**Name** : Aditya Sumbaraju  
**Date** : 02/26/21  
**Title** : Predicting Hospital Readmission of Diabetic patients

**Section 2:**

* **Data importing and cleaning steps are explained in the text and in the Github exercises. (Tell me why you are doing the data cleaning activities that you perform). Follow a logical process.**

Data importing and cleaning steps

filename <- 'C:/BU/DSC520/assignment\_repo/dsc520/data/final\_project/diabetic\_data.csv'  
diabetic\_data <- read.table(filename, sep = ",", header = T, na.strings = "?")  
#head(data)  
dim(diabetic\_data)

## [1] 101766 50

We don't need many columns; hence, we should be selecting only relevant values to make it simpler. Also, we need to remove those columns which mainly have NA values.

diabetic\_data$admission\_type\_id <- as.factor(diabetic\_data$admission\_type\_id)  
diabetic\_data$discharge\_disposition\_id <- as.factor(diabetic\_data$discharge\_disposition\_id)  
diabetic\_data$admission\_source\_id <- as.factor(diabetic\_data$admission\_source\_id)

Cleaning Missing Values Below is the logic to replace insufficient data "?" and "unknown/invalid values" with NA. I am passing respective columns from dataset "diabetic\_data" using ncol and identifying the wrong data by passing the columns in the loop and thereby replacing the data with NA.

count <- 0  
for(i in 1:ncol(diabetic\_data))  
 {  
 if(is.factor(diabetic\_data[,i]))  
 {  
 for(j in 1:nrow(diabetic\_data))  
 {  
 if(diabetic\_data[j,i]== "?" | diabetic\_data[j,i]== "Unknown/Invalid" )  
 {  
 count <- count + 1  
 diabetic\_data[j,i] <- NA #replace "?" and "Unknown/Invalid" values with NA  
 }  
 }  
 if(count > 0)  
 {  
 print(c(colnames(diabetic\_data)[i],count))  
 }  
 }  
 count <- 0  
 }

As the data is huge, it leads to performance issue; hence created new file represents clean data.

write.csv(diabetic\_data, file = "diabetic\_data\_clean.csv")

Below are the variables removed: Payer code, weight and Medical Specialty are not included since they have many missing values.

Minor variability variables are excluded. That includes the following variables acetohexamide, glimepiride.pioglitazone, metformin.rosiglitazone, metformin.pioglitazone, chlorpropamide, acetohexamide, tolbutamide, acarbose, miglitor, troglitazone, tolazamide, examide, citoglipton, glyburide.metformin, glipizide.metformin, and glimepiride.pioglitazone.

diabetic\_data\_clean <- read.csv("./diabetic\_data\_clean.csv")  
diabetic\_data\_clean$X <- NULL  
diabetic\_data\_clean$weight <- NULL  
diabetic\_data\_clean$payer\_code <- NULL  
#diabetic\_data\_clean$age <- NULL  
#diabetic\_data\_clean$admission\_type\_id <- NULL  
#diabetic\_data\_clean$discharge\_disposition\_id <- NULL  
#diabetic\_data\_clean$admission\_source\_id <- NULL   
diabetic\_data\_clean$medical\_specialty <- NULL  
diabetic\_data\_clean$encounter\_id <- NULL  
#diabetic\_data\_clean$diag\_1 <- NULL  
#diabetic\_data\_clean$diag\_2 <- NULL  
#diabetic\_data\_clean$diag\_3 <- NULL  
diabetic\_data\_clean$examide <- NULL  
diabetic\_data\_clean$citoglipton <- NULL  
diabetic\_data\_clean$acetohexamide <- NULL  
diabetic\_data\_clean$repaglinide <- NULL  
diabetic\_data\_clean$nateglinide <- NULL  
diabetic\_data\_clean$metformin.pioglitazone <- NULL   
diabetic\_data\_clean$metformin.rosiglitazone <- NULL  
diabetic\_data\_clean$chlorpropamide <- NULL  
diabetic\_data\_clean$acetohexamide <- NULL  
diabetic\_data\_clean$miglitol <- NULL   
diabetic\_data\_clean$tolbutamide <- NULL  
diabetic\_data\_clean$acarbose <- NULL  
diabetic\_data\_clean$miglitor <- NULL  
diabetic\_data\_clean$troglitazone <- NULL  
diabetic\_data\_clean$tolazamide <- NULL  
diabetic\_data\_clean$examide <- NULL  
diabetic\_data\_clean$citoglipton <- NULL  
diabetic\_data\_clean$glyburide.metformin <- NULL  
diabetic\_data\_clean$glipizide.metformin <- NULL  
diabetic\_data\_clean$glimepiride.pioglitazone <- NULL  
diabetic\_data\_clean <- na.omit(diabetic\_data\_clean)  
write.csv(diabetic\_data, file = "diabetic\_data\_clean1.csv")  
dim(diabetic\_data\_clean)

## [1] 98053 30

* **With a clean dataset, show what the final data set looks like. However, do not print off a data frame with 200+ rows; show me the data in the most condensed form possible.**

Below is the Summary & Structure of the data;

str(diabetic\_data\_clean)

## 'data.frame': 98053 obs. of 30 variables:  
## $ patient\_nbr : int 55629189 86047875 82442376 42519267 82637451 84259809 114882984 48330783 63555939 89869032 ...  
## $ race : chr "Caucasian" "AfricanAmerican" "Caucasian" "Caucasian" ...  
## $ gender : chr "Female" "Female" "Male" "Male" ...  
## $ age : chr "[10-20)" "[20-30)" "[30-40)" "[40-50)" ...  
## $ admission\_type\_id : int 1 1 1 1 2 3 1 2 3 1 ...  
## $ discharge\_disposition\_id: int 1 1 1 1 1 1 1 1 3 1 ...  
## $ admission\_source\_id : int 7 7 7 7 2 2 7 4 4 7 ...  
## $ time\_in\_hospital : int 3 2 2 1 3 4 5 13 12 9 ...  
## $ num\_lab\_procedures : int 59 11 44 51 31 70 73 68 33 47 ...  
## $ num\_procedures : int 0 5 1 0 6 1 0 2 3 2 ...  
## $ num\_medications : int 18 13 16 8 16 21 12 28 18 17 ...  
## $ number\_outpatient : int 0 2 0 0 0 0 0 0 0 0 ...  
## $ number\_emergency : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ number\_inpatient : int 0 1 0 0 0 0 0 0 0 0 ...  
## $ diag\_1 : chr "276" "648" "8" "197" ...  
## $ diag\_2 : chr "250.01" "250" "250.43" "157" ...  
## $ diag\_3 : chr "255" "V27" "403" "250" ...  
## $ number\_diagnoses : int 9 6 7 5 9 7 8 8 8 9 ...  
## $ max\_glu\_serum : chr "None" "None" "None" "None" ...  
## $ A1Cresult : chr "None" "None" "None" "None" ...  
## $ metformin : chr "No" "No" "No" "No" ...  
## $ glimepiride : chr "No" "No" "No" "No" ...  
## $ glipizide : chr "No" "Steady" "No" "Steady" ...  
## $ glyburide : chr "No" "No" "No" "No" ...  
## $ pioglitazone : chr "No" "No" "No" "No" ...  
## $ rosiglitazone : chr "No" "No" "No" "No" ...  
## $ insulin : chr "Up" "No" "Up" "Steady" ...  
## $ change : chr "Ch" "No" "Ch" "Ch" ...  
## $ diabetesMed : chr "Yes" "Yes" "Yes" "Yes" ...  
## $ readmitted : chr ">30" "NO" "NO" "NO" ...  
## - attr(\*, "na.action")= 'omit' Named int [1:3713] 1 20 21 22 55 66 67 88 100 112 ...  
## ..- attr(\*, "names")= chr [1:3713] "1" "20" "21" "22" ...

#head (diabetic\_data\_clean)  
# Raw diabetic\_data\_clean:  
dim(diabetic\_data)

## [1] 101766 50

# Clean diabetic\_data\_clean  
dim(diabetic\_data\_clean)

## [1] 98053 30

* **What do you not know how to do right now that you need to learn to import and cleanup your dataset?**

I have removed low latency variables at this point, but I am unsure if this can impact my model in predicting the readmittance rate.

Regarding the performance, I am not sure how to tune the performace in R while cleaning the huge datasets. As an interim solution, I have created a new file that contains clean data.

* **Discuss how you plan to uncover new information in the data that is not self-evident.**

A logistic regression model is considered a better fit for the current dataset. It demonstrates an improvement over a model with fewer predictors. I hope it can be achieved using the Likelihood Ratio Test (LR) and Analysis of Variance (ANOVA). In chapter 8 of "Discovering Statistics Using R", we have learned that this test compares the probability of the data under the full model against the likelihood of the data under a model with fewer predictors. Removing predictor variables from a model will almost always make the model fit less well. Still, it is necessary to test whether the observed difference in model fit is statistically significant. I will also look at all the datasets and find common factors and relevant data that can be combined to answer my research questions.

* **What are different ways you could look at this data to answer the questions you want to answer?**

I would be considering box plots for

1. Length of stay (days) vs. readmitted within <30 days
2. Number of lab procedures vs. readmitted within <30 days
3. Number of non-lab procedures vs. readmitted within <30 days
4. Number of Diagnosis vs. readmitted within <30 days

Most importantly, it will analyze the patients' medical history by checking outpatient visits, inpatient visits, and emergencies.

* **Do you plan to slice and dice the data in different ways, create new variables, or join separate data frames to create new summary information? Explain.**

I plan to split the data w.r.t patient's hospital readmittance rate > 30 days and < 30 days.

I would also be slicing the Patient demographics, Patient medical History, and clinical results and creating a data frame to publish the relationship between medication details and patients' discharge details.

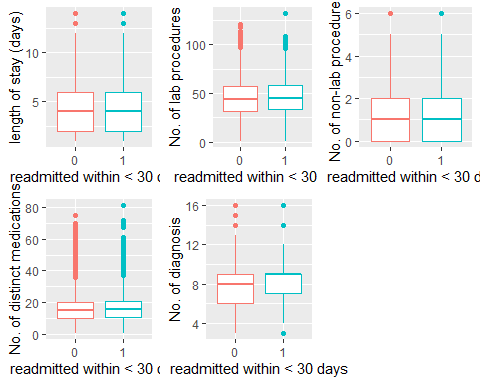
* **How could you summarize your data to answer key questions?**

After establishing the relationship (From an output of the slice and dice process), I would be evaluating the models in terms of accuracy, Sensitivity, precision, and AUC(Area under the curve), and then I would be handy with the variables responsible for the readmittance rate.

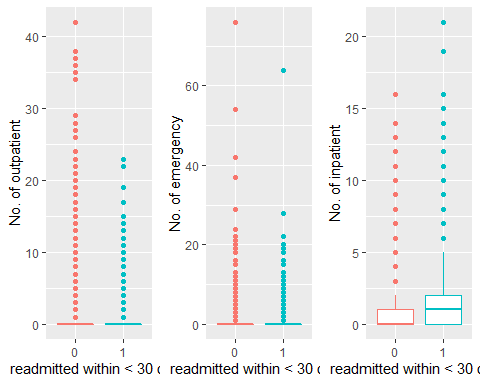
* **What types of plots and tables will help you to illustrate the findings to your questions? Ensure that all graph plots have axis titles, legend if necessary, scales are appropriate, appropriate geoms used, etc.).**

I would be using box plots to summarize a set of readmittance data on an interval of 30days

# Plots  
plot1 <- ggplot(data, aes(x = y, y = hosp, color = y)) + geom\_boxplot() + theme(legend.position = "none") +  
 labs(y = "length of stay (days)", x = "readmitted within < 30 days")   
plot2 <- ggplot(data, aes(x = y, y = labp, color = y)) + geom\_boxplot() + theme(legend.position = "none") +  
 labs(y = "No. of lab procedures", x = "readmitted within < 30 days")   
plot3 <- ggplot(data, aes(x = y, y = proc, color = y)) + geom\_boxplot() + theme(legend.position = "none") +  
 labs(y = "No. of non-lab procedures", x = "readmitted within < 30 days")   
plot4 <- ggplot(data, aes(x = y, y = nmed, color = y)) + geom\_boxplot() + theme(legend.position = "none") +  
 labs(y = "No. of distinct medications", x = "readmitted within < 30 days")   
plot8 <- ggplot(data, aes(x = y, y = diag, color = y)) + geom\_boxplot() + theme(legend.position = "none") +  
 labs(y = "No. of diagnosis", x = "readmitted within < 30 days")   
grid.arrange(plot1, plot2, plot3, plot4, plot8, nrow = 2)



plot5 <- ggplot(data, aes(x = y, y = nout, color = y)) + geom\_boxplot() + theme(legend.position = "none") +  
 labs(y = "No. of outpatient", x = "readmitted within < 30 days")   
plot6 <- ggplot(data, aes(x = y, y = emer, color = y)) + geom\_boxplot() + theme(legend.position = "none") +  
 labs(y = "No. of emergency", x = "readmitted within < 30 days")   
plot7 <- ggplot(data, aes(x = y, y = ninp, color = y)) + geom\_boxplot() + theme(legend.position = "none") +  
 labs(y = "No. of inpatient", x = "readmitted within < 30 days")   
grid.arrange(plot5, plot6, plot7, nrow = 1)



* **What do you not know how to do right now that you need to learn to answer your questions?**

Right now, I am in a gray space to figure out how to create a preliminary model and refine a model using glm and p-value criteria. I am anticipating it could be challenging to examine the model performance with the current dataset.

* **Do you plan on incorporating any machine learning techniques to answer your research questions? Explain.**

Yes, based on the identified parameters, I would like to estimate Positive Predictive Values and Negative Predictive Value, and to process model evaluation, I would be using the Bayes rule.