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.

Answer:

- `PatientID`: **PatientID identifies each patient uniquely, who have scheduled to book a medical appointment.**
- `AppointmentID`: **AppointmentID uniquely identifies each appointment. It could be based on combination of a type of medical appointment along with the PatientID.**
- `Gender`: **A categorical variable specifying the gender of the patient. Wherein, 'M' corresponds to male and 'F' corresponds to female.**
- `ScheduledDate`: **Specifies the date and time on which a medical appointment was scheduled by a particular patient.**
- `AppointmentDate`: **Specifies the date of the actual medical appointment.**
- `Age`: **Describes the patient's numerical age.**
- `Neighbourhood`: **Specifies the District of Vitória (the capital city of Espírito Santo, Brazil) in which the medical appointment corresponding to the patient is scheduled.**
- `SocialWelfare`: **Specifies if the patient is a recipient of Bolsa Família welfare payments. Wherein, '0' corresponds to *not a recipient* and '1' corresponds to *is a recipient.***
- `Hypertension`: **Indicates if the patient was previously diagnosed with hypertension. The value, '0' corresponds to *not diagnosed with hypertension* and '1' corresponds to *diagnosed with hypertension.***
- `Diabetes`: **Indicates if the patient was previously diagnosed with diabetes. The value, '0' corresponds to *not diagnosed with diabetes* and '1' corresponds to *diagnosed with diabetes.***

- **`AlcoholUseDisorder`: Indicates if the patient was previously diagnosed with alcohol use disorder. The value, '0' corresponds to *not diagnosed with alcohol use disorder* and '1' corresponds to *diagnosed with alcohol use disorder*.**
- **`Disability`: Categorically indicates if the patient was previously diagnosed with disability. The value (category) ranges based on a severity measure from 0 to 4.**
- **`SMSReceived`: Indicates if the patient received at least 1 reminder text message before appointment. The value, '0' corresponds to *SMS not received* and '1' corresponds to *SMS received*.**
- **`NoShow`: Indicates if the patient did not attend scheduled appointment. The value, 'Yes' corresponds to *appointment not attended* and 'No' corresponds to *appointment attended*.**

`NoShow` is a dependent variable which depends on 13 other independent variables from `PatientID` to `SMSReceived`.

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

Answer:

- **Occurrence of unforeseen circumstances, such as illness or another conflicting job.**
- **Higher chances of forgetting the appointment date or time.**
- **Ineligibility or expiry of government issued medical scheme(s).**

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?

Answer:

- **Time of appointment is not included along with the appointment date.**
- **Mode of appointment: Is the scheduled appointment phone-based, online, or in-person?**
- **Nature of appointment: Is the scheduled appointment a new appointment or followup of a previous appointment?**

Data Parsing and Cleaning

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

```
#Reading the dataset from Public GitHub Repository
raw.data <- read_csv('https://raw.githubusercontent.com/Sai97-Ravi/CSCI6410/Lab_1/2016_05v2_VitoriaAppo

#Printing thr dimensions (number of rows and columns in the data)
dim(raw.data)
```

```
## [1] 110527     14
```

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 > 100)
```

```
## # A tibble: 7 x 14
##   PatientID   AppointmentID Gender ScheduledDate       AppointmentDate       Age
##   <fct>       <fct>         <fct>  <dttm>              <dttm>              <int>
## 1 9762947997~ 5651757       F      2016-05-03 09:14:53 2016-05-03 00:00:00   102
## 2 3196321161~ 5700278       F      2016-05-16 09:17:44 2016-05-19 00:00:00   115
## 3 3196321161~ 5700279       F      2016-05-16 09:17:44 2016-05-19 00:00:00   115
## 4 3196321161~ 5562812       F      2016-04-08 14:29:17 2016-05-16 00:00:00   115
## 5 3196321161~ 5744037       F      2016-05-30 09:44:51 2016-05-30 00:00:00   115
## 6 2342835965~ 5751563       F      2016-05-31 10:19:49 2016-06-02 00:00:00   102
## 7 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 some patients older than 100 which seems suspicious but we can't actually say if this is impossible.

5 Are there any individuals with impossible ages?

Answer:

There is an occurrence of a patient with Patient ID: 465943158731293 and Appointment ID: 5775010 having an age of "-1". Thus, we drop this false record from the dataset. Our cleaned data contains patients' age which is equal to or above 0 years.

```
raw.data %>% filter(Age == -1)
```

```
## # A tibble: 1 x 14
##   PatientID   AppointmentID Gender ScheduledDate       AppointmentDate       Age
##   <fct>       <fct>         <fct>  <dttm>              <dttm>              <int>
## 1 4659431587~ 5775010       F      2016-06-06 08:58:13 2016-06-06 00:00:00    -1
## # i 8 more variables: Neighbourhood <fct>, SocialWelfare <lgl>,
## #   Hypertension <lgl>, Diabetes <lgl>, AlcoholUseDisorder <lgl>,
## #   Disability <fct>, SMSReceived <lgl>, NoShow <fct>
```

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

```
## [1] 110526     14
```

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?

Answer:

Patients are uniquely identified by PatientID. The following code fetches us the count of appointments booked by each patient (since there are no duplicate Appointment IDs). In order to know the maximum number of appointments from the same patient, the count of appointments are listed in descending order.

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

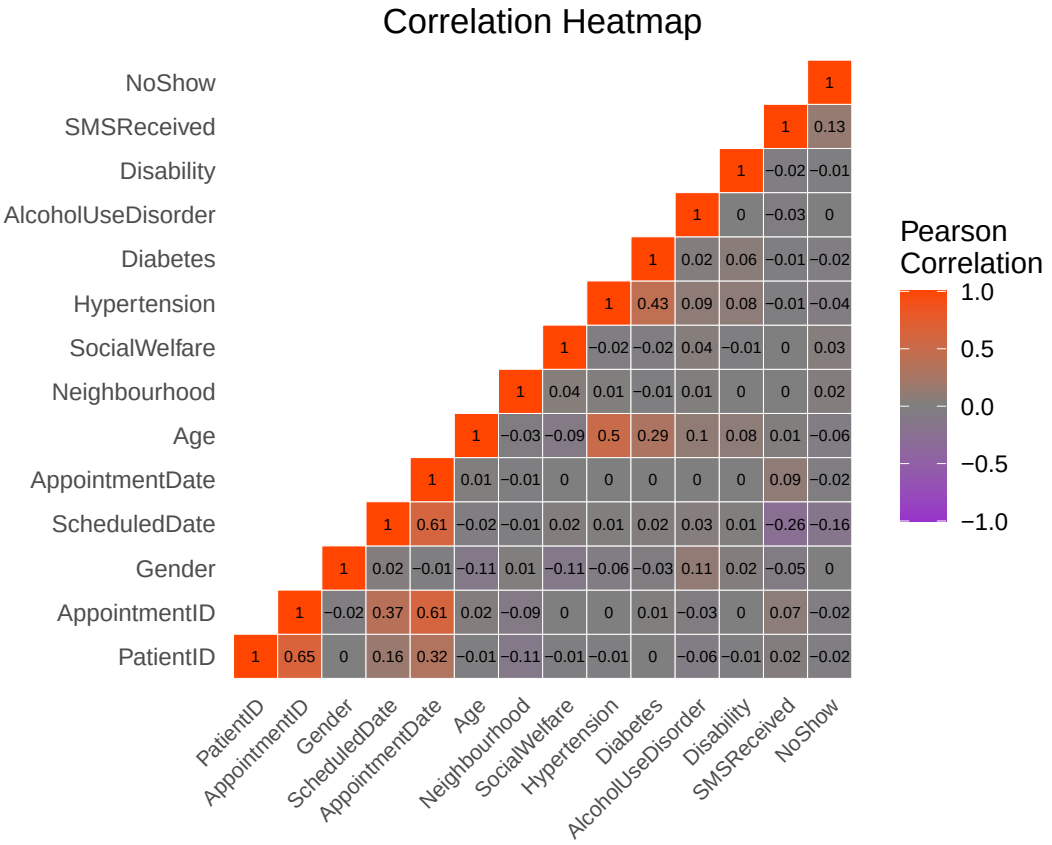
```
## # A tibble: 62,298 x 2
##    PatientID           n
##    <fct>           <int>
##  1 822145925426128    88
##  2 99637671331        84
##  3 26886125921145     70
##  4 33534783483176     65
##  5 258424392677       62
##  6 871374938638855    62
##  7 6264198675331      62
##  8 75797461494159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

We observe that the patient with PatientID: 822145925426128 has booked a total of 88 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`)?

Answer:

Out of all the variables, the `SMSReceived` variable slightly correlates with `NoShow`, although the correlation coefficient of `0.13` is not a strong correlation.

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

Answer:

There are no variables which strongly correlates with one another. For variables to be strongly correlated the correlation coefficient should be greater than `0.8 (for string positive correlation)` and `-0.8 (for strong negative correlation)` [1].

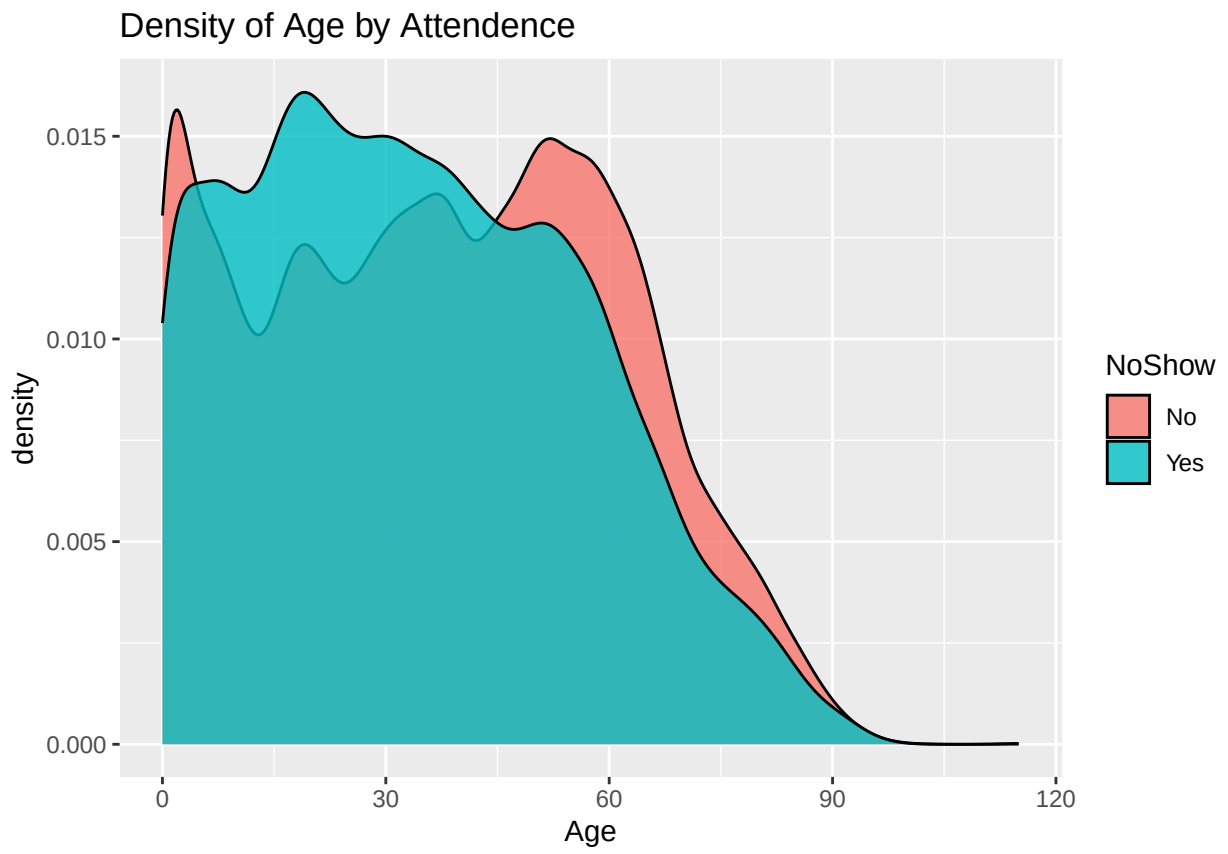
9 Do you see any issues with PatientID/AppointmentID being included in this plot?

Answer:

The variables, `AppointmentID` with `PatientID` are independent variables. The correlation between both of these variables is misleading.

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

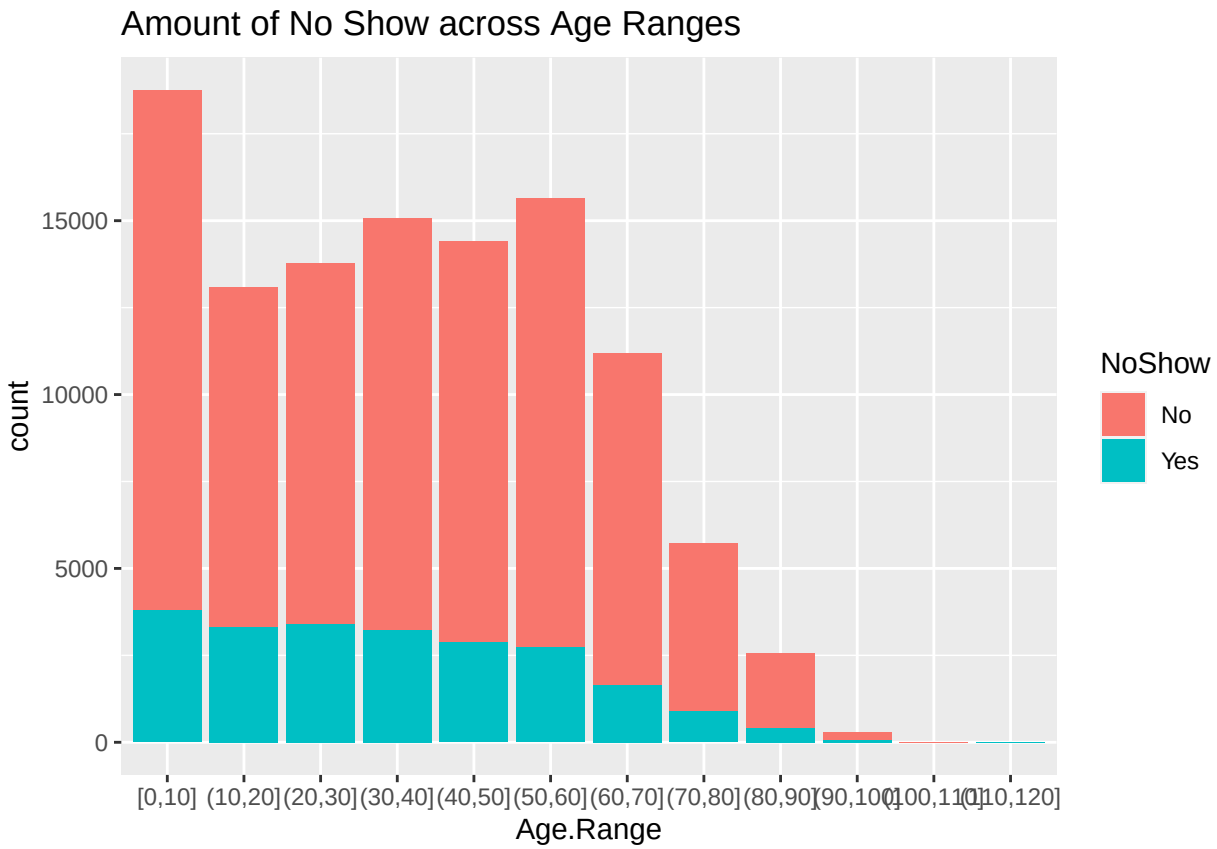
```
ggplot(raw.data) +
  geom_density(aes(x=Age, fill=NoShow), alpha=0.8) +
  ggtitle("Density of Age by Attendence")
```



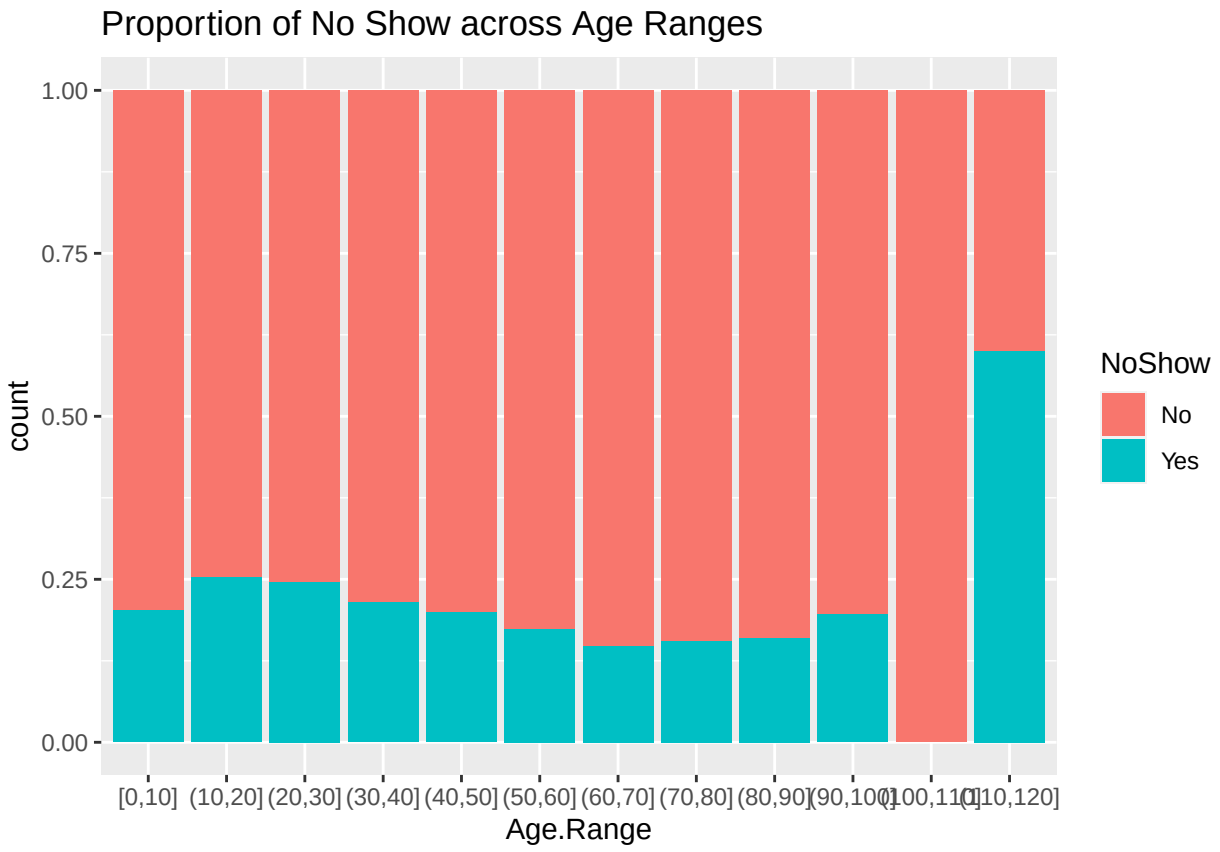
There does seem to be a difference in the distribution of ages of people that miss and don't miss appointments. However, the shape of this distribution means the actual correlation is near 0 in the heatmap above. This highlights the need to look at individual variables.

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?

Answer:

In the first plot, there is no count of patients within the ranges `100-110` and `110-120`. However, in the second plot, there exists proportion of patients within the age group `100-110`, taking up the full proportion of showing up. While patients in the age group `110-120`, about 60% of patients would not show up, and the remaining proportion will show up for their appointment.

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.

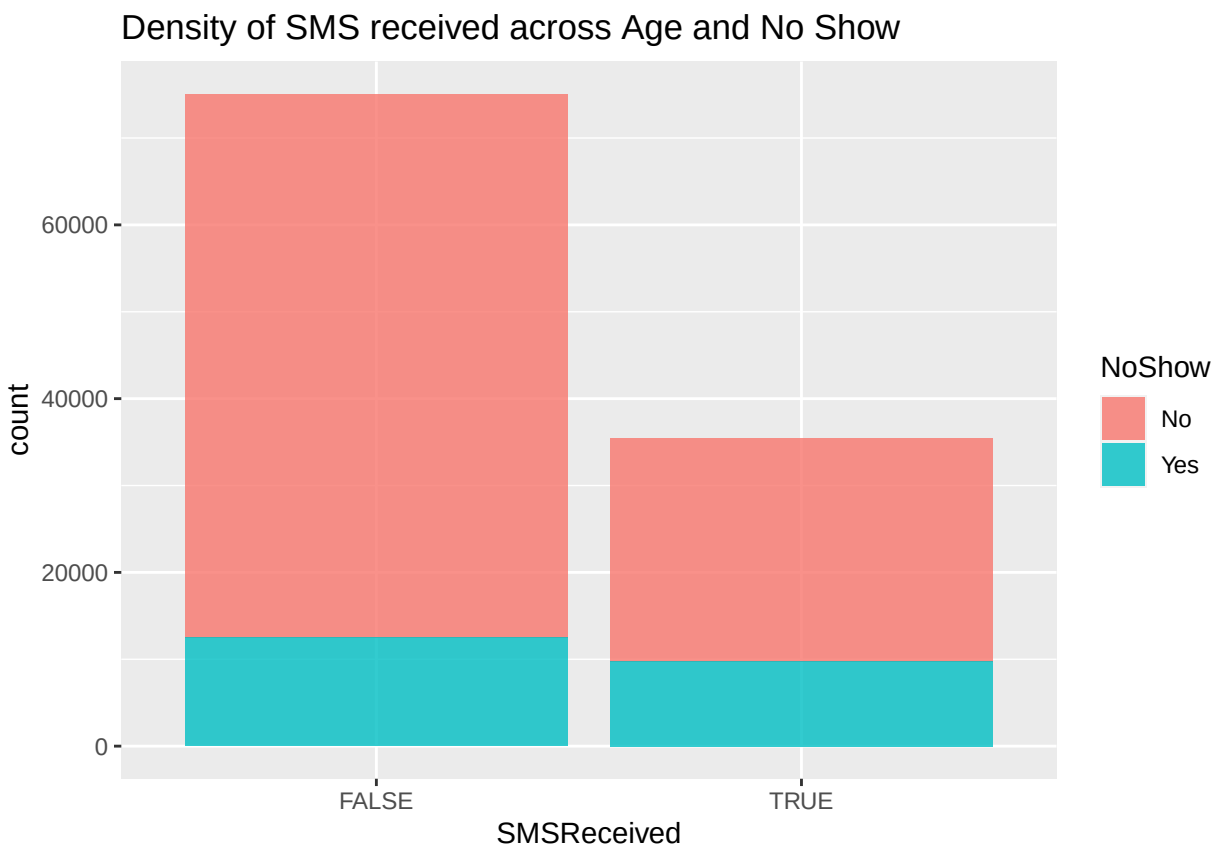
Another interesting finding is the `NA` group, they are the result of trying to assign age of 0 to groups and represent missing data.

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

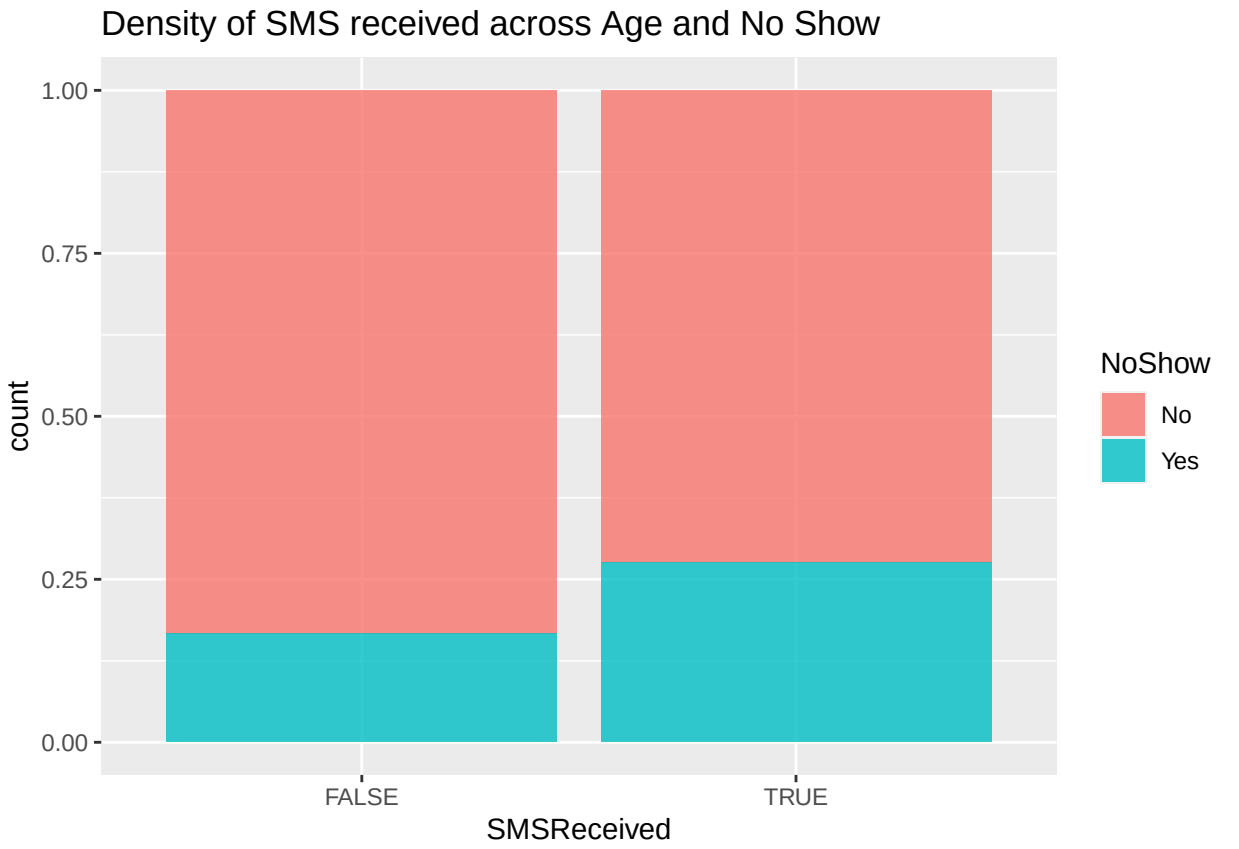
```
## # A tibble: 1 x 1
##       n
##   <int>
## 1  3539
```

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)?

Answer:

Based on the proportion between `SMSReceived` and `NoShow`, there does not seem to have significant influence of SMS on patient attendance.

```
count(raw.data %>% filter(SMSReceived == TRUE, NoShow == 'Yes', Age > 60))
```

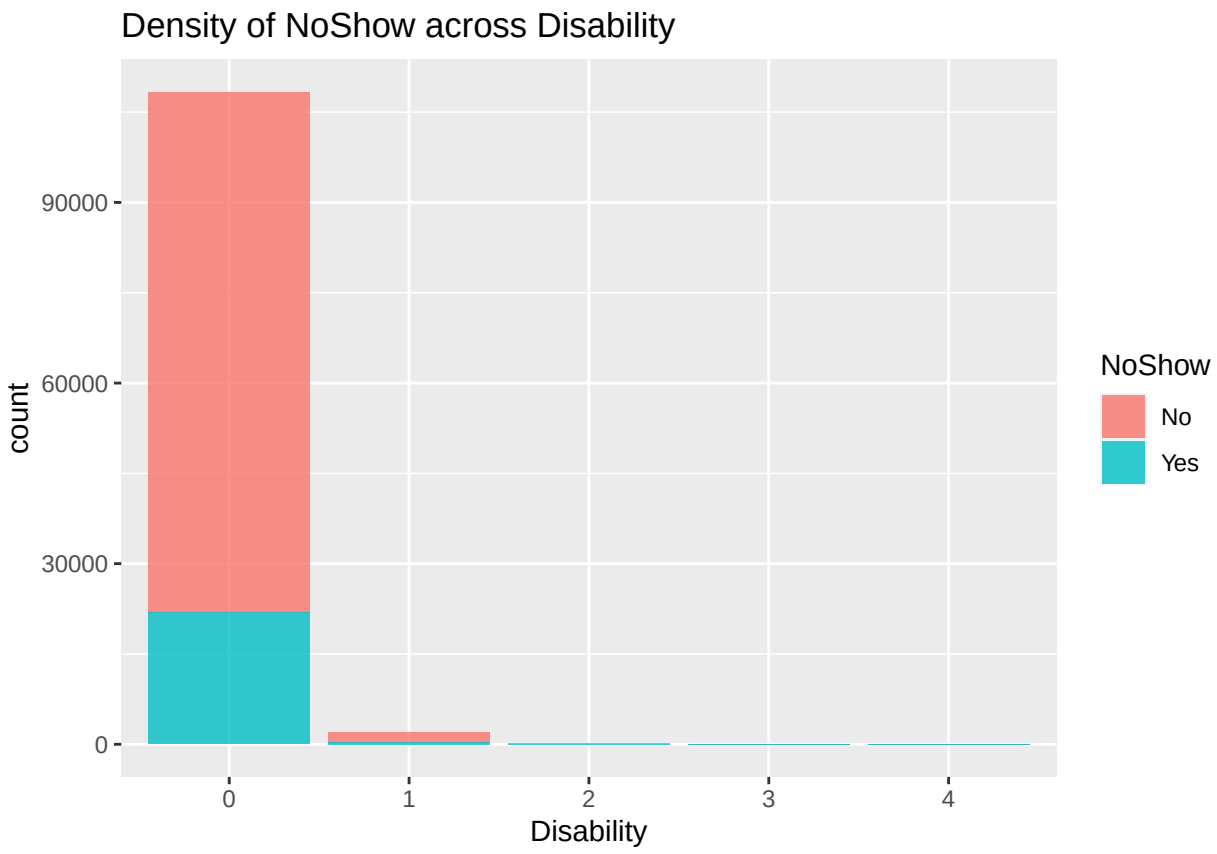
```
## # A tibble: 1 x 1
##       n
##   <int>
## 1  1179
```

One plausible reason is due to unforeseen circumstanses for a patient on their appointment day. For example, based on the above code, we see that older patients (greater than 60 years of age) are not able to attend their appointment, in-spite of receiving SMS reminder.

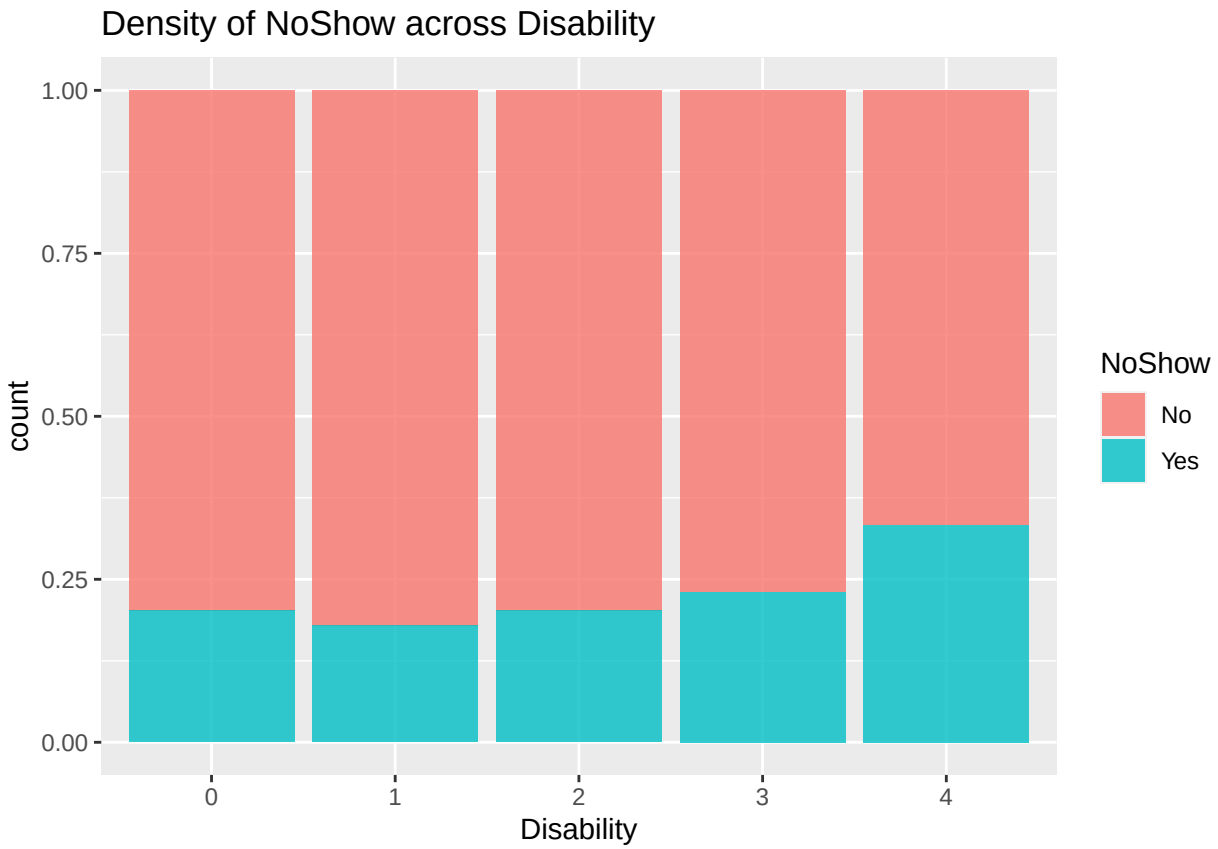
12 Create a similar plot which compares the the density of `NoShow` across the values of disability

Answer:

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

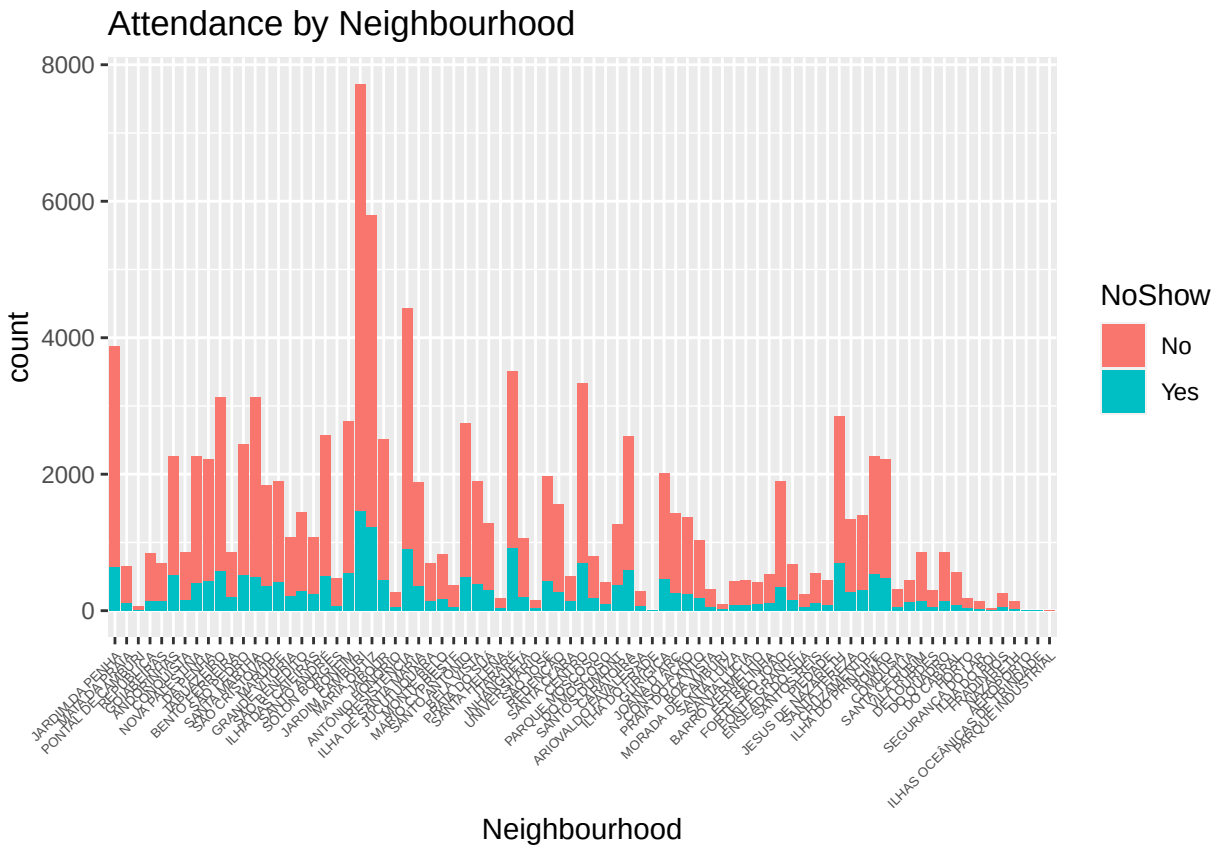



```
ggplot(raw.data) +
  geom_bar(aes(x=Disability, fill=NoShow), position='fill', alpha=0.8) +
  ggtitle("Density of NoShow across Disability")
```

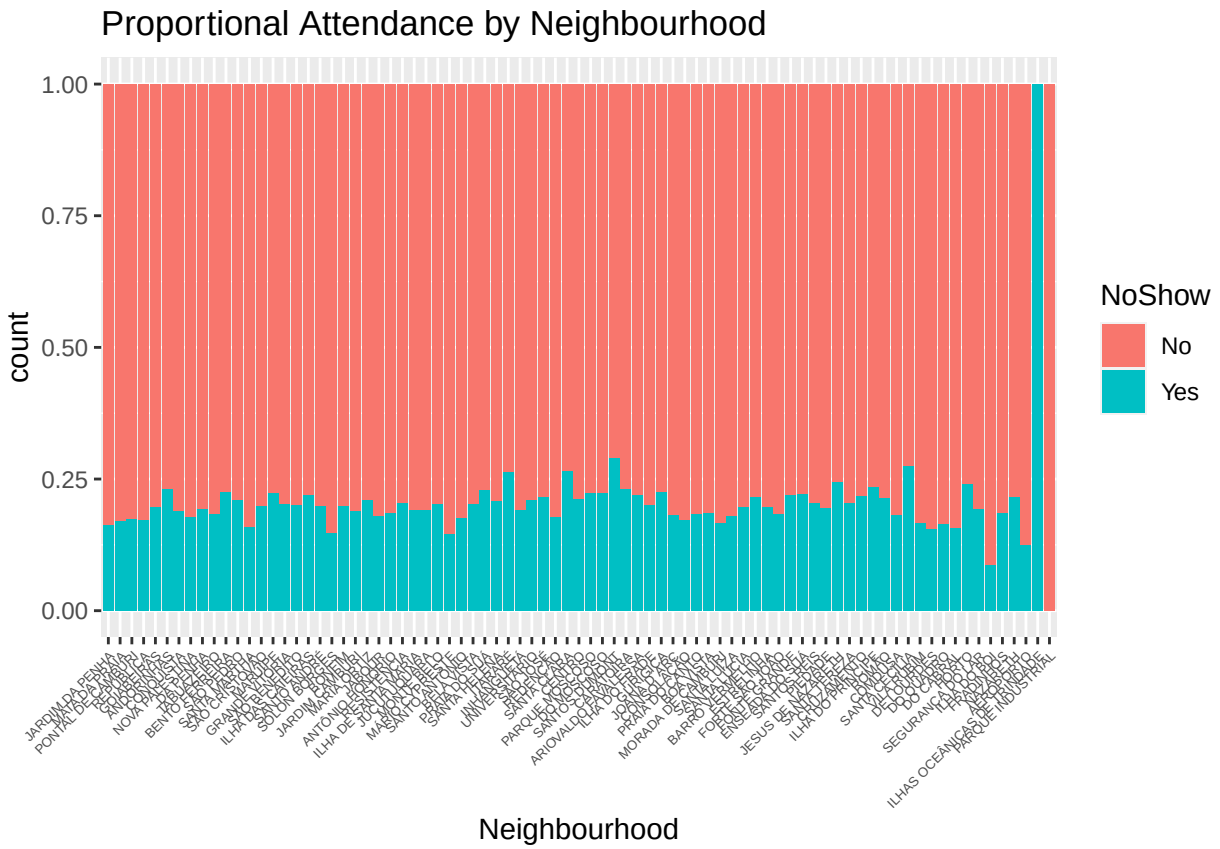


Now let's look at the neighbourhood data as location can correlate highly with many social determinants of health.

```
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.

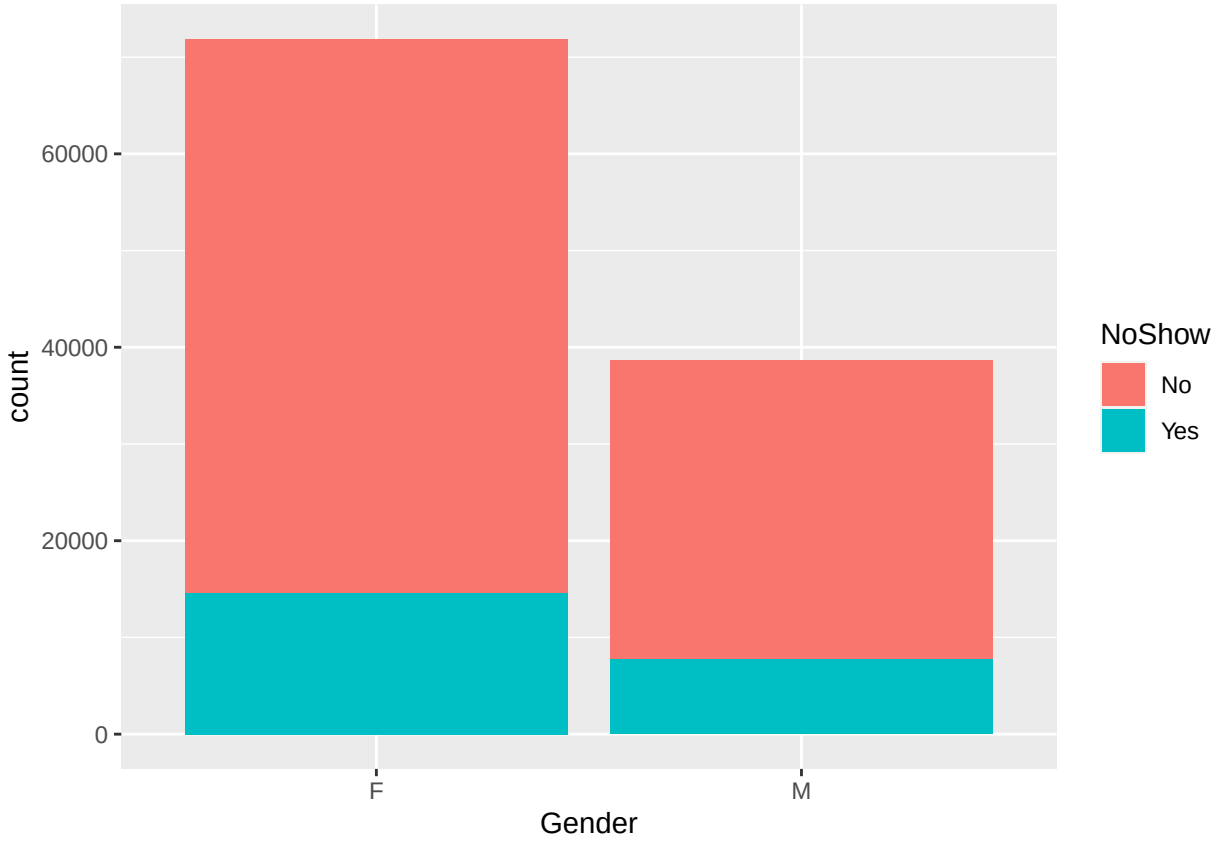
13 Suggest a reason for differences in attendance rates across neighbourhoods.

Answer:

Time to reach the healthcare center depends on the location of the neighborhood and any unforseen circumstances while travelling to the healthcare center such as detour, would account for attendance during the scheduled appointment.

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

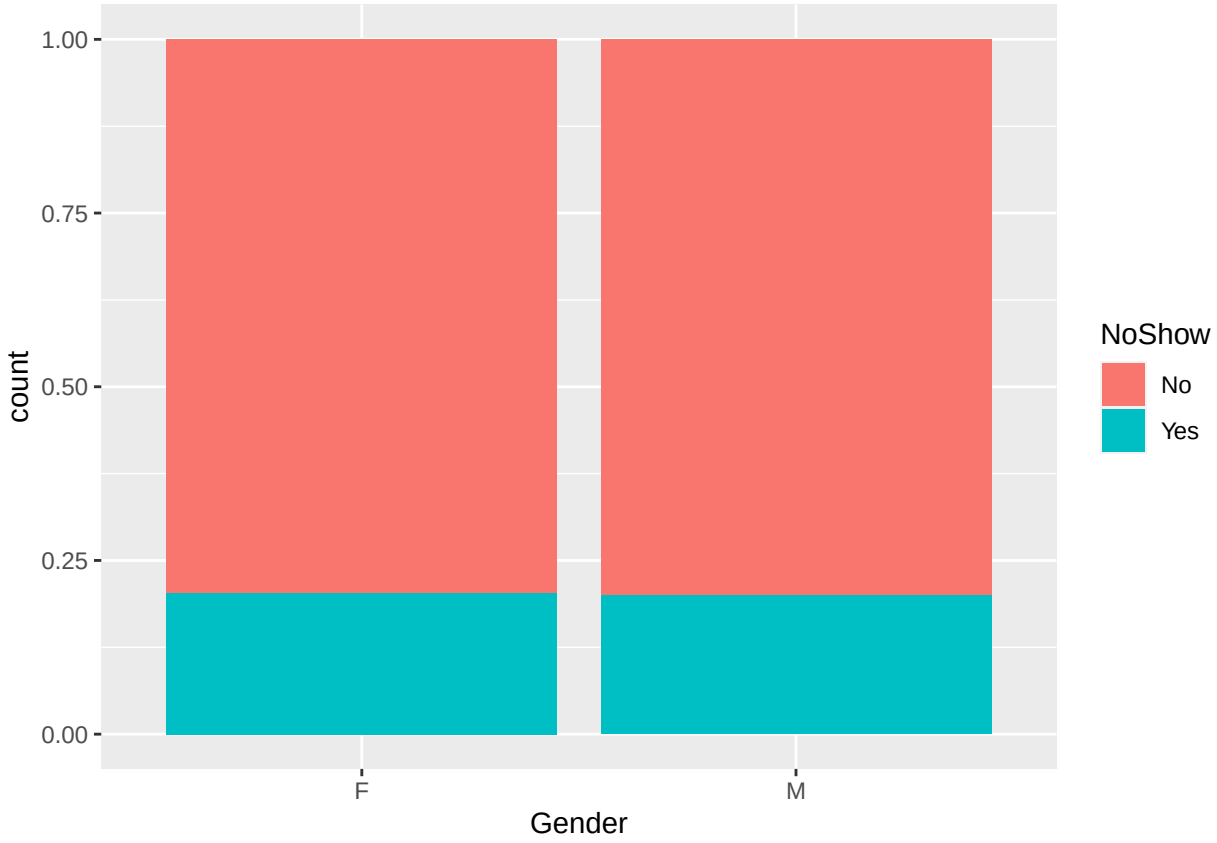
```
ggplot(raw.data) +
  geom_bar(aes(x=Gender, fill=NoShow))
```




```
  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')
```



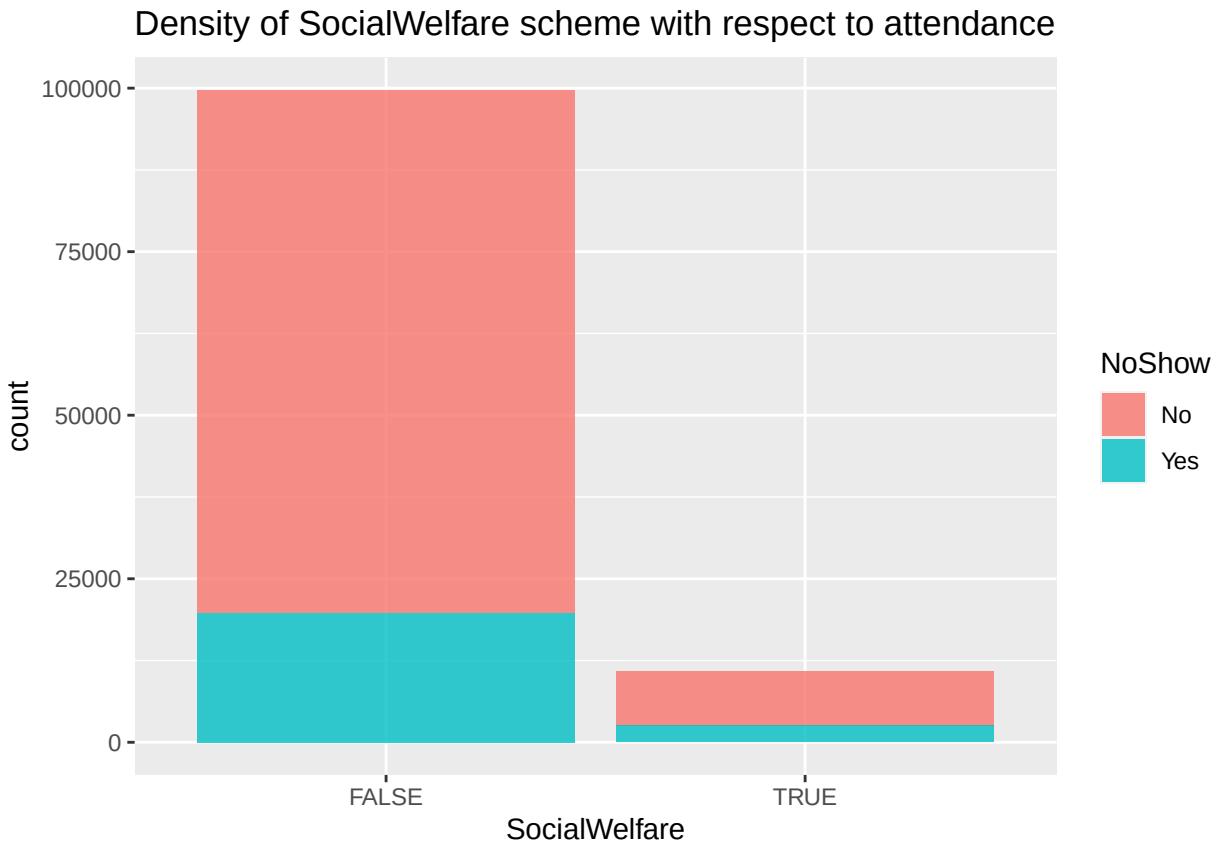

```
  ggtitle("Gender by attendance")
```

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

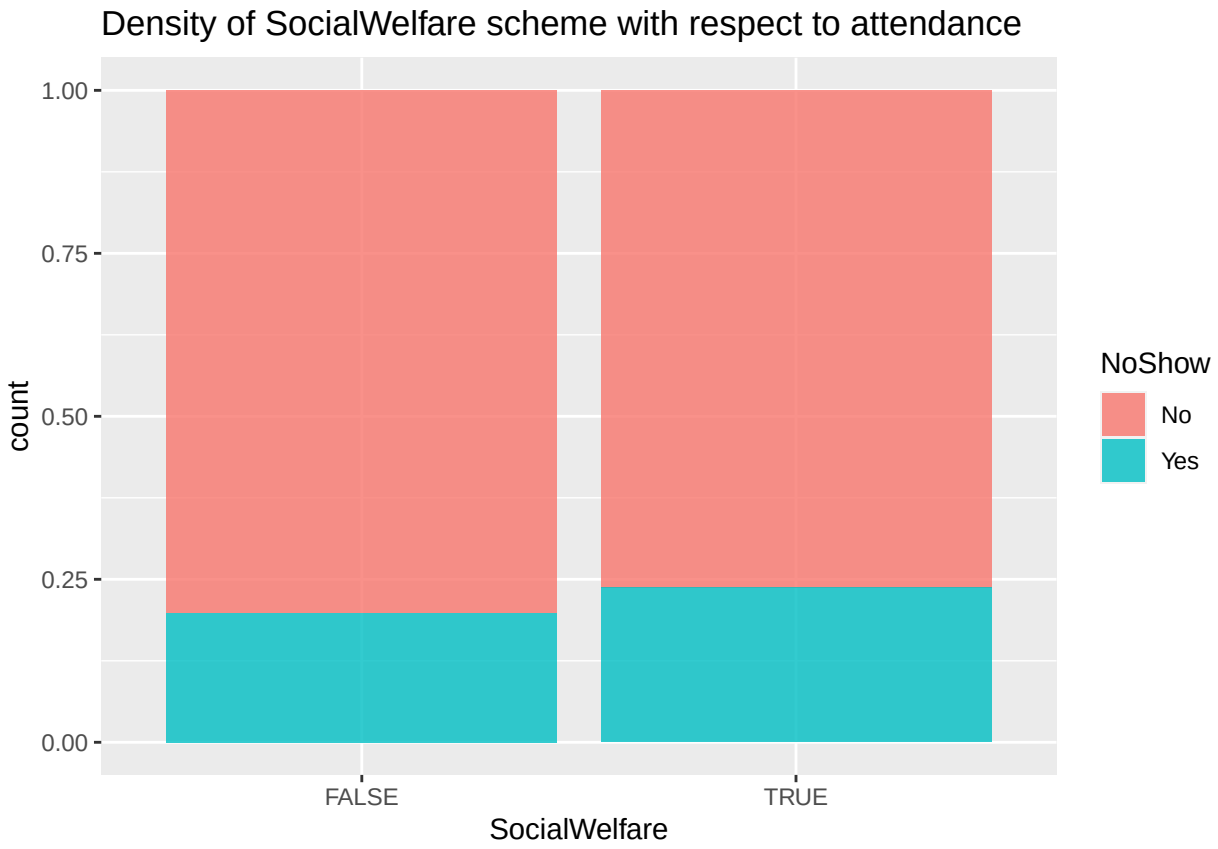
14 Create a similar plot using `SocialWelfare`

Answer:

```
ggplot(raw.data) +
  geom_bar(aes(x=SocialWelfare, fill=NoShow), alpha=0.8) +
  ggtitle("Density of SocialWelfare scheme with respect to attendance")
```




```
ggplot(raw.data) +
  geom_bar(aes(x=SocialWelfare, fill=NoShow), position='fill', alpha=0.8) +
  ggtitle("Density of SocialWelfare scheme with respect to attendance")
```



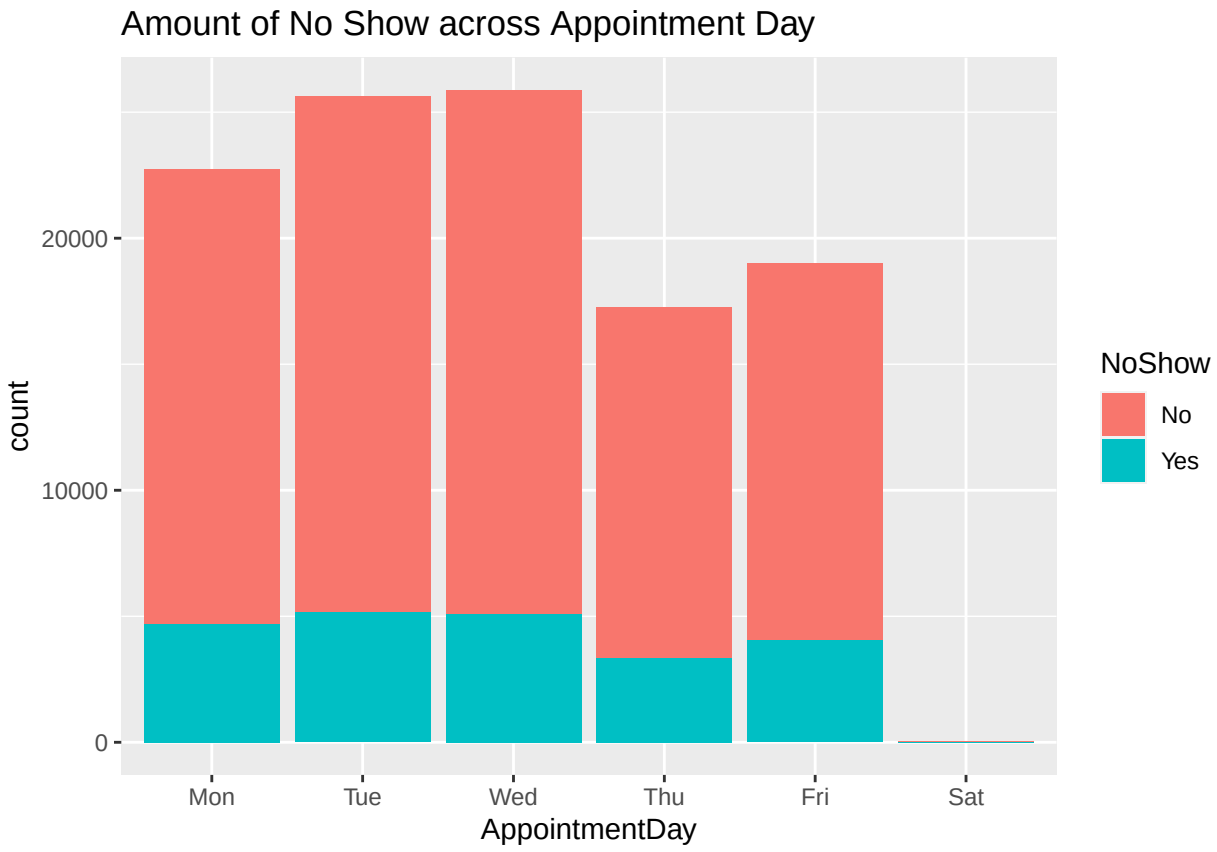
Far more exploration could still be done, including dimensionality reduction approaches but although we have found some patterns there is no major/striking patterns on the data as it currently stands.

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

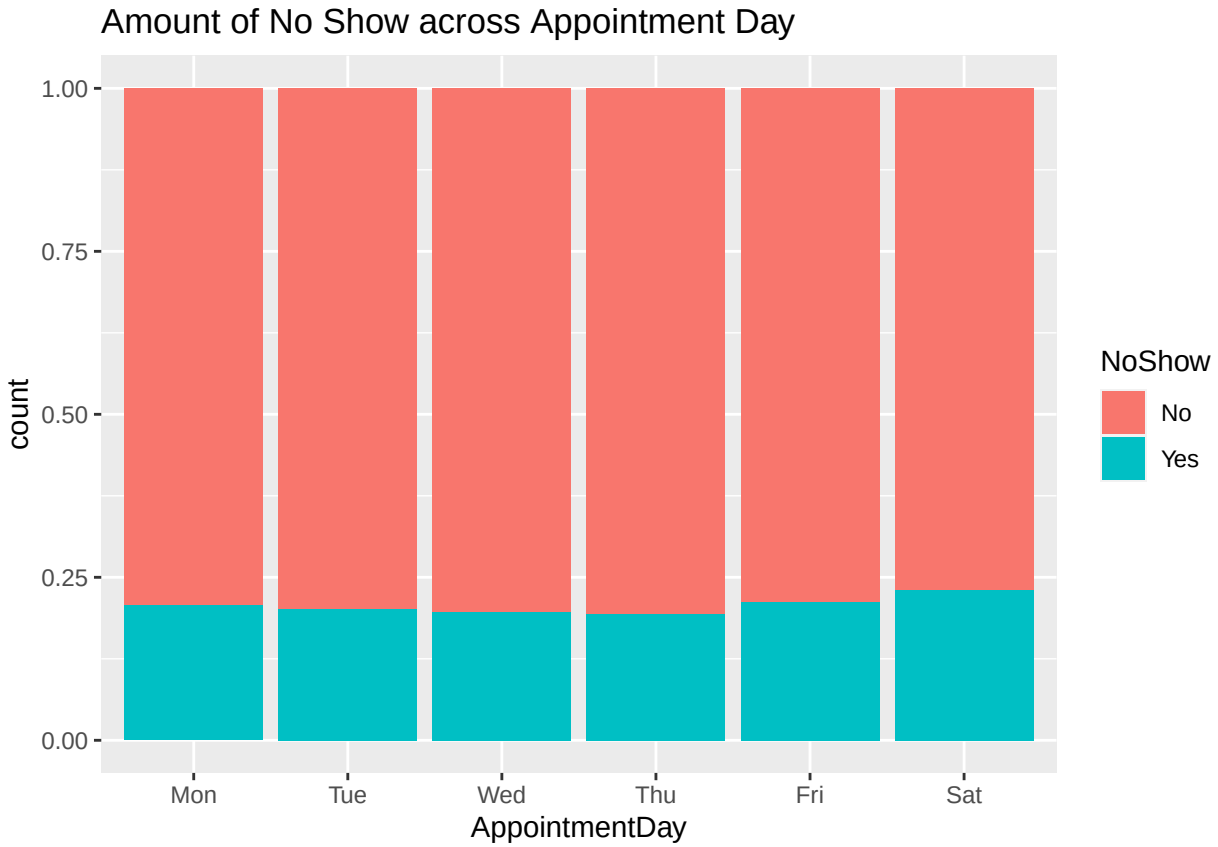
Feature Engineering

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!

```
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")
```



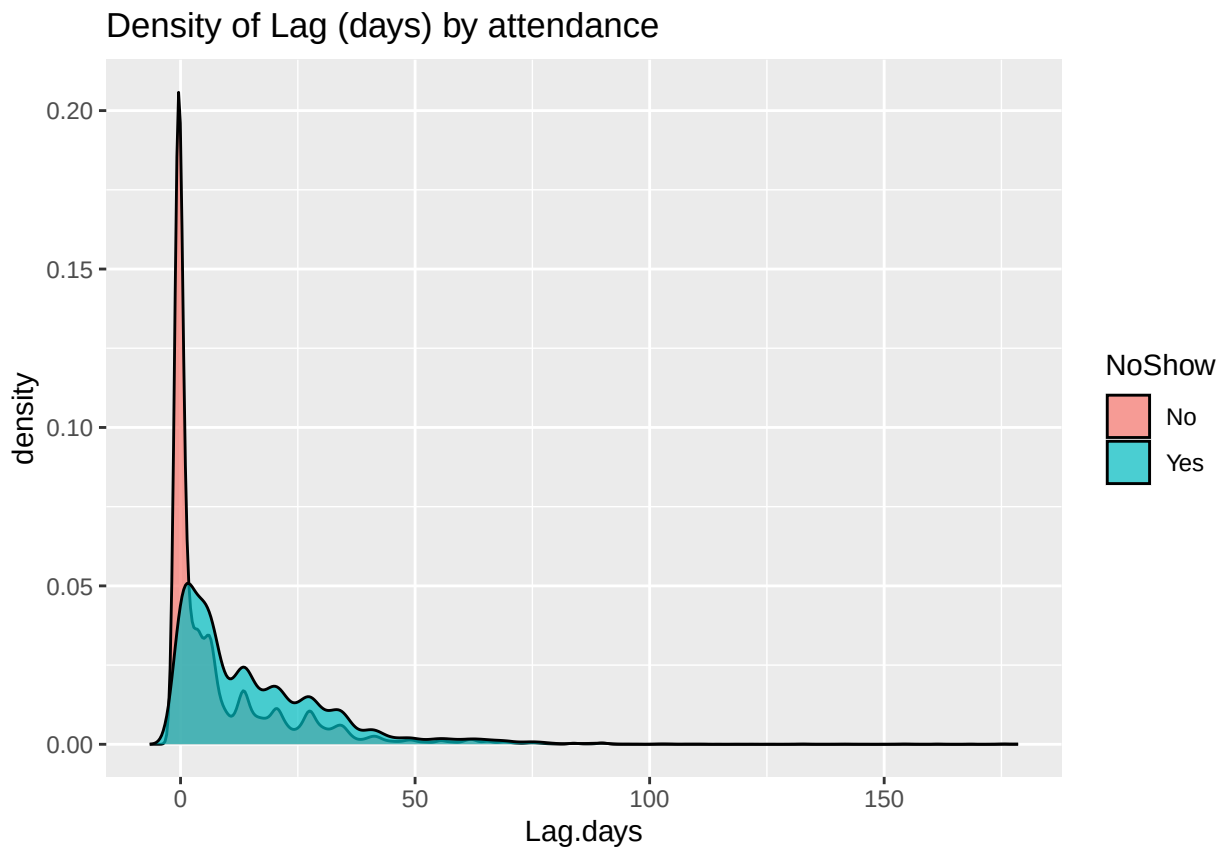

```
ggplot(raw.data) +
  geom_bar(aes(x=AppointmentDay, fill=NoShow), position = 'fill') +
  ggtitle("Amount of No Show across Appointment Day")
```



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?

Answer:

Based on the plot, we observe that patients show up for their appointment, if the `Lag` value is `0 days` (a case of same day appointment). Patients doesn't not show up, even if the `Lag` value is greater than `1 day`.

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.

```
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?

Answer:

Predictive modelling would not perform well, due to the following reasons:

- **Lacking sufficient number of features to predict the `NoShow` variable.**
- **Weak to negligible correlation between the dependent variable, `NoShow` and other independent variables (based on the correlation heatmap).**

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
##
```

```
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
```

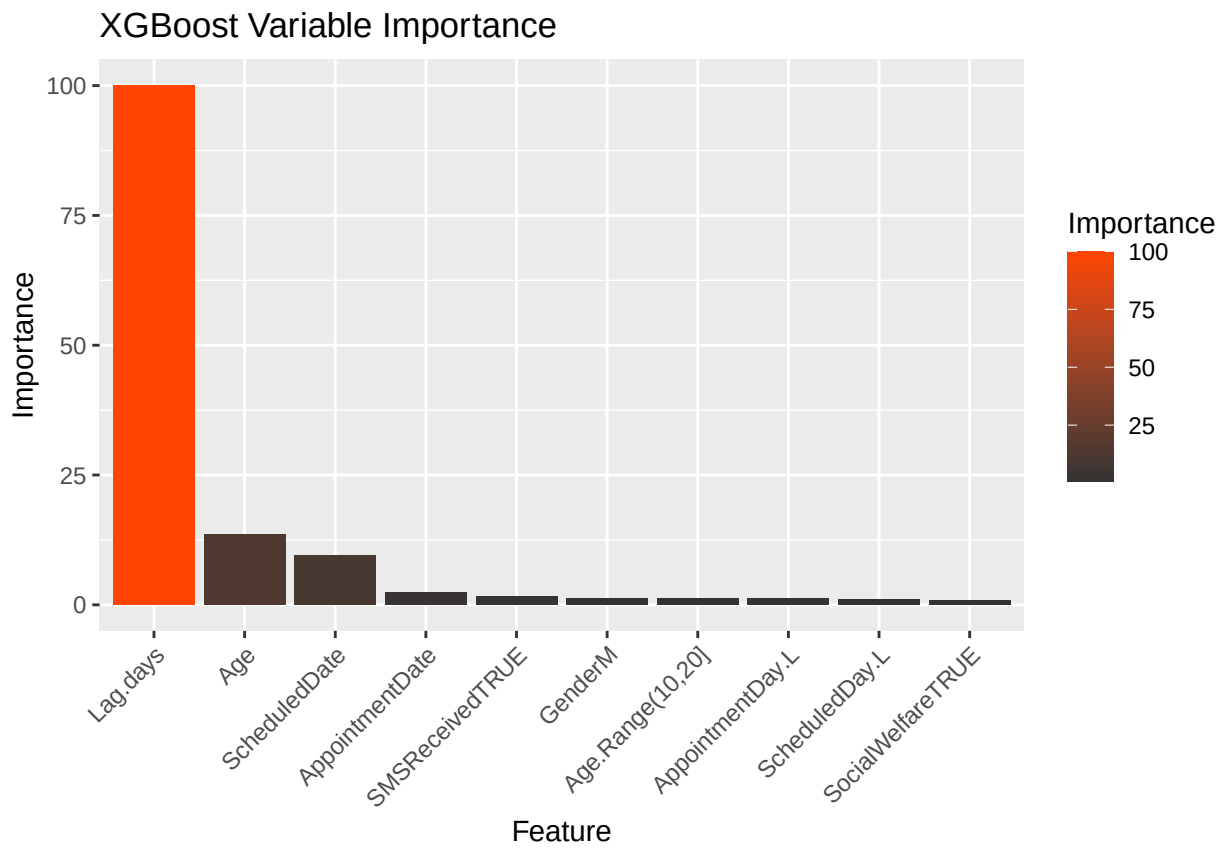
```
## [1] "XGBoost Area under ROC Curve: 0.742"
```

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

```
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.

Answer:

Implementing Random Forest on the data.

```
#Defining control parameters with 3-fold cross validation
fit.control <- trainControl(method="cv",number=3,
                            classProbs = TRUE, summaryFunction = twoClassSummary)
```

```
rf.model <- train(NoShow ~ .,data=train, method="rf", metric="ROC", trControl=fit.control)

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

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

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    No   Yes
##        No  25423  5411
##        Yes  1122  1202
##
##                Accuracy : 0.803
##                  95% CI : (0.7987, 0.8072)
##     No Information Rate : 0.8006
##     P-Value [Acc > NIR] : 0.1372
##
##                   Kappa : 0.1844
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.18176
##             Specificity : 0.95773
##          Pos Pred Value : 0.51721
##          Neg Pred Value : 0.82451
##              Prevalence : 0.19944
##          Detection Rate : 0.03625
##    Detection Prevalence : 0.07009
##       Balanced Accuracy : 0.56975
##
##        'Positive' Class : Yes
##
```

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

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

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

```
## [1] "Random Forest Area under ROC Curve: 0.745"
```

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

Answer:

The dataset is limited in terms of a good analysis because of the following downsides:

- **Lack of sufficient number of features and missing contextual various information.**
- **Chances of discrepancies and errors within the data. For example, we saw impossible ages of -1 within the data.**
- **Weak to negligible correlation between variables.**

Reference

[1] Correlation and Simple Linear Regression, Kelly H. Zou, Kemal Tuncali, and Stuart G. Silverman, Radiology 2003 227:3, 617-628

Credits

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