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 is unique identifier for each patient therefore they cannot be directly identified from their name etc. for ethical purposes.

AppointmentID is a unique identifier to each appointment for confidentiality, as described above

Gender indicates the patient's gender, which is Male or Female

ScheduledDate is the date when the appointment was scheduled

AppointmentDate is the date of the actual appointment when the individual comes in

Age is the Patient's age (in years)

Neighbourhood is the District of Vitória where the appointment takes place (Southeastern coast of Brazil)

SocialWelfare is if the Patient is a recipient of Bolsa Família welfare payments (payments from the social welfare of the government of Brazil)

Hypertension is if the Patient was previously diagnoised with hypertension before appointment (1 yes/ 0 no)

Diabetes is if the Patient was previously diagnosed with diabetes before appointment (1 yes/ 0 no)

AlcoholUseDisorder is if the Patient was previously diagnosed with alcohol use disorder before appointment (1 yes/ 0 no)

Disability is if the Patient was previously diagnosed with a disability before the appointment (severity rated 0-4)

SMSReceived is at least 1 reminder text sent before appointment (1 yes/ 0 no)

NoShow is if the Patient did not show up scheduled appointment (Yes/No)

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

1. Low socioeconomic status (may not have transportation or ability to get there)
2. Older Age (may forget about appointment or may not be able to physically get there without assistance)

3. Mental health and addiction issues (may not want to go there (embarassment/stigma surrounding addictions) or may not have resourses/transportation to get there)

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?
1. Distance from place of residence to appointment site (could be a major factor to predicting no shows, as the more time individuals have to spend to get to their appointment, maybe the less likely they are to attend it)

2. Number of admissions to hospital (this would be helpful to know if patients were admitted to the hospital following an appointment and could be the reason for many appointments if they have a disease or chronic condition)

3. Type of disability (this could be helpful for predicting no shows and could be a proxy to determine if individuals need assistance getting to their appointments)

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/allyseamone/AppliedHealthDataScience/mai
```

```
## `curl` package not installed, falling back to using `url()`
```

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)`

While there are two people that are 115 years old, we cannot actually tell if this is an impossible age. One of the individuals does have hypertension, which can be managed using medication, but it may not be biologically plausible for an individual to live to 115 years while having hypertension. It also says that the individual did receive a text message reminder, in this day and age a lot of individuals do have cell phones; an individual that is 115 years old may not have a cell phone. This age may be an imputation error or it could be correct, but these should be considered with caution as they are borderline biologically implausible. These individuals will be kept in the data set but when interpreting results, there should be considerations for these individuals. Since there are 110 527 observations, these 2 individuals should not skew the data or analyses.

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?

The maximum number of appointments from the same patient (822145925426128) is 88.They did not show up to a lot of their appointments though.

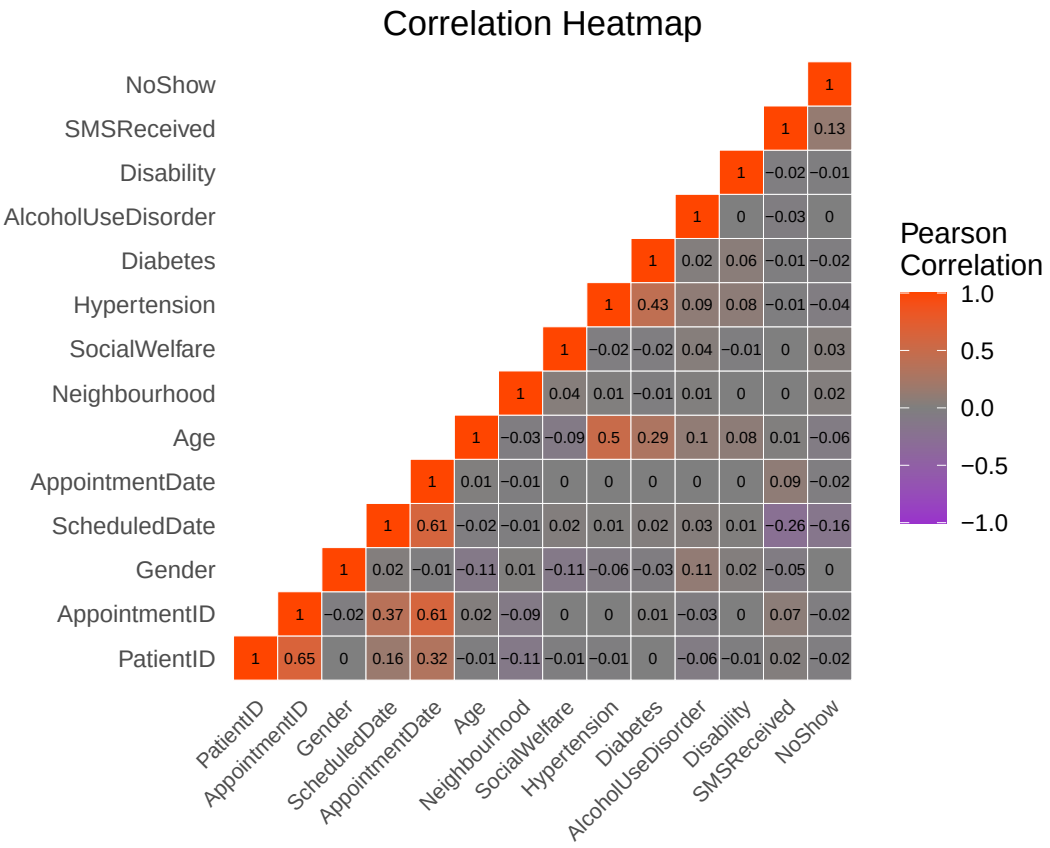
```
count(raw.data, PatientID, sort = TRUE)
```

```
## # 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
```

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`)?

SMSRecieved and ScheduledDate are most strongly correlated with NoShow, even though these are not above 0.5 or below -0.5 they have the highest values on the column. The strongly correlated is based on the legend on the heat map.

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

Consider anything above 0.5 or below -0.5 strongly correlated based on the Pearson Correlation scale to the left of the heat map.

AppointmentID and AppointmentDate are strongly correlated. PatientID and AppointmentID are strongly correlated. Age and Hypertension are strongly correlated.

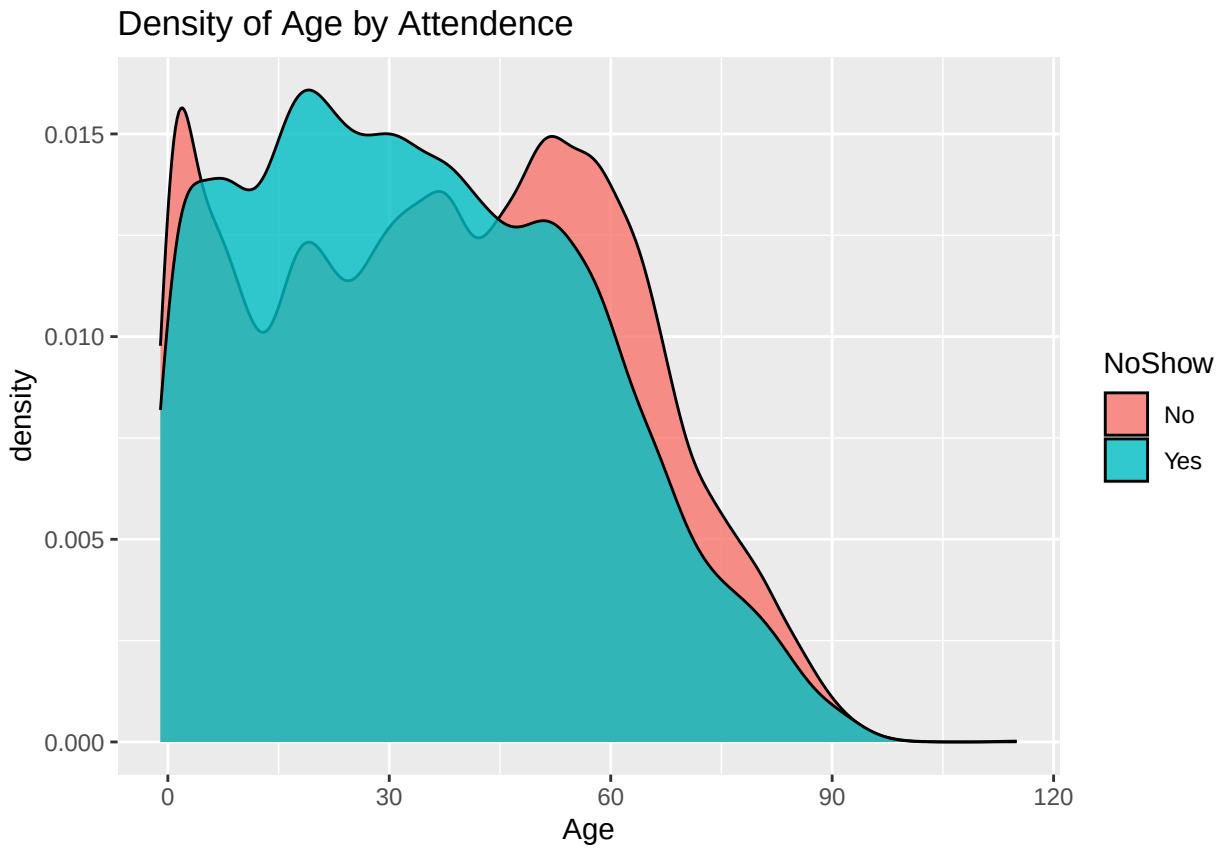
9 Do you see any issues with PatientID/AppointmentID being included in this plot?

Yes, there is an issue with PatientID/AppointmentID being included in this plot; they are variables that are supposed to be random identifiers so they should not have any correlation.

For example a singular patient (PatientID) could have more than 1 appointment, which would include more than 1 AppointmentID. Therefore AppointmentID could only ever have one PatientID associated to it, but a PatientID could have more than one AppointmentID associated with it.

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

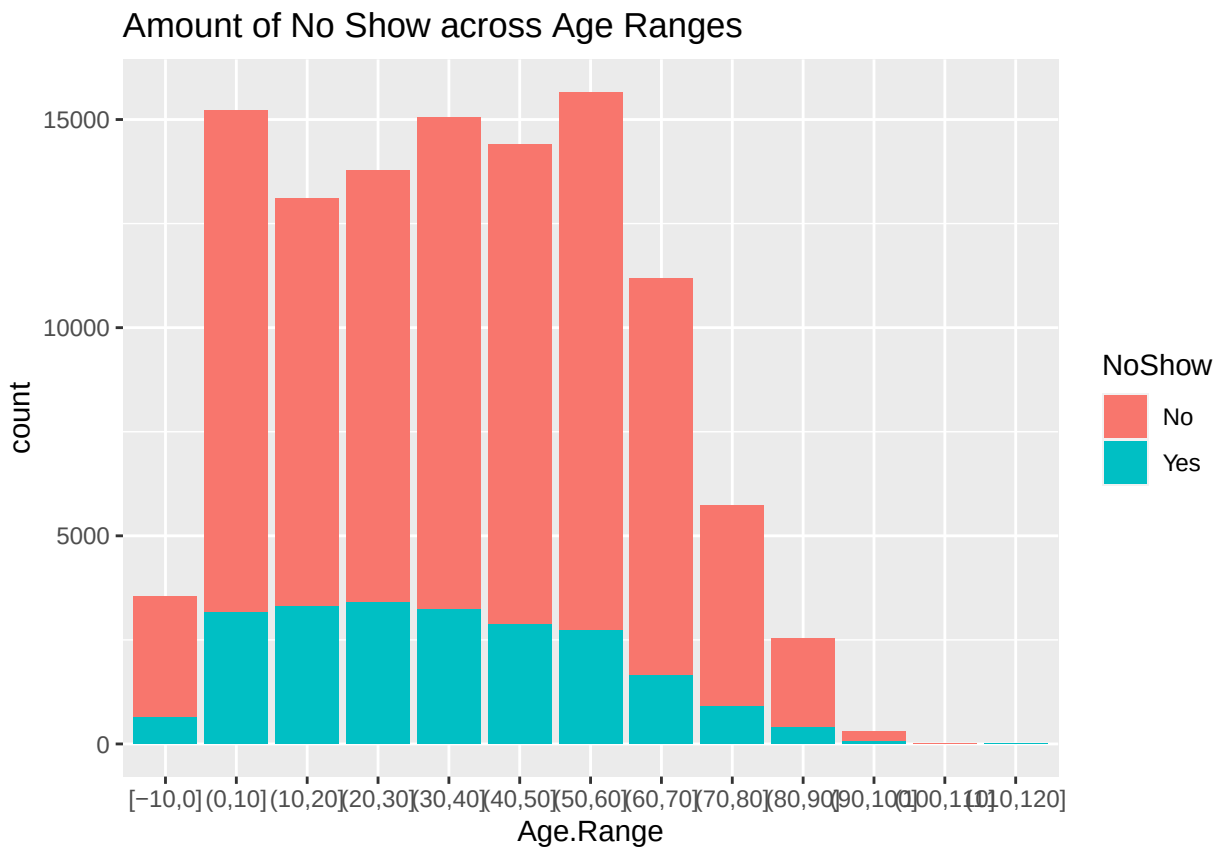
```
ggplot(raw.data) +
  geom_density(aes(x=Age, fill=NoShow), alpha=0.8) +
  ggtitle("Density of Age by Attendence")
```



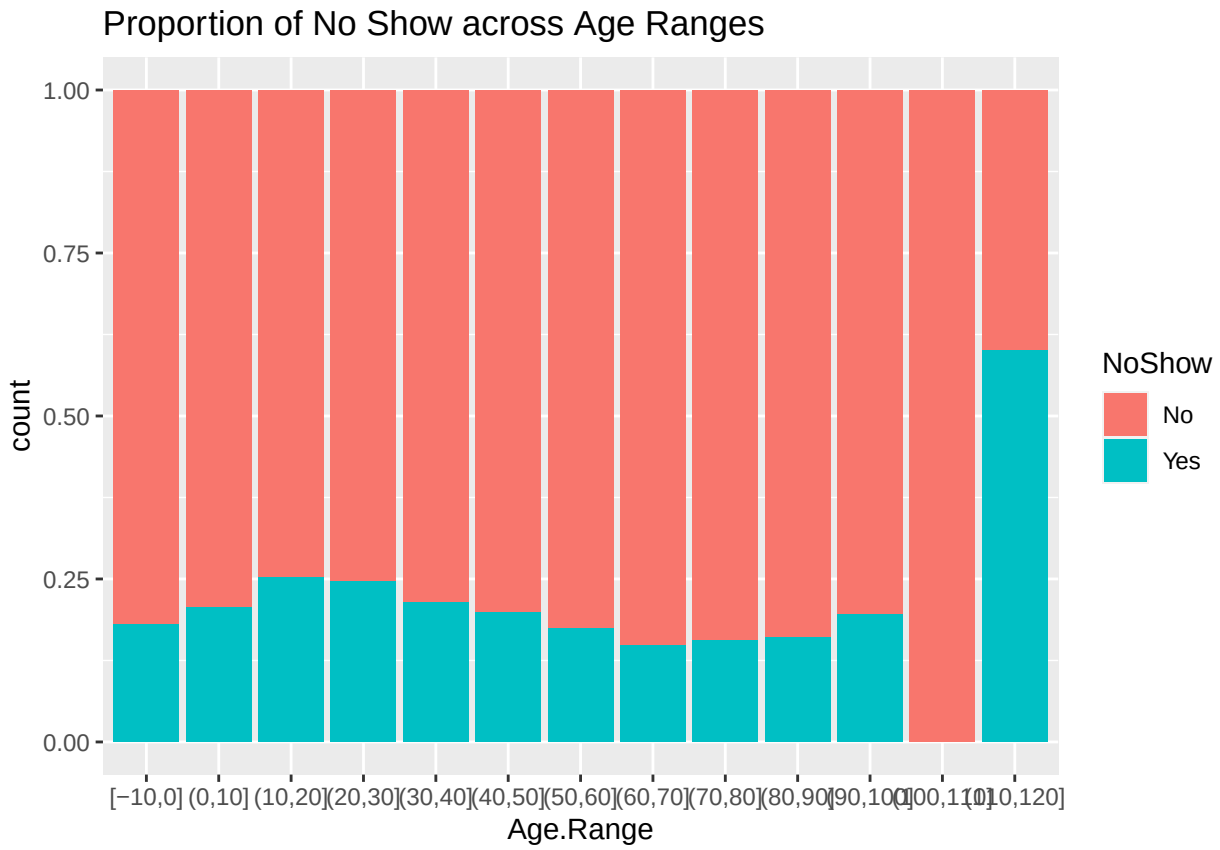
There does seem to be a difference in the distribution of ages of people that miss and don't miss appointments. However, the shape of this distribution means the actual correlation is near 0 in the heatmap above. This highlights the need to look at individual variables.

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?

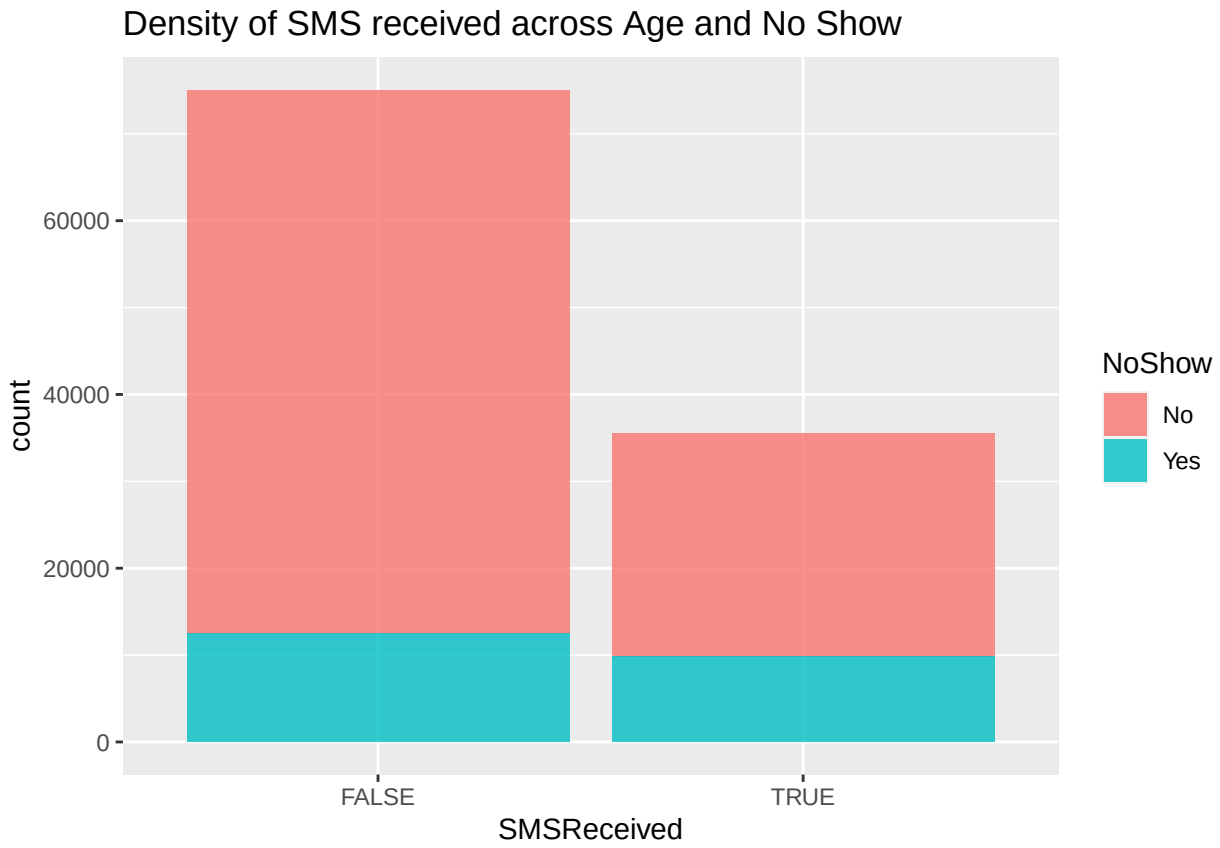
You could be misled if you only plotted 1 of these 2 plots of attendance by age group because 1 plot is showing counts and one plot is showing proportions. In plot 1, it looks as if individuals aged 0 to 30 have the most no shows, and no shows decrease with age, since there are very few individuals over 90. In plot 2, it looks as if individuals aged 10 to 30 have the most no shows and decreases with age until 60, then increases rapidly as there is a smaller number of older individuals, but there is a larger proportion of no shows here. There are very few individuals older than 90 years old from plot 1, therefore this does not really impact the overall distributions. There are twice as many missed appointments in the 10-20 age group compared to the 60-70 age group. The zero age group has some no shows since the observations with missing age will be in that category.

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

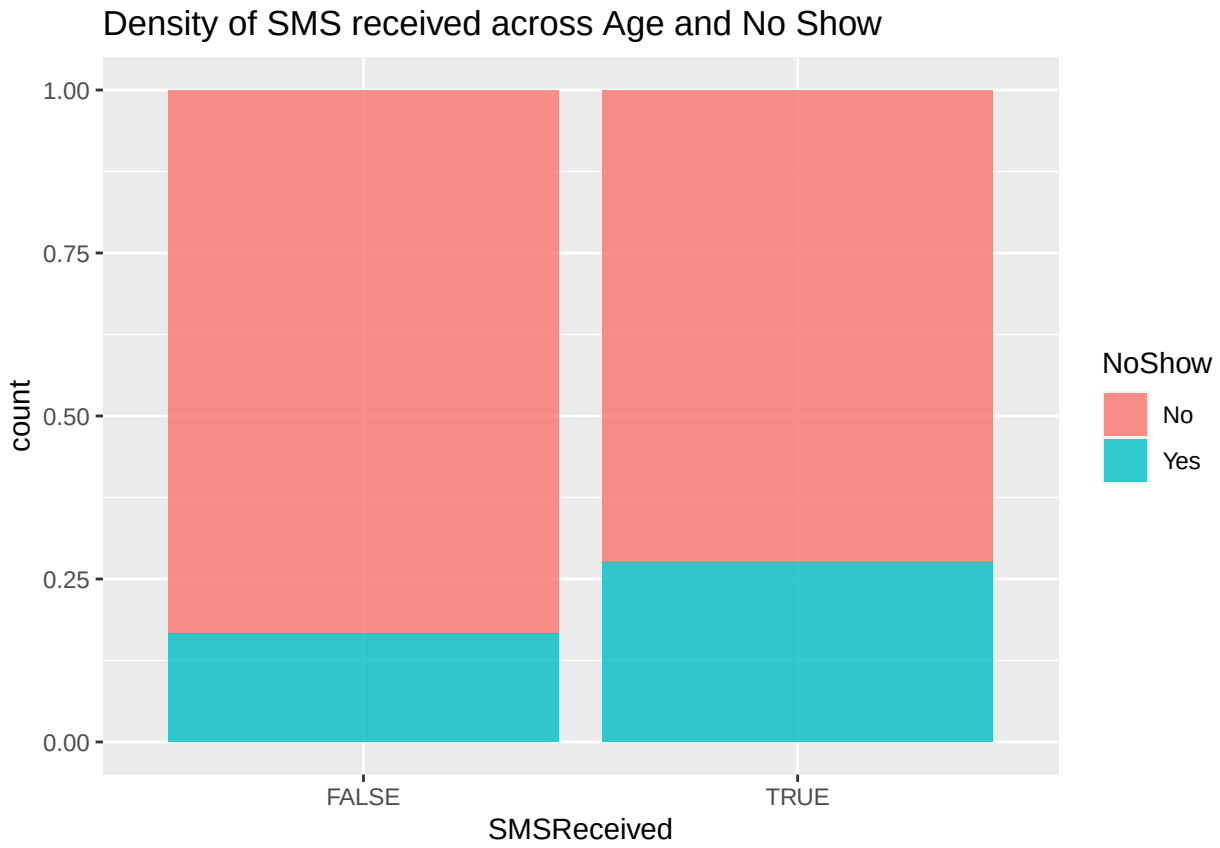
```
## # A tibble: 1 x 1
##       n
##   <int>
## 1  3539
```

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")
```



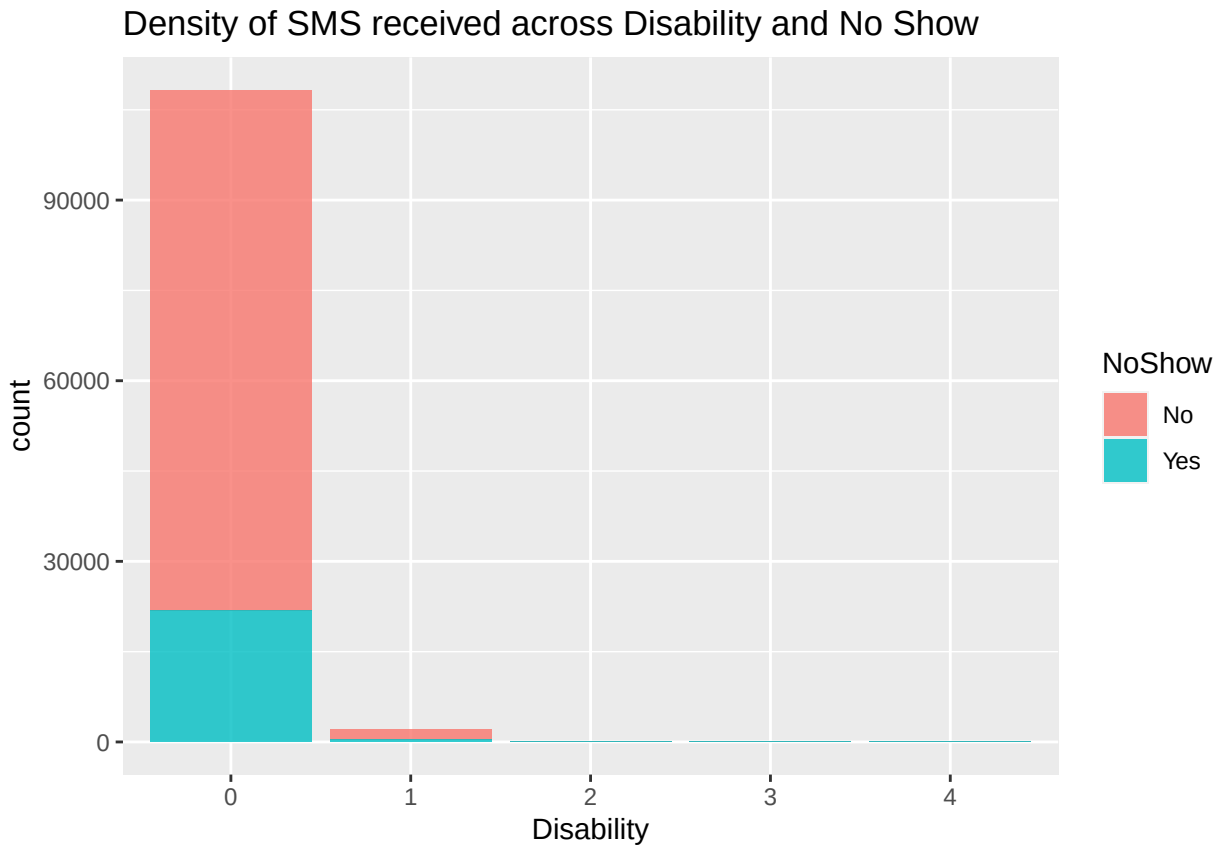
11 From this plot does it look like SMS reminders increase or decrease the chance of someone not attending an appointment? Why might the opposite actually be true (hint: think about biases)?

From the plot it does look like SMS reminders increase the chance of someone not attending an appointment. This could be the case because individuals who have same day or walk-in appointments may not be receiving a text message reminder, and they do show up. This would make the SMS Received FALSE. This could also be biased because older individuals may be less likely to have cell phones, and therefore may not be able to receive text messages, which would therefore make the SMS Received variable FALSE.

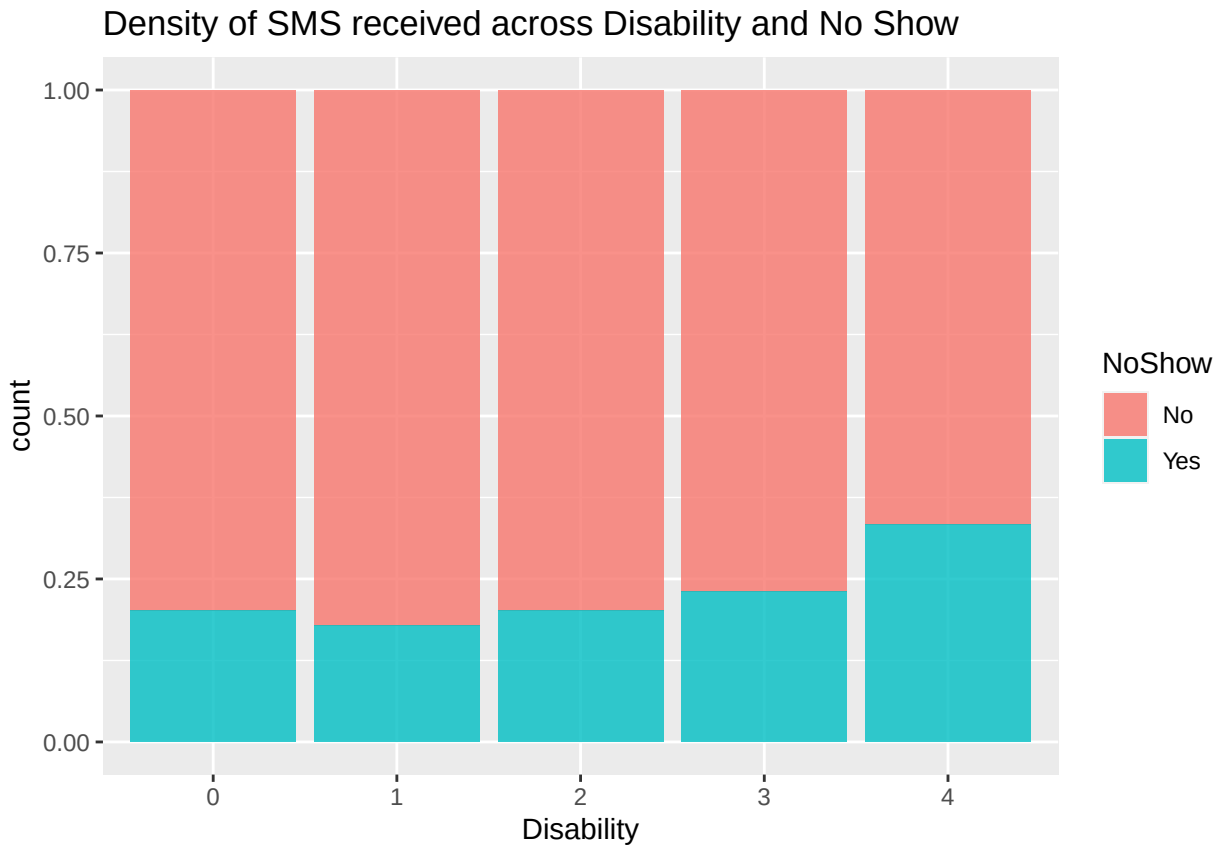
Therefore the individuals that do not received a SMS reminder may be more likely to show up to their appointment since it may be on the same day or they just do not have a cell phone. This is biased according to the plot and thats why the opposite (individuals that do not received a SMS reminder may be more likely to show up to their appointment) may be true.

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 SMS received across Disability and No Show")
```

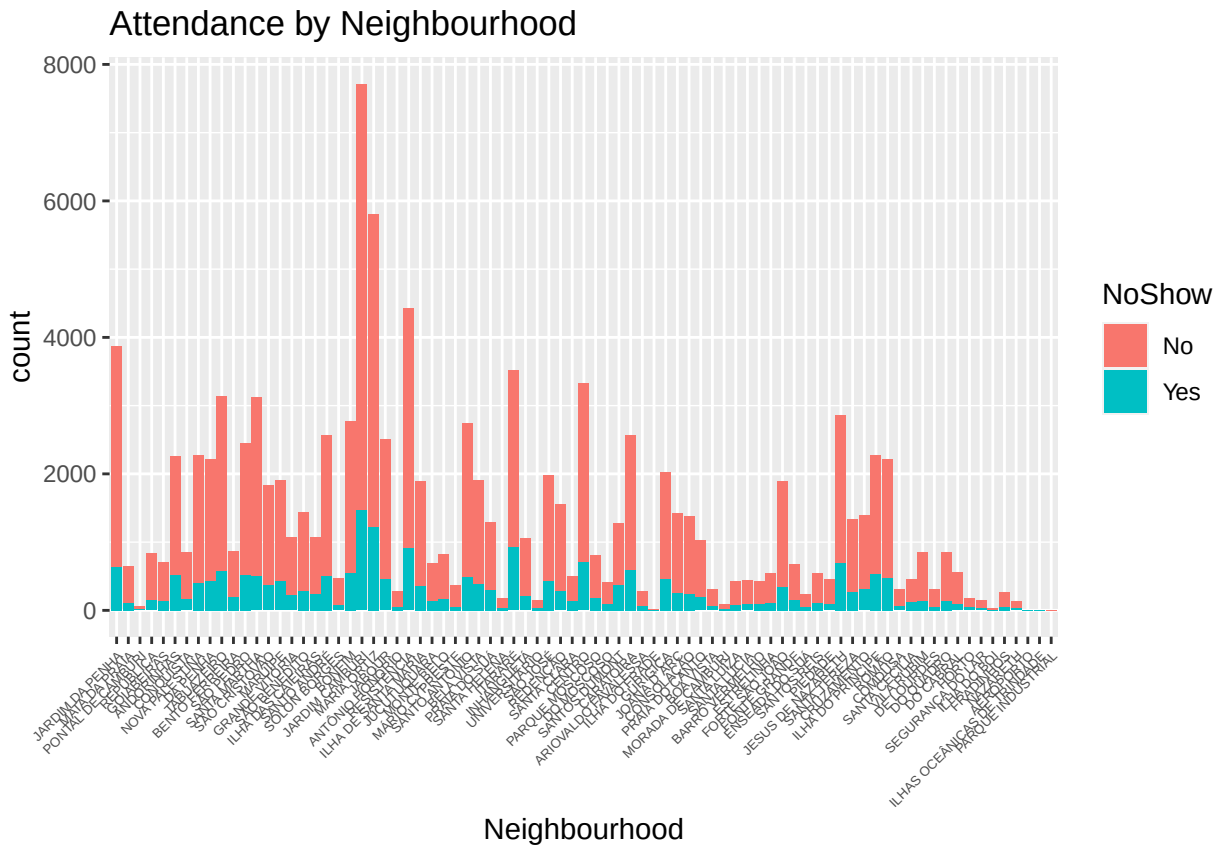



```
ggplot(raw.data) +
  geom_bar(aes(x=Disability, fill=NoShow), position='fill', alpha=0.8) +
  ggtitle("Density of SMS received across Disability and No Show")
```

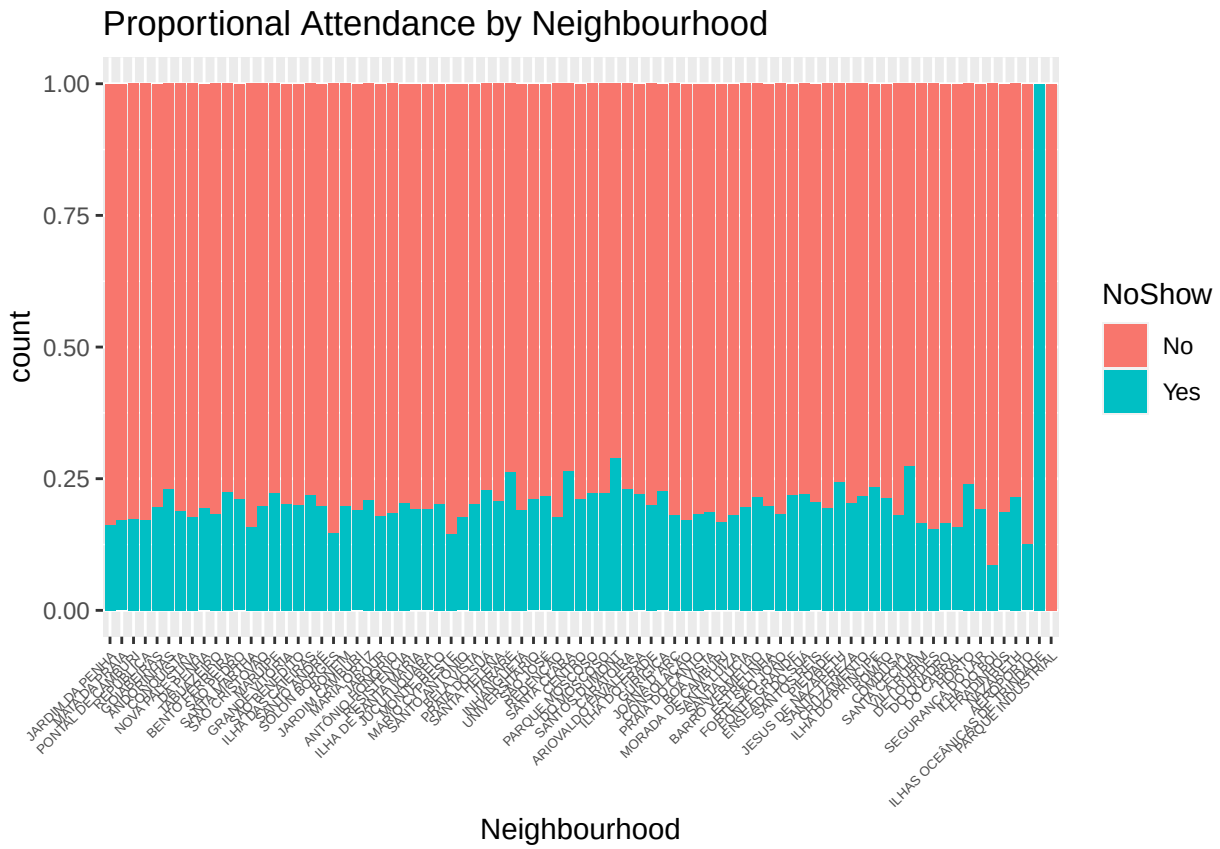


Now let's look at the neighbourhood data as location can correlate highly with many social determinants of health.

```
ggplot(raw.data) +
  geom_bar(aes(x=Neighbourhood, fill=NoShow)) +
  theme(axis.text.x = element_text(angle=45, hjust=1, size=5)) +
  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=5)) +
  ggtitle('Proportional Attendance by Neighbourhood')
```



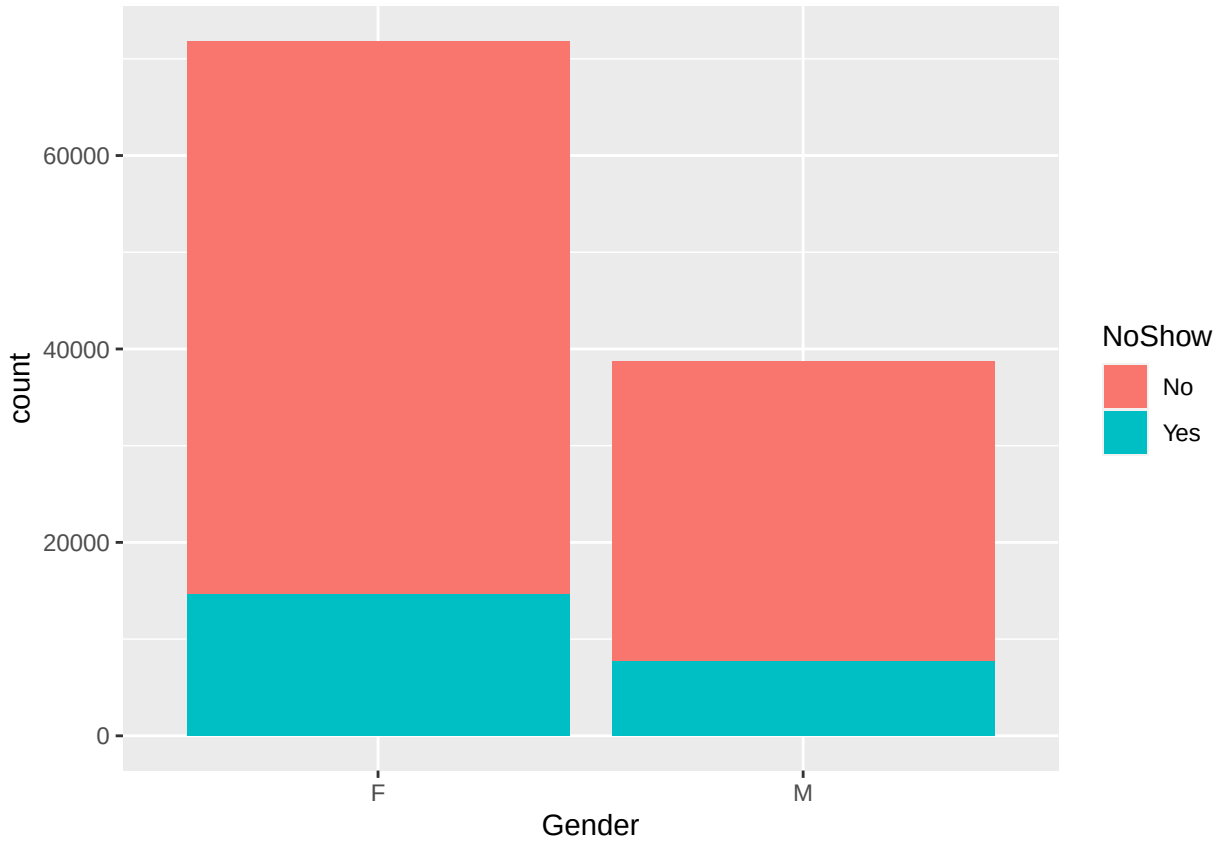
Most neighborhoods have similar proportions of no-show but some have much higher and lower rates.

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

A possible reason for differences in attendance rates across neighborhoods could be differences in socioeconomic status or distance from appointment site. If individuals are from a certain low SES neighborhood then they may not have access to a car and this could be a barrier to overcome when getting to their appointment. Individuals that live in a certain neighborhood that is really far away from the appointment site may be less likely to get to their appointment due to the time it takes to get their (due to traffic, etc.).

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

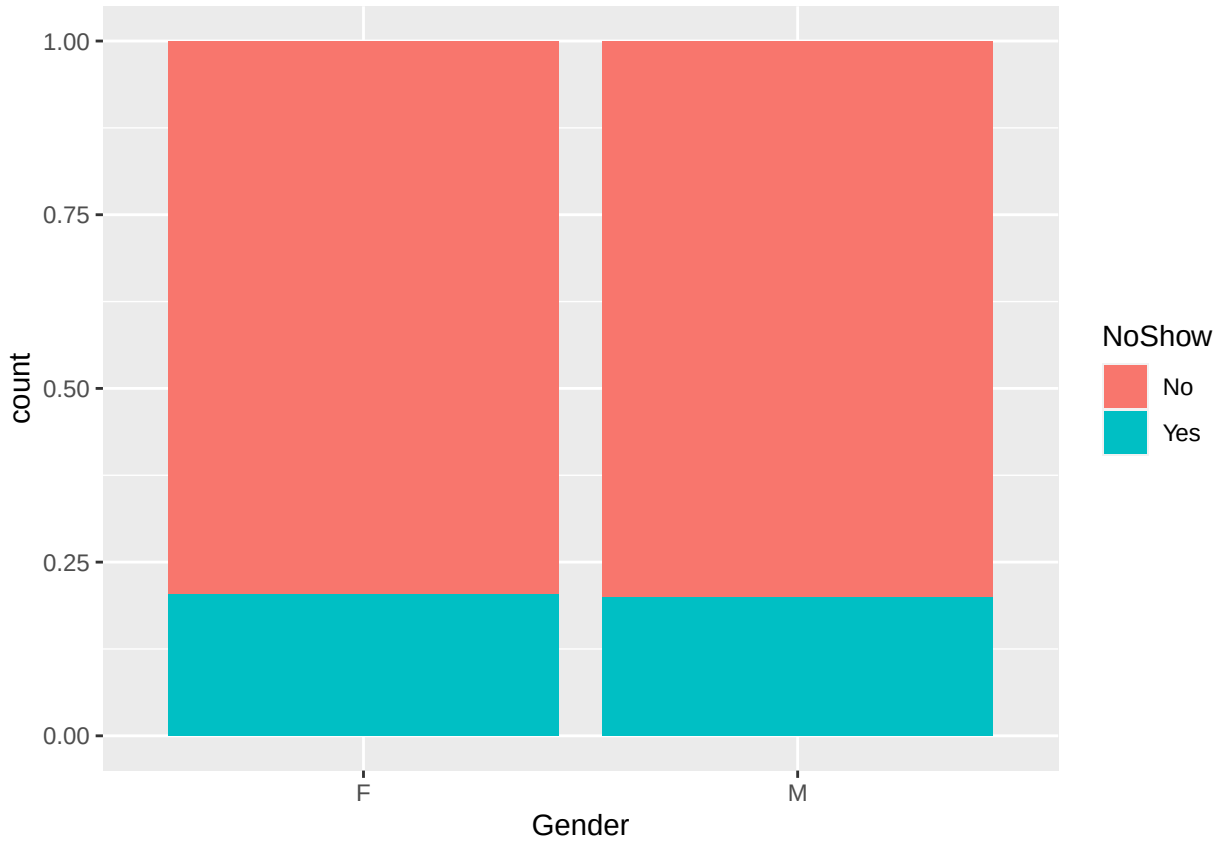
```
ggplot(raw.data) +
  geom_bar(aes(x=Gender, fill=NoShow))
```




```
  ggtitle("Gender by attendance")
```

```
## $title
## [1] "Gender by attendance"
##
## attr(,"class")
## [1] "labels"
```

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

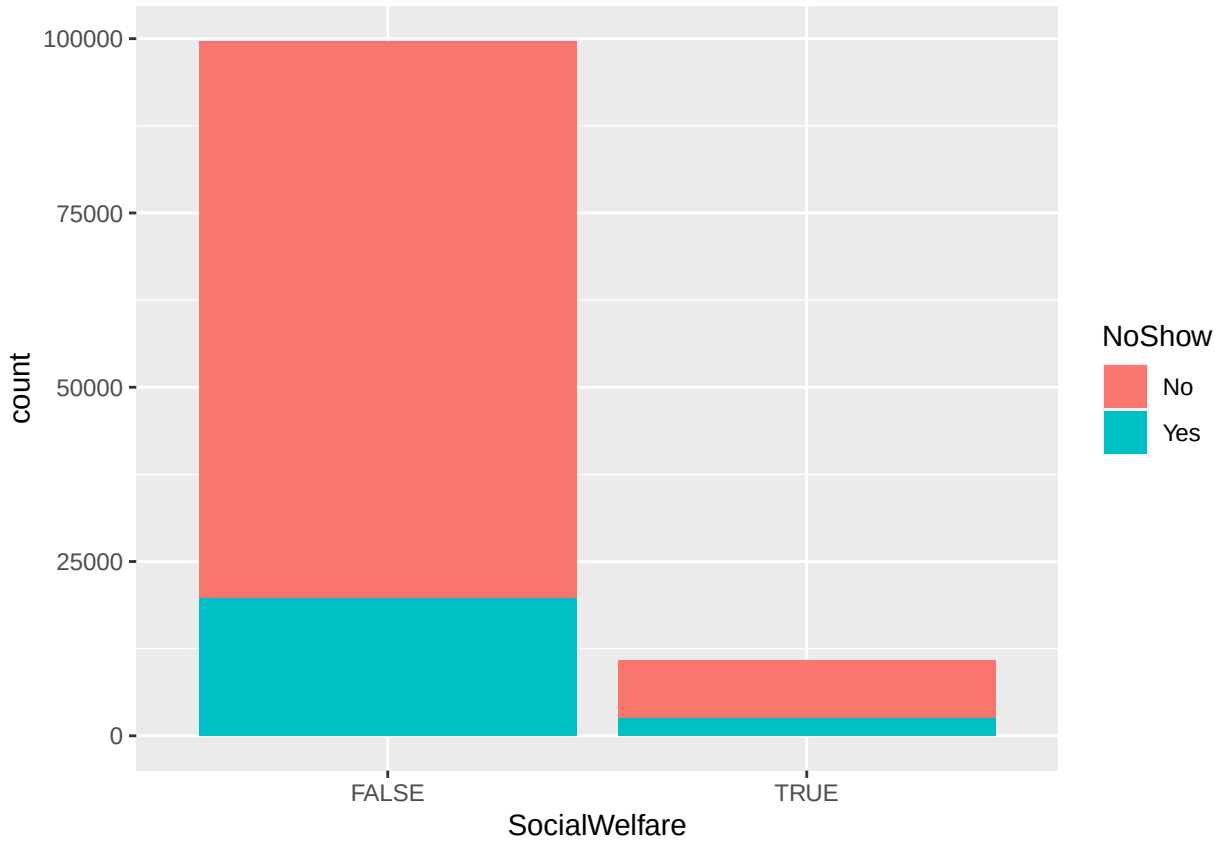



```
  ggtitle("Gender by attendance")
```

```
## $title
## [1] "Gender by attendance"
##
## attr(,"class")
## [1] "labels"
```

14 Create a similar plot using `SocialWelfare`

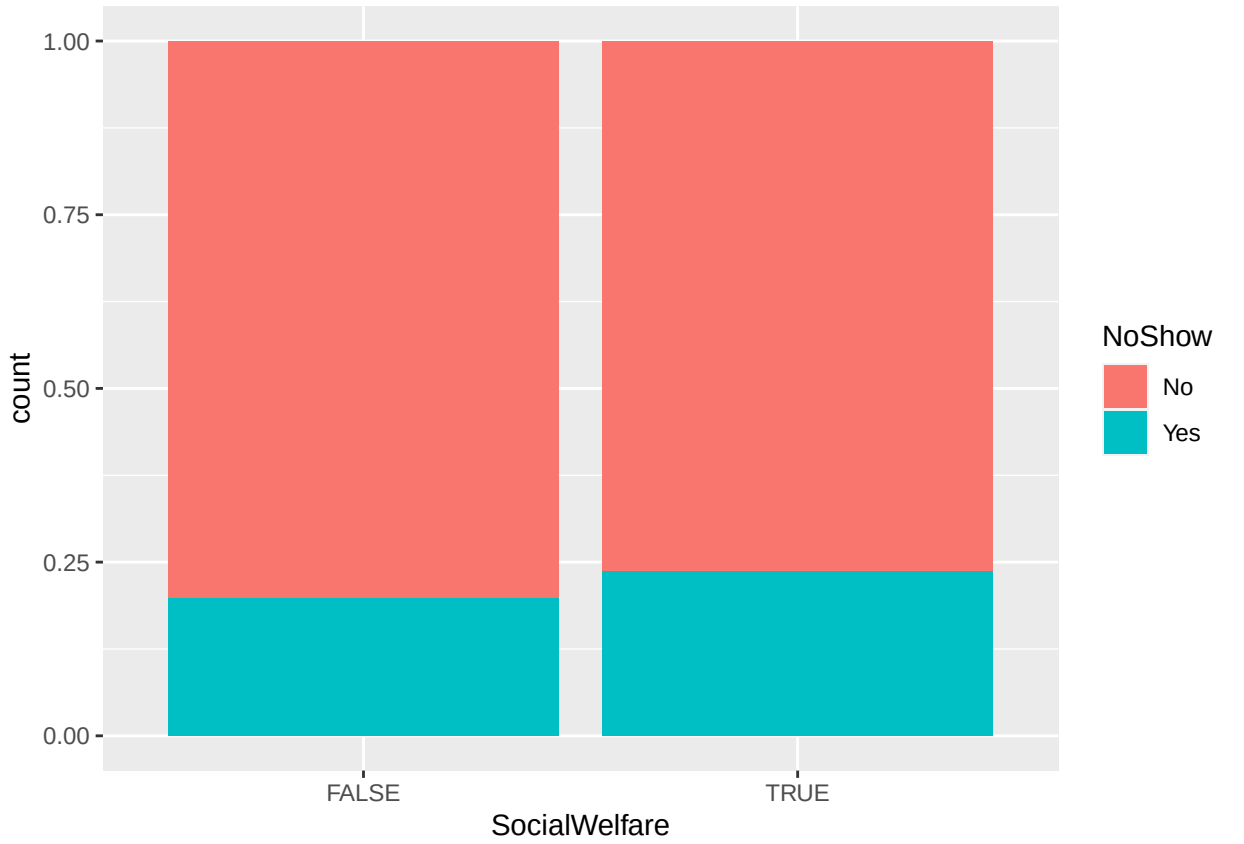
```
ggplot(raw.data) +
  geom_bar(aes(x=SocialWelfare, fill=NoShow))
```




```
  ggtitle("Social Welfare by attendance")
```

```
## $title
## [1] "Social Welfare by attendance"
##
## attr(,"class")
## [1] "labels"
```

```
ggplot(raw.data) +
  geom_bar(aes(x=SocialWelfare, fill=NoShow), position='fill')
```




```
  ggtitle("Social Welfare by attendance")
```

```
## $title
## [1] "Social Welfare by attendance"
##
## attr(,"class")
## [1] "labels"
```

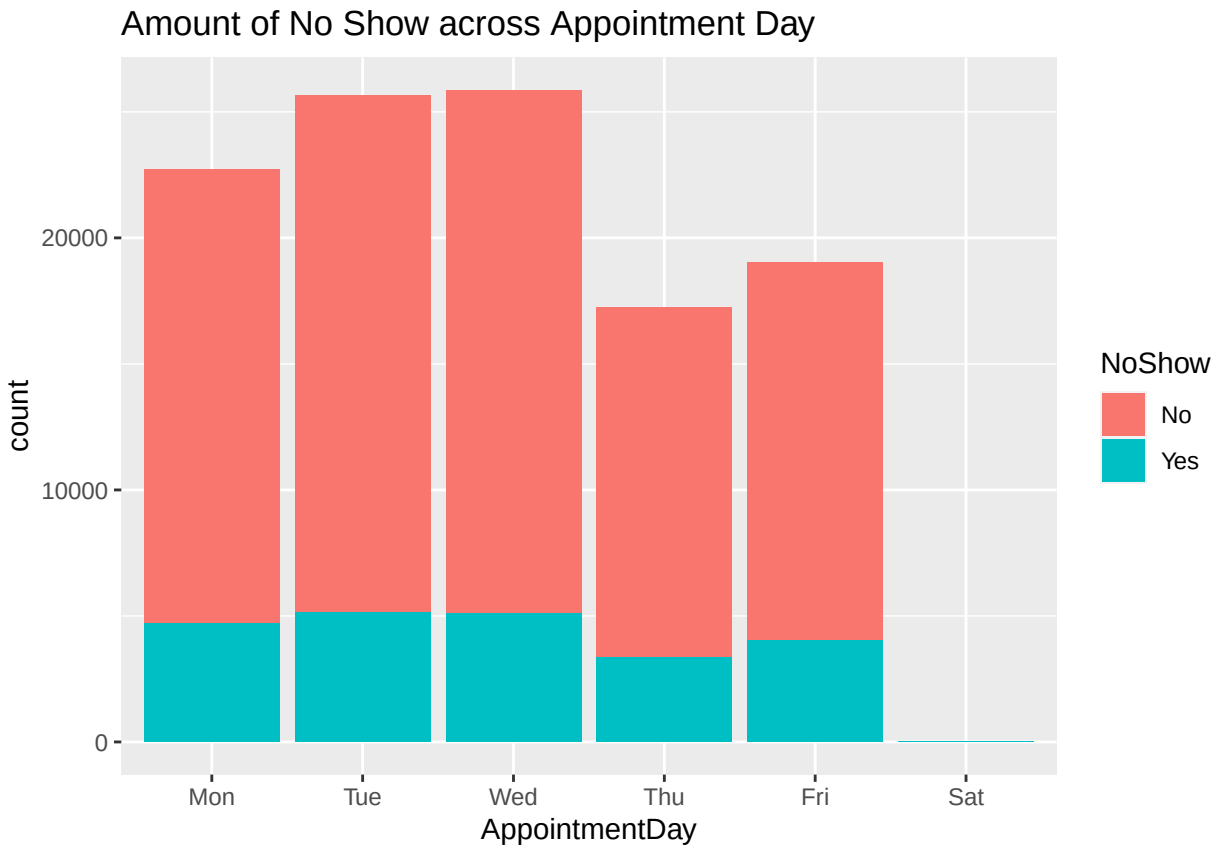
Far more exploration could still be done, including dimensionality reduction approaches but although we have found some patterns there is no major/striking patterns on the data as it currently stands.

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

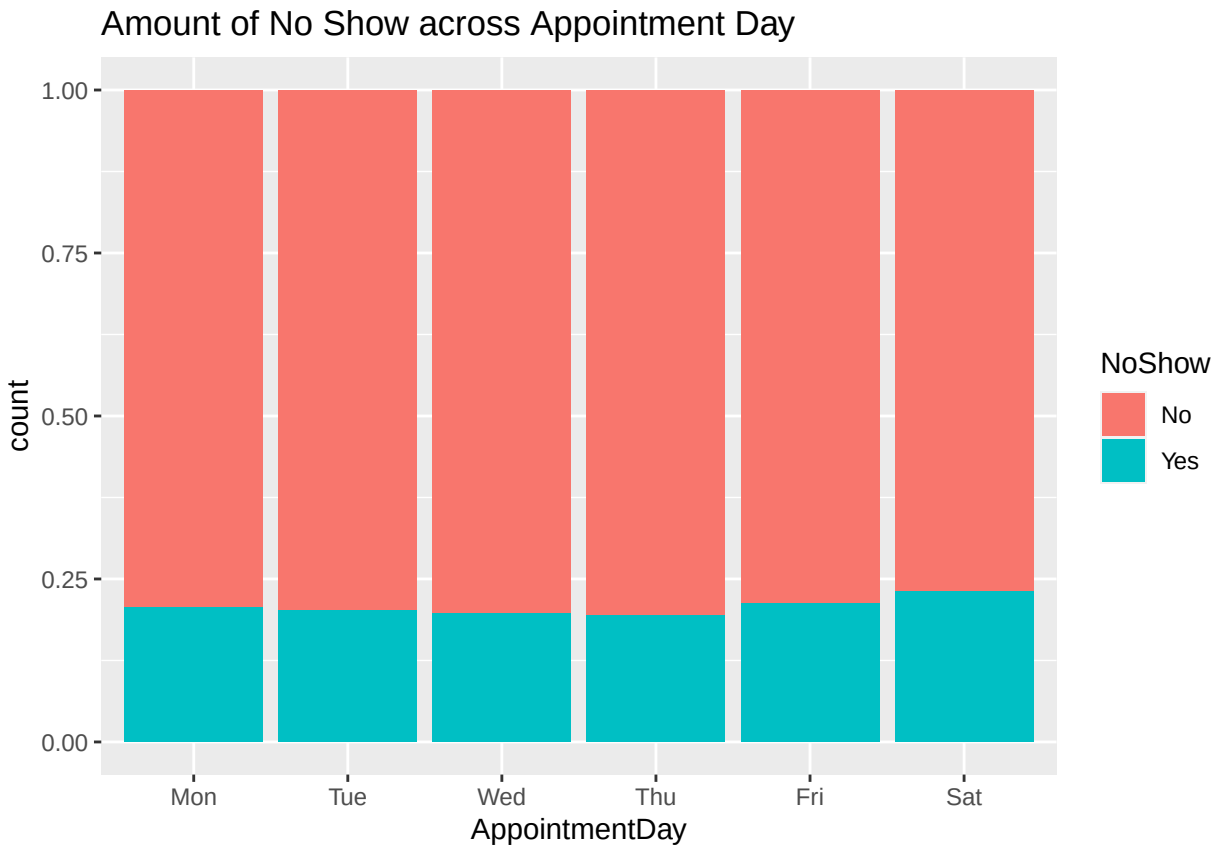
Feature Engineering

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")
```



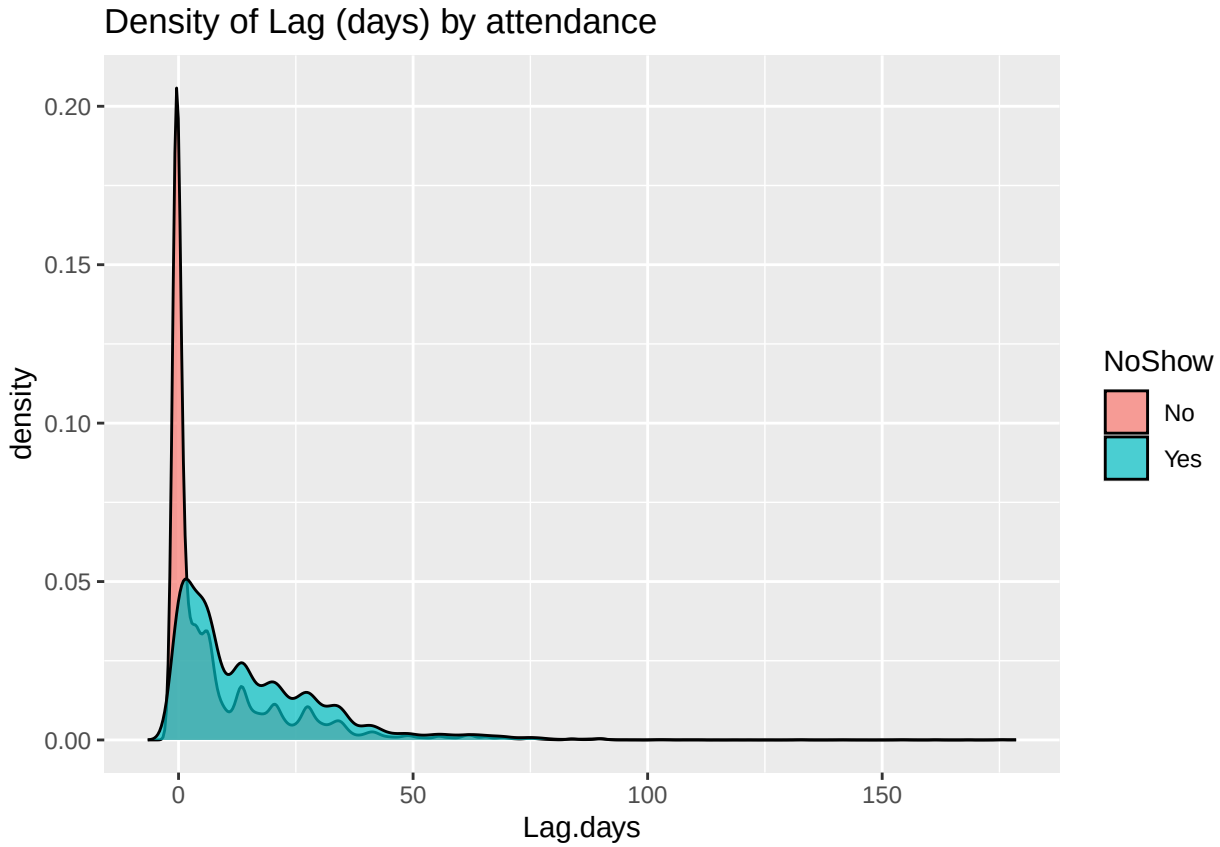

```
ggplot(raw.data) +
  geom_bar(aes(x=AppointmentDay, fill=NoShow), position = 'fill') +
  ggtitle("Amount of No Show across Appointment Day")
```



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?

In the graph, it does show that there are lag days that are less than zero and there is a huge number of lag days that are zero. Negative lag days means that the scheduled date was a later date than when the appointment date was. This could be an error when entering the dates and interpreting the days and months in reverse. When considering the lag days that are zero, this could be appointments that are scheduled (or individuals just show up) and occur on the same day; which could be walk in appointments. When interpreting models, it will be necessary to distinguish between walk in and scheduled appointments. Since these two types of appointments are included here, it may be difficult to develop a good prediction model.

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)

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 exploratory data analysis, there have been some issues identified. There are some important variables missing from the data set when predicting no shows such as: distance from site, a more direct identifier of income, and type of disability. The variables in the data set may do a decent job of predicting no shows, but there could be more options that could be better predictors of whether an individual will not show up to their appointment. Another issue is that there are a few outliers in the data set that are hard to distinguish if they are biologically plausible or not. Keeping these individuals in the data set may skew the results very slightly, but since there are 110 527 observations this should not be an issue, but should be considered. There is a bias that was associated with SMS reminders that could impact how well the prediction model will work. Prediction may also be difficult since walk-in and scheduled appointments are being consdiered here, based on the lag time variable. Therefore, based on the issues mentioned above, this prediction model may not predict no shows very well.

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="")
```

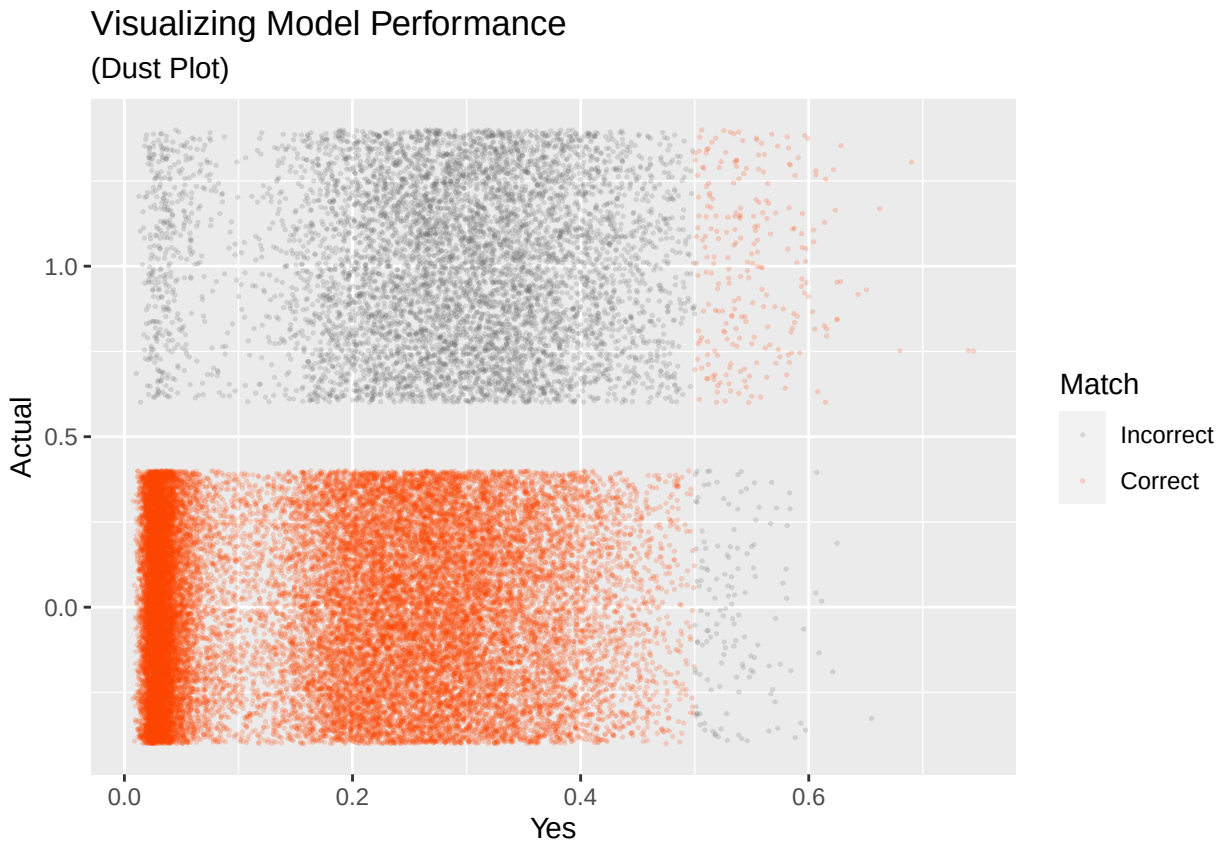
```
## Setting levels: control = 0, case = 1
```

```
## 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,

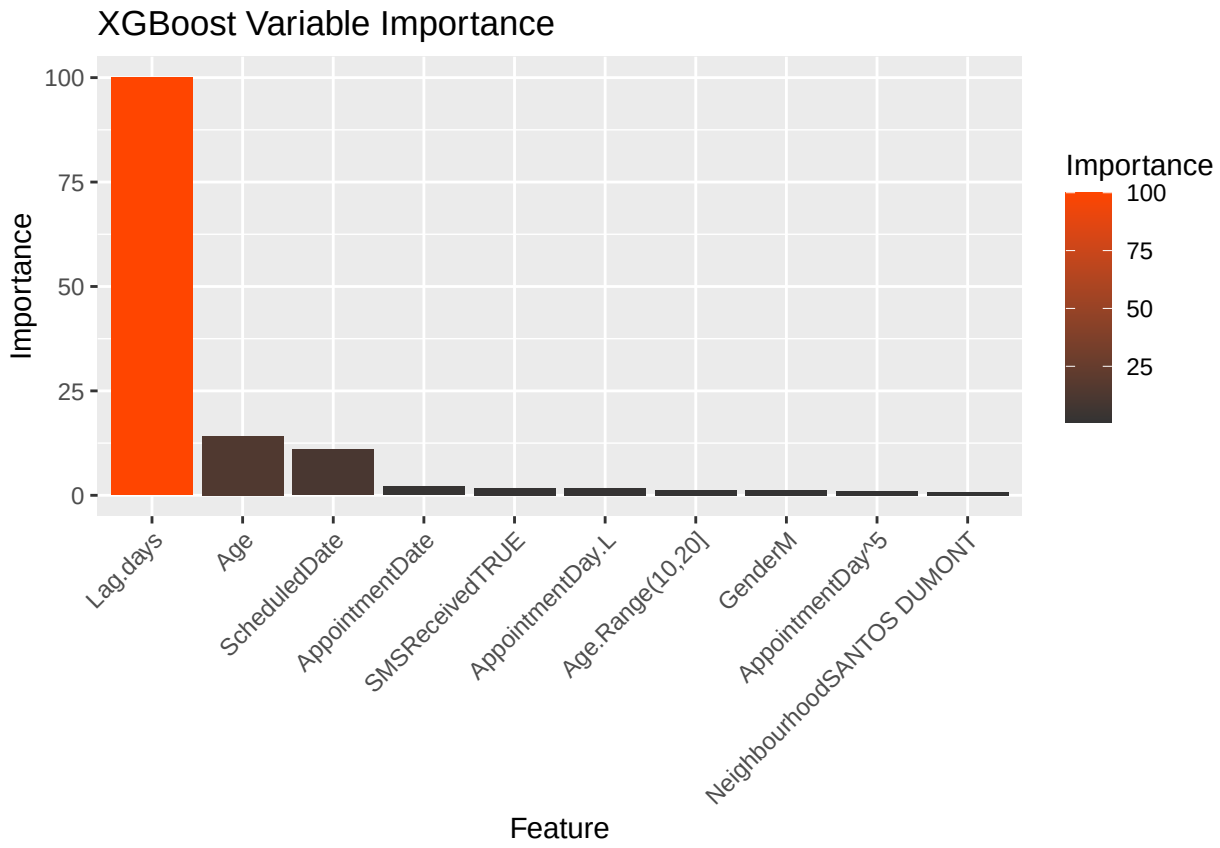
```
xgb.probs$Actual = test$NoShow.numerical
xgb.probs$ActualClass = test$NoShow
xgb.probs$PredictedClass = xgb.pred
xgb.probs$Match = ifelse(xgb.probs$ActualClass == xgb.probs$PredictedClass,
                         "Correct","Incorrect")
# [4.8] Plot Accuracy
xgb.probs$Match = factor(xgb.probs$Match,levels=c("Incorrect","Correct"))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

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

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

```
fit.control <- trainControl(method="repeatedcv", number=6,repeats=6)
```

```
crt.tree <- train(NoShow ~., data=train, method="rpart", trControl=fit.control)
```

```
crt.tree
```

```
## CART
##
## 77368 samples
##    16 predictor
##     2 classes: 'No', 'Yes'
```

```
##
## No pre-processing
## Resampling: Cross-Validated (6 fold, repeated 6 times)
## Summary of sample sizes: 64474, 64474, 64473, 64473, 64473, 64473, ...
## Resampling results across tuning parameters:
##
##   cp            Accuracy   Kappa
##   0.0003612333  0.7974119  0.018698452
##   0.0003824823  0.7973064  0.010475966
##   0.0003888570  0.7972978  0.009649892
##
## 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 previous exploratory data analysis and the prediction models developed above, I would say that these analyses should be considered with caution. Since there were biases with the SMS received variable, further variables that may be better predictors (mentioned in question 16), and other limitations explained in the lag time variable question (including walk-in and scheduled appointment).

The two prediction models performed decently with an accuracy value around 0.8, however the first prediction model had a very low sensitivity and a very high specificity. When creating a good prediction model, a balance between the two (sensitivity and specificity) is more important than having a very high one and a very low one.

With the issues and statistics described above, I would say that we cannot fully trust analyses based on this data set, but interpretations should consider and describe the limitations. Interpretations of the models should consider the different types of appointments included here, based on the lag time variable there were walk-in appointments and scheduled appointments in the dataset. Predicting no shows for walk-in appointments may be more difficult as there would only be time independent data used to predict this, and this would be different for scheduled appointments.

The main factors that make the data set trustworthy for analysis would be the large number of observations and number of variables included.

Credits

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