Health Data Science Practical 1

Jenna Hepburn

May 28, 2023

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 - Number for each patient that uniquely identifies them.

AppointmentID - Number for each appointment that uniquely identifies it.

Gender - Patient gender (Male or Female).

ScheduledDate - Date that the patient scheduled the appointment.

AppointmentDate - Date the appointment occurred.

Age - Patient age at time of appointment.

Neighbourhood - The neighbourhood/district of Vitória (capital city of Espírito Santo, Brazil) in which the appointment occurred.

SocialWelfare - Binary indicator of whether the patient is a recipient of Bolsa Família welfare payments, which are welfare payments of the country of Brazil for people who fall below the poverty line.

Hypertension - Binary indicator of whether the patient has been diagnosed with hypertension.

Diabetes - Binary indicator of whether the patient has been diagnosed with diabetes.

AlcoholUseDisorder - Binary indicator of whether the patient has been diagnosed with alcohol use disorder.

Disability - Whether the patient previously diagnosed with a disability. 0 value if no disability and severity rating of 1-4 if the patient has a disability.

SMSReceived - Binary indicator of whether a reminder text sent to the patient before appointment.

NoShow - Indicator of whether the patient attended scheduled appointment (yes/no).

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

- Unable to get time off work to attend appointment as medical appointments typically happen during weekday working hours which may be a barrier for people to attend.
- Disability or chronic illness could make it difficult for patients to physically make it to appointments.
- Transportation issues including not having a vehicle, not being able to afford public transit, or improper, unavailable, or unsafe public transit.

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?

- More information on comorbidities such as chronic illnesses and diagnoses other than the 4 included (hypertension, diabetes, alcohol use disorder, and disability)
- Access to a vehicle
- Level of educational attainment or socioeconomic status
- Hospitalizations

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('2016_05v2_VitoriaAppointmentData.csv', col_types='fffTTifllllflf')

raw.data <- readr::read_csv('https://raw.githubusercontent.com/jennahepburn/healthdatasci/main/Practical
```

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 %>% 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 100 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 two individuals who are 115 years old. This is on the edge of being biologically plausible, as the oldest ever person lived to 122 years old. However, while this may seem unlikely, 115 is not old enough to be biologically impossible. Therefore, I will not drop these patients.

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?

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

```
## # A tibble: 62,299 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 62,289 more rows
```

```
raw.data %>% filter(PatientID == 822145925426128) %>% unique()
```

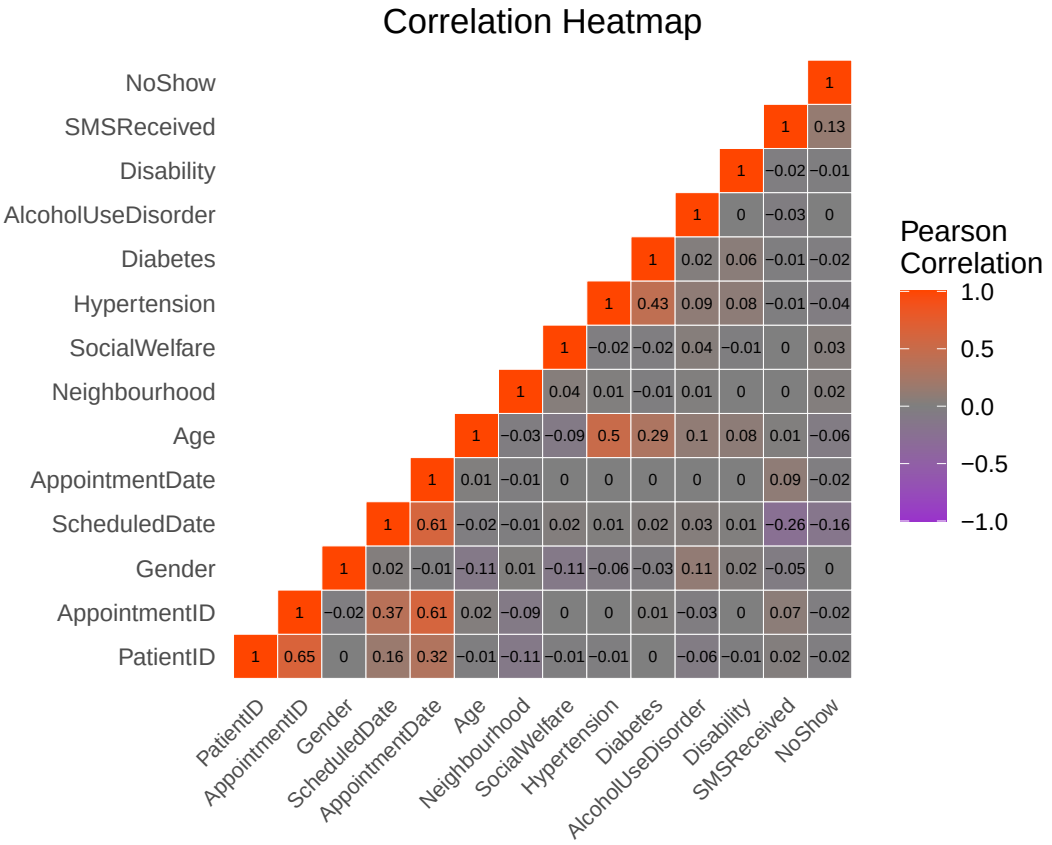
```
## # A tibble: 88 x 14
##    PatientID  AppointmentID Gender ScheduledDate       AppointmentDate       Age
##    <fct>      <fct>         <fct>  <dttm>              <dttm>              <int>
##  1 822145925~ 5638995       M      2016-04-29 08:38:44 2016-04-29 00:00:00    38
##  2 822145925~ 5642878       M      2016-04-29 18:02:42 2016-04-29 00:00:00    38
##  3 822145925~ 5640809       M      2016-04-29 11:27:34 2016-04-29 00:00:00    38
##  4 822145925~ 5705135       M      2016-05-16 18:38:11 2016-05-16 00:00:00    38
##  5 822145925~ 5668887       M      2016-05-06 09:54:32 2016-05-06 00:00:00    38
##  6 822145925~ 5735078       M      2016-05-24 16:56:06 2016-05-24 00:00:00    38
##  7 822145925~ 5710752       M      2016-05-17 17:47:53 2016-05-17 00:00:00    38
##  8 822145925~ 5736811       M      2016-05-25 08:48:46 2016-05-25 00:00:00    38
##  9 822145925~ 5711913       M      2016-05-18 08:13:00 2016-05-18 00:00:00    38
## 10 822145925~ 5682166       M      2016-05-10 15:56:47 2016-05-13 00:00:00    38
## # i 78 more rows
## # i 8 more variables: Neighbourhood <fct>, SocialWelfare <lgl>,
## #   Hypertension <lgl>, Diabetes <lgl>, AlcoholUseDisorder <lgl>,
## #   Disability <fct>, SMSReceived <lgl>, NoShow <fct>
```

The max number of appointments booked by the same patient is 88. However, it appears he did not show up to most of the appointments. He attended 3 of the 88 booked appointments.

Let's explore the correlation between variables:

```
# 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 variables that are most strongly correlated with missing appointments are SMSReceived and ScheduledDate.

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

No variables are overly highly correlated with each other, however, based on the heat map scale, we will consider correlations greater than or equal to ± 0.5 as a strong correlation.

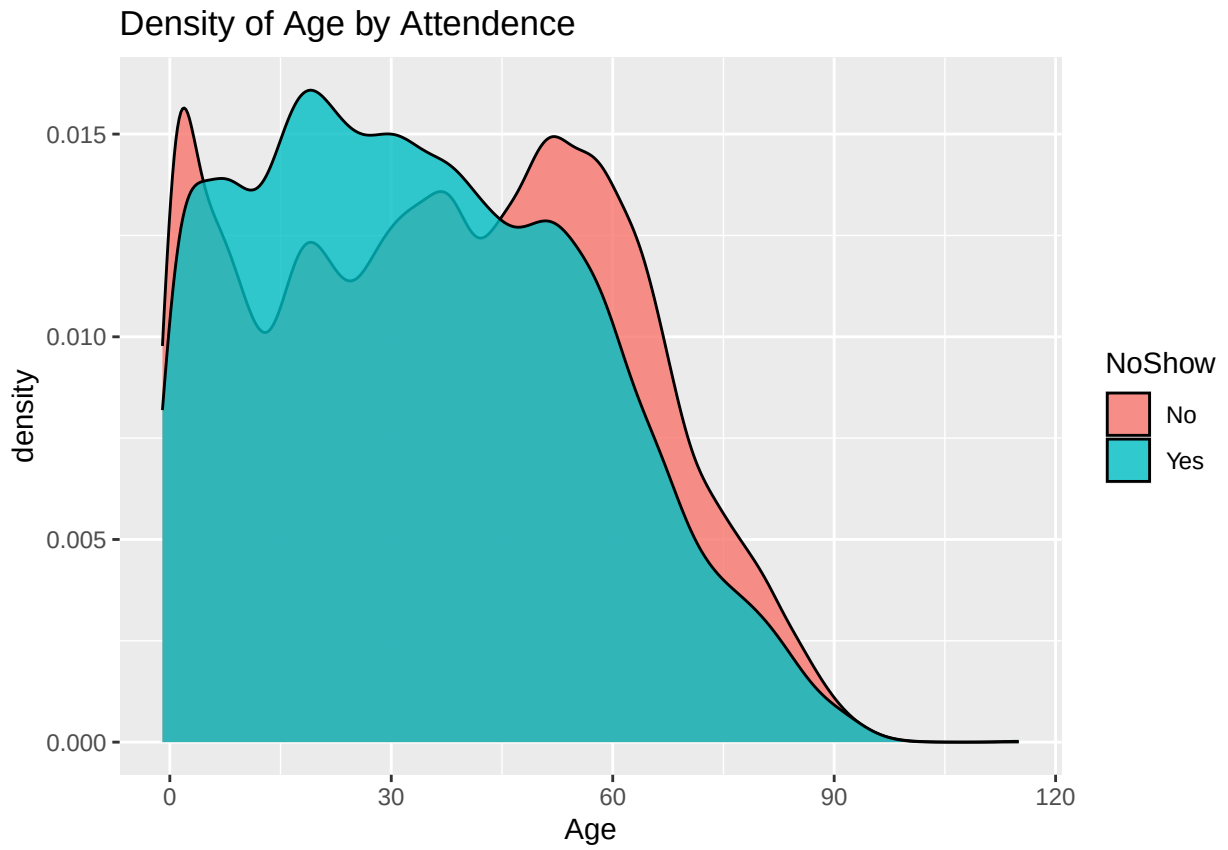
- Appointment ID and patient ID
- Appointment ID and appointment date
- Appointment date and scheduled date
- Age and hypertension

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

Yes, I do not think patient ID and appointment ID should be included in this plot as they are dependent on each other and therefore expected to be correlated. Appointment IDs will only ever have 1 patient ID associated with it. Additionally, there is no useful information to be gained from these values as they are randomly generated and cannot give us any prediction power to determine something such as missing an appointment.

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

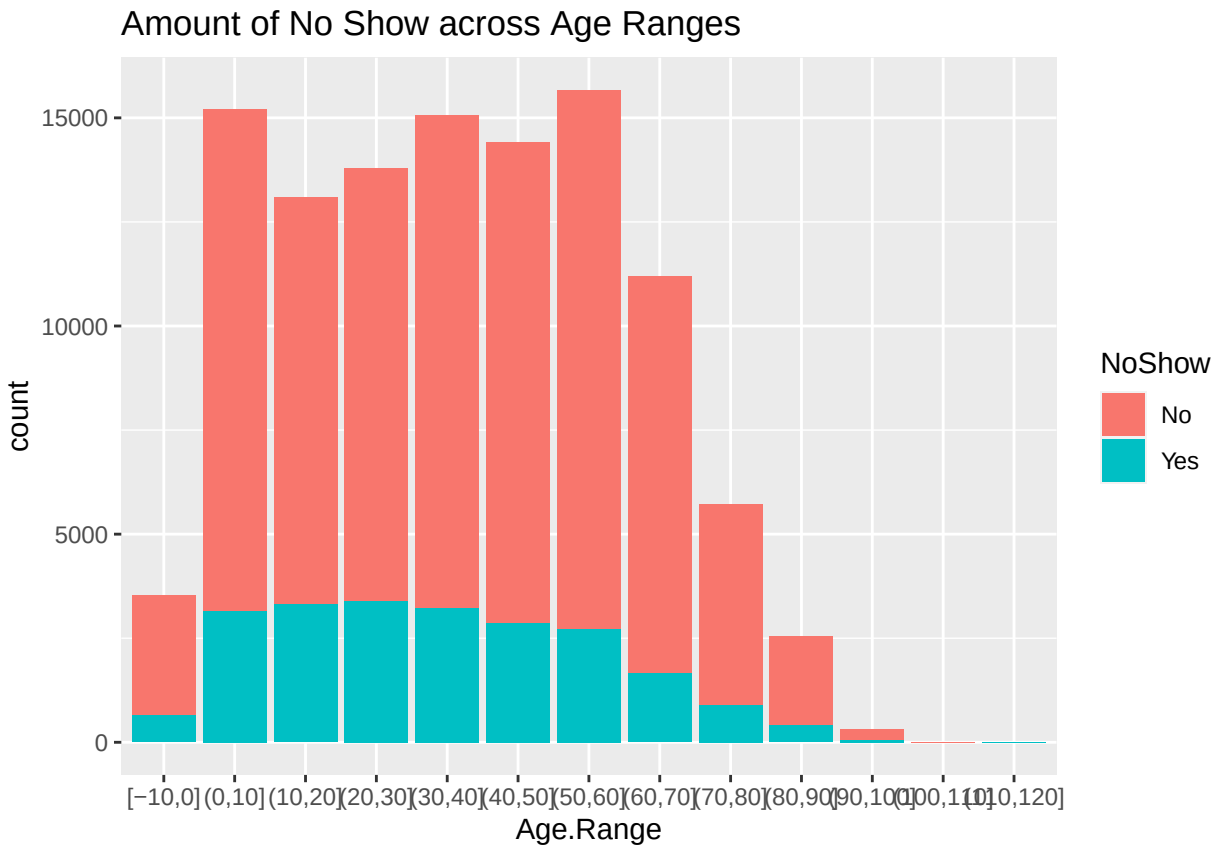
```
ggplot(raw.data) +
  geom_density(aes(x=Age, fill=NoShow), alpha=0.8) +
  ggtitle("Density of Age by Attendence")
```



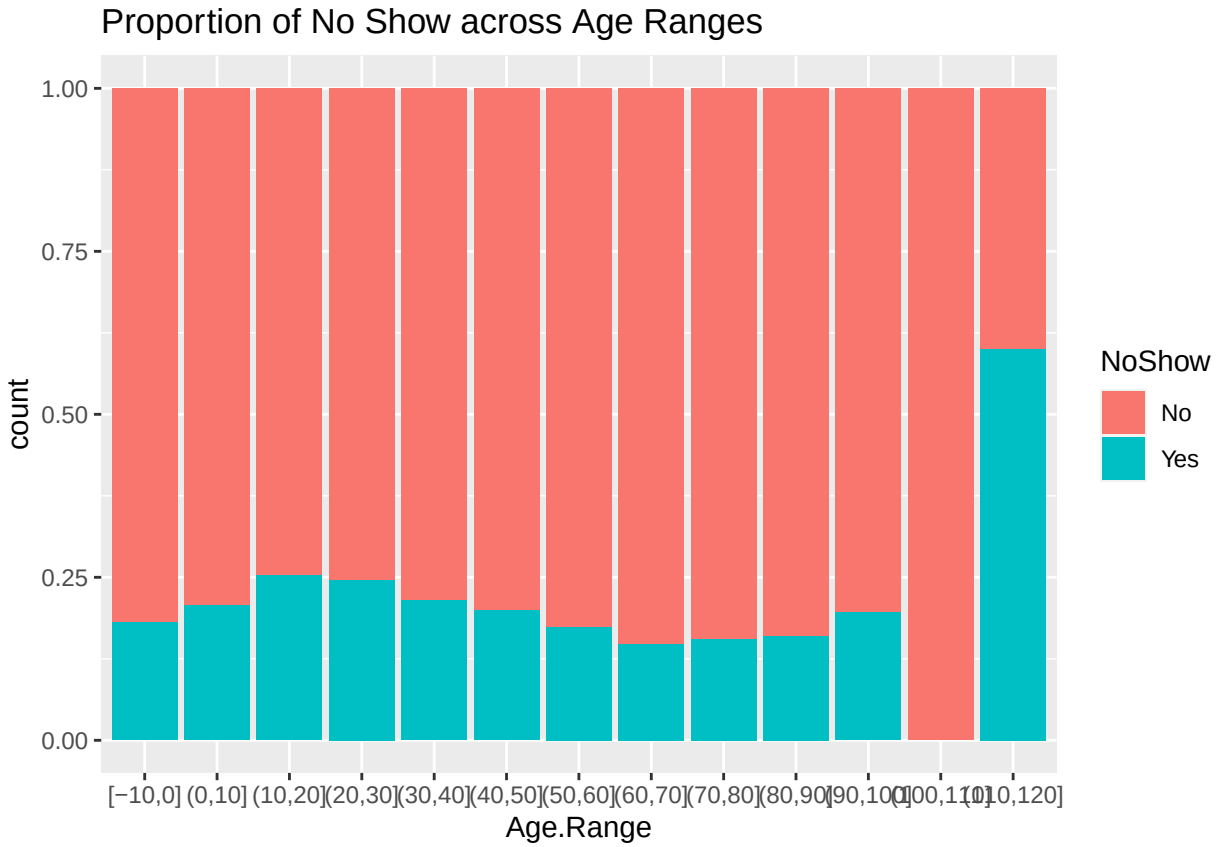
There does seem to be a difference in the distribution of ages of people that miss and don't miss appointments. However, the shape of this distribution means the actual correlation is near 0 in the heatmap above. This highlights the need to look at individual variables.

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

```
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")
```




```
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 I had only plotted the second plot, it may seem that there was a large number of individuals no-showing appointments. For example, there are only 2 individuals (5 appointments) for the 110-120 age group, but one of the individuals missed 3 of 4 appointments. These 3 of 5 missed appointments for this age group makes it look like a massive burden of no-showed appointments, but in reality the effect is fairly small since there are only 2 individuals represented in this age group. This is much better represented in the first plot, which demonstrates the size of the groups to better understand the effect size. Additionally, trends are more easily seen with the first plot, which shows the no-shows appear to decrease with age starting at age 40-50.

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.

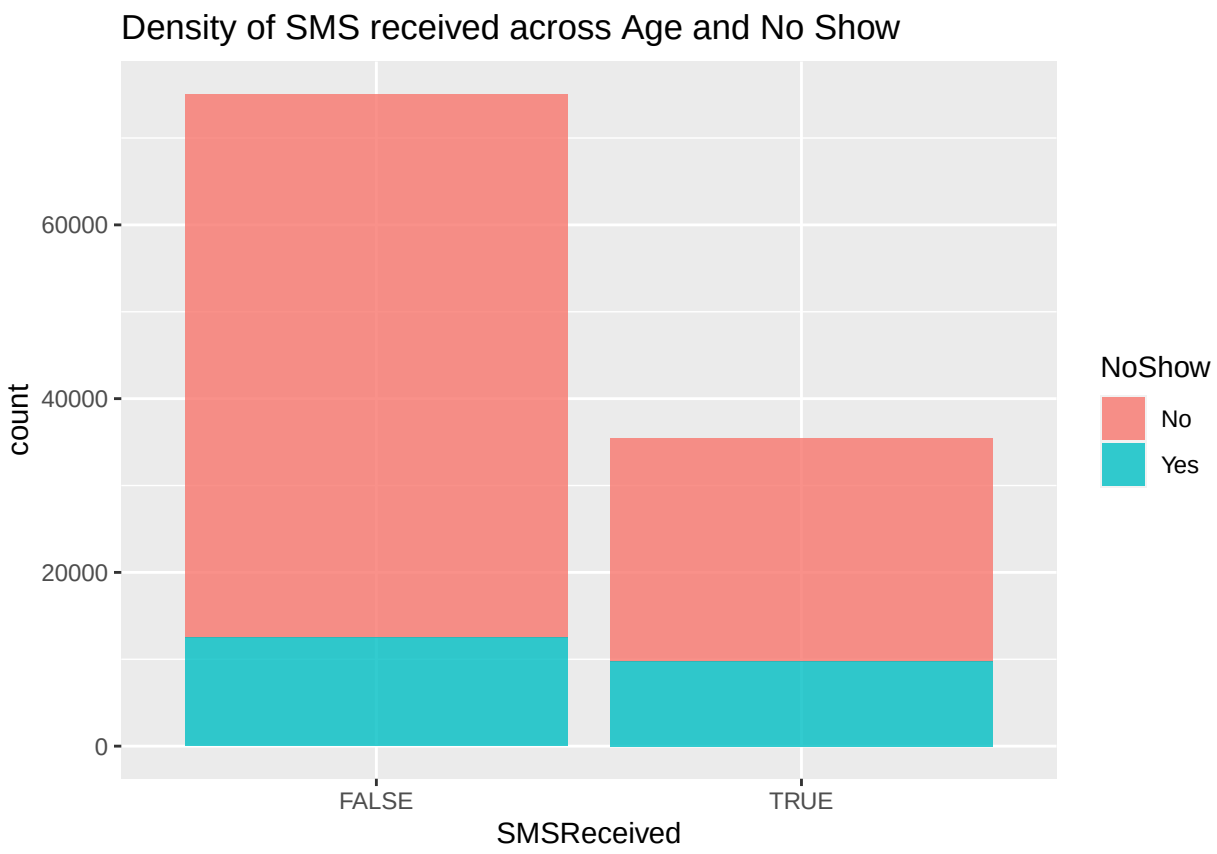
Another interesting finding is the `NA` group, they are the result of trying to assign age of 0 to groups and represent missing data.

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

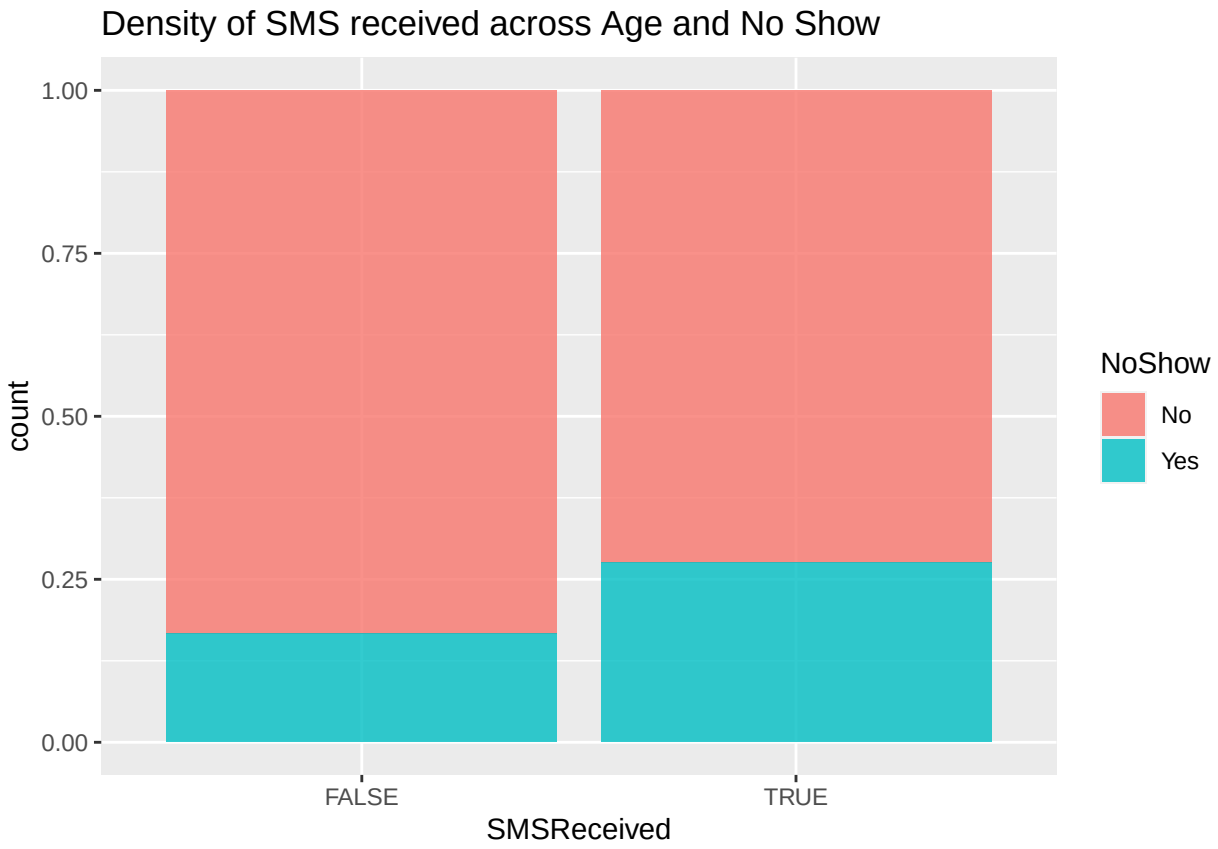
```
## # A tibble: 1 x 1
##       n
##   <int>
## 1  3539
```

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

```
ggplot(raw.data) +
  geom_bar(aes(x=SMSReceived, fill=NoShow), alpha=0.8) +
  ggtitle("Density of SMS received across Age and No Show")
```




```
ggplot(raw.data) +
  geom_bar(aes(x=SMSReceived, fill=NoShow), position='fill', alpha=0.8) +
  ggtitle("Density of SMS received across Age and No Show")
```

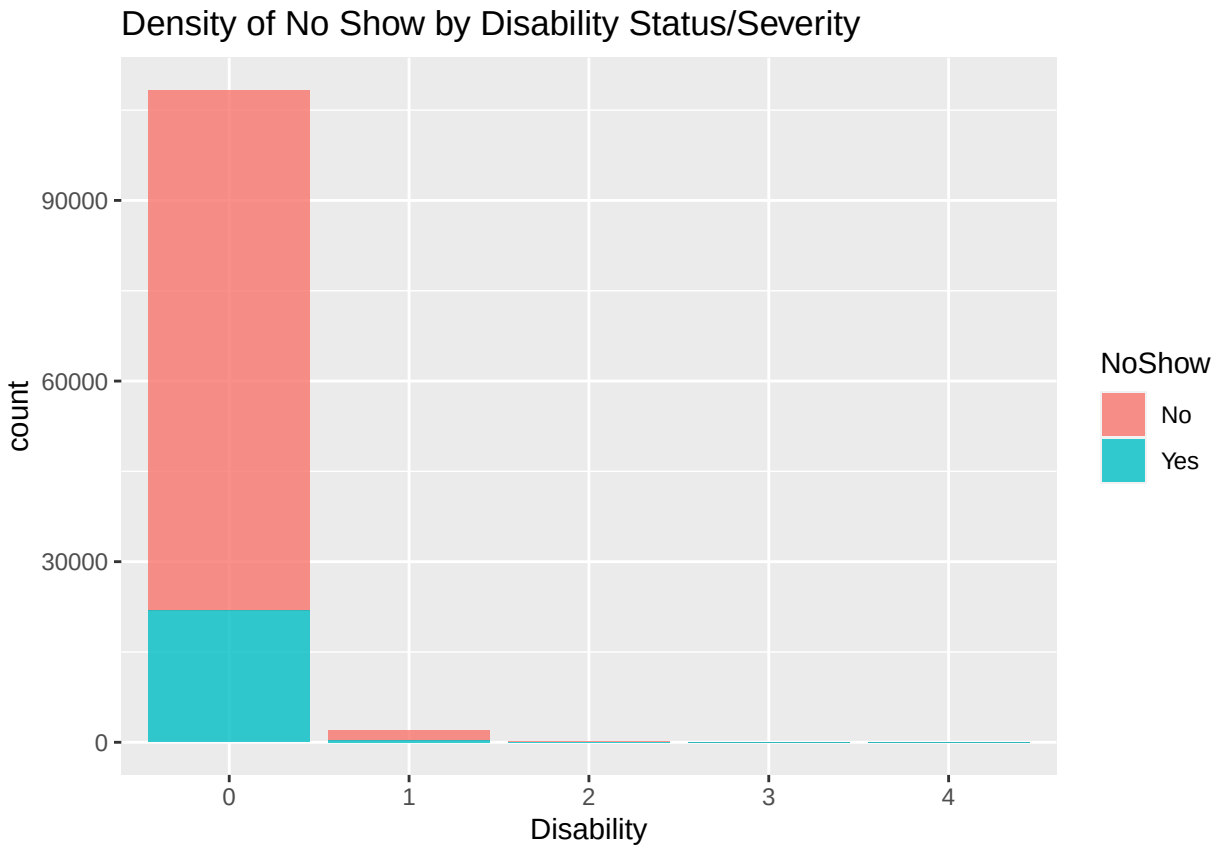


11 From this plot does it look like SMS reminders increase or decrease the chance of someone not attending an appointment? Why might the opposite actually be true (hint: think about biases)?

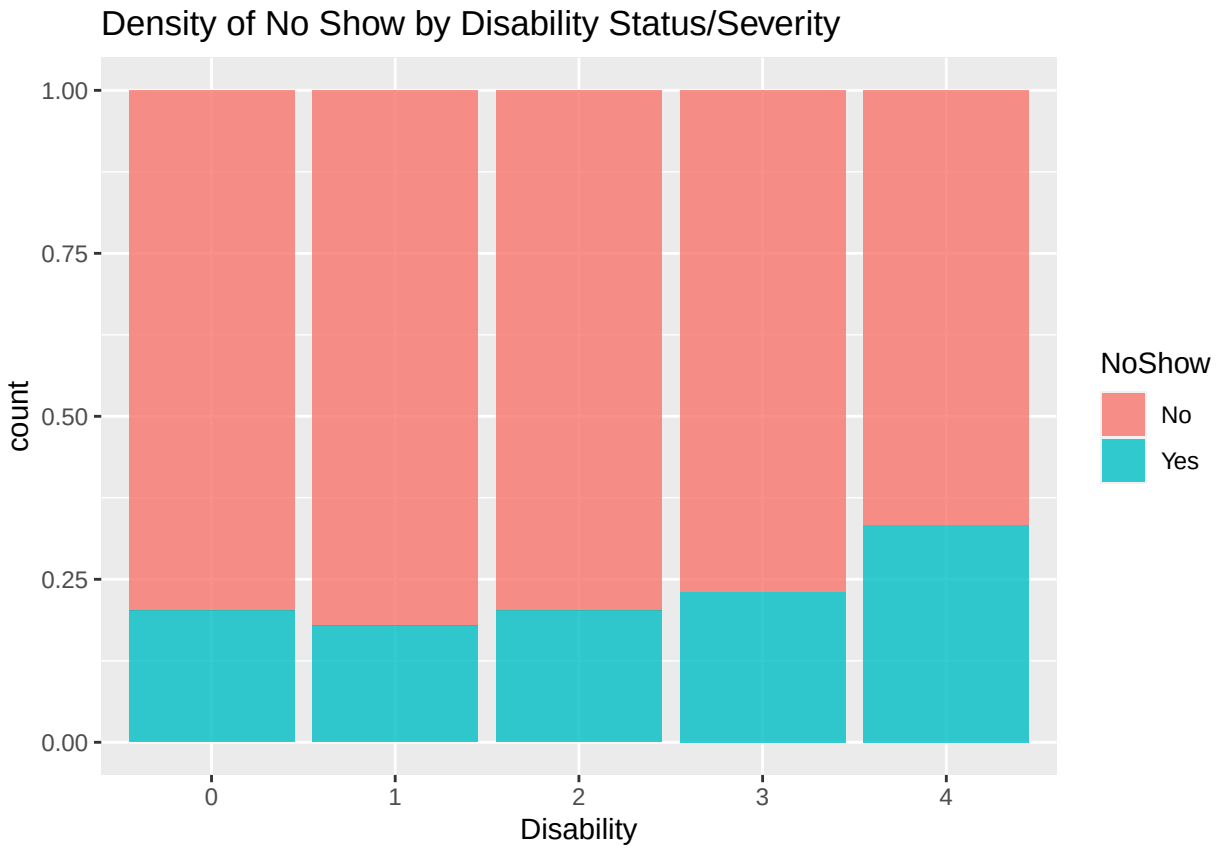
It appears that receiving an SMS reminder reduces the chances of attending an appointment. However, this may not be entirely accurate. Firstly, from the previous question, we can see that age groups under 40 are more likely to miss an appointment. This younger age group would also be more likely to have a cell phone to receive SMS reminders compared to older ages. Additionally, same day appointments or walk-in appointments would likely not send patients a reminder. If an appointment is booked same day or through walk-in, the patient would be less likely to miss the appointment, despite not receiving a reminder. These factors may bias the results we see with the graphs.

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

```
ggplot(raw.data) +
  geom_bar(aes(x=Disability, fill=NoShow), alpha=0.8) +
  ggtitle("Density of No Show by Disability Status/Severity")
```




```
ggplot(raw.data) +
  geom_bar(aes(x=Disability, fill=NoShow), position='fill', alpha=0.8) +
  ggtitle("Density of No Show by Disability Status/Severity")
```



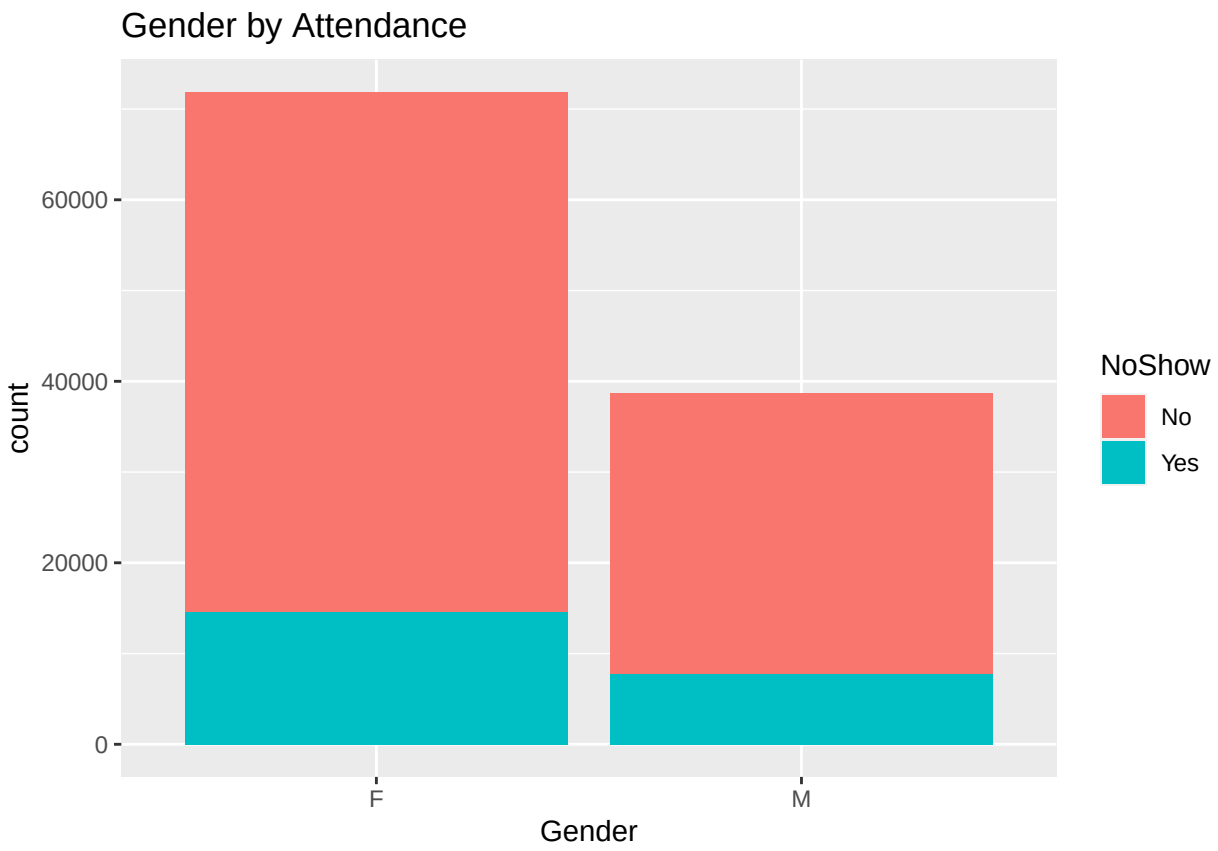
Most neighborhoods have similar proportions of no-show but some have much higher and lower rates.

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

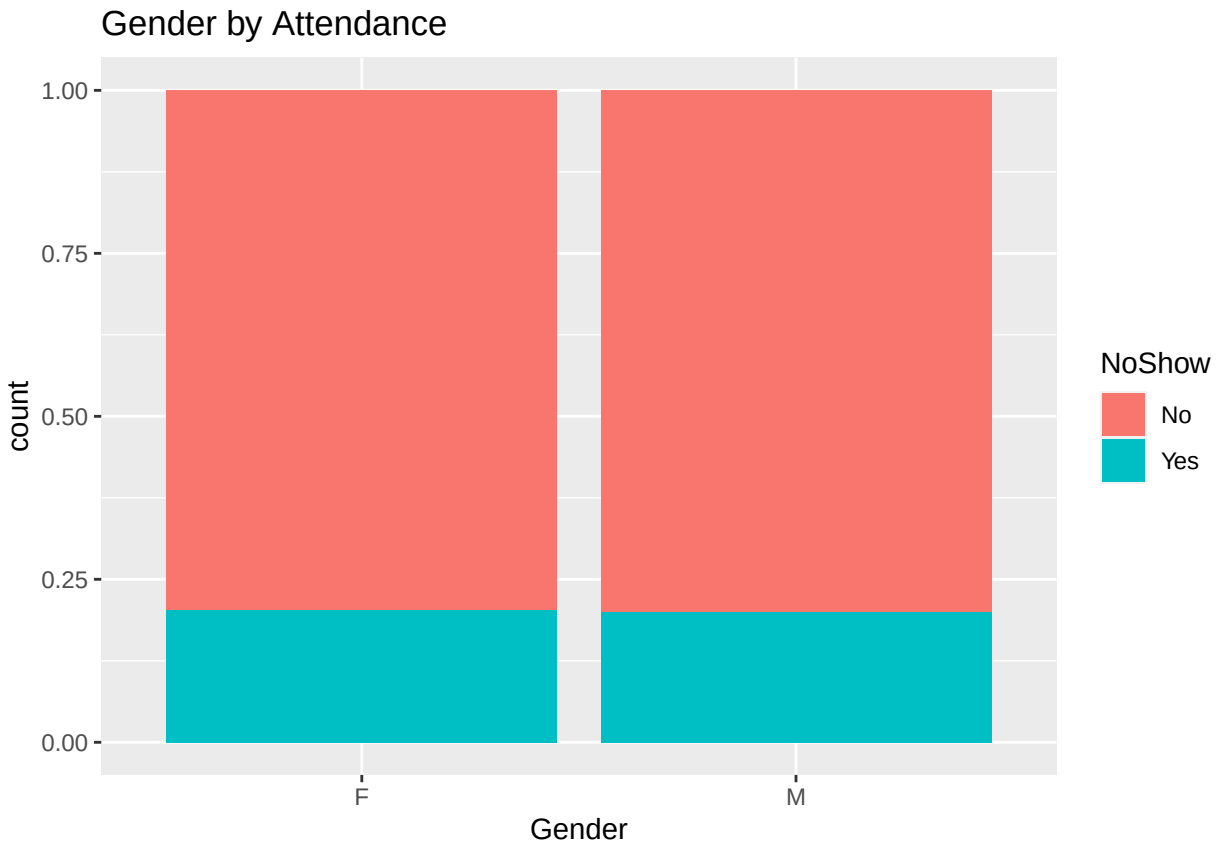
Differences in attendance by neighbourhood may be explained by differences in socioeconomic status by location. Often in research postal code or location can be used as a proxy for socioeconomic status, and this may be valid in this case as well. There may also be differences in distance to a medical clinic by neighbourhood that makes it more difficult to attend appointments due to increased transportation and time requirements.

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

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

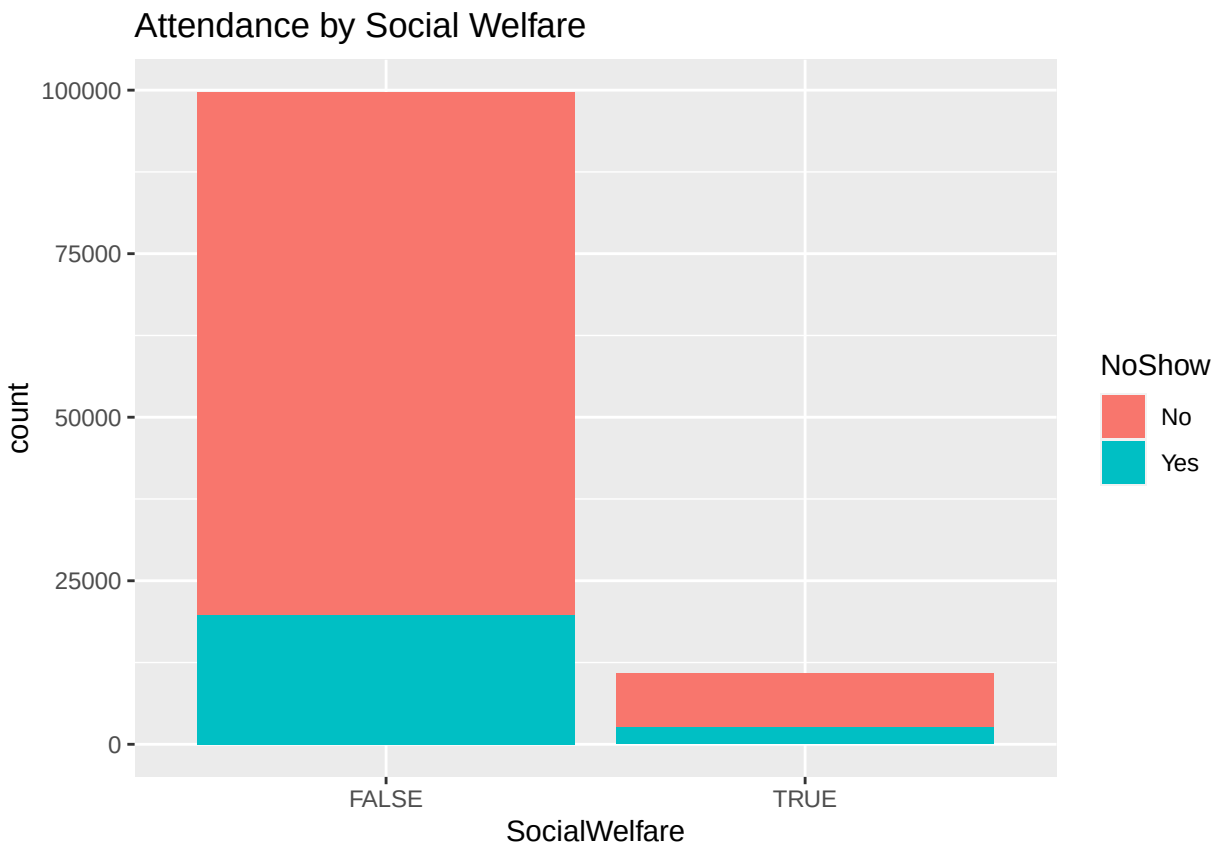



```
ggplot(raw.data) +
  geom_bar(aes(x=Gender, fill=NoShow), position='fill') +
  ggtitle("Gender by Attendance")
```

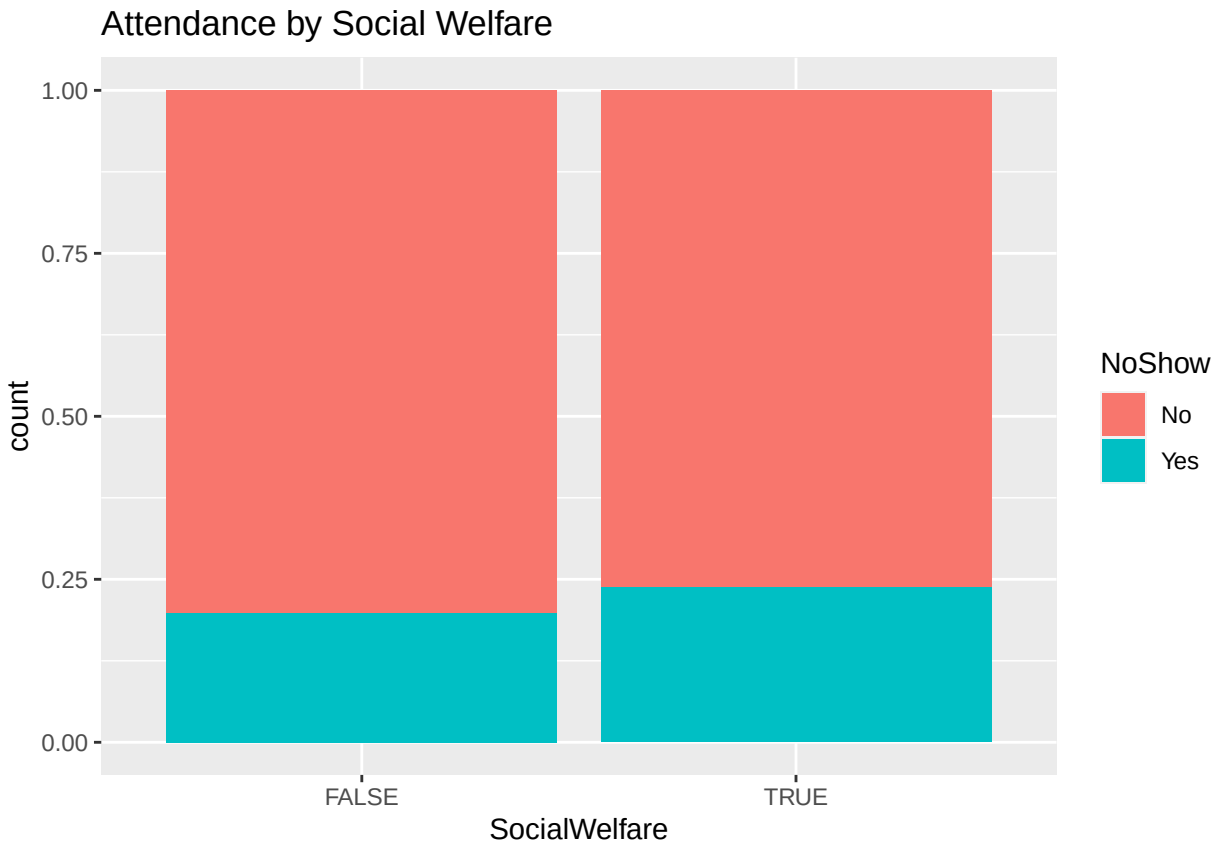


14 Create a similar plot using `SocialWelfare`

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




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



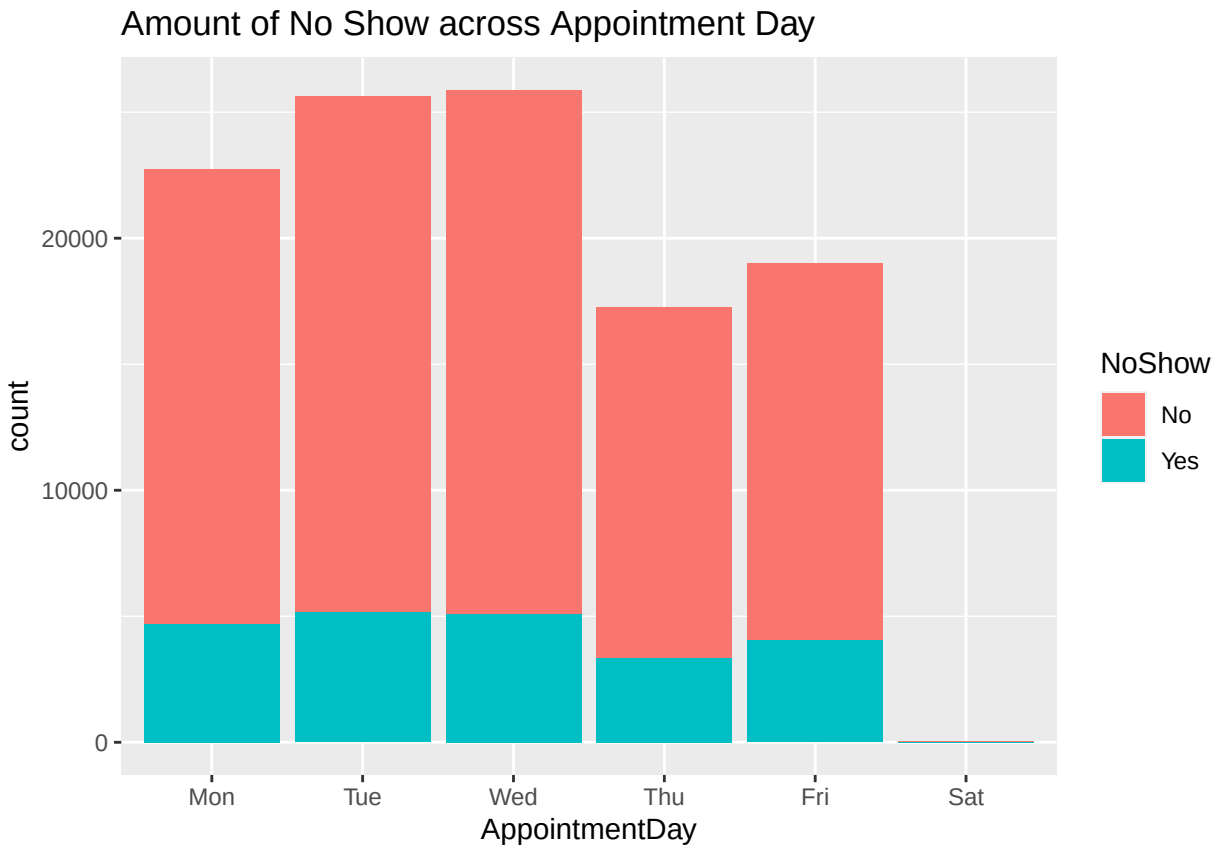
Far more exploration could still be done, including dimensionality reduction approaches but although we have found some patterns there is no major/striking patterns on the data as it currently stands.

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

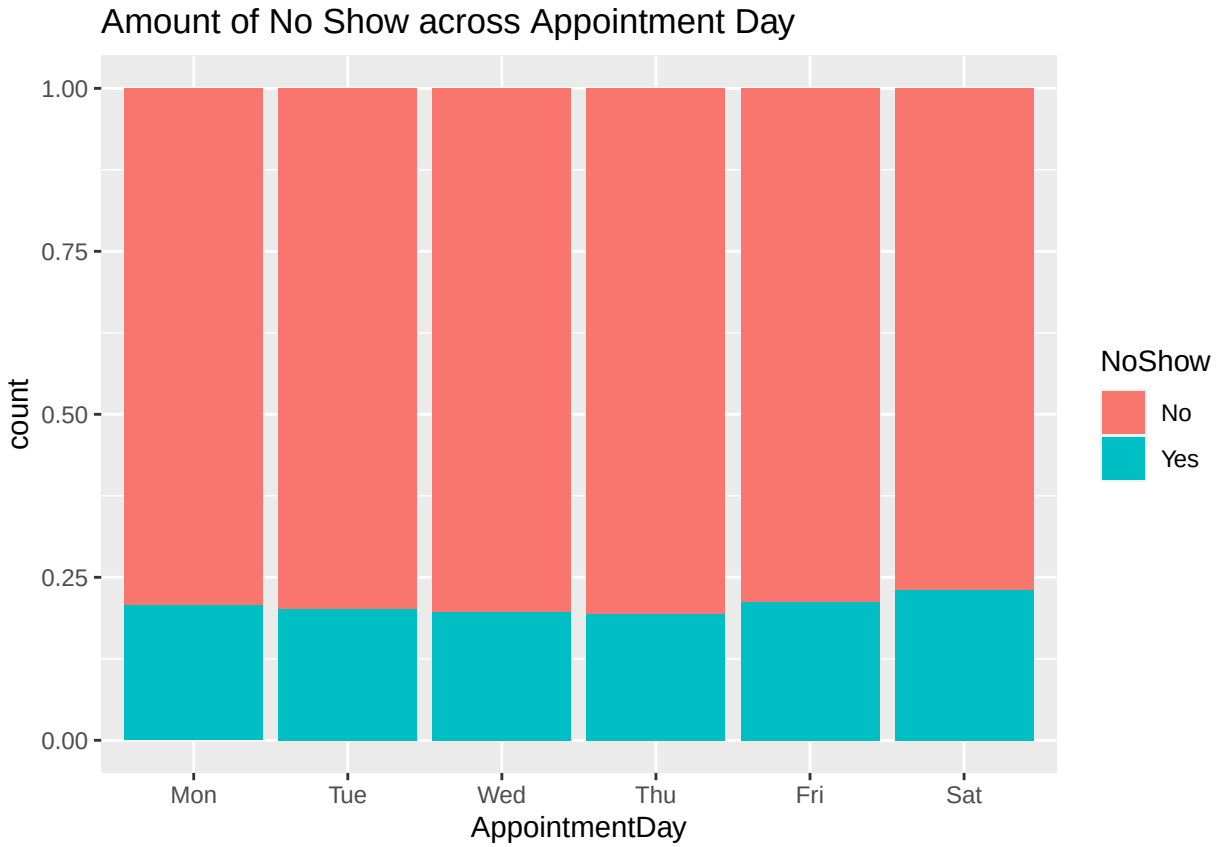
Feature Engineering

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!

```
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")
```



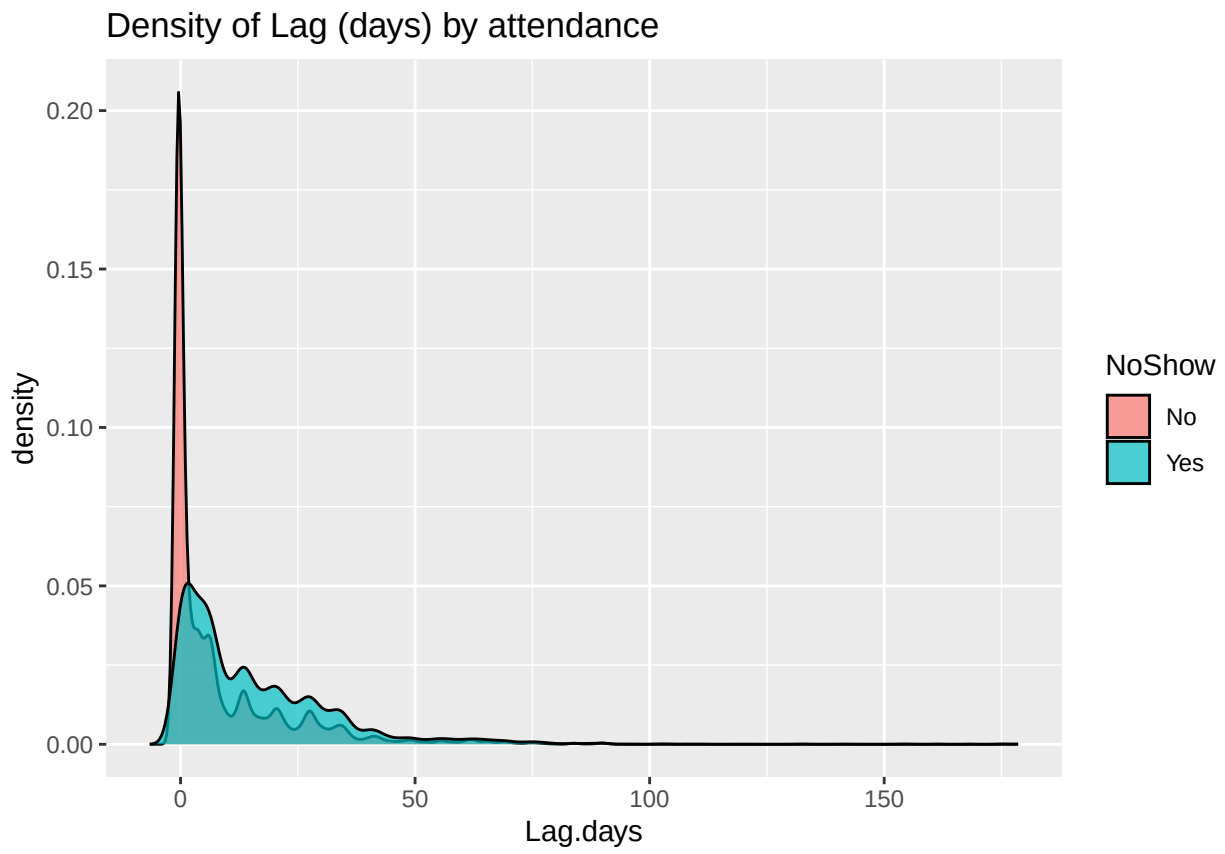

```
ggplot(raw.data) +
  geom_bar(aes(x=AppointmentDay, fill=NoShow), position = 'fill') +
  ggtitle("Amount of No Show across Appointment Day")
```



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?

Yes, it seems odd that that a huge portion of the patients that showed up to their appointment had 0 days between making the appointment and attending the appointment. This makes me think that some of these visits were likely walk-in visits (which by nature have a 0% no-show rate), or same day booking which would likely have a reduced rate of no-shows compared to appointments booked ahead of time.

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.

```
data.prep <- raw.data %>% select(-AppointmentID, -PatientID)

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 the issues and biases we identified in previous questions including effect of walk-in appointments, missing info that may be important, bias in SMS reminders, etc., it is likely that this model will also be biased and therefore not overly effective as a predictive tool. However, there are over 110,000 observations which is a large dataset to base this off of, which may work in favour of the model despite potential bias.

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  26385  6390
##        Yes   142   242
##
##                Accuracy : 0.803
##                  95% CI : (0.7987, 0.8073)
##     No Information Rate : 0.8
##     P-Value [Acc > NIR] : 0.08578
##
##                   Kappa : 0.0481
##
##  Mcnemar's Test P-Value : < 2e-16
##
##             Sensitivity : 0.036490
##             Specificity : 0.994647
##          Pos Pred Value : 0.630208
##          Neg Pred Value : 0.805034
##              Prevalence : 0.200006
##          Detection Rate : 0.007298
##    Detection Prevalence : 0.011581
##       Balanced Accuracy : 0.515568
##
##        'Positive' Class : Yes
##
```

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

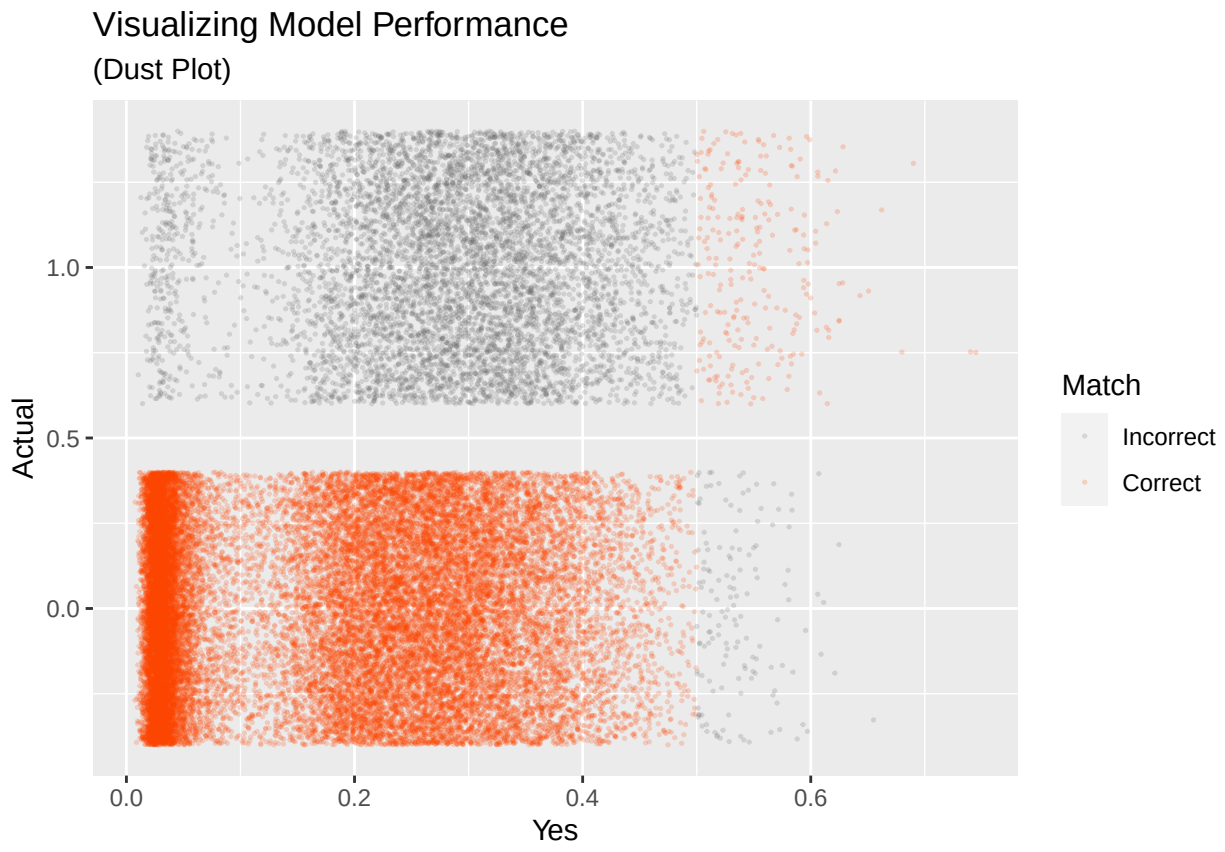
```
## Setting levels: control = 0, case = 1
```

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

```
## [1] "XGBoost Area under ROC Curve: 0.74"
```

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

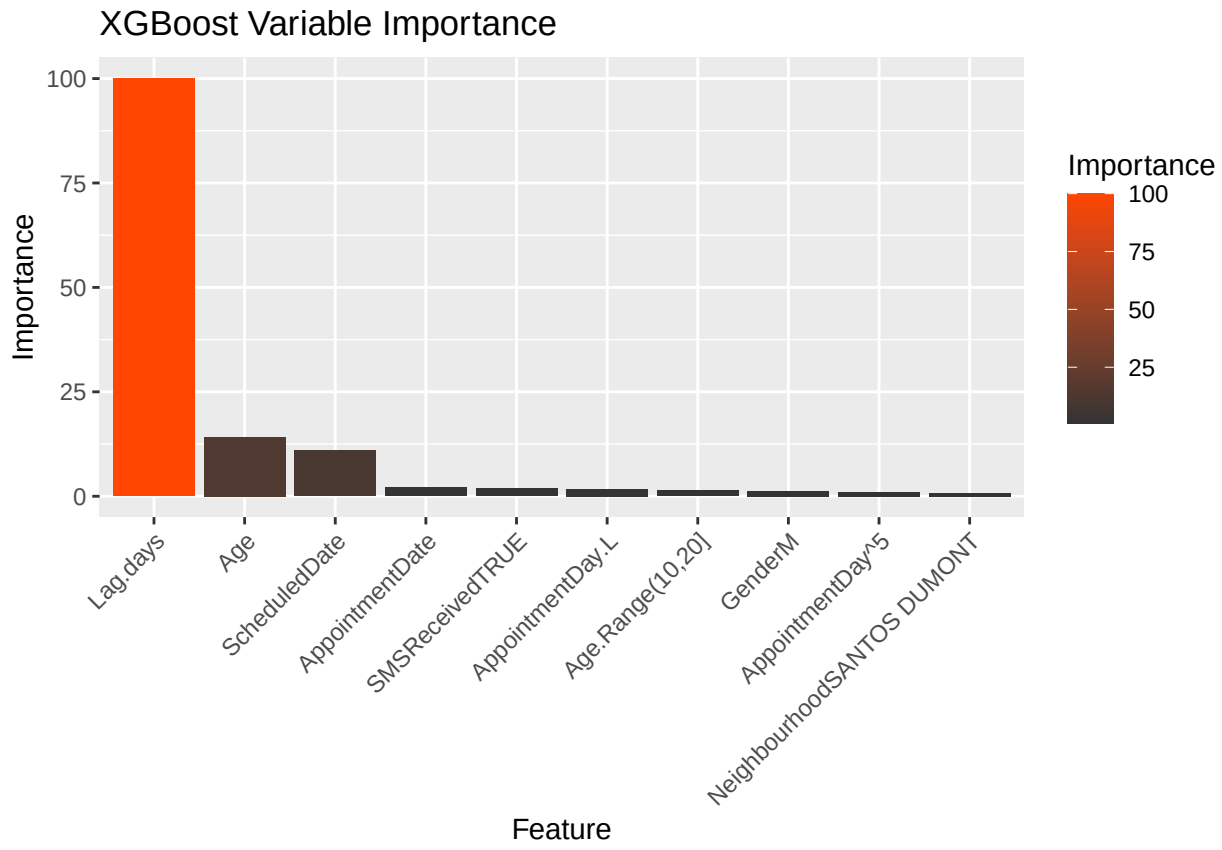
```
xgb.probs$Actual = test$NoShow.numerical
xgb.probs$ActualClass = test$NoShow
xgb.probs$PredictedClass = xgb.pred
xgb.probs$Match = ifelse(xgb.probs$ActualClass == xgb.probs$PredictedClass,
                         "Correct","Incorrect")
# [4.8] Plot Accuracy
xgb.probs$Match = factor(xgb.probs$Match,levels=c("Incorrect","Correct"))
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.

```
train.ctl <- trainControl(
  method = "repeatedcv",
  number = 10,
  repeats = 10)

d.tree <- train(NoShow ~ .,
                data = train,
                method = "rpart",
                trControl = train.ctl)

d.tree
```

```
## CART
```

```
##
## 77368 samples
##    16 predictor
##     2 classes: 'No', 'Yes'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 10 times)
## Summary of sample sizes: 69632, 69631, 69631, 69631, 69631, 69631, ...
## Resampling results across tuning parameters:
##
##   cp            Accuracy   Kappa
##   0.0003612333  0.7974615  0.01595638
##   0.0003824823  0.7973762  0.01128389
##   0.0003888570  0.7973775  0.01065938
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was cp = 0.0003612333.
```

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

Based on everything, I think this data has severe limitations that affect the usefulness and accuracy of our results. The fact that there are both walk-in and booked appointments in this dataset add a challenge for our models, as all predictors may not be applicable for both types of appointments (ex. SMS reminder). The models perform decently given the quality of the data, but I do not believe these analyses can be trusted at face value. It is important to interpret these models with caution, and clearly outline the limitations that we have discussed.

Credits

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