Practical 1 - Medical Databases

Alexandra Del Favero-Campbell

2024-05-17

This practical is based on exploratory data analysis and prediction of a dataset derived from a municipal database of healthcare administrative data. This dataset is derived from Vitoria, the capital city of Espírito Santo, Brazil (population 1.8 million) and was freely shared under a creative commons license.

Generate an rmarkdown report that contains all the necessary code to document and perform: EDA, prediction of no-shows using XGBoost, and an analysis of variable/feature importance using this data set. Ensure your report includes answers to any questions marked in bold. Please submit your report via brightspace as a link to a git repository containing the rmarkdown and compiled/knitted html version of the notebook.

Introduction

The Brazilian public health system, known as SUS for Unified Health System in its acronym in Portuguese, is one of the largest health system in the world, representing government investment of more than 9% of GDP. However, its operation is not homogeneous and there are distinct perceptions of quality from citizens in different regions of the country. Non-attendance of medical appointments contributes a significant additional burden on limited medical resources. This analysis will try and investigate possible factors behind non-attendance using an administrative database of appointment data from Vitoria, Espírito Santo, Brazil.

The data required is available via the course website.

Understanding the data

1. Use the data dictionary describe each of the variables/features in the CSV in your report.

PatientID: A unique identifier used to identify each patient

AppointmentID: A unique identifier to identify and differentiate each appointment

Gender: Patient's Gender (i.e., Male or Female)

ScheduledDate: Date for which the appointment was scheduled

AppointmentDate: Date of the actual appointment

Age: Patient's age

Neighbourhood: District of Vitória in which the appointment will take place

SocialWelfare: Indicates whether a patient is a recipient of Bolsa Família welfare payments

Hypertension: Indicates whether a patient has been previously diagnosed with hypertension (Boolean)

Diabetes: Indicates whether a patient has been previously diagnosed with diabetes (Boolean)

AlcoholUseDisorder: Indicates whether a patient has been previously diagnosed with alcohol use disorder (Boolean)

Disability: Indicates whether a patient has been previously diagnosed with a disability (severity rated 0-4)

SMSReceived: Indicates whether a patient was given at least 1 reminder via text sent to them before their appointment (Boolean)

NoShow: Indicates whether the patient did not attend their scheduled appointment (Boolean: Yes/No)

2. Can you think of 3 hypotheses for why someone may be more likely to miss a medical appointment?

A. Patients that did not receive at least 1 SMS reminder about their appointment were more likely to miss their medical appointment.

B. Patients who had their appointments scheduled more than 2 days in advance were more likely to miss their medical appointment.

C. Patients that were over the age of 50 years old were more likely to miss their medical appointment.

3. Can you provide 3 examples of important contextual information that is missing in this data dictionary and dataset that could impact your analyses e.g., what type of medical appointment does each `AppointmentID` refer to?

A. The type of medical appointment would be important to further categorize and define. Different appointment types require different amounts of time at the doctor's office, which some people may not have enough time for. The type of medical appointment would also be important to further categorize and define. For example, some appointments may require more prep time to prepare for, which could be why some people also do not show up for their appointments.

B. Whether a patient even has SMS receiving capabilities would also be an important factor to consider because they may not receive a reminder at all, which may affect their remembering to come for their appointment.

C. It may also be important to contextually know which neighborhood each patient lives in. If the patient does not live in the same neighborhood, it may be too far for them to get to their appointment.

Data Parsing and Cleaning

4. Modify the following to make it reproducible i.e., downloads the data file directly from version control

```
#raw.data <- read_csv('C:/Users/ajdel/Desktop/Alex/PhD Stuff/CSCI6410/Practicals/Practical 1/2016_05v2_
raw.data <- readr::read_csv('https://maguire-lab.github.io/health_data_science_research_2024/static_fil
```

Now we need to check data is valid: because we specified col_types and the data parsed without error most of our data seems to at least be formatted as we expect i.e., ages are integers

```
raw.data %>% dplyr::filter(Age > 110)
```

```
## # A tibble: 5 x 14
##   PatientID   AppointmentID Gender ScheduledDate       AppointmentDate       Age
##   <fct>       <fct>         <fct>  <dttm>              <dttm>              <int>
## 1 3196321161~ 5700278       F      2016-05-16 09:17:44 2016-05-19 00:00:00   115
## 2 3196321161~ 5700279       F      2016-05-16 09:17:44 2016-05-19 00:00:00   115
## 3 3196321161~ 5562812       F      2016-04-08 14:29:17 2016-05-16 00:00:00   115
## 4 3196321161~ 5744037       F      2016-05-30 09:44:51 2016-05-30 00:00:00   115
## 5 7482345792~ 5717451       F      2016-05-19 07:57:56 2016-06-03 00:00:00   115
## # i 8 more variables: Neighbourhood <fct>, SocialWelfare <lgl>,
## #   Hypertension <lgl>, Diabetes <lgl>, AlcoholUseDisorder <lgl>,
## #   Disability <fct>, SMSReceived <lgl>, NoShow <fct>
```

We can see there are 2 patient's older than 110 which seems suspicious but we can't actually say if this is impossible.

5. Are there any individuals with impossible ages? If so we can drop this row using `filter` i.e., `data <- data %>% filter(CRITERIA)`

There are people who are above 110 years old and also at least one patient that is -1 years old and several that are 0 years old. To remove impossible ages (i.e., someone aged -1 years old) we will use the filter function to drop that patient.

```
raw.data <- raw.data %>% filter(Age>=0)
```

Exploratory Data Analysis

First, we should get an idea if the data meets our expectations, there are newborns in the data (`Age==0`) and we wouldn't expect any of these to be diagnosed with Diabetes, Alcohol Use Disorder, and Hypertension (although in theory it could be possible). We can easily check this:

```
raw.data %>% filter(Age == 0) %>% select(Hypertension, Diabetes, AlcoholUseDisorder) %>% unique()
```

```
## # A tibble: 1 x 3
##   Hypertension Diabetes AlcoholUseDisorder
##   <lgl>        <lgl>    <lgl>
## 1 FALSE        FALSE    FALSE
```

We can also explore things like how many different neighborhoods are there and how many appoints are from each?

```
count(raw.data, Neighbourhood, sort = TRUE)
```

```
## # A tibble: 81 x 2
##    Neighbourhood         n
##    <fct>             <int>
##  1 JARDIM CAMBURI     7717
##  2 MARIA ORTIZ        5805
##  3 RESISTÊNCIA        4431
##  4 JARDIM DA PENHA    3877
##  5 ITARARÉ            3514
##  6 CENTRO             3334
##  7 TABUAZEIRO         3132
##  8 SANTA MARTHA       3131
##  9 JESUS DE NAZARETH  2853
## 10 BONFIM             2773
## # i 71 more rows
```

6. What is the maximum number of appointments from the same patient?

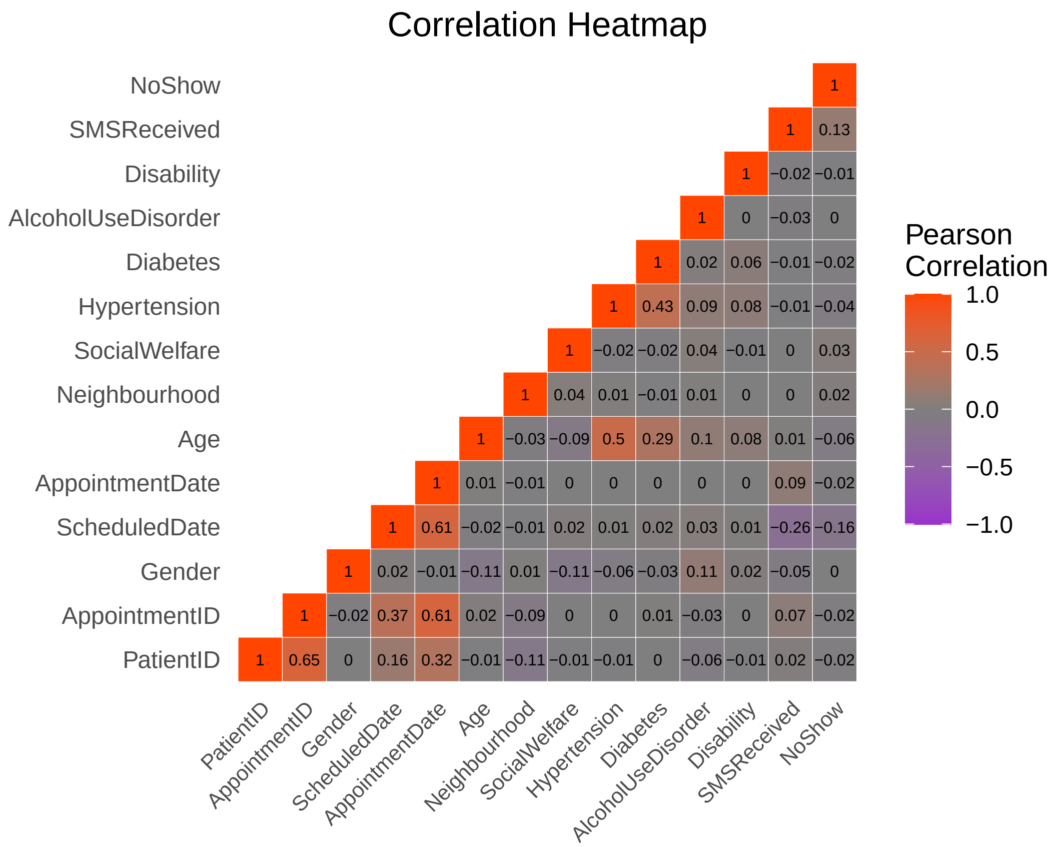
```r
count(raw.data, PatientID, sort = TRUE)
```

```
## # A tibble: 62,298 x 2
##    PatientID           n
##    <fct>           <int>
##  1 822145925426128    88
##  2 99637671331        84
##  3 26886125921145     70
##  4 33534783483176     65
##  5 258424392677       62
##  6 871374938638855    62
##  7 6264198675331      62
##  8 75797461494159     62
##  9 66844879846766     57
## 10 872278549442       55
## # i 62,288 more rows
```

The maximum number of appointments from the same patient is 88 appointments by the patient with PatientID number 822145925426128.

Let's explore the correlation between variables:

```r
# let's define a plotting function
corplot = function(df){

  cor_matrix_raw <- round(cor(df),2)
  cor_matrix <- melt(cor_matrix_raw)


  #Get triangle of the correlation matrix
  #Lower Triangle
  get_lower_tri<-function(cor_matrix_raw){
    cor_matrix_raw[upper.tri(cor_matrix_raw)] <- NA
    return(cor_matrix_raw)
  }

  # Upper Triangle
  get_upper_tri <- function(cor_matrix_raw){
    cor_matrix_raw[lower.tri(cor_matrix_raw)]<- NA
    return(cor_matrix_raw)
  }

  upper_tri <- get_upper_tri(cor_matrix_raw)

  # Melt the correlation matrix
  cor_matrix <- melt(upper_tri, na.rm = TRUE)

  # Heatmap Plot
  cor_graph <- ggplot(data = cor_matrix, aes(Var2, Var1, fill = value))+
    geom_tile(color = "white")+
    scale_fill_gradient2(low = "darkorchid", high = "orangered", mid = "grey50",
                         midpoint = 0, limit = c(-1,1), space = "Lab",
                         name="Pearson\nCorrelation") +
    theme_minimal()+
    theme(axis.text.x = element_text(angle = 45, vjust = 1,
                                     size = 8, hjust = 1))+
    coord_fixed()+ geom_text(aes(Var2, Var1, label = value), color = "black", size = 2) +
    theme(
      axis.title.x = element_blank(),
      axis.title.y = element_blank(),
      panel.grid.major = element_blank(),
      panel.border = element_blank(),
      panel.background = element_blank(),
      axis.ticks = element_blank())+
      ggtitle("Correlation Heatmap")+
      theme(plot.title = element_text(hjust = 0.5))

  cor_graph
}

numeric.data = mutate_all(raw.data, function(x) as.numeric(x))

# Plot Correlation Heatmap
corplot(numeric.data)
```



Correlation heatmaps are useful for identifying linear relationships between variables/features. In this case, we are particularly interested in relationships between `NoShow` and any specific variables.

7. Which parameters most strongly correlate with missing appointments (`NoShow`)?

The parameters that seem to be the most strongly correlated with missing appointments are SMSReceived (i.e., whether a patient received at least one SMS reminder about their appointment) (r=0.13), ScheduledDate (i.e., the date in which the appointment was scheduled) (r=-0.16), and Age (i.e., the age of the patient) (r=-0.06). However, these do not seem like too strong of correlations.

8. Are there any other variables which strongly correlate with one another?

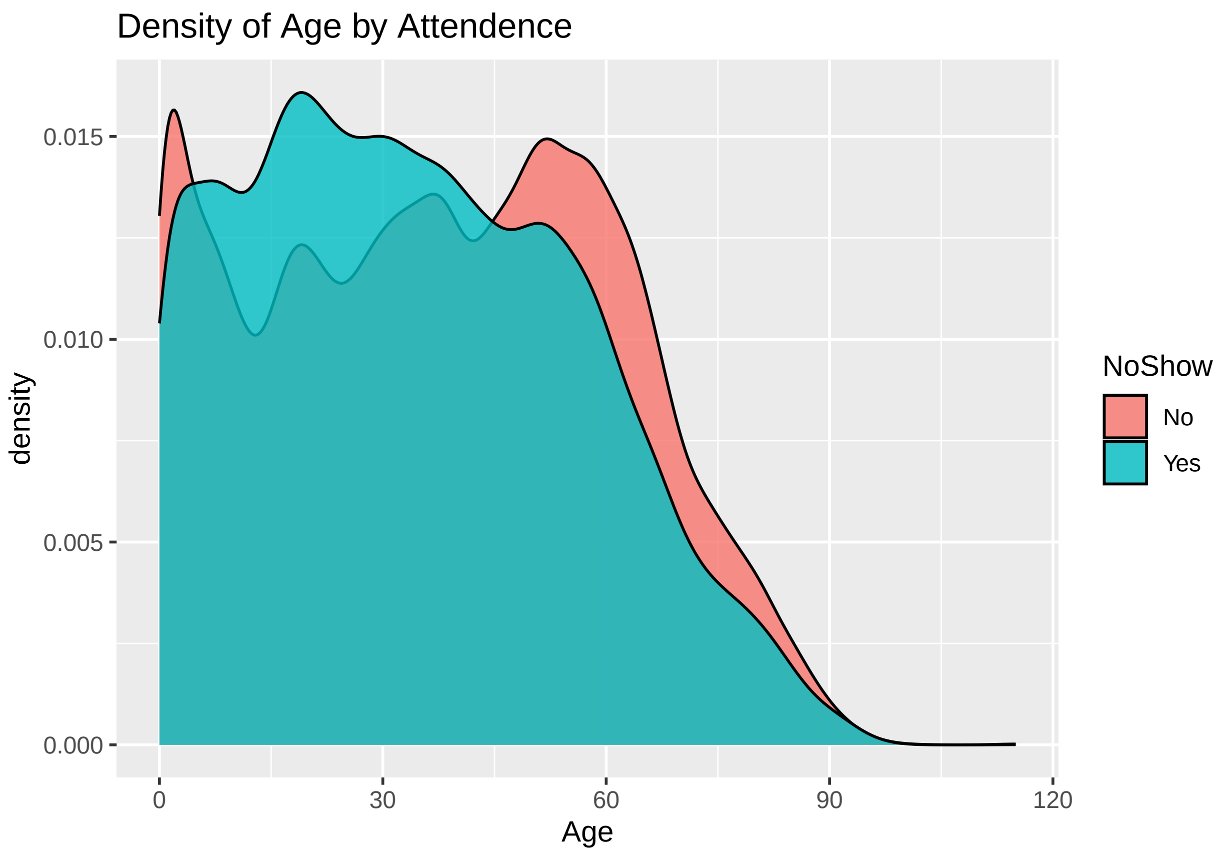
Yes, there are other variables which strongly correlate (i.e., r > +/-0.4) with one another. These include: AppointmentDate and AppointmentID (r=0.61), ScheduledDate and AppointmentDate (r=0.61), AppointmentID and PatientID (r=0.65), Hypertension and Age (r=0.5), and Hypertension and Diabetes (r=0.43).

9. Do you see any issues with PatientID/AppointmentID being included in this plot?

PatientID and AppointmentID are strongly correlated and both are categorized as numbers that uniquely identify a patient and a type of appointment. But this does not really tell us or help us differentiate whether it is one person with the same patientID that has the same type of multiple same appointmentID appointments or if it is different patients that all have the same appointmentID appointments. We would need to further differentiate this to see if the correlation is actually true or not.

Let's look at some individual variables and their relationship with `NoShow`.

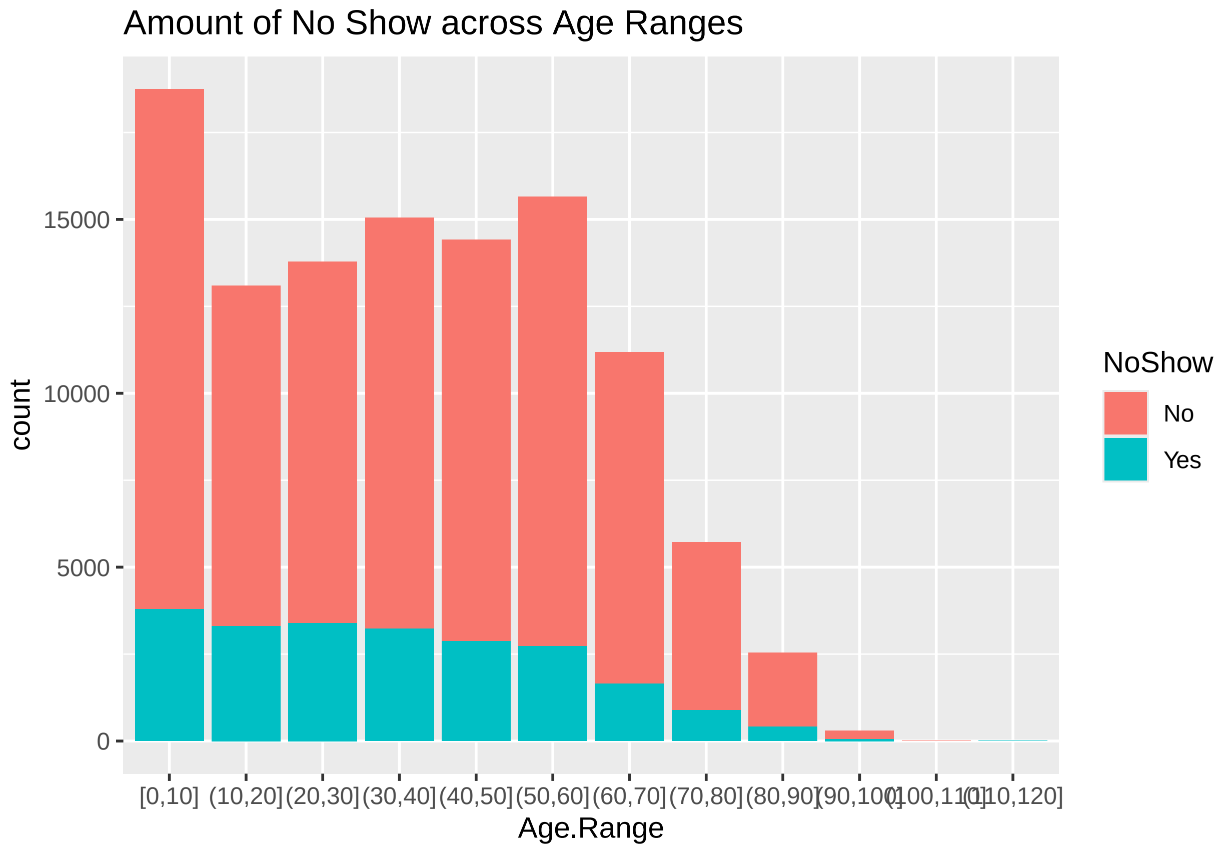
```r
ggplot(raw.data) +
  geom_density(aes(x=Age, fill=NoShow), alpha=0.8) +
  ggtitle("Density of Age by Attendence")
```



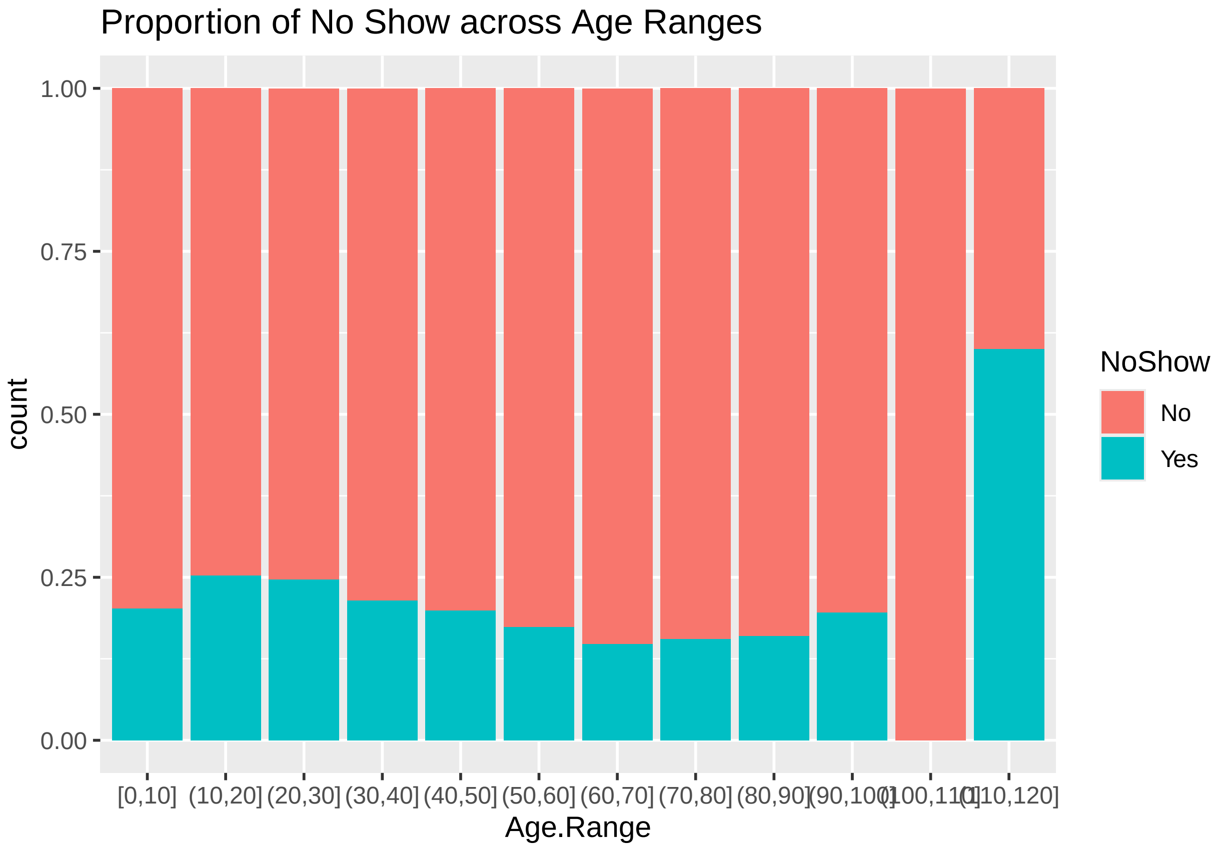
There does seem to be a difference in the distribution of ages of people that miss and don't miss appointments. However, the shape of this distribution means the actual correlation is near 0 in the heatmap above. This highlights the need to look at individual variables.

Let's take a closer look at age by breaking it into categories.

```r
raw.data <- raw.data %>% mutate(Age.Range=cut_interval(Age, length=10))

ggplot(raw.data) +
  geom_bar(aes(x=Age.Range, fill=NoShow)) +
  ggtitle("Amount of No Show across Age Ranges")
```



```r
ggplot(raw.data) +
  geom_bar(aes(x=Age.Range, fill=NoShow), position='fill') +
  ggtitle("Proportion of No Show across Age Ranges")
```



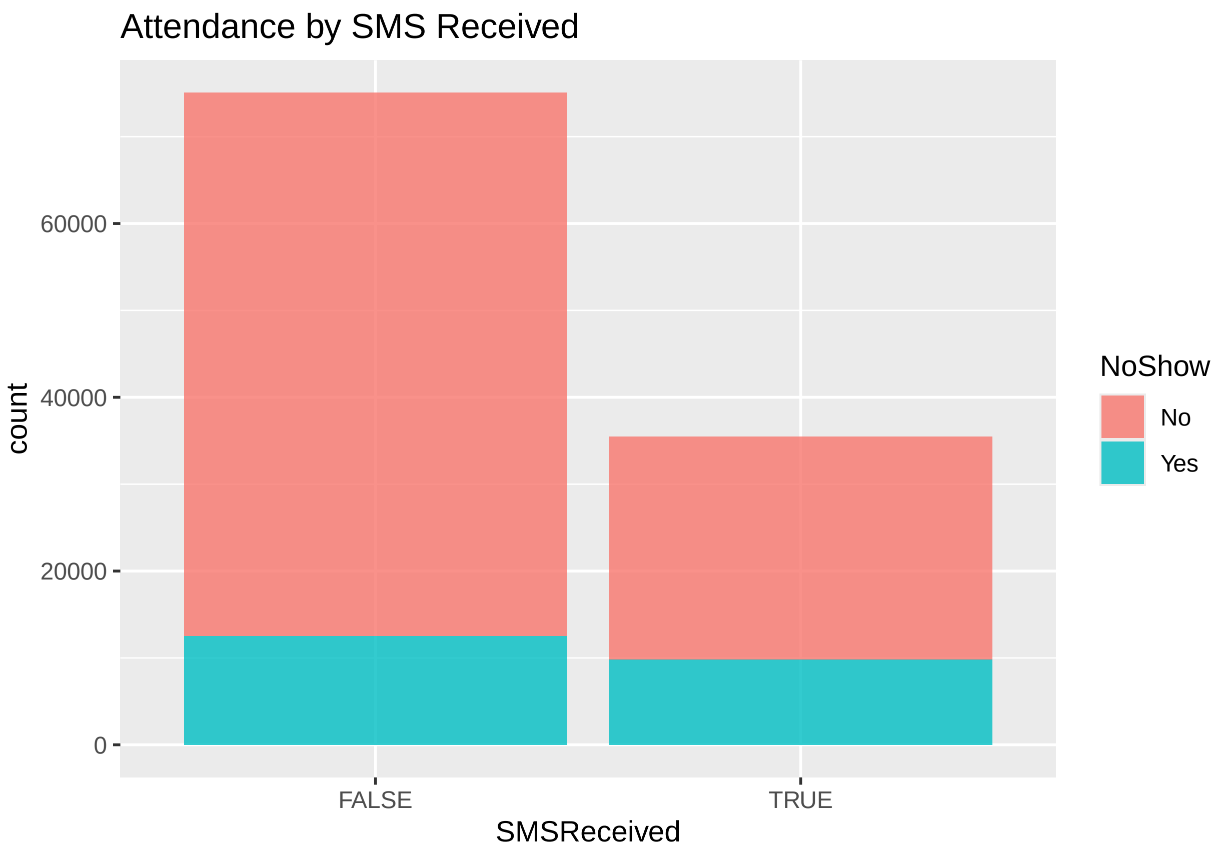
10. How could you be misled if you only plotted 1 of these 2 plots of attendance by age group?

If you only plotted plot 1 of these 2 plots you would, first of all, think that there were no patients at all within the age range categories of 100-110 and 110-120. We know that this is not the case. We see that in plot 2. Additionally, although the first plot gives us some ideas of how many people in different age groups may have appointments (i.e., there are more people who fit into those younger age group categories and these people have more appointments compared to the amount of people who are very old that are involved in this study), it does not really give us an idea of proportion of shows vs. no-shows. In the first plot, it almost looks like there are very few people that come to their appointments at all. We see this is not necessarily the case in the second plot.

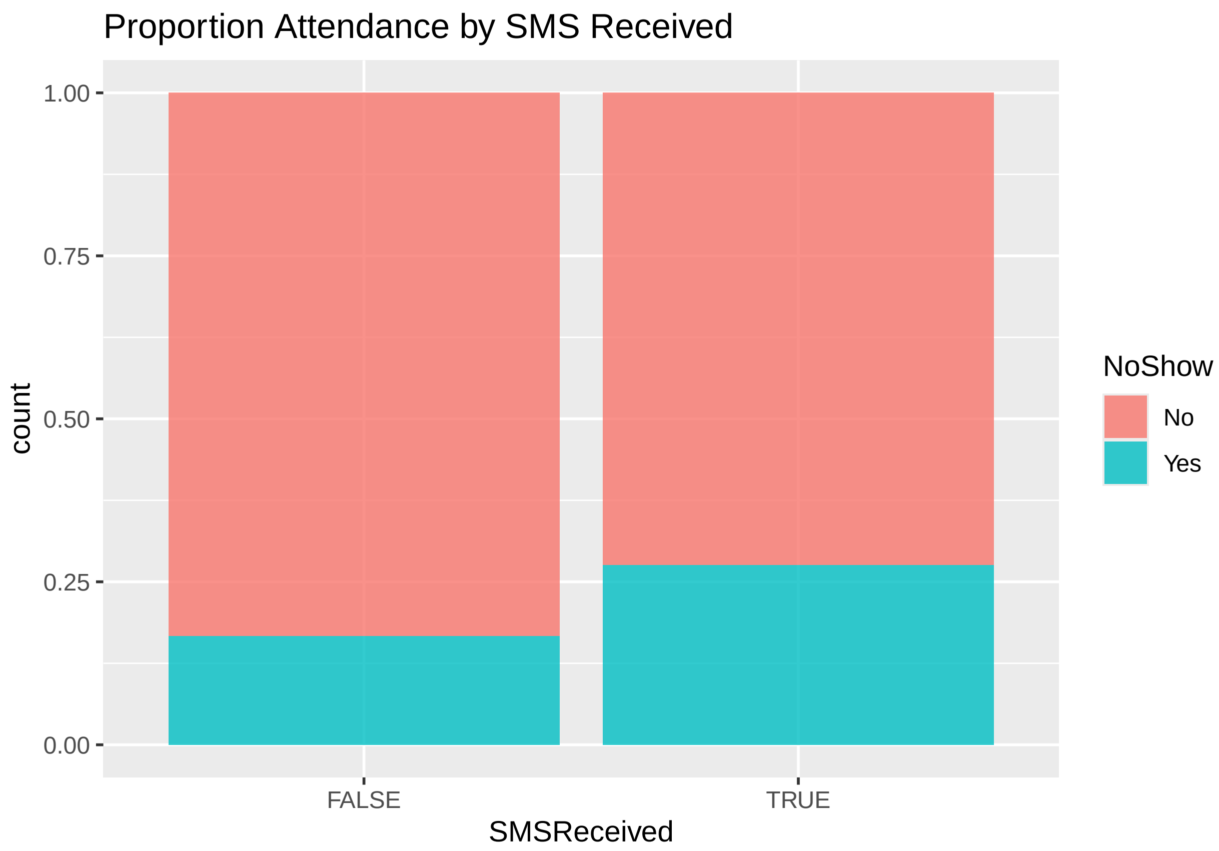
The key takeaway from this is that number of individuals $>$ 90 are very few from plot 1 so probably are very small so unlikely to make much of an impact on the overall distributions. However, other patterns do emerge such as 10-20 age group is nearly twice as likely to miss appointments as the 60-70 years old.

Next, we'll have a look at `SMSReceived` variable:

```
ggplot(raw.data) +
  geom_bar(aes(x=SMSReceived, fill=NoShow), alpha=0.8) +
  ggtitle("Attendance by SMS Received")
```



```r
ggplot(raw.data) +
  geom_bar(aes(x=SMSReceived, fill=NoShow), position='fill', alpha=0.8) +
  ggtitle("Proportion Attendance by SMS Received")
```



11. From this plot does it look like SMS reminders increase or decrease the chance of someone not attending an appointment? Why might the opposite actually be true (hint: think about biases)?

First of all, from the first plot, it seems like less people are receiving SMS reminders and that more people who are not receiving reminders are actually showing up for their appointments. However, when we look at the second plot, we see that, proportionally, the people that did receive an SMS reminder did end up showing up for their appointments. Perhaps the people that did not receive an SMS potentially do not have a mobile number listed or do not have a phone that has SMS capabilities. This could be a bias in the data.

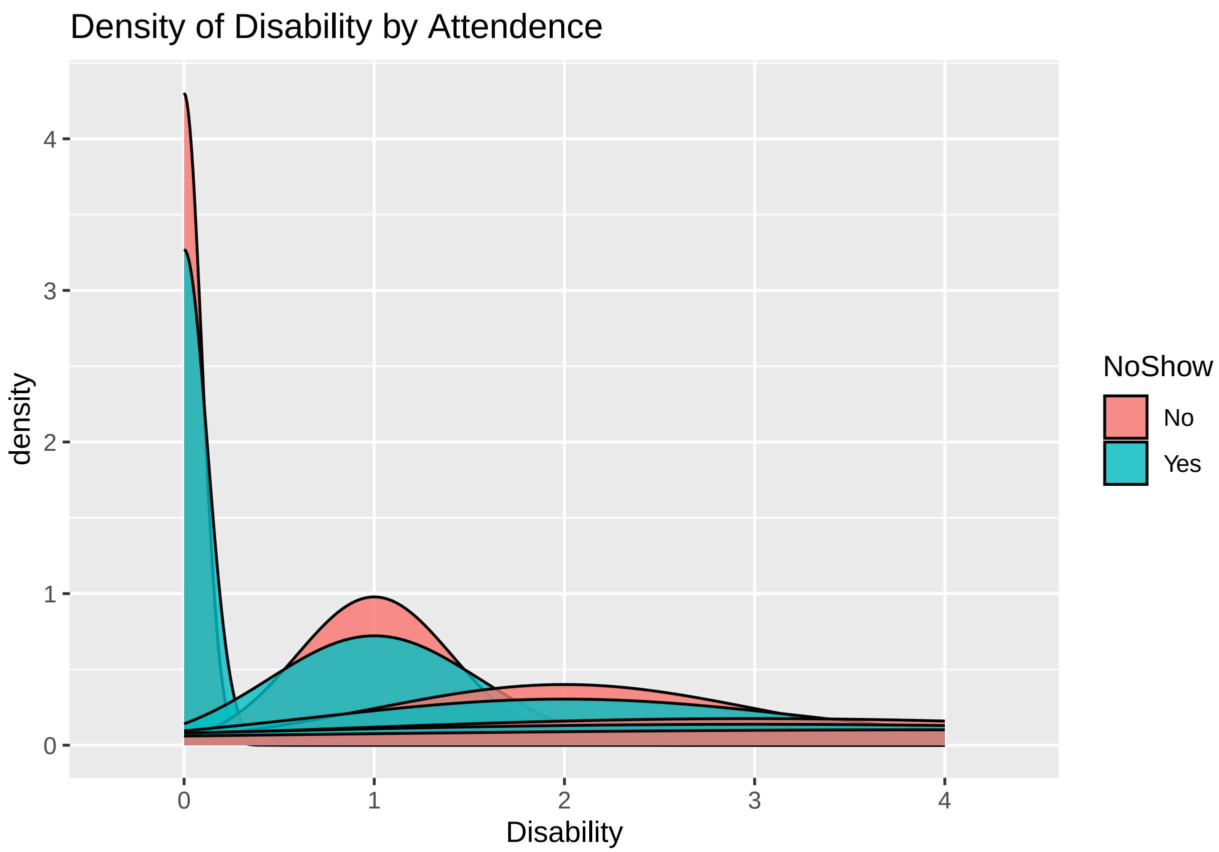
12. Create a similar plot which compares the the density of `NoShow` across the values of disability

```
#Insert plot
```

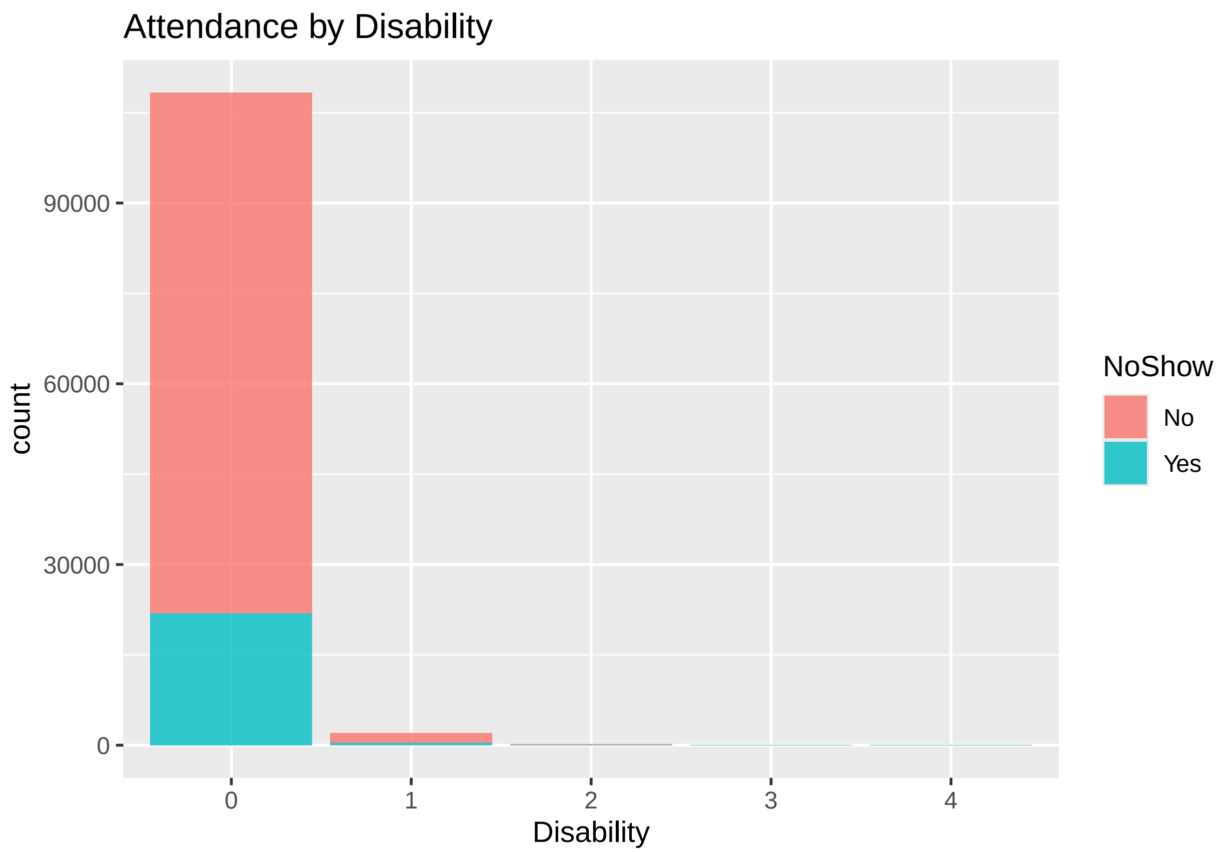
```
ggplot(raw.data) +
  geom_density(aes(x=Disability, fill=NoShow), alpha=0.8) +
  ggtitle("Density of Disability by Attendence")
```

```
## Warning: Groups with fewer than two data points have been dropped.
```

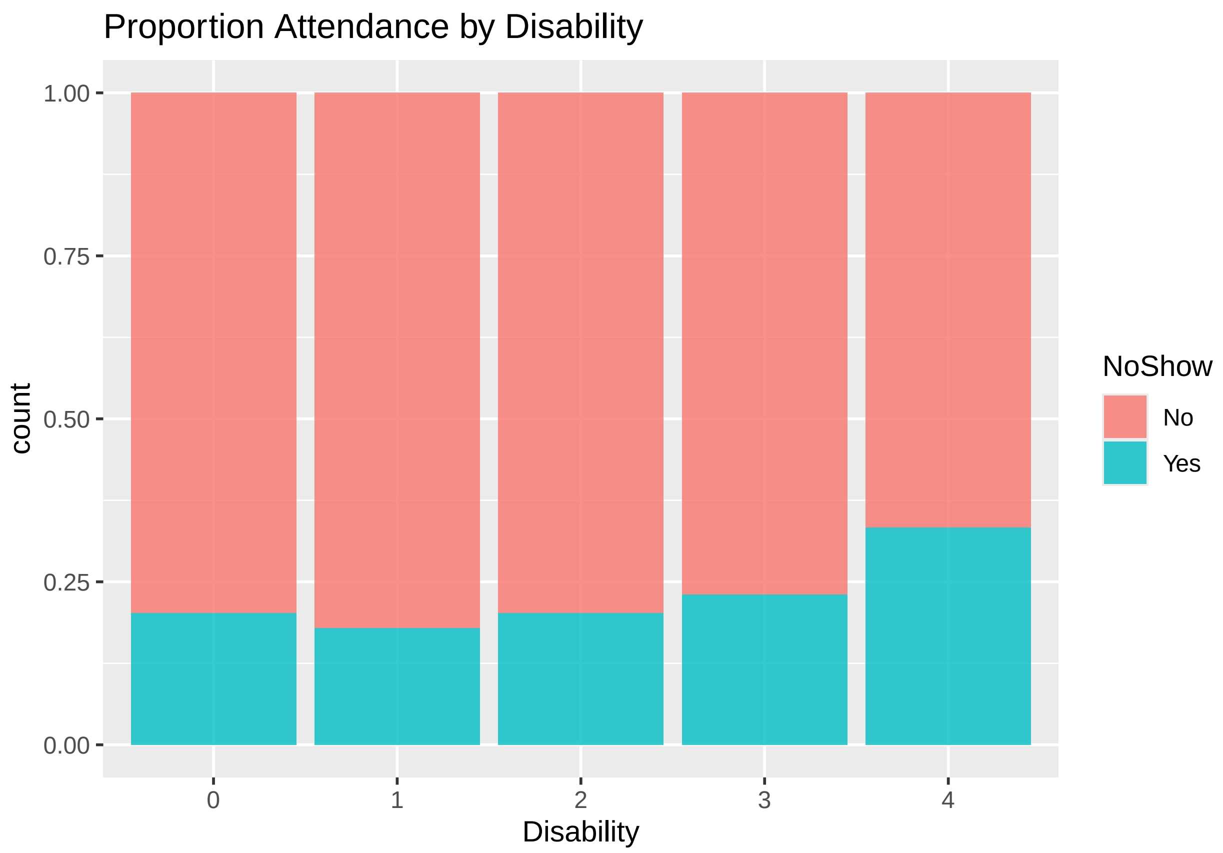
```
## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning
## -Inf
```



```
ggplot(raw.data) +
  geom_bar(aes(x=Disability, fill=NoShow), alpha=0.8) +
  ggtitle("Attendance by Disability")
```

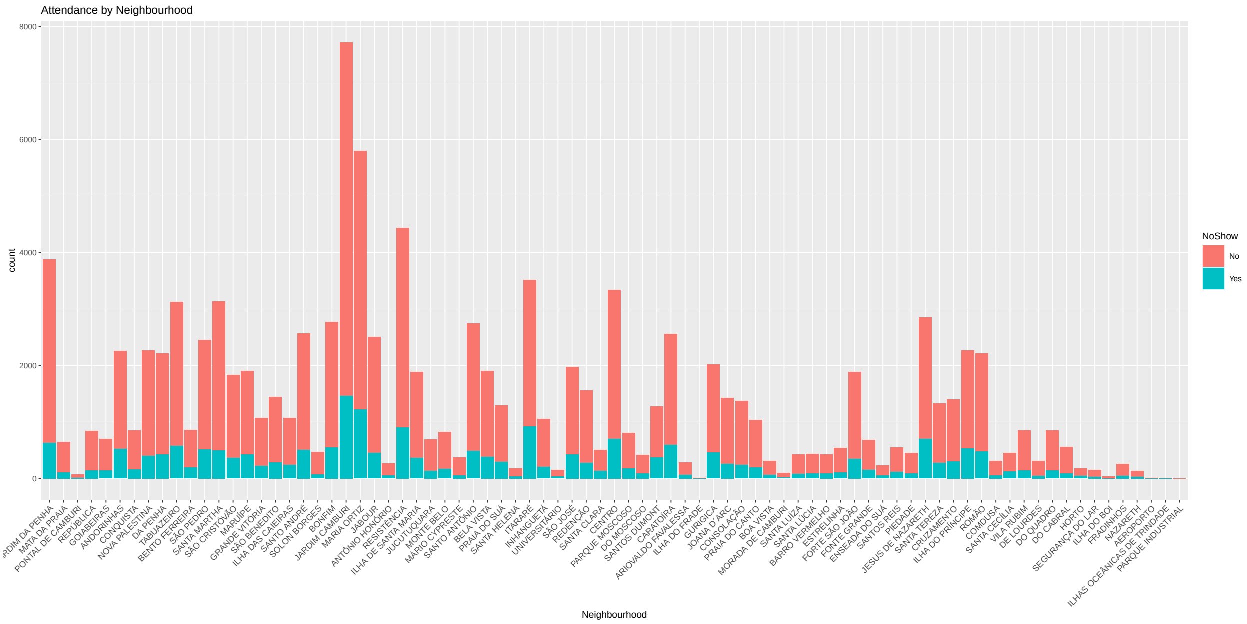


```
ggplot(raw.data) +
  geom_bar(aes(x=Disability, fill=NoShow), position='fill', alpha=0.8) +
  ggtitle("Proportion Attendance by Disability")
```

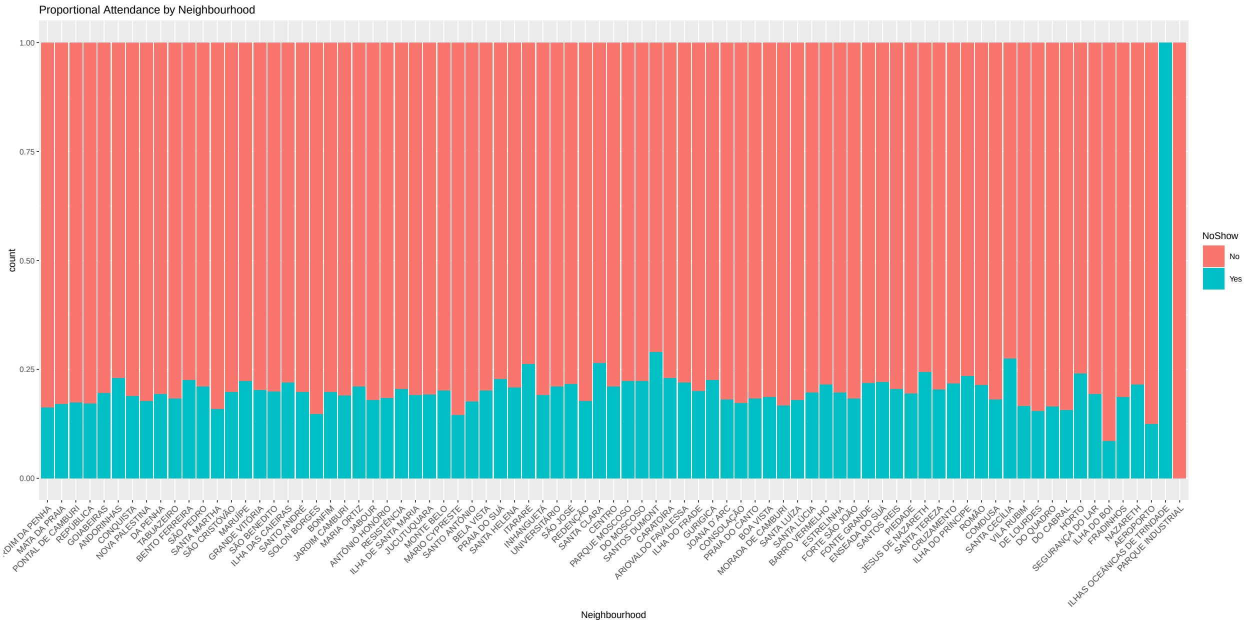


Now let's look at the neighbourhood data as location can correlate highly with many social determinants of health.

```
par(cex.lab=2,cex.axis=2)
ggplot(raw.data) +
  geom_bar(aes(x=Neighbourhood, fill=NoShow)) +
  theme(axis.text.x = element_text(angle=45, hjust=1, size=10)) +
  ggtitle('Attendance by Neighbourhood')
```



```
ggplot(raw.data) +
  geom_bar(aes(x=Neighbourhood, fill=NoShow), position='fill') +
  theme(axis.text.x = element_text(angle=45, hjust=1, size=10)) +
  ggtitle('Proportional Attendance by Neighbourhood')
```



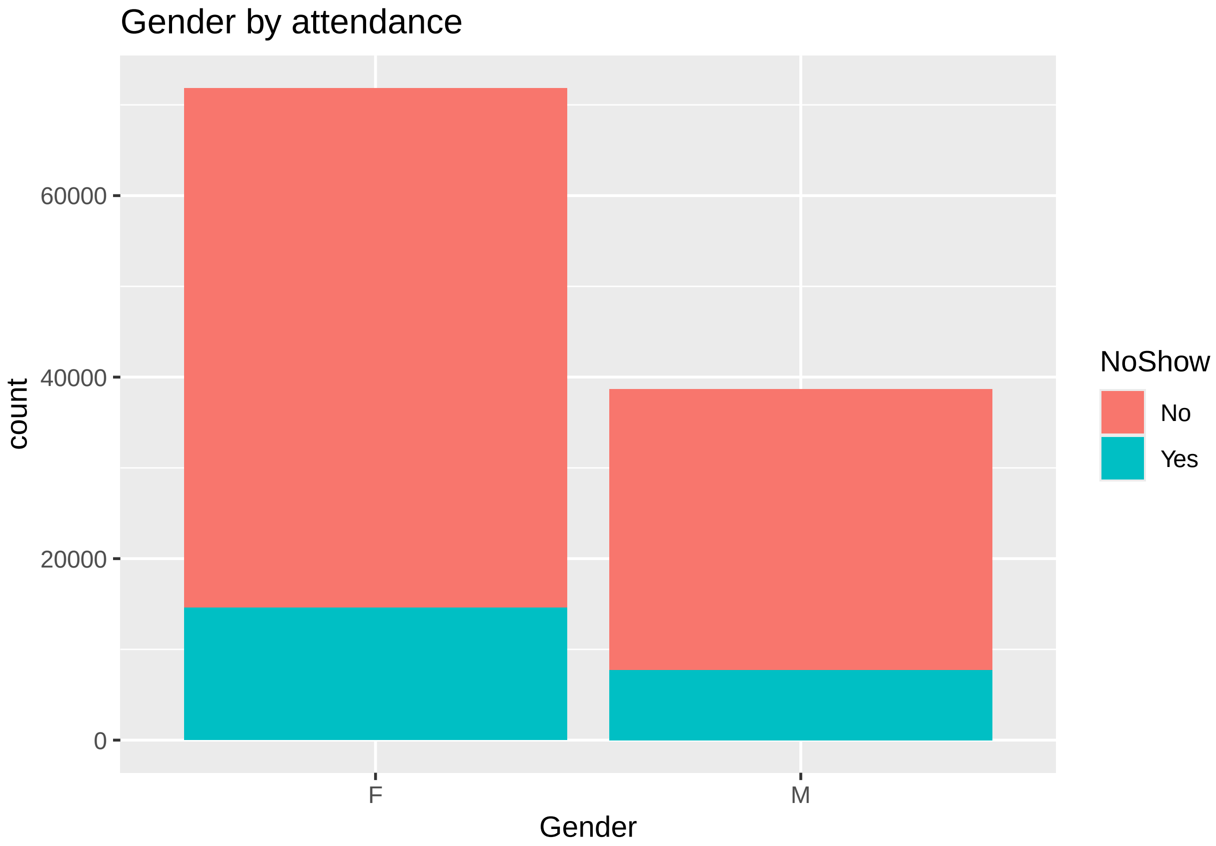
Most neighborhoods have similar proportions of no-show but some have much higher and lower rates.

13. Suggest a reason for differences in attendance rates across neighbourhoods.

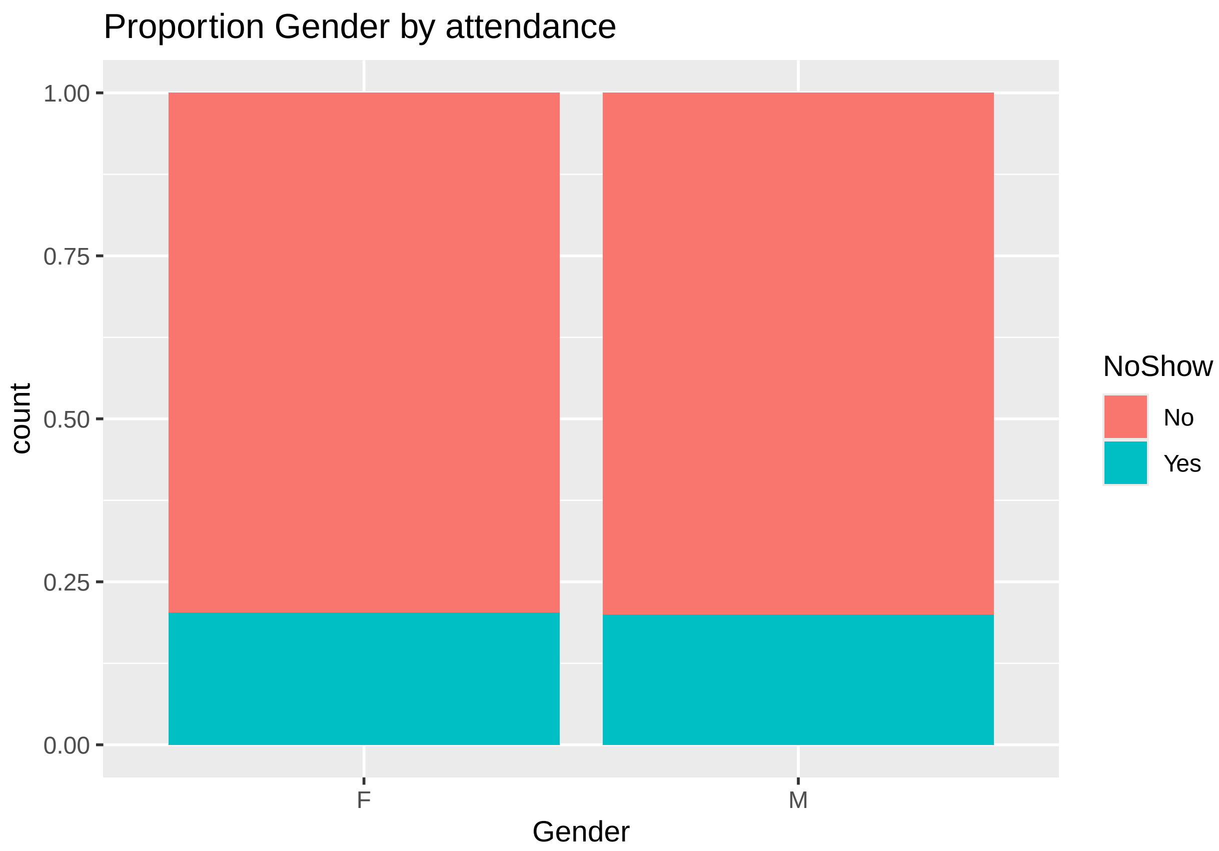
It could be that neighborhoods are different sizes. Perhaps some are very small and the people that live in that neighborhood find it much easier to get to their appointments. It also could be that some places have better transportation available, so it is much easier for patients to make it to their appointments.

Now let's explore the relationship between gender and NoShow.

```
ggplot(raw.data) +
  geom_bar(aes(x=Gender, fill=NoShow))+
  ggtitle("Gender by attendance")
```



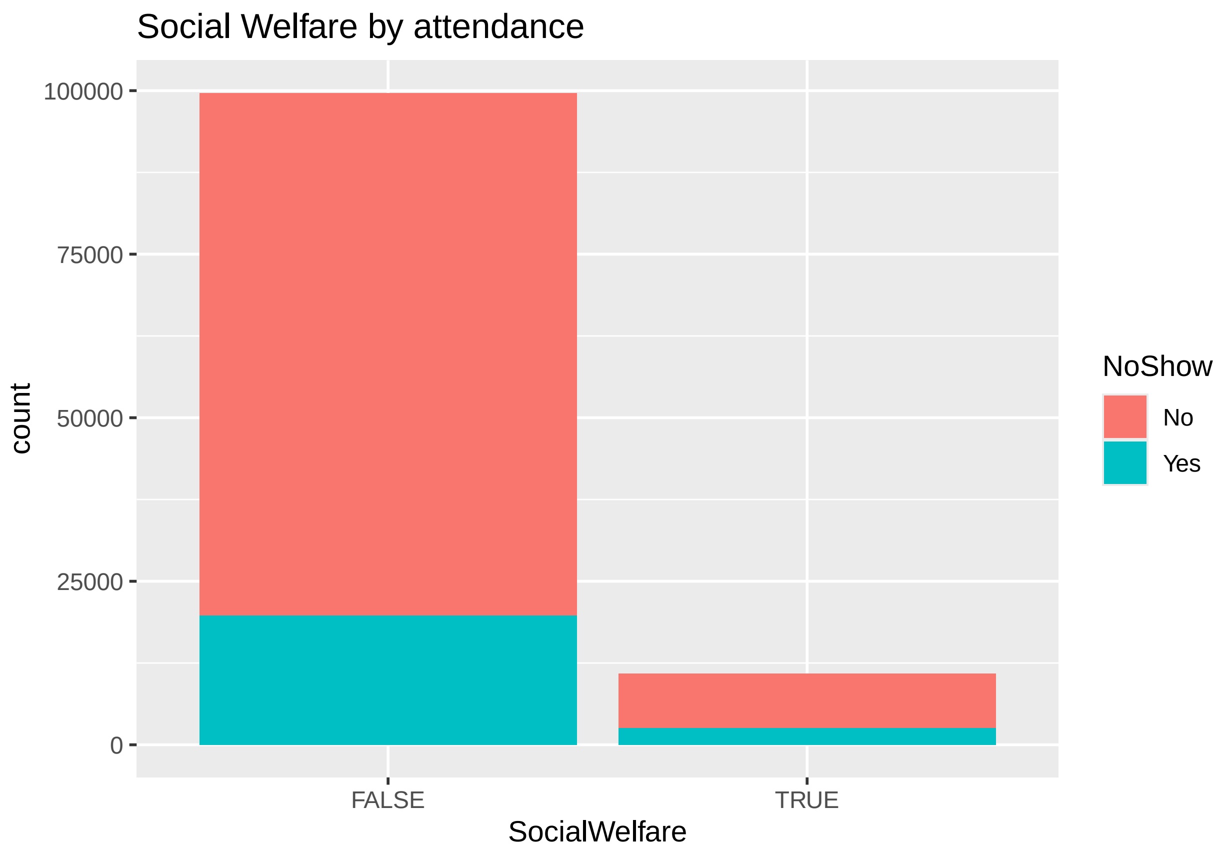
```
ggplot(raw.data) +
  geom_bar(aes(x=Gender, fill=NoShow), position='fill')+
  ggtitle("Proportion Gender by attendance")
```



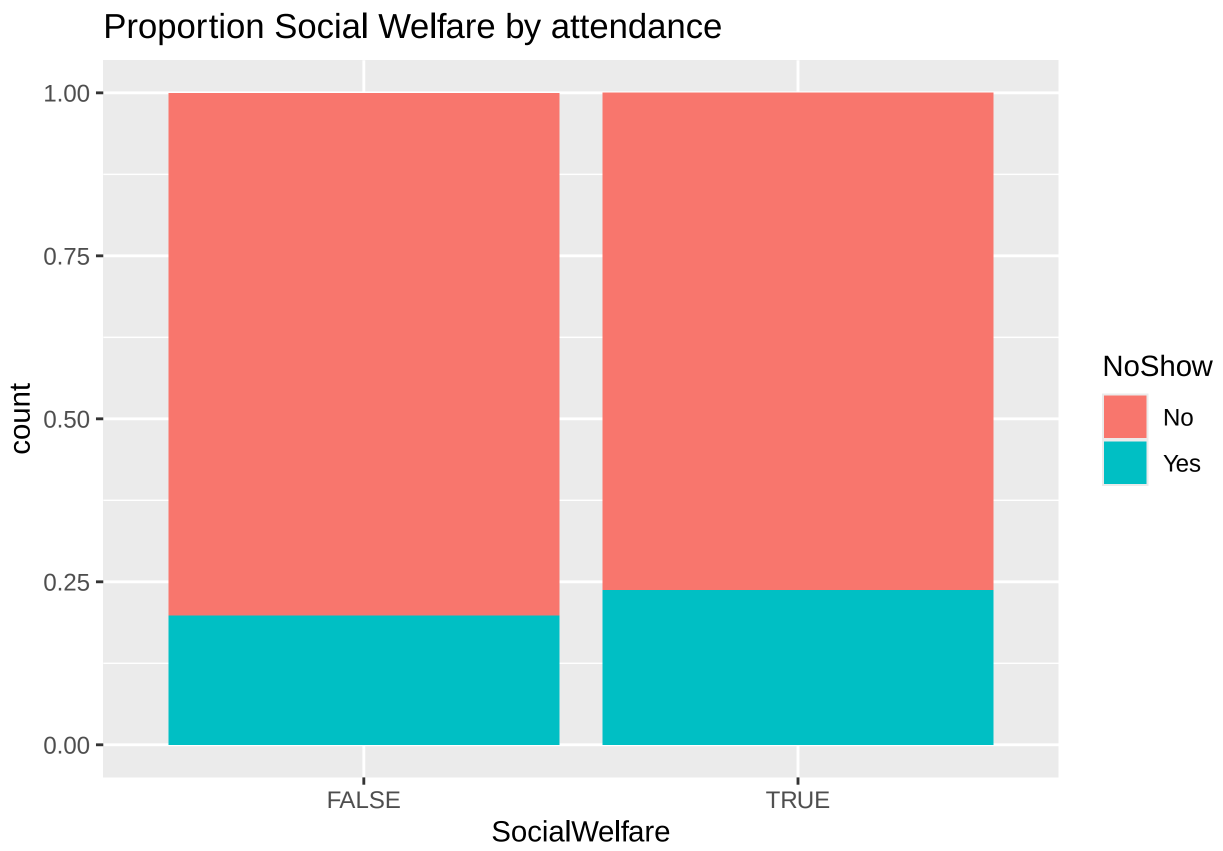
14. Create a similar plot using `SocialWelfare`

```
#Insert plot
```

```
ggplot(raw.data) +
  geom_bar(aes(x=SocialWelfare, fill=NoShow))+
  ggtitle("Social Welfare by attendance")
```



```
ggplot(raw.data) +
  geom_bar(aes(x=SocialWelfare, fill=NoShow), position='fill')+
  ggtitle("Proportion Social Welfare by attendance")
```



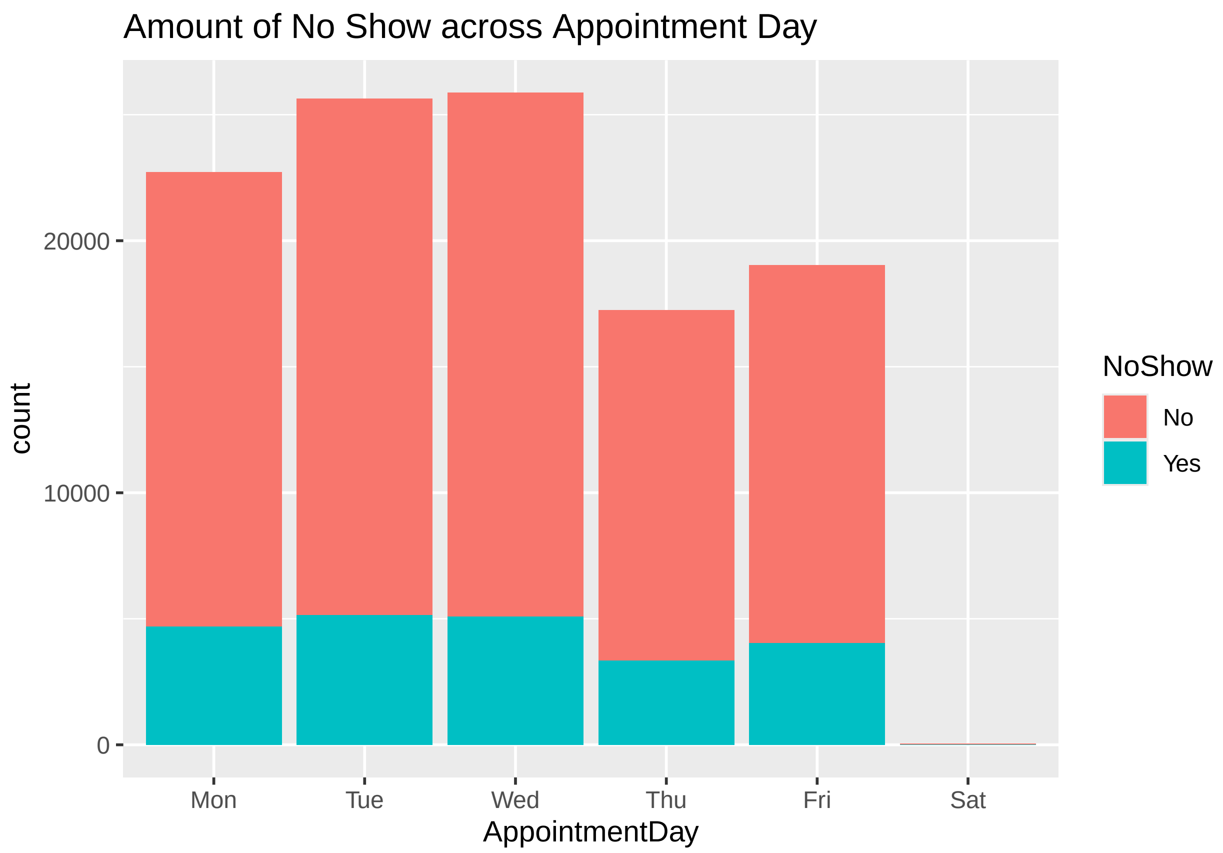
Far more exploration could still be done, including dimensionality reduction approaches but although we have found some patterns there is no major/striking patterns on the data as it currently stands.

However, maybe we can generate some new features/variables that more strongly relate to the `NoShow`.

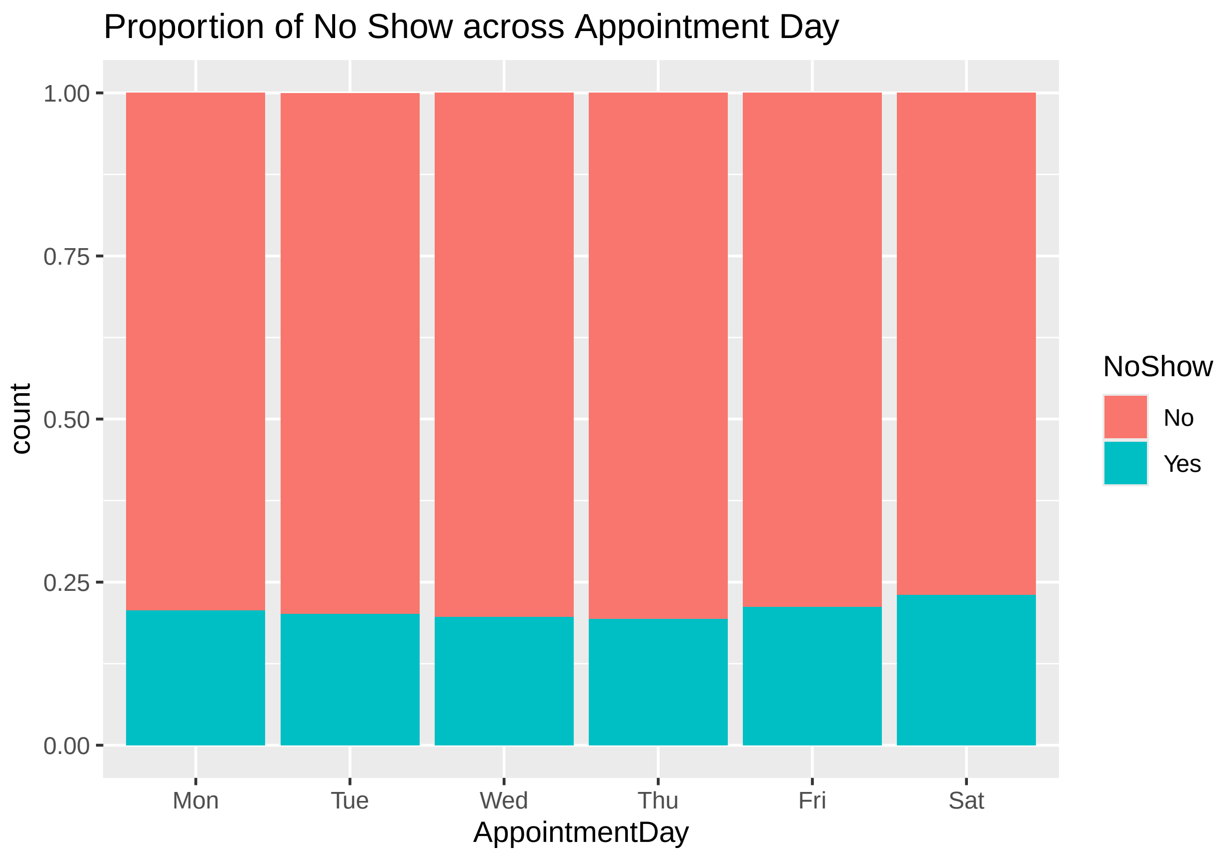
Feature Engineering

Let's begin by seeing if appointments on any day of the week has more no-show's. Fortunately, the `lubridate` library makes this quite easy!

```r
raw.data <- raw.data %>% mutate(AppointmentDay = wday(AppointmentDate, label=TRUE, abbr=TRUE),
                                 ScheduledDay = wday(ScheduledDate,  label=TRUE, abbr=TRUE))

ggplot(raw.data) +
  geom_bar(aes(x=AppointmentDay, fill=NoShow)) +
  ggtitle("Amount of No Show across Appointment Day")
```



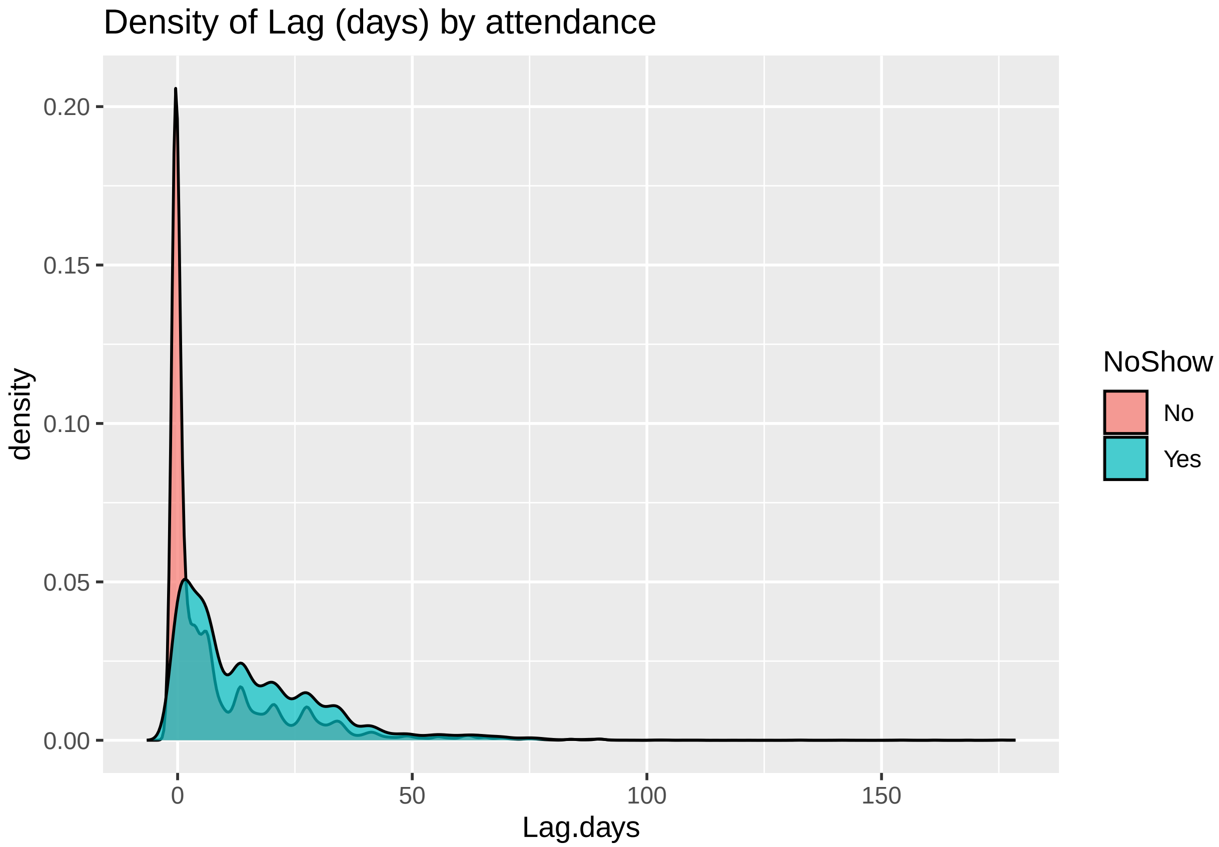
```r
ggplot(raw.data) +
  geom_bar(aes(x=AppointmentDay, fill=NoShow), position = 'fill') +
  ggtitle("Proportion of No Show across Appointment Day")
```



Let's begin by creating a variable called `Lag`, which is the difference between when an appointment was scheduled and the actual appointment.

```
raw.data <- raw.data %>% mutate(Lag.days=difftime(AppointmentDate, ScheduledDate, units = "days"),
                                Lag.hours=difftime(AppointmentDate, ScheduledDate, units = "hours"))

ggplot(raw.data) +
  geom_density(aes(x=Lag.days, fill=NoShow), alpha=0.7)+
  ggtitle("Density of Lag (days) by attendance")
```

```
## Don't know how to automatically pick scale for object of type <difftime>.
## Defaulting to continuous.
```



15. Have a look at the values in lag variable, does anything seem odd?

```
summarize(raw.data,Lag.days)
```

```
## Warning: Returning more (or less) than 1 row per ‘summarise()‘ group was deprecated in
## dplyr 1.1.0.
## i Please use ‘reframe()‘ instead.
## i When switching from ‘summarise()‘ to ‘reframe()‘, remember that ‘reframe()‘
##   always returns an ungrouped data frame and adjust accordingly.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```
## # A tibble: 110,526 x 1
##    Lag.days
##    <drtn>
##  1 -0.7764815 days
##  2 -0.6725347 days
##  3 -0.6799074 days
##  4 -0.7288310 days
##  5 -0.6717940 days
##  6  1.6410764 days
##  7  1.3713889 days
##  8  1.3472454 days
##  9 -0.3349074 days
## 10  1.4663773 days
## # i 110,516 more rows
```

We see that some of the lag day values are negative. This is odd. If we also take a closer look at our raw dataset, we see that some of the appointment dates scheduled are actually set for a time that is even before the time and date that they were actually scheduled. For example, the scheduled date for one patient is 2016-04-29T18:38:08Z, but, within the dataset the appointment date listed is 2016-04-29T00:00:00Z. This does not make a lot of sense. The data likely needs to be cleaned a bit more to fix this issue. This could be a data entry issue. Additionally, there are a lot of no shows when the scheduled date and appointment date are less than or equal to 0.

Predictive Modeling

Let's see how well we can predict NoShow from the data.

We'll start by preparing the data, followed by splitting it into testing and training set, modeling and finally, evaluating our results. For now we will subsample but please run on full dataset for final execution.

```
### REMOVE SUBSAMPLING FOR FINAL MODEL
data.prep <- raw.data %>% select(-AppointmentID, -PatientID) #%>% sample_n(10000) #Removes PatientID an

set.seed(42)
data.split <- initial_split(data.prep, prop = 0.7)
train  <- training(data.split)
test <- testing(data.split)
```

Let's now set the cross validation parameters, and add classProbs so we can use AUC as a metric for xgboost.

```
fit.control <- trainControl(method="cv",number=3,
                            classProbs = TRUE, summaryFunction = twoClassSummary)
```

16. Based on the EDA, how well do you think this is going to work?

Based on what we have seen so far, I think that this could work, but there are some things that may some make challenges for us to predict as well as we may want to. For example, we saw that our correlations between variables are somewhat weak, especially when looking at our "NoShow" variable. We did not see any particular feature that really strongly determined whether someone attended their appointment. Due to weak individual correlations between variables of interest to us, perhaps the sensitivity of the model may not be as strong as we would like. There also may be some imbalances in the dataset. This could make it harder to predict no-shows as accurately as possible.

Now we can train our XGBoost model

```
xgb.grid <- expand.grid(eta=c(0.05),
                        max_depth=c(4),colsample_bytree=1,
                        subsample=1, nrounds=500, gamma=0, min_child_weight=5)

xgb.model <- train(NoShow ~ .,data=train, method="xgbTree",metric="ROC",
                   tuneGrid=xgb.grid, trControl=fit.control)

xgb.pred <- predict(xgb.model, newdata=test)
xgb.probs <- predict(xgb.model, newdata=test, type="prob")
```

```
test <- test %>% mutate(NoShow.numerical = ifelse(NoShow=="Yes",1,0))
confusionMatrix(xgb.pred, test$NoShow, positive="Yes")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    No   Yes
##        No  26423  6409
##        Yes   122   204
##
##                Accuracy : 0.803
##                  95% CI : (0.7987, 0.8073)
##     No Information Rate : 0.8006
##     P-Value [Acc > NIR] : 0.1313
##
##                   Kappa : 0.0408
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.030848
##             Specificity : 0.995404
##          Pos Pred Value : 0.625767
##          Neg Pred Value : 0.804794
##              Prevalence : 0.199439
##          Detection Rate : 0.006152
##    Detection Prevalence : 0.009832
##       Balanced Accuracy : 0.513126
##
##        'Positive' Class : Yes
##
```

```r
paste("XGBoost Area under ROC Curve: ", round(auc(test$NoShow.numerical, xgb.probs[,2]),3), sep="")
```

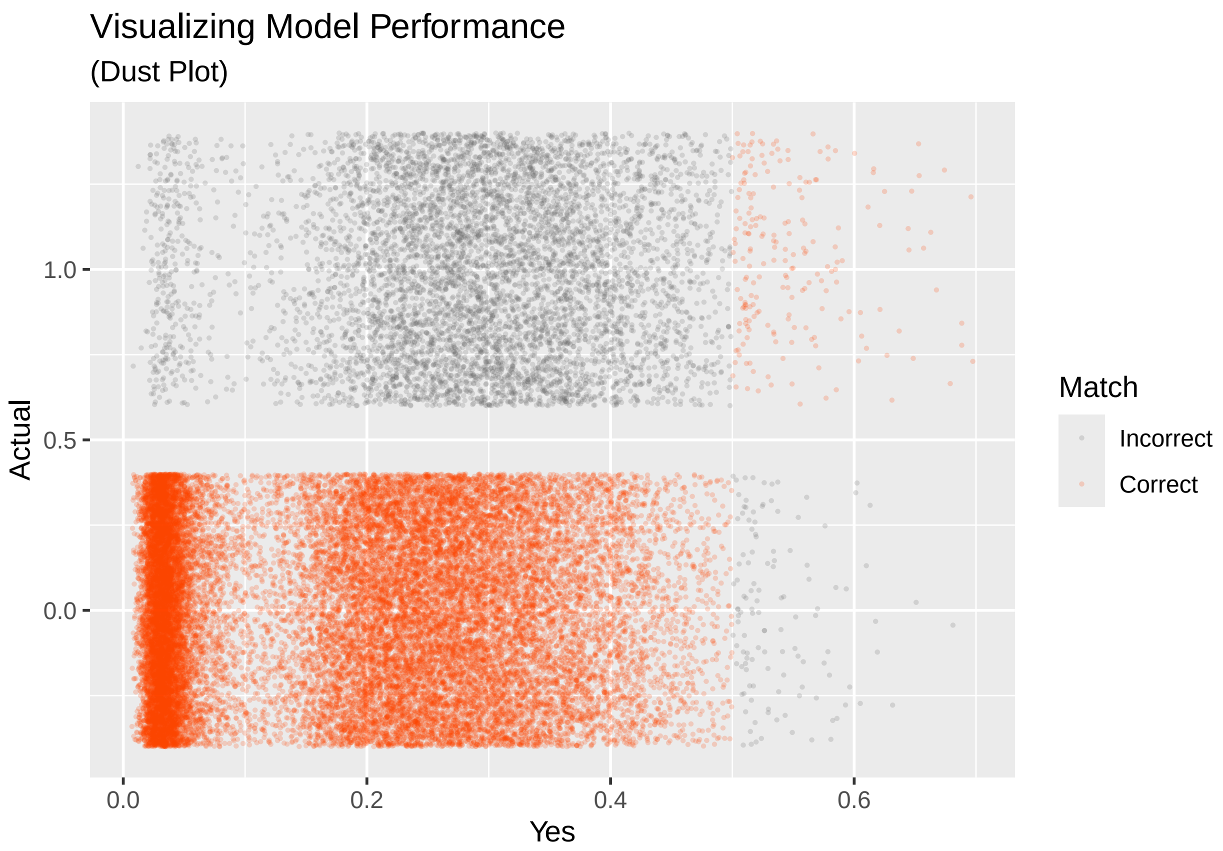
```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
## [1] "XGBoost Area under ROC Curve: 0.742"
```

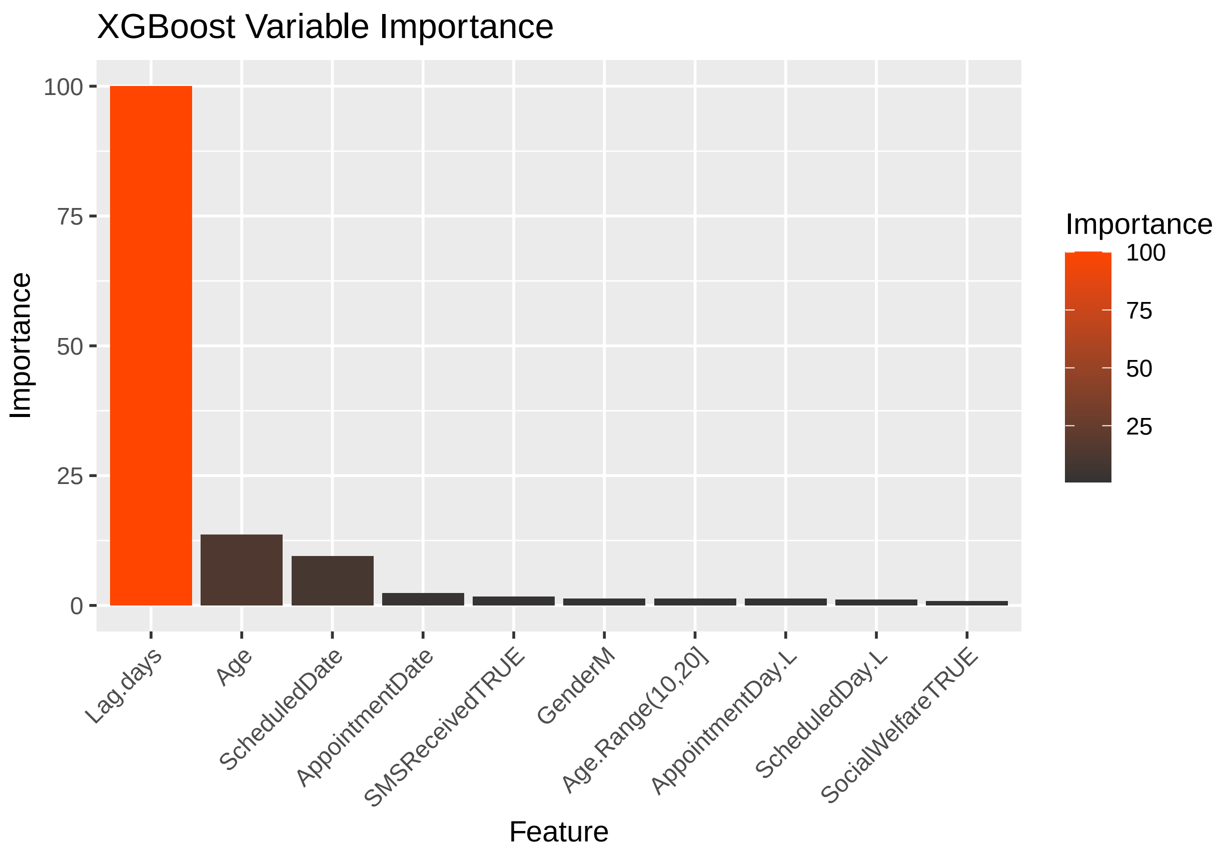
This isn't an unreasonable performance, but let's look a bit more carefully at the correct and incorrect predictions,

```r
xgb.probs$Actual = test$NoShow.numerical
xgb.probs$ActualClass = test$NoShow
xgb.probs$PredictedClass = xgb.pred
xgb.probs$Match = ifelse(xgb.probs$ActualClass == xgb.probs$PredictedClass,
                         "Correct","Incorrect")
# [4.8] Plot Accuracy
xgb.probs$Match = factor(xgb.probs$Match,levels=c("Incorrect","Correct"))
ggplot(xgb.probs,aes(x=Yes,y=Actual,color=Match))+
  geom_jitter(alpha=0.2,size=0.25)+
  scale_color_manual(values=c("grey40","orangered"))+
  ggtitle("Visualizing Model Performance", "(Dust Plot)")
```



Finally, let's close it off with the variable importance of our model:

```
results = data.frame(Feature = rownames(varImp(xgb.model)$importance)[1:10],
                     Importance = varImp(xgb.model)$importance[1:10,])

results$Feature = factor(results$Feature,levels=results$Feature)


# [4.10] Plot Variable Importance
ggplot(results, aes(x=Feature, y=Importance,fill=Importance))+
  geom_bar(stat="identity")+
  scale_fill_gradient(low="grey20",high="orangered")+
  ggtitle("XGBoost Variable Importance")+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



17. Using the caret package fit and evaluate 1 other ML model on this data.

```r
library(caret)
library(dplyr)
library(pROC)

# Load the data
data.prep <- dplyr::select(raw.data, -AppointmentID, -PatientID)

# Convert 'NoShow' to a binary factor
data.prep$NoShow <- factor(data.prep$NoShow, levels = c("No", "Yes"))

set.seed(42)
data.split <- initial_split(data.prep, prop = 0.7)
train  <- training(data.split)
test <- testing(data.split)

# Set up cross-validation
fitControl <- trainControl(method = "cv", number = 3, classProbs = TRUE, summaryFunction = twoClassSumma

# Train Random Forest model
set.seed(42)
rf_model <- train(NoShow ~ ., data = train, method = "rf", metric = "ROC", trControl = fitControl)

# Make predictions on the test set
rf_pred <- predict(rf_model, newdata = test)
```

```
rf_probs <- predict(rf_model, newdata = test, type = "prob")

# Create NoShow.numerical column in the test set
test <- test %>% mutate(NoShow.numerical = ifelse(NoShow == "Yes", 1, 0))

# Evaluate performance
rf_cm <- confusionMatrix(rf_pred, test$NoShow, positive = "Yes")
rf_auc <- round(auc(test$NoShow.numerical, rf_probs[,2]), 3)
```

```
## Setting levels: control = 0, case = 1

## Setting direction: controls < cases
```

```
# Print results
print(rf_cm)
```

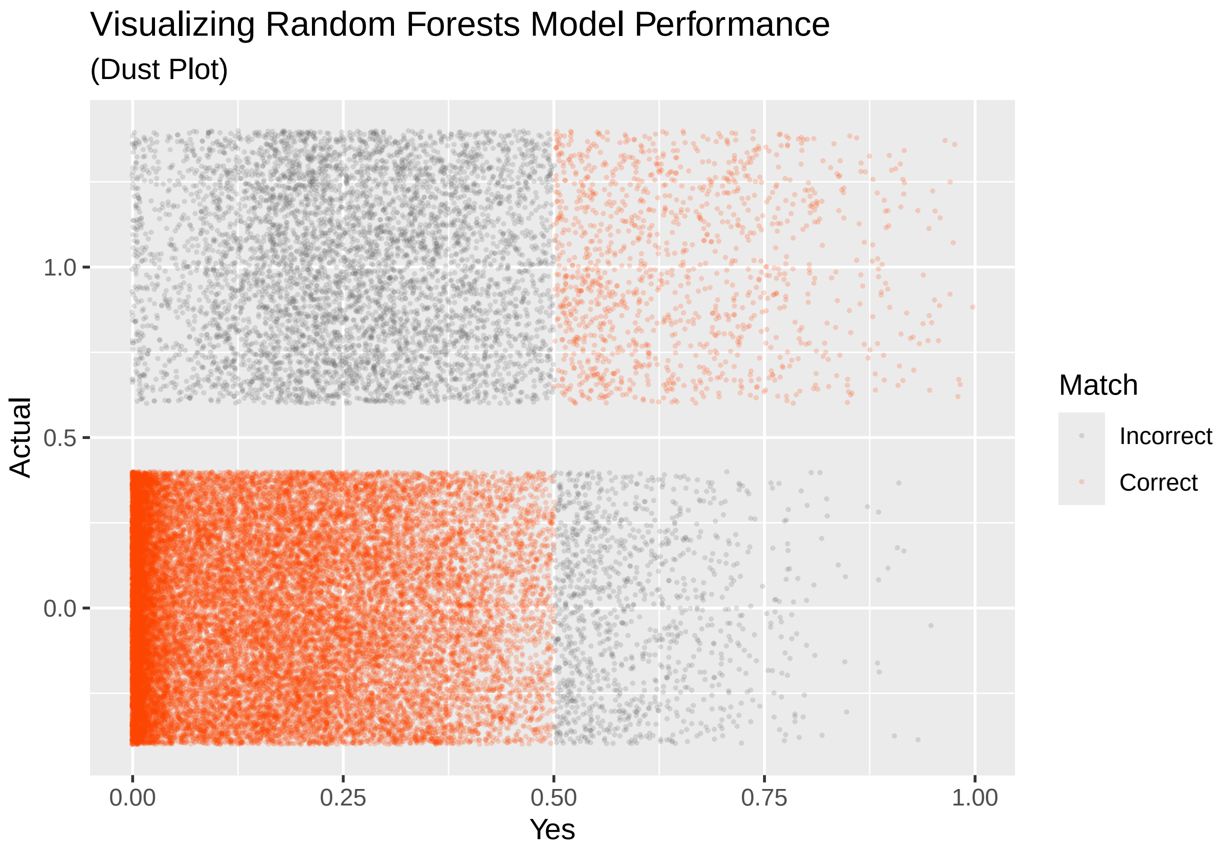
```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction    No   Yes
##        No  25432  5389
##        Yes  1113  1224
##
##                Accuracy : 0.8039
##                  95% CI : (0.7996, 0.8082)
##     No Information Rate : 0.8006
##     P-Value [Acc > NIR] : 0.06419
##
##                   Kappa : 0.1891
##
##  Mcnemar’s Test P-Value : < 2e-16
##
##             Sensitivity : 0.18509
##             Specificity : 0.95807
##          Pos Pred Value : 0.52375
##          Neg Pred Value : 0.82515
##              Prevalence : 0.19944
##          Detection Rate : 0.03691
##    Detection Prevalence : 0.07048
##       Balanced Accuracy : 0.57158
##
##        ’Positive’ Class : Yes
##
```

```
print(paste("Random Forest Area under ROC Curve: ", rf_auc, sep=""))
```

```
## [1] "Random Forest Area under ROC Curve: 0.744"
```

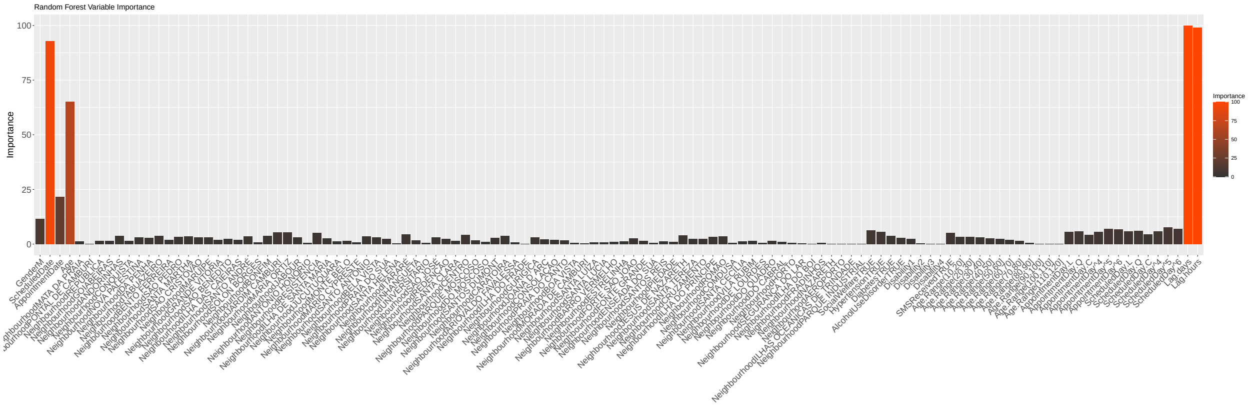
```
##Visualize Random Forests Model Performance
# Make predictions using Random Forest model
rf.pred <- predict(rf_model, newdata = test)
```

```r
rf.probs <- predict(rf_model, newdata = test, type = "prob")

# Add the actual and predicted classes to the rf.probs data frame
rf.probs$Actual <- test$NoShow.numerical
rf.probs$ActualClass <- test$NoShow
rf.probs$PredictedClass <- rf.pred
rf.probs$Match <- ifelse(rf.probs$ActualClass == rf.probs$PredictedClass, "Correct", "Incorrect")

# Plot Accuracy
rf.probs$Match <- factor(rf.probs$Match, levels = c("Incorrect", "Correct"))
ggplot(rf.probs, aes(x = Yes, y = Actual, color = Match)) +
  geom_jitter(alpha = 0.2, size = 0.25) +
  scale_color_manual(values = c("grey40", "orangered")) +
  ggtitle("Visualizing Random Forests Model Performance", "(Dust Plot)")
```



```r
# Plot Variable Importance
library(ggplot2)

# Extract variable importance from the Random Forest model
var_importance <- varImp(rf_model)$importance
results <- data.frame(Feature = rownames(var_importance), Importance = var_importance$Overall)

# Convert Feature to a factor for proper ordering in the plot
results$Feature <- factor(results$Feature, levels = results$Feature)
```

```
# Plot Variable Importance
ggplot(results, aes(x = Feature, y = Importance, fill = Importance)) +
  geom_bar(stat = "identity") +
  scale_fill_gradient(low = "grey20", high = "orangered") +
  ggtitle("Random Forest Variable Importance") +theme(axis.title=element_text(size=16))+theme(axis.text
  theme(axis.text.x = element_text(angle = 45, hjust = 1, size=15))
```



18. Based on everything, do you think we can trust analyses based on this dataset? Explain your reasoning.

I would say based on everything that we can somewhat trust the analyses based on this dataset, but with some apprehensions and limitations. For example, as mentioned previously, there are some noticeable imbalances in the data; this is especially true when looking at the target variable of no-show appointments. The higher prevalence of no-shows definitely had an influence on both models' performances when it came to things like sensitivity. However, we were able to predict things with a moderately acceptable predictive power (i.e., an AUC score of around 0.7). This is confirmed by our dust plot results where both models seem to perform well at low actual values, but do not predict that reliably at higher actual values. There is a noticeable difference in the variables that are found to be of importance between both different models run. For example, while Gender is found to be a variable of some importance in the Random Forest model, it is not found to be important in the XGBoost model. Moreover, the presence or absence of receiving an SMS reminder was found to be more of an important variable in the XGBoost model than in the Random Forest model. We also find that there were many more variables in the Random Forest model that showed minimal importance that were not present at all in the XGBoost model. Although somewhat acceptable predictive power is present, there seems to be a lack of consistency between both models. As the dataset does show some limitations, such as the imbalance in shows vs. no-shows for appointments, we could likely do a few more tweaks to the dataset and perform more modeling techniques to try and address these and other similar issues to make the analyses on this dataset and subsequent analyses more reliable.

Credits

This notebook was based on a combination of other notebooks e.g., 1, 2, 3