-99                                         3.210e-01   1.240
## admission_type_id2                               6.140e-02  -1.702
## admission_type_id3                               7.595e-02  -1.995
## admission_type_id4                               2.400e+03  -0.006
## admission_type_id5                               1.458e-01  -0.457
## admission_type_id6                               9.940e-02   2.259
## admission_type_id8                               2.773e-01  -0.642
## discharge_disposition_id2                        1.035e-01   6.318
## discharge_disposition_id3                        5.475e-02   9.810
## discharge_disposition_id4                        1.916e-01   0.824
```

```
## discharge_disposition_id5                                    1.201e-01   8.210
## discharge_disposition_id6                                    6.046e-02   3.186
## discharge_disposition_id7                                    2.342e-01   0.350
## discharge_disposition_id8                                    5.168e-01   1.012
## discharge_disposition_id9                                    1.232e+00   1.434
## discharge_disposition_id11                                   9.887e+01  -0.147
## discharge_disposition_id13                                   5.154e-01  -2.633
## discharge_disposition_id14                                   4.740e-01  -1.284
## discharge_disposition_id15                                   4.980e-01   4.280
## discharge_disposition_id18                                   1.338e-01   3.426
## discharge_disposition_id22                                   9.374e-02  11.819
## discharge_disposition_id23                                   3.210e-01  -1.245
## discharge_disposition_id24                                   1.177e+00   0.760
## discharge_disposition_id25                                   1.729e-01  -1.303
## discharge_disposition_id28                                   5.494e-01   3.994
## admission_source_id2                                         3.255e-01  -1.634
## admission_source_id3                                         6.267e-01  -0.112
## admission_source_id4                                         1.312e-01  -2.414
## admission_source_id5                                         1.832e-01  -2.808
## admission_source_id6                                         1.294e-01  -1.528
## admission_source_id7                                         6.795e-02  -2.031
## admission_source_idOther                                     6.964e+02  -0.021
## admission_source_id9                                         4.731e-01   2.009
## admission_source_id17                                        1.191e-01  -2.059
## time_in_hospital                                             7.119e-03   0.399
## medical_specialtyCardiology                                  2.901e-01  -0.949
## medical_specialtyEmergency/Trauma                            2.890e-01  -0.897
## medical_specialtyEndocrinology                               5.878e-01  -1.656
## medical_specialtyFamily/GeneralPractice                      2.865e-01  -0.311
## medical_specialtyGastroenterology                            3.317e-01  -0.580
## medical_specialtyGynecology                                  3.776e+02  -0.038
## medical_specialtyHematology                                  4.351e-01   2.156
## medical_specialtyHematology/Oncology                         3.610e-01   2.028
## medical_specialtyHospitalist                                 6.072e-01  -0.388
## medical_specialtyInfectiousDiseases                          5.925e-01   1.012
## medical_specialtyInternalMedicine                            2.848e-01  -0.307
## medical_specialtyNephrology                                  2.957e-01   0.689
## medical_specialtyNeurology                                   4.502e-01  -1.357
## medical_specialtyObsterics&Gynecology-GynecologicOnco        8.060e-01  -0.070
## medical_specialtyObstetricsandGynecology                     3.555e-01  -1.719
## medical_specialtyOncology                                    3.295e-01   1.622
## medical_specialtyOphthalmology                               1.058e+00  -1.062
## medical_specialtyOrthopedics                                 3.059e-01  -1.148
## medical_specialtyOrthopedics-Reconstructive                  3.175e-01  -2.252
## medical_specialtyOsteopath                                   6.840e-01  -0.450
## medical_specialtyOtolaryngology                              1.047e+00  -2.078
## medical_specialtyPediatrics                                  4.170e-01  -0.941
```

```
## medical_specialtyPediatrics-CriticalCare                1.085e+00  -1.122
## medical_specialtyPediatrics-Endocrinology                1.078e+00  -1.663
## medical_specialtyPediatrics-Pulmonology                  8.409e-01  -0.176
## medical_specialtyPhysicalMedicineandRehabilitation       3.419e-01   1.105
## medical_specialtyPodiatry                                5.197e-01  -0.878
## medical_specialtyPsychiatry                              3.115e-01   0.152
## medical_specialtyPsychology                              5.102e-01  -0.927
## medical_specialtyPulmonology                             3.158e-01  -0.859
## medical_specialtyRadiologist                             3.121e-01  -0.972
## medical_specialtyRadiology                               5.676e-01   0.566
## medical_specialtySurgeon                                 6.714e-01  -0.458
## medical_specialtySurgery-Cardiovascular                  6.592e-01  -1.533
## medical_specialtySurgery-Cardiovascular/Thoracic         3.594e-01  -2.368
## medical_specialtySurgery-General                         2.921e-01  -0.450
## medical_specialtySurgery-Neuro                           3.654e-01  -1.379
## medical_specialtySurgery-Plastic                         6.373e-01   0.058
## medical_specialtySurgery-Thoracic                        4.628e-01  -0.124
## medical_specialtySurgery-Vascular                        3.212e-01   0.241
## medical_specialtySurgicalSpecialty                       7.977e-01  -0.084
## medical_specialtyUrology                                 3.209e-01   0.014
## num_lab_procedures                                       1.092e-03  -2.144
## num_procedures                                           1.234e-02  -0.515
## num_medications                                          2.941e-03   1.292
## number_outpatient                                        1.800e-02  -0.499
## number_emergency                                         1.170e-02   1.611
## number_inpatient                                         1.112e-02  23.758
## number_diagnoses                                         1.090e-02   4.954
## metforminNo                                              2.118e-01  -2.918
## metforminSteady                                          2.129e-01  -2.905
## metforminUp                                              2.764e-01  -2.544
## glipizideNo                                              2.076e-01  -2.554
## glipizideSteady                                          2.108e-01  -2.270
## glipizideUp                                              2.791e-01  -1.719
## glyburideNo                                              2.959e-01   1.032
## glyburideSteady                                          2.995e-01   1.284
## glyburideUp                                              3.528e-01   1.484
## insulinNo                                                7.622e-02  -4.147
## insulinSteady                                            6.617e-02  -2.630
## insulinUp                                                6.590e-02  -0.284
## changeNo                                                 5.905e-02   0.962
##                                                         Pr(>|z|)
## (Intercept)                                             0.000689 ***
## raceAsian                                               0.204267
## raceCaucasian                                           0.727099
## raceHispanic                                            0.386161
## raceOther                                               0.778927
## genderMale                                              0.249277
```

```
## age20-39                                        0.184722
## age40-59                                        0.351964
## age60-79                                        0.206433
## age80-99                                        0.214988
## admission_type_id2                              0.088722 .
## admission_type_id3                              0.046090 *
## admission_type_id4                              0.995416
## admission_type_id5                              0.647967
## admission_type_id6                              0.023871 *
## admission_type_id8                              0.520573
## discharge_disposition_id2                       2.64e-10 ***
## discharge_disposition_id3                        < 2e-16 ***
## discharge_disposition_id4                       0.409945
## discharge_disposition_id5                        < 2e-16 ***
## discharge_disposition_id6                       0.001442 **
## discharge_disposition_id7                       0.726351
## discharge_disposition_id8                       0.311501
## discharge_disposition_id9                       0.151636
## discharge_disposition_id11                      0.883448
## discharge_disposition_id13                      0.008469 **
## discharge_disposition_id14                      0.198993
## discharge_disposition_id15                      1.87e-05 ***
## discharge_disposition_id18                      0.000614 ***
## discharge_disposition_id22                       < 2e-16 ***
## discharge_disposition_id23                      0.213175
## discharge_disposition_id24                      0.447452
## discharge_disposition_id25                      0.192527
## discharge_disposition_id28                      6.50e-05 ***
## admission_source_id2                            0.102357
## admission_source_id3                            0.910460
## admission_source_id4                            0.015774 *
## admission_source_id5                            0.004978 **
## admission_source_id6                            0.126448
## admission_source_id7                            0.042298 *
## admission_source_idOther                        0.983379
## admission_source_id9                            0.044571 *
## admission_source_id17                           0.039466 *
## time_in_hospital                                0.689611
## medical_specialtyCardiology                     0.342760
## medical_specialtyEmergency/Trauma               0.369861
## medical_specialtyEndocrinology                  0.097681 .
## medical_specialtyFamily/GeneralPractice         0.755614
## medical_specialtyGastroenterology               0.561933
## medical_specialtyGynecology                     0.970026
## medical_specialtyHematology                     0.031103 *
## medical_specialtyHematology/Oncology            0.042554 *
## medical_specialtyHospitalist                    0.698142
```

```
## medical_specialtyInfectiousDiseases                    0.311423
## medical_specialtyInternalMedicine                       0.759166
## medical_specialtyNephrology                             0.490514
## medical_specialtyNeurology                              0.174871
## medical_specialtyObsterics&Gynecology-GynecologicOnco 0.944354
## medical_specialtyObstetricsandGynecology                0.085600 .
## medical_specialtyOncology                               0.104790
## medical_specialtyOphthalmology                          0.288298
## medical_specialtyOrthopedics                            0.250924
## medical_specialtyOrthopedics-Reconstructive             0.024299 *
## medical_specialtyOsteopath                              0.652464
## medical_specialtyOtolaryngology                         0.037677 *
## medical_specialtyPediatrics                             0.346582
## medical_specialtyPediatrics-CriticalCare                0.262060
## medical_specialtyPediatrics-Endocrinology               0.096254 .
## medical_specialtyPediatrics-Pulmonology                 0.860071
## medical_specialtyPhysicalMedicineandRehabilitation      0.269331
## medical_specialtyPodiatry                               0.379881
## medical_specialtyPsychiatry                             0.878798
## medical_specialtyPsychology                             0.353926
## medical_specialtyPulmonology                            0.390598
## medical_specialtyRadiologist                            0.331141
## medical_specialtyRadiology                              0.571530
## medical_specialtySurgeon                                0.646665
## medical_specialtySurgery-Cardiovascular                 0.125264
## medical_specialtySurgery-Cardiovascular/Thoracic        0.017886 *
## medical_specialtySurgery-General                        0.652434
## medical_specialtySurgery-Neuro                          0.167877
## medical_specialtySurgery-Plastic                        0.953410
## medical_specialtySurgery-Thoracic                       0.901570
## medical_specialtySurgery-Vascular                       0.809359
## medical_specialtySurgicalSpecialty                      0.933259
## medical_specialtyUrology                                0.988731
## num_lab_procedures                                      0.032024 *
## num_procedures                                          0.606887
## num_medications                                         0.196208
## number_outpatient                                       0.618042
## number_emergency                                        0.107225
## number_inpatient                                         < 2e-16 ***
## number_diagnoses                                        7.25e-07 ***
## metforminNo                                             0.003527 **
## metforminSteady                                         0.003672 **
## metforminUp                                             0.010957 *
## glipizideNo                                             0.010649 *
## glipizideSteady                                         0.023185 *
## glipizideUp                                             0.085607 .
## glyburideNo                                             0.301851
```

```
## glyburideSteady                                          0.199014
## glyburideUp                                              0.137917
## insulinNo                                                3.37e-05 ***
## insulinSteady                                            0.008541 **
## insulinUp                                                0.776509
## changeNo                                                 0.336245
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 24330  on 35482  degrees of freedom
## Residual deviance: 22728  on 35377  degrees of freedom
## AIC: 22940
##
## Number of Fisher Scoring iterations: 15
```

Logistic regression models are usually more interpretable when looking at the odds, so let's calculate the odds based on the probabilities. However, I only display the categories that have a probability of less than 0.01. In other words, I am only interested in the categories that had a less than 1% chance of obtaining the values they did, simply due to random error. This resulted in 27 rows - these are the variable categories that are most significant and most likely contribute to the readmission differences we see.

```
logreg.odds <- round(data.frame(summary(logit.reg)$coefficients,
    odds = exp(coef(logit.reg))), 5)
  #only selecting rows that are significant, based on alpha of 0.01
logreg.odds <- logreg.odds[logreg.odds $Pr...z.. <= 0.01,]
logreg.odds
```

```
##                            Estimate Std..Error  z.value Pr...z..    odds
## (Intercept)                -1.96872    0.58006 -3.39397  0.00069 0.13964
## discharge_disposition_id2   0.65373    0.10346  6.31843  0.00000 1.92271
## discharge_disposition_id3   0.53712    0.05475  9.80969  0.00000 1.71107
## discharge_disposition_id5   0.98571    0.12006  8.21010  0.00000 2.67971
## discharge_disposition_id6   0.19264    0.06046  3.18616  0.00144 1.21244
## discharge_disposition_id13 -1.35684    0.51536 -2.63278  0.00847 0.25747
## discharge_disposition_id15  2.13117    0.49797  4.27970  0.00002 8.42469
## discharge_disposition_id18  0.45836    0.13381  3.42552  0.00061 1.58148
## discharge_disposition_id22  1.10786    0.09374 11.81886  0.00000 3.02787
## discharge_disposition_id28  2.19409    0.54938  3.99372  0.00007 8.97181
## admission_source_id5       -0.51447    0.18319 -2.80843  0.00498 0.59782
## number_inpatient            0.26425    0.01112 23.75764  0.00000 1.30246
## number_diagnoses            0.05402    0.01090  4.95445  0.00000 1.05550
## metforminNo                -0.61803    0.21183 -2.91762  0.00353 0.53900
## metforminSteady            -0.61853    0.21292 -2.90503  0.00367 0.53874
## insulinNo                  -0.31603    0.07621 -4.14660  0.00003 0.72904
```

```
## insulinSteady               -0.17402    0.06617 -2.62991  0.00854 0.84028
```

In the above output, I take a very numeric approach to interpreting the data. I cannot interpret the estimates & standard errors in isolation. What's most important is the z-value. Whenever I see a high absolute z-value, the correspoding probability is lower, as compared to when I see a low absolute z-value. This is because, the higher the z-value, the further away the value is from the reference variable. The further away from the reference, the lower the probability, because there's a lower chance of obtaining the estimate I did simply by chance. If I use the standard alpha of 0.05, then only the rows with a probability of less than 0.05 are considered significant.

Then, for the rows with a signficant p-value, I can interpret the odds column as follows:

1) Numerical variable: Let's take number_inpatient as the example. The probability is 0.00000, so it is definitely significant. The odds are 1.3, which means that for every 1 unit increase in number_inpatient, the odds of a patient being readmitted increase by a multiplicative factor of 1.3. The odds were obtained by calculating the following: e^(0.26425). In other words, to obtain the odds, I took the natural log of the coefficient associated with this variable.

2) Categorical variable: Let's take discharge_disposition_id13 as the example. Although not clear from the output, I can simply refer back to the original data set & the original categories for each column to infer what the reference variable is. In this case, the dropped reference was discharge_disposition_id1, which means the patient was discharged to home. The odds of a patient who was discharged to a hospice being readmitted to the hospital in less than 30 days is 1.71, as compared to a patient who was discharged home. This makes sense - a patient who is discharged to a hospice is already in a formal patient care setting, and therefore is already relatively unhealthy, and more likely to be readmitted to a hospital.

One interesting point: when the coefficient estimate is positive, the odds are greater than 1. When the coefficient estimate is negative, the odds are less than 1.

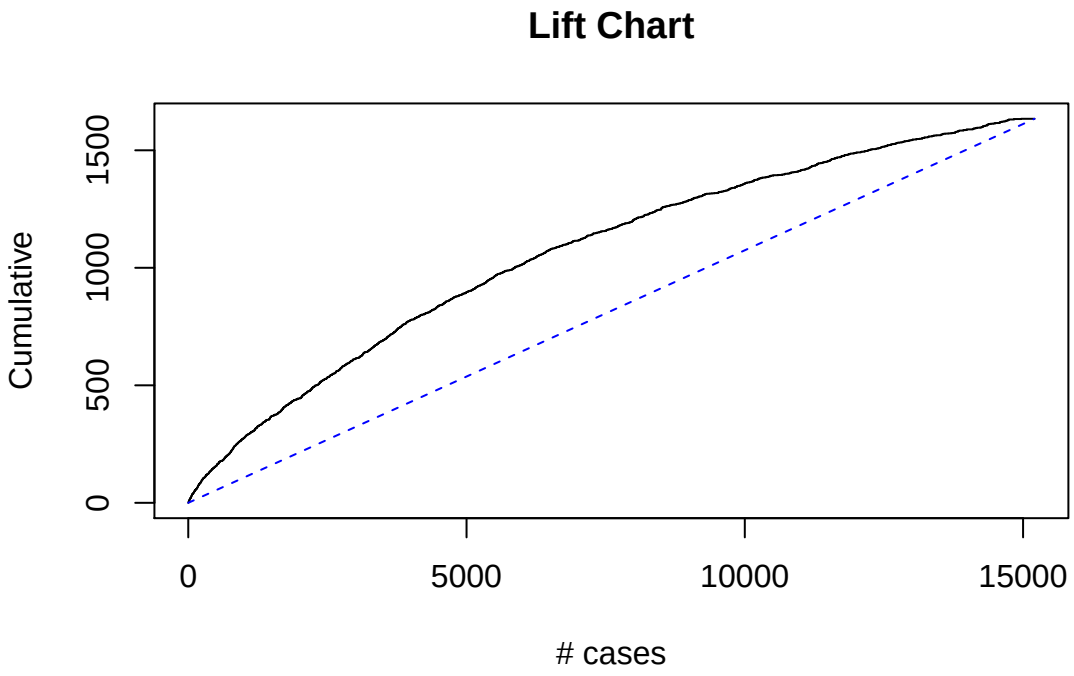
To evaluate model accuracy, I created the confusion matrix below.

```
  #predicting on test data set
log.predict.test <- predict(logit.reg, newdata = test.df.log,
                            type = "response")

  # Confusion Matrix - test data
log_predict1 <- ifelse(log.predict.test>0.5,1,0)

log_predict1 <- as.factor(log_predict1)
isReadmit <- as.factor(test.df.log$isReadmit)

confusionMatrix(log_predict1, isReadmit, positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
```

```
## Prediction     0     1
##          0 13530  1595
##          1    42    39
##
##                Accuracy : 0.8923
##                  95% CI : (0.8873, 0.8972)
##     No Information Rate : 0.8925
##     P-Value [Acc > NIR] : 0.5379
##
##                   Kappa : 0.0357
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.023868
##             Specificity : 0.996905
##          Pos Pred Value : 0.481481
##          Neg Pred Value : 0.894545
##              Prevalence : 0.107458
##          Detection Rate : 0.002565
##    Detection Prevalence : 0.005327
##       Balanced Accuracy : 0.510387
##
##        'Positive' Class : 1
##
```

While the accuracy of this model is high at 89.23%, the sensitivity is only 2.4%, which renders the model useless for predicting whether or not a patient ends up readmitted. Still, below is a (misleading) lift chart.




Let's run a second iteration of the logistic regression model. This time, let's use fewer predictor variables, but also create a couple of interaction variables. The interaction variables allow further refinement of the effect each of the individual variables has on the readmission probability. I created interaction variables for:

1) race and gender - it is often hypothesized in the literature that females belonging to a minority race are at increased economic and health disadvantages.

2) number_inpatient and number_emergency - it makes intuitive sense that the greater the number of inpatient visits a patient has had, the less likely they are to have an emergency hospital visit. If they have a high number of inpatient visits, they likely receive all the care they need during those visits, making an emergency visit less of a possibility.

```
options(max.print=1000)
logit.reg2 <- glm(isReadmit ~ age + admission_type_id + time_in_hospital +
                    num_medications + number_emergency + number_diagnoses +
                    (race * gender) + (number_inpatient * number_emergency) +
                    (time_in_hospital * number_emergency),
                  data = train.df.log, family = "binomial")
options(scipen = 999)
summary(logit.reg2)
```

```
##
## Call:
## glm(formula = isReadmit ~ age + admission_type_id + time_in_hospital +
```

```
##     num_medications + number_emergency + number_diagnoses + (race *
##     gender) + (number_inpatient * number_emergency) + (time_in_hospital *
##     number_emergency), family = "binomial", data = train.df.log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5036  -0.4843  -0.4311  -0.3829   2.6877
##
## Coefficients:
##                                    Estimate Std. Error z value
## (Intercept)                       -3.766635   0.265155 -14.205
## age20-39                           0.799384   0.269743   2.964
## age40-59                           0.692834   0.262882   2.636
## age60-79                           0.851126   0.262123   3.247
## age80-99                           0.913434   0.263994   3.460
## admission_type_id2                -0.043467   0.044795  -0.970
## admission_type_id3                -0.029925   0.046783  -0.640
## admission_type_id4                -7.022308 119.468454  -0.059
## admission_type_id5                 0.049819   0.098300   0.507
## admission_type_id6                 0.045743   0.069114   0.662
## admission_type_id8                -0.158594   0.270569  -0.586
## time_in_hospital                   0.031381   0.006376   4.921
## num_medications                    0.001890   0.002438   0.775
## number_emergency                   0.112894   0.025531   4.422
## number_diagnoses                   0.058756   0.010166   5.780
## raceAsian                         -0.071629   0.286905  -0.250
## raceCaucasian                     -0.013152   0.056513  -0.233
## raceHispanic                      -0.196163   0.181785  -1.079
## raceOther                          0.105384   0.204976   0.514
## genderMale                        -0.005174   0.078099  -0.066
## number_inpatient                   0.295542   0.011202  26.384
## raceAsian:genderMale               0.387810   0.377445   1.027
## raceCaucasian:genderMale           0.014271   0.087943   0.162
## raceHispanic:genderMale            0.562361   0.244876   2.297
## raceOther:genderMale              -0.266098   0.300073  -0.887
## number_emergency:number_inpatient -0.015286   0.003745  -4.081
## time_in_hospital:number_emergency -0.009326   0.004857  -1.920
##                                              Pr(>|z|)
## (Intercept)                      < 0.0000000000000002 ***
## age20-39                                      0.00304 **
## age40-59                                      0.00840 **
## age60-79                                      0.00117 **
## age80-99                                      0.00054 ***
## admission_type_id2                            0.33187
## admission_type_id3                            0.52240
## admission_type_id4                            0.95313
## admission_type_id5                            0.61229
```

```
## admission_type_id6                              0.50807
## admission_type_id8                              0.55777
## time_in_hospital                          0.00000085912 ***
## num_medications                                 0.43806
## number_emergency                          0.00000978502 ***
## number_diagnoses                          0.00000000749 ***
## raceAsian                                       0.80285
## raceCaucasian                                   0.81598
## raceHispanic                                    0.28055
## raceOther                                       0.60716
## genderMale                                      0.94718
## number_inpatient                    < 0.0000000000000002 ***
## raceAsian:genderMale                            0.30420
## raceCaucasian:genderMale                        0.87109
## raceHispanic:genderMale                         0.02165 *
## raceOther:genderMale                            0.37520
## number_emergency:number_inpatient         0.00004474828 ***
## time_in_hospital:number_emergency               0.05487 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 24330  on 35482  degrees of freedom
## Residual deviance: 23351  on 35456  degrees of freedom
## AIC: 23405
##
## Number of Fisher Scoring iterations: 9
```

```
logreg.odds2 <- round(data.frame(summary(logit.reg2)$coefficients, odds = exp(coef(logit.r
  #only selecting the rows that are significant, based on an alpha of 0.01
logreg.odds2 <- logreg.odds2[logreg.odds2$Pr...z.. <= 0.01,]
logreg.odds2
```

```
##                                  Estimate Std..Error   z.value Pr...z..
## (Intercept)                      -3.76664    0.26516 -14.20540  0.00000
## age20-39                          0.79938    0.26974   2.96350  0.00304
## age40-59                          0.69283    0.26288   2.63553  0.00840
## age60-79                          0.85113    0.26212   3.24705  0.00117
## age80-99                          0.91343    0.26399   3.46006  0.00054
## time_in_hospital                  0.03138    0.00638   4.92143  0.00000
## number_emergency                  0.11289    0.02553   4.42187  0.00001
## number_diagnoses                  0.05876    0.01017   5.77966  0.00000
## number_inpatient                  0.29554    0.01120  26.38403  0.00000
## number_emergency:number_inpatient -0.01529    0.00375  -4.08149  0.00004
##                                      odds
## (Intercept)                       0.02313
```

```
## age20-39                              2.22417
## age40-59                              1.99937
## age60-79                              2.34228
## age80-99                              2.49287
## time_in_hospital                      1.03188
## number_emergency                      1.11951
## number_diagnoses                      1.06052
## number_inpatient                      1.34386
## number_emergency:number_inpatient 0.98483
```

Confusion matrix of test data set:

```
  #predicting on test data set
log.predict.test2 <- predict(logit.reg2, newdata = test.df.log, type = "response")

  # Confusion Matrix - test data
log_predict2 <- ifelse(log.predict.test2>0.5,1,0)

log_predict2 <- as.factor(log_predict2)
isReadmit2 <- as.factor(test.df.log$isReadmit)

confusionMatrix(log_predict2, isReadmit2, positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 13546  1607
##          1    26    27
##
##                Accuracy : 0.8926
##                  95% CI : (0.8876, 0.8975)
##     No Information Rate : 0.8925
##     P-Value [Acc > NIR] : 0.4961
##
##                   Kappa : 0.0254
##
##  Mcnemar's Test P-Value : <0.0000000000000002
##
##             Sensitivity : 0.016524
##             Specificity : 0.998084
##          Pos Pred Value : 0.509434
##          Neg Pred Value : 0.893948
##              Prevalence : 0.107458
##          Detection Rate : 0.001776
##    Detection Prevalence : 0.003485
##       Balanced Accuracy : 0.507304
```

```
##
##        'Positive' Class : 1
##
```

Unfortunately, we see a decrease in the sensitivity down to 1.65% when using fewer predictor variables. Hence, logistic regression is definitely not a candidate for the final model.

***Model 5: Decision Tree***

The last model I will implement is a classification tree, which is a simple, but powerful algorithm. One of the major benefits of this model is that it can handle missing data, so there is no need to omit NA's. However, one of the disadvantages of this model is that it is easy to overfit it to the training data. Thus, I will keep a special eye on the training vs. test accuracies.

All iterations of the decision tree will include all predictor variables. But what differs across the iterations is the maximum depth. I will compare the test model accuracies for the varying maximum depths.

It should also be noted that cp (complexity parameter) tells the model: if the next best split in a decision tree does not reduce the tree's complexity by the given amount, then the split should not be pursued. I set this parameter to 0.001. I also set the minsplit parameter to 10, which means that there must be a minimum of 20 observations in a node for a split to be attempted. I let xval = 5, which means I would like for there to be a 5-fold cross-validation in the data set. Lastly, I set maxdepth to 4, which means that a node can only have a maximum depth of 5 splits (since the root node is counted as depth 0).

```
train.ct1 <- rpart(isReadmit ~ ., data = train.df.tree, method = "class",
                   minsplit = 10, cp = 0.001, maxdepth = 4, xval = 5)
test.ct.pred1 <- predict(train.ct1, test.df.tree, type = "class")
confusionMatrix(test.ct.pred1, as.factor(test.df.tree$isReadmit), positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1553 1095
##          1 1445 2721
##
##                Accuracy : 0.6272
##                  95% CI : (0.6156, 0.6387)
##     No Information Rate : 0.56
##     P-Value [Acc > NIR] : < 0.00000000000000022
##
##                   Kappa : 0.234
##
##  Mcnemar's Test P-Value : 0.000000000004365
##
##             Sensitivity : 0.7131
##             Specificity : 0.5180
```

```
##          Pos Pred Value : 0.6531
##          Neg Pred Value : 0.5865
##              Prevalence : 0.5600
##          Detection Rate : 0.3993
##    Detection Prevalence : 0.6114
##       Balanced Accuracy : 0.6155
##
##        'Positive' Class : 1
##
```

```
printcp(train.ct1)
```

```
##
## Classification tree:
## rpart(formula = isReadmit ~ ., data = train.df.tree, method = "class",
##     minsplit = 10, cp = 0.001, maxdepth = 4, xval = 5)
##
## Variables actually used in tree construction:
## [1] discharge_disposition_id medical_specialty
## [3] number_inpatient
##
## Root node error: 7063/15900 = 0.44421
##
## n= 15900
##
##         CP nsplit rel error  xerror      xstd
## 1 0.057766      0   1.00000 1.00000 0.0088707
## 2 0.023078      2   0.88447 0.88744 0.0087244
## 3 0.005097      3   0.86139 0.86819 0.0086899
## 4 0.003398      4   0.85629 0.86479 0.0086836
## 5 0.001345      5   0.85290 0.86366 0.0086814
## 6 0.001000      7   0.85021 0.86295 0.0086801
```

In this model, the accuracy is 62.72%, and the sensitivity is 71.31%. This is not too much better than the previous models. In addition, if we look at the output of cross-validation errors, we see that the 'xerror' is fairly high even at nsplit = 7. When we multiply the 'xerror' value of 0.86295 by the root node error of 0.44421, we obtain (1 - accuracy rate), which is the misclassification rate. So, the lower 'xerror' is the better. Let's run another iteration of this model, this time with a much greater maxdepth of 20.

```
train.ct2 <- rpart(isReadmit ~ ., data = train.df.tree, method = "class",
                   minsplit = 50, cp = 0.001, maxdepth = 20, xval = 5)
test.ct.pred2 <- predict(train.ct2, test.df.tree, type = "class")
confusionMatrix(test.ct.pred2, as.factor(test.df.tree$isReadmit), positive = "1")
```

```
## Confusion Matrix and Statistics
```

```
##
##           Reference
## Prediction    0    1
##          0 1393  943
##          1 1605 2873
##
##                Accuracy : 0.6261
##                  95% CI : (0.6145, 0.6376)
##     No Information Rate : 0.56
##     P-Value [Acc > NIR] : < 0.00000000000000022
##
##                   Kappa : 0.2228
##
##  Mcnemar's Test P-Value : < 0.00000000000000022
##
##             Sensitivity : 0.7529
##             Specificity : 0.4646
##          Pos Pred Value : 0.6416
##          Neg Pred Value : 0.5963
##              Prevalence : 0.5600
##          Detection Rate : 0.4216
##    Detection Prevalence : 0.6572
##       Balanced Accuracy : 0.6088
##
##        'Positive' Class : 1
##
```

```
printcp(train.ct2)
```

```
##
## Classification tree:
## rpart(formula = isReadmit ~ ., data = train.df.tree, method = "class",
##     minsplit = 50, cp = 0.001, maxdepth = 20, xval = 5)
##
## Variables actually used in tree construction:
##  [1] admission_source_id      admission_type_id
##  [3] age                      discharge_disposition_id
##  [5] glipizide                glyburide
##  [7] insulin                  medical_specialty
##  [9] metformin                num_lab_procedures
## [11] num_medications          num_procedures
## [13] number_diagnoses         number_inpatient
## [15] patient_nbr              race
## [17] time_in_hospital
##
## Root node error: 7063/15900 = 0.44421
##
```

```
## n= 15900
##
##           CP nsplit rel error  xerror      xstd
## 1  0.0577658      0   1.00000 1.00000 0.0088707
## 2  0.0230780      2   0.88447 0.88617 0.0087222
## 3  0.0050970      3   0.86139 0.86989 0.0086931
## 4  0.0038227      4   0.85629 0.86762 0.0086889
## 5  0.0033980      5   0.85247 0.86946 0.0086923
## 6  0.0021709      6   0.84907 0.86380 0.0086817
## 7  0.0020294      9   0.84256 0.86224 0.0086787
## 8  0.0019822     14   0.83208 0.86125 0.0086768
## 9  0.0012742     16   0.82812 0.86281 0.0086798
## 10 0.0012389     17   0.82684 0.87187 0.0086967
## 11 0.0012035     37   0.79343 0.87258 0.0086980
## 12 0.0011327     45   0.78310 0.87187 0.0086967
## 13 0.0010383     48   0.77970 0.87158 0.0086962
## 14 0.0010000     51   0.77658 0.87895 0.0087095
```

Although the accuracy rate decreased slightly to 62.61%, it was more than made up for by the fact that the sensitivity rate increased to a healthy 75.29%. Since for this purpose, I am most interested in predicting which patients are most likely to be readmitted rather than the other way around, I am ok with a lower-than-average specificity, which for this model was 46.9%.

**Part F: Conclusion**

***Please see supplement for summary of all models, their accuracies, sensitivities and specificities.***

In sum, I used five different algorithms to see which model would produce the best accuracy and sensitivity in terms of identifying patients who ended up being readmitted to the hospital in less than 30 days. I found that the second iteration of the decision tree, with a maxdepth of 20, gives the best sensitivity. Since for this purpose, I am most interested in predicting which patients are most likely to be readmitted, I pick the model with the highest sensitivity. Hospital administrators can use decision tree algorithms to help them see which patient characteristics are likely to put a patient in the 'readmit' category. The hospital can then implement policies and procedures to minimize the proportion of readmitted patients.

**Sources**

Beata Strack, Jonathan P. DeShazo, Chris Gennings, Juan L. Olmo, Sebastian Ventura, Krzysztof J. Cios, and John N. Clore, "Impact of HbA1c Measurement on Hospital Readmission Rates: Analysis of 70,000 Clinical Database Patient Records," BioMed Research International, vol. 2014, Article ID 781670, 11 pages, 2014.

Cms.gov. (2019). Guide to Reducing Disparities in Readmissions. [online] Available at: https://www.cms.gov/About-CMS/Agency-Information/OMH/Downloads/OMH_Readmissions_Guide.pdf.

Rubin, D. J. (2015). Hospital Readmission of Patients with Diabetes. Current Diabetes Reports, 15(4). doi: 10.1007/s11892-015-0584-7

Shmueli, G., Bruce, P. C., Yahav, I., Patel, N. R., & Lichtendahl, K. C. (2018). Data Mining for Business Analytics: Concepts, Techniques, and Applications in R. Hoboken, NJ: John Wiley & Sons, Inc.

Sinha-Gregory, N., Seley, J., Kochhar, S., DeJesus, J., Lubanksy, S., Nikolova, M., Wei, E., Gerber, L., Mauer, E., Hastu, R., Galla, N. and Greene, R. (2015). Decreasing 30-day Readmission Rates in High-Risk Diabetes Patients Using a Transitional Care Program from Inpatient to Outpatient. [online] Nyspfp.org. Available at: https://www.nyspfp.org/Materials/2017_05_Prev_DM_Readm.pdf.