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: Unique identifier for each patient The patientID is their unique identifier used to be able to tell each patient apart from the others

AppointmentID: Unique identifier to each appointment AppointmentID is used to be able to keep track of the different appointment date and time for each patient reason for its unique identifer number its to make sure their isn't someone with an appointment on the same day or time.

Gender: Patient Gender (limited to Male or Female) Gender variable is used to identify the patient if they are male or female.

ScheduledDate: date on which the appointment was scheduled

AppointmentDate: date of the actual appointment

Age: Patient age

Neighbourhood: District of Vitória in which the appointment

The neighbourhood is where the appointments are located or where patient ar located

SocialWelfare: Patient is a recipient of Bolsa Família welfare payments

social welfare is for low income individuals that might not be able to afford health care for this example, and they are getting funds from Bolsa Familia to help pay for medicine

Hypertension: Patient previously diagnoised with hypertensio (Boolean)

Hypertension is patient with high blood pressure higher than whats normal

Diabetes: Patient previously diagnosed with diabetes (Boolean)

patient is diagnosed with diabetes because their body doesnt make enough insulin

AlcoholUseDisorder: Patient previously diagnosed with alcohol use disorder (Boolean)

Patient who is addicted to drinking, over the regular amount

Disability: Patient previously diagnosed with a disability (severity rated 0-4)

patient who may have disability can be someone with physical disabilitie

SMSReceived: At least 1 reminder text sent before appointment (Boolean)

SMS is when an email or a text message is send out to patient to remind them of their appointment

NoShow: Patient did not attend scheduled appointment (Boolean: Yes/No)

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

one hypotheses, they forgot about their appointment even with the reminder if the appointment is early in the morning.

second hypotheses they have a conflict with their appointment with other appointments

third they forgot to cancel that appointment, and they were still in system for that appointment.

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

one what type of disability does the patient have physical or mental, what does it refer to when mentioning disability

second SocialWelfare what does this mean in relation to the pateint does this count towards insurance for reference in helping patient with low income still get the treatment.

third missing information could be a variable on how many patient answer their SMSRecieved, and how many from those that answer their message actually showed up, having a SMS for those that said yes and showed up.

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('lab1/2016_05v2_VitoriaAppointmentData.csv', col_types='fffTTifllllflf')
raw.data <- readr::read_csv('https://raw.githubusercontent.com/maguire-lab/health_data_science_research_
```

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)`

```
raw.data %>% filter(Age > 100 ) %>% filter(Age >= 0 ) %>% drop_na()
```

```
## # 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>
```

```
#raw.data %>% filter(Age > 100)
```

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?

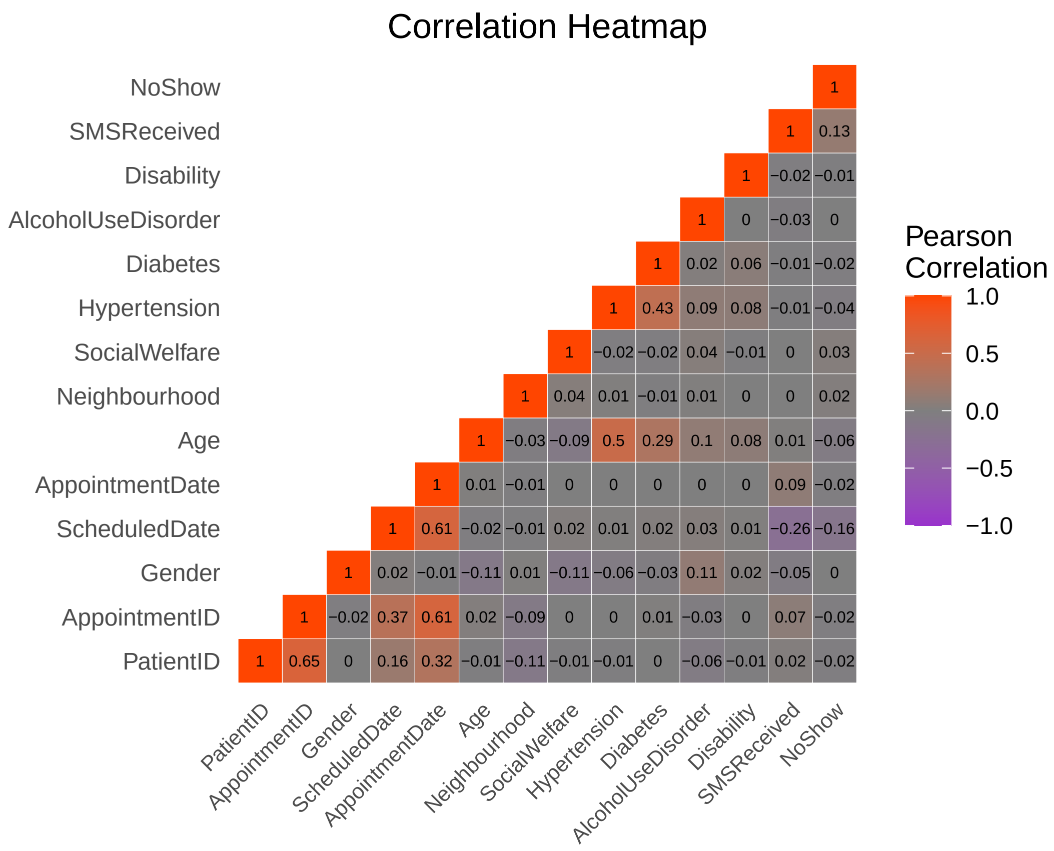
```
count(raw.data, PatientID, sort = TRUE)
```

```
## # 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
```

88 is the max amount of appointments a patient can have

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`)?

SMSRecievd with a postive correlation of 0.13. The patient that didn't answer were mostly likely to not show up for their appointments

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

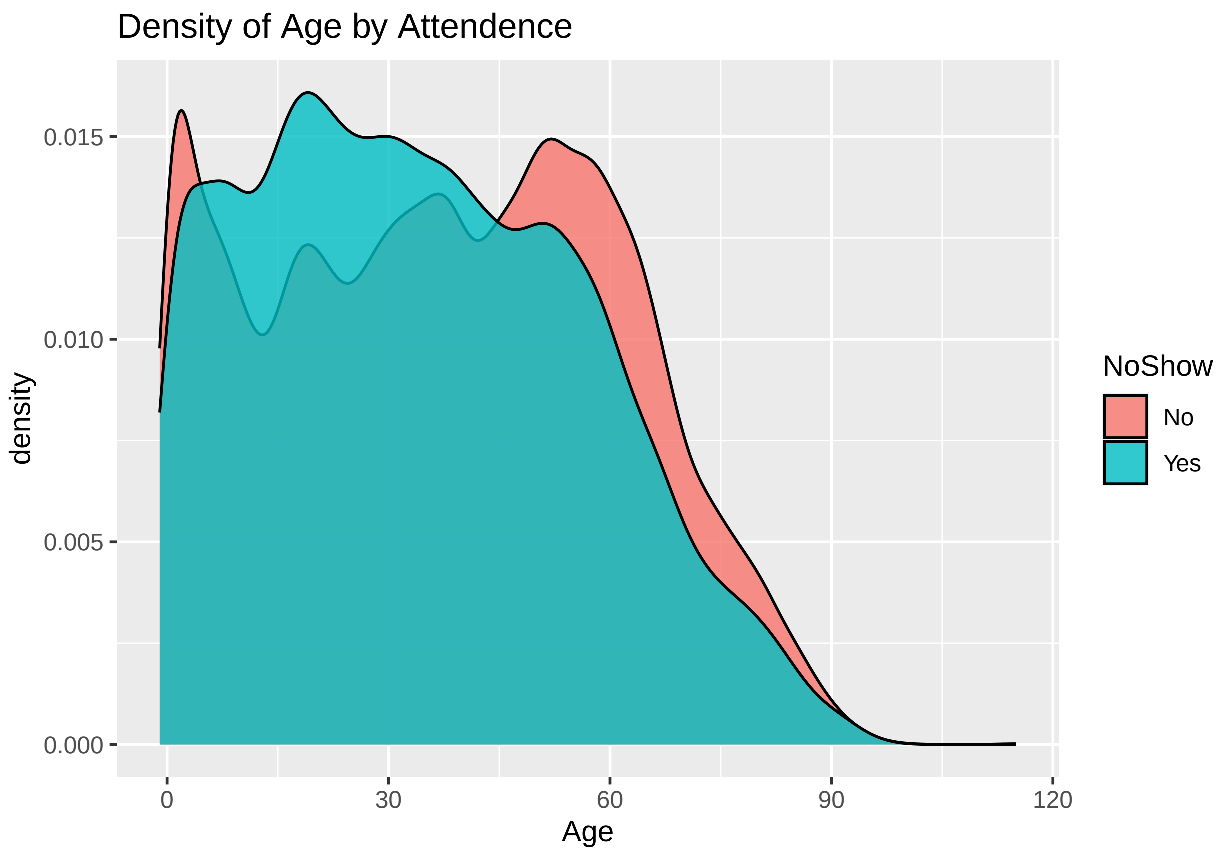
AppointmentID and appointmentDate have a strong correlation of 0.61, since the dates of the appointment are different, for each appointmentID its more likely that if we see appointmentID there will be an appointment data

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

No because patientID and appointmentID have a strong correlation since each patient is asign to a appointmentID. To be able to distinguish different appointment-ID with patients

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

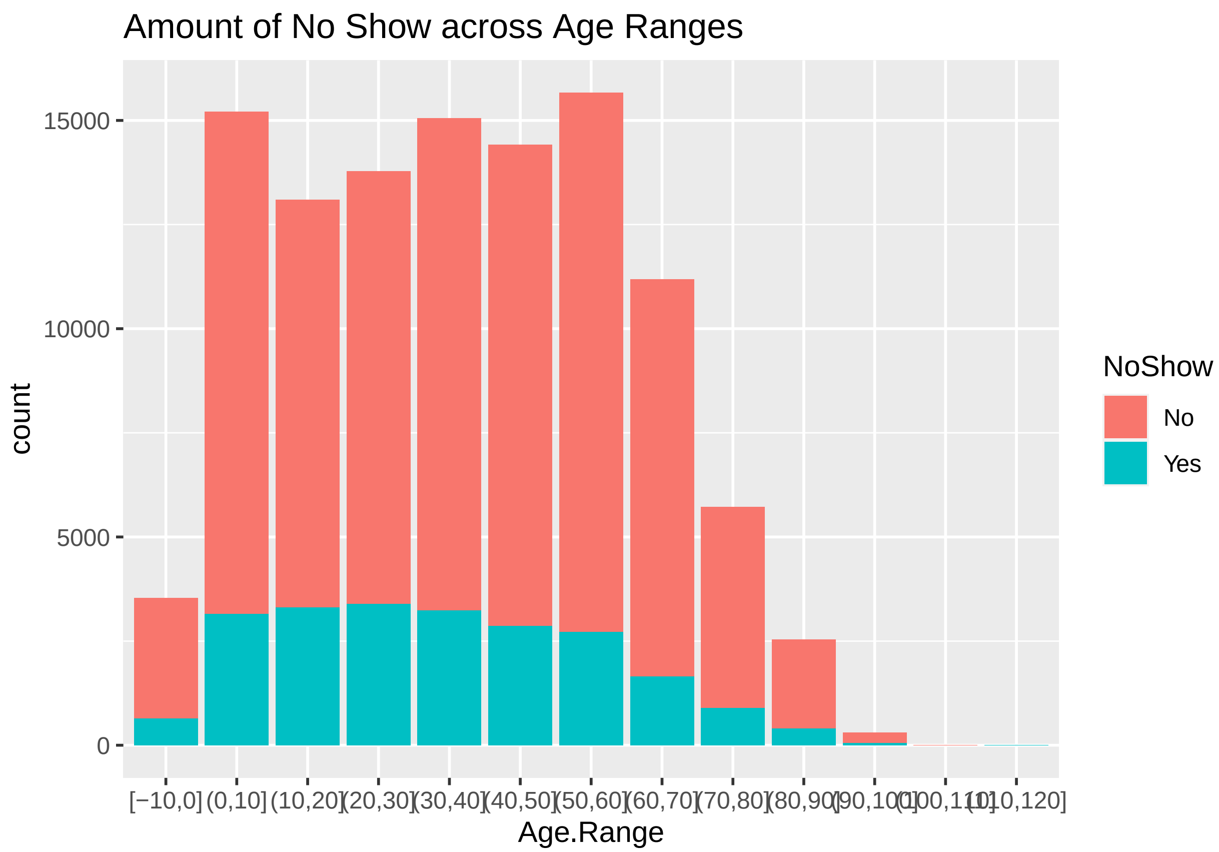
```
ggplot(raw.data) +
  geom_density(aes(x=Age, fill=NoShow), alpha=0.8) +
  ggtitle("Density of Age by Attendence")
```



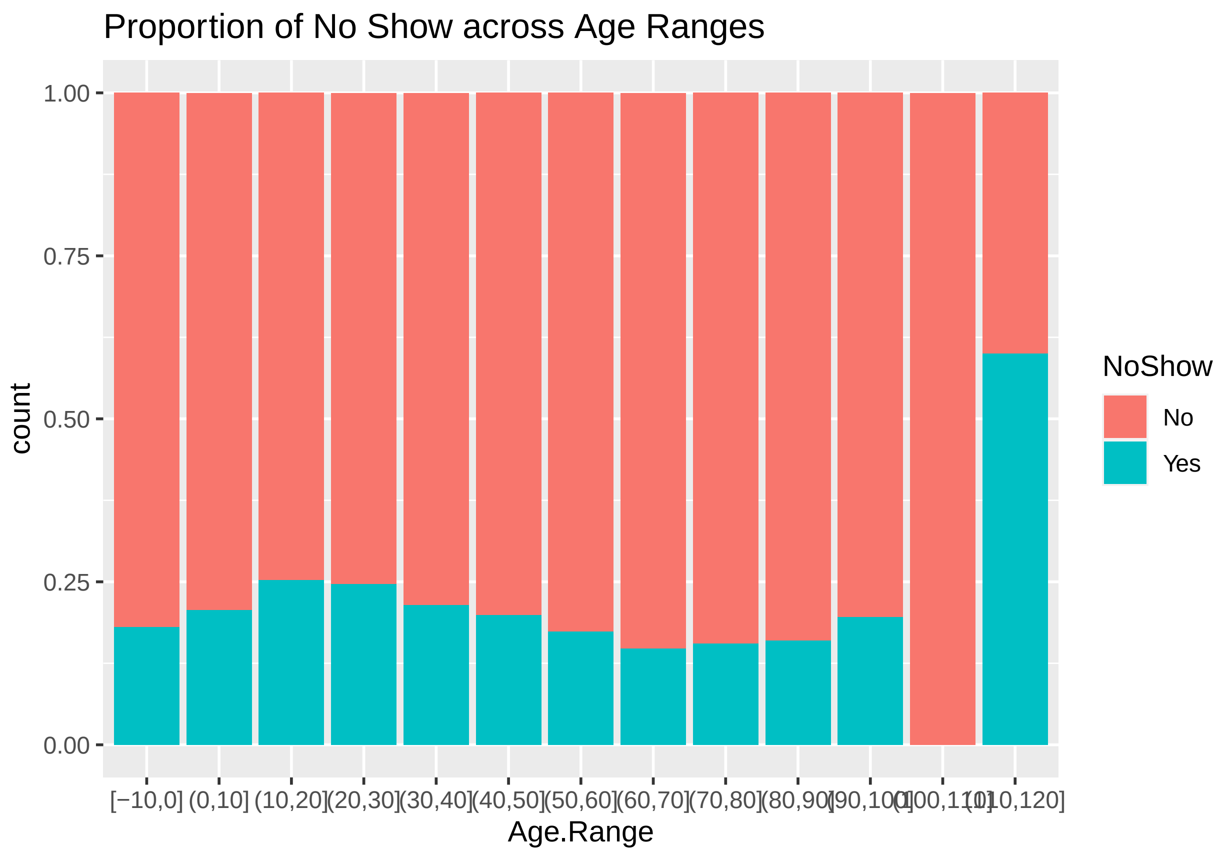
There does seem to be a difference in the distribution of ages of people that miss and don't miss appointments. However, the shape of this distribution means the actual correlation is near 0 in the heatmap above. This highlights the need to look at individual variables.

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

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

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



```
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?

It can be misleading for example if we looked at plot 2, it makes it seem like the age group of 110-120 are showing up when clearly they aren't compared to the age groups below 100. While plot 1 which is reasonable in showcasing the age group that do show up for their appointments. while also excluding none reasonable age group below 0 and above 100 years.

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.

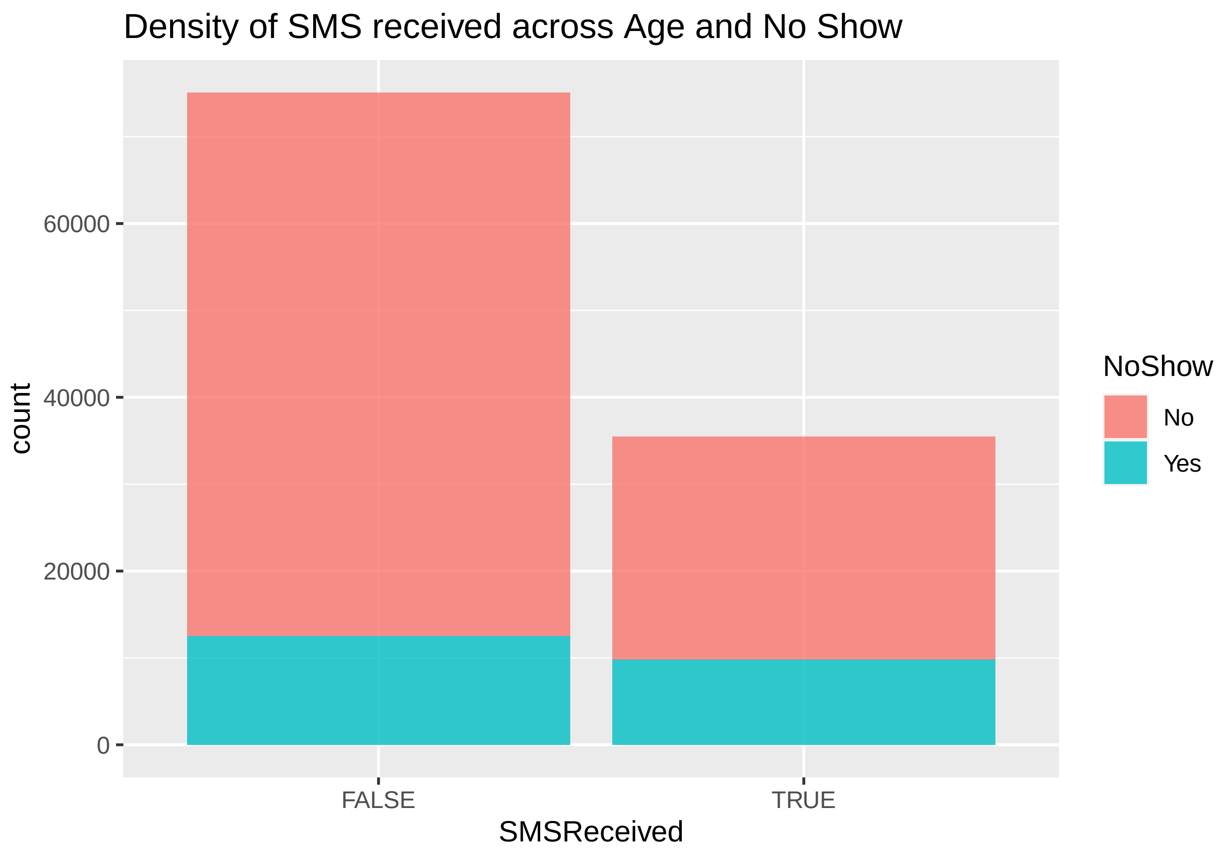
Another interesting finding is the `NA` group, they are the result of trying to assign age of 0 to groups and represent missing data.

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

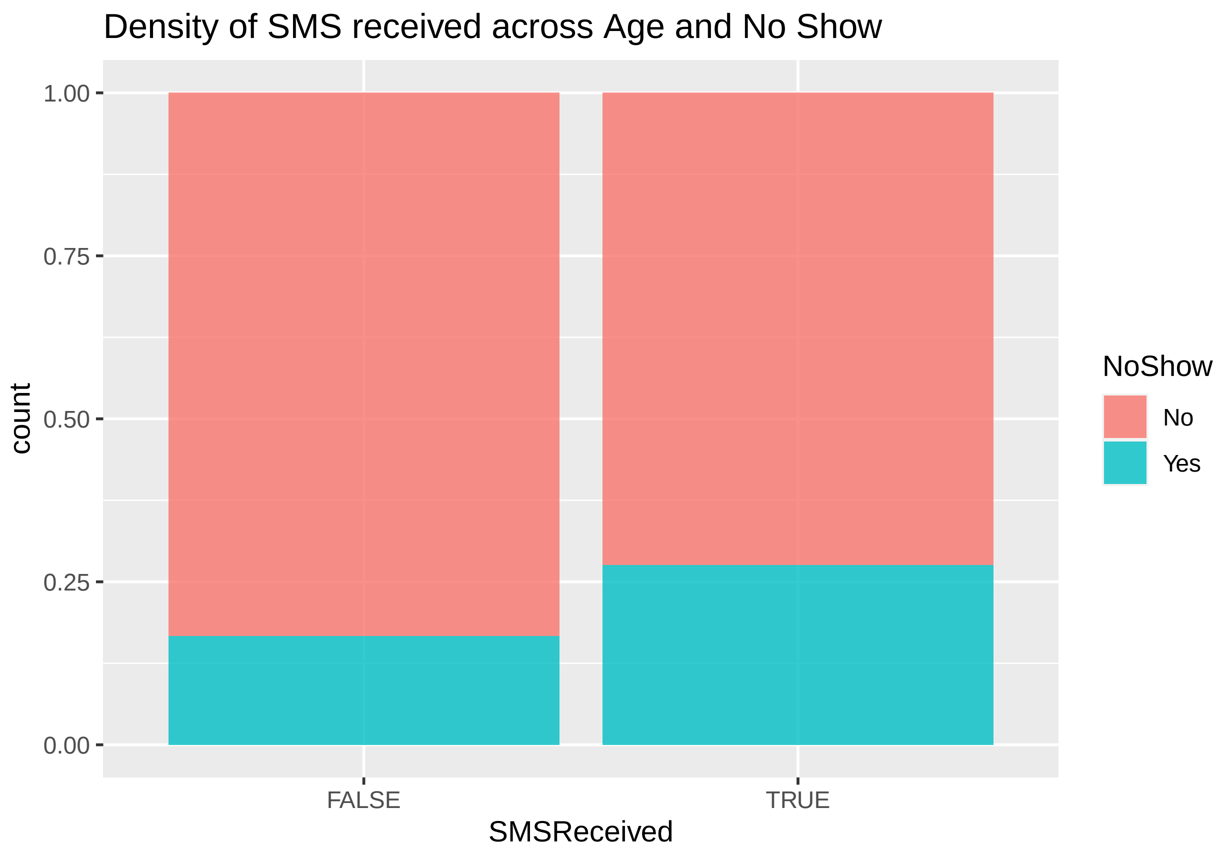
```
## # A tibble: 1 x 1
##       n
##   <int>
## 1  3539
```

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")
```



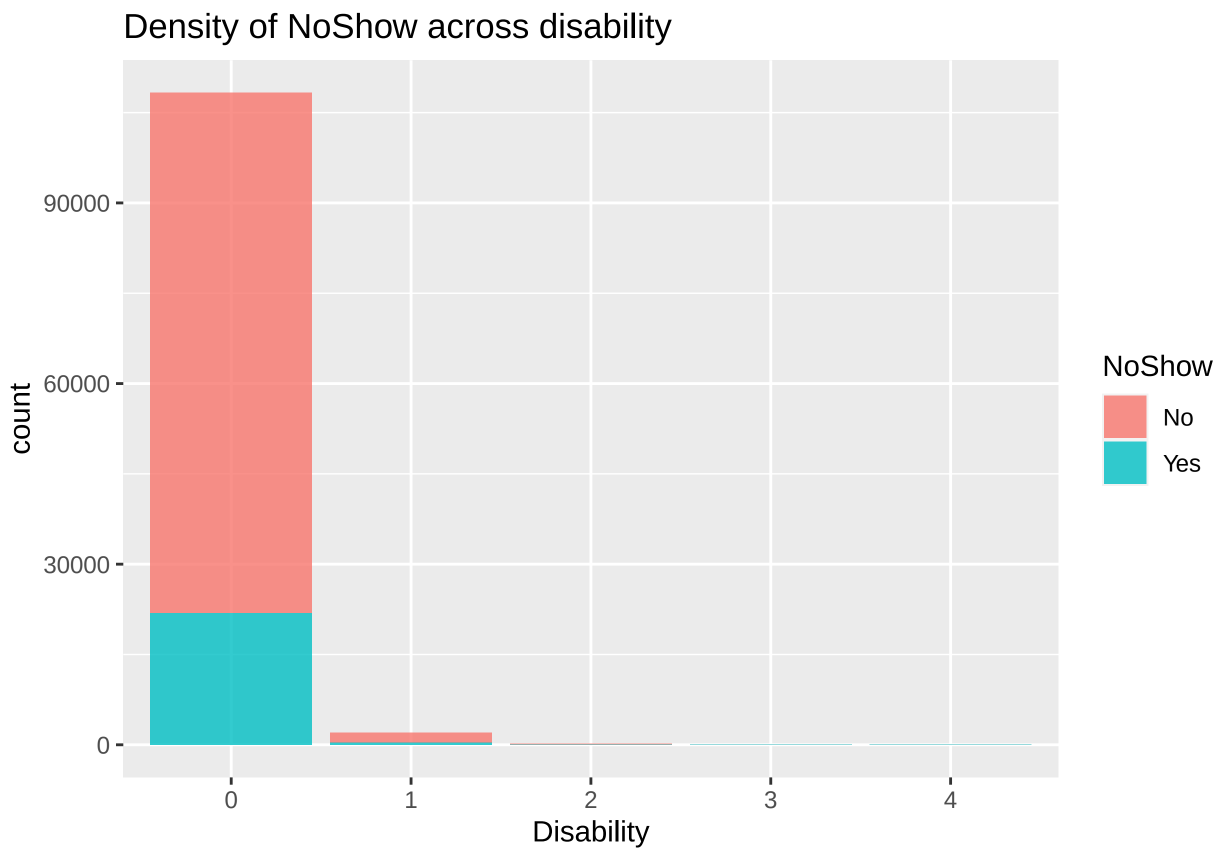
```
ggplot(raw.data) +
  geom_bar(aes(x=SMSReceived, fill=NoShow), position='fill', alpha=0.8) +
  ggtitle("Density of SMS received across Age and No Show")
```



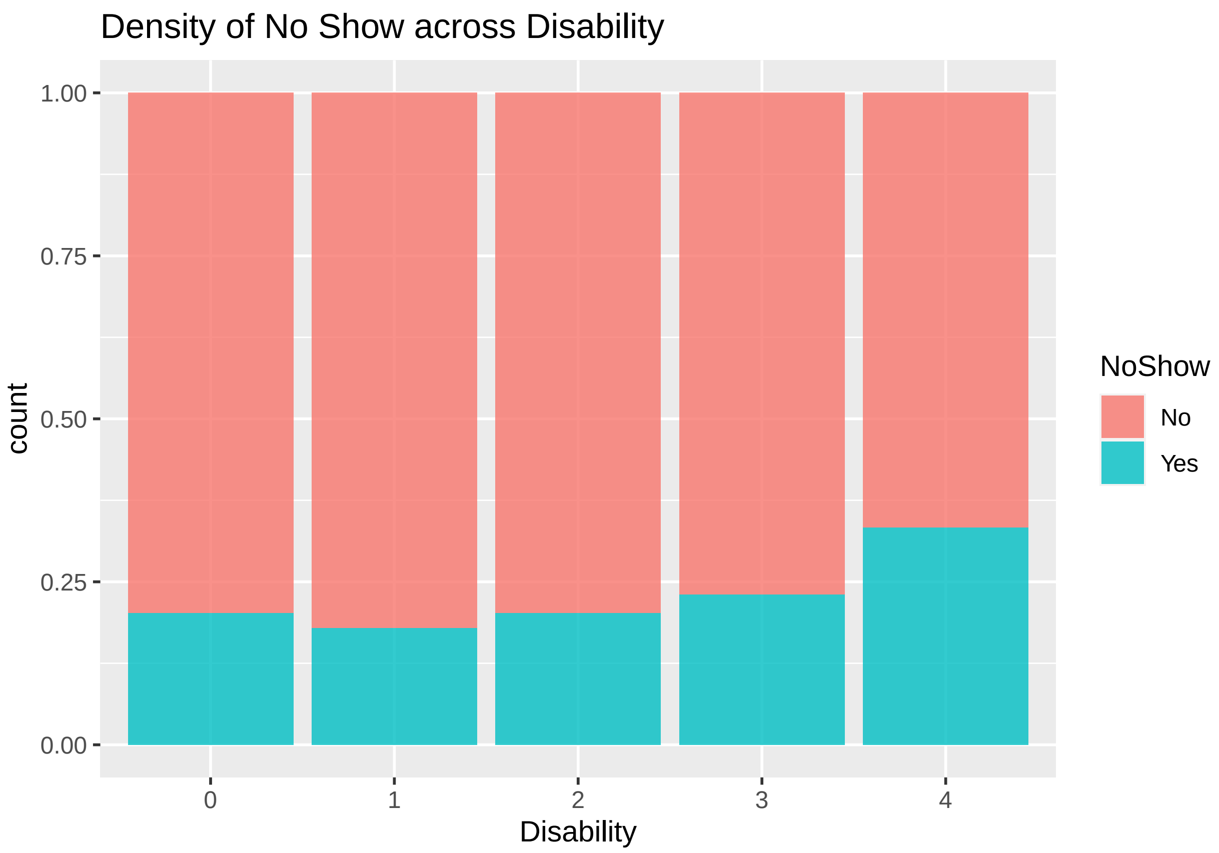
11 From this plot does it look like SMS reminders increase or decrease the chance of someone not attending an appointment? Why might the opposite actually be true (hint: think about biases)? The reminder seems to have decreased do to the amount of no. The opposite can be true because of the biases of including age group that might not have actually shown up, and thats due to the adding or filling of everything. Its also not likely that individual age 60-90 are going to answer a text message where they have to response compared to if it was a phone call.

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

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

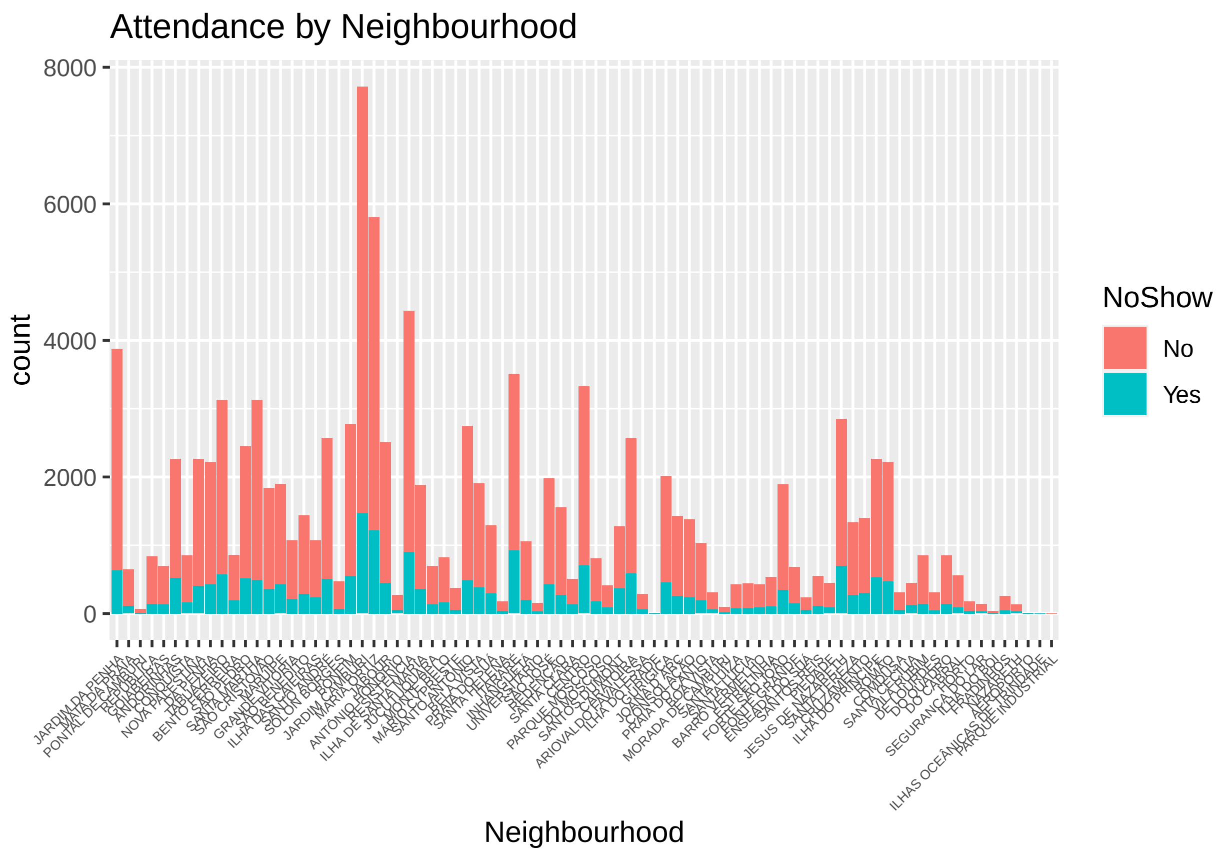


```
ggplot(raw.data) +
  geom_bar(aes(x=Disability, fill=NoShow), position='fill', alpha=0.8) +
  ggtitle("Density of No Show across Disability")
```

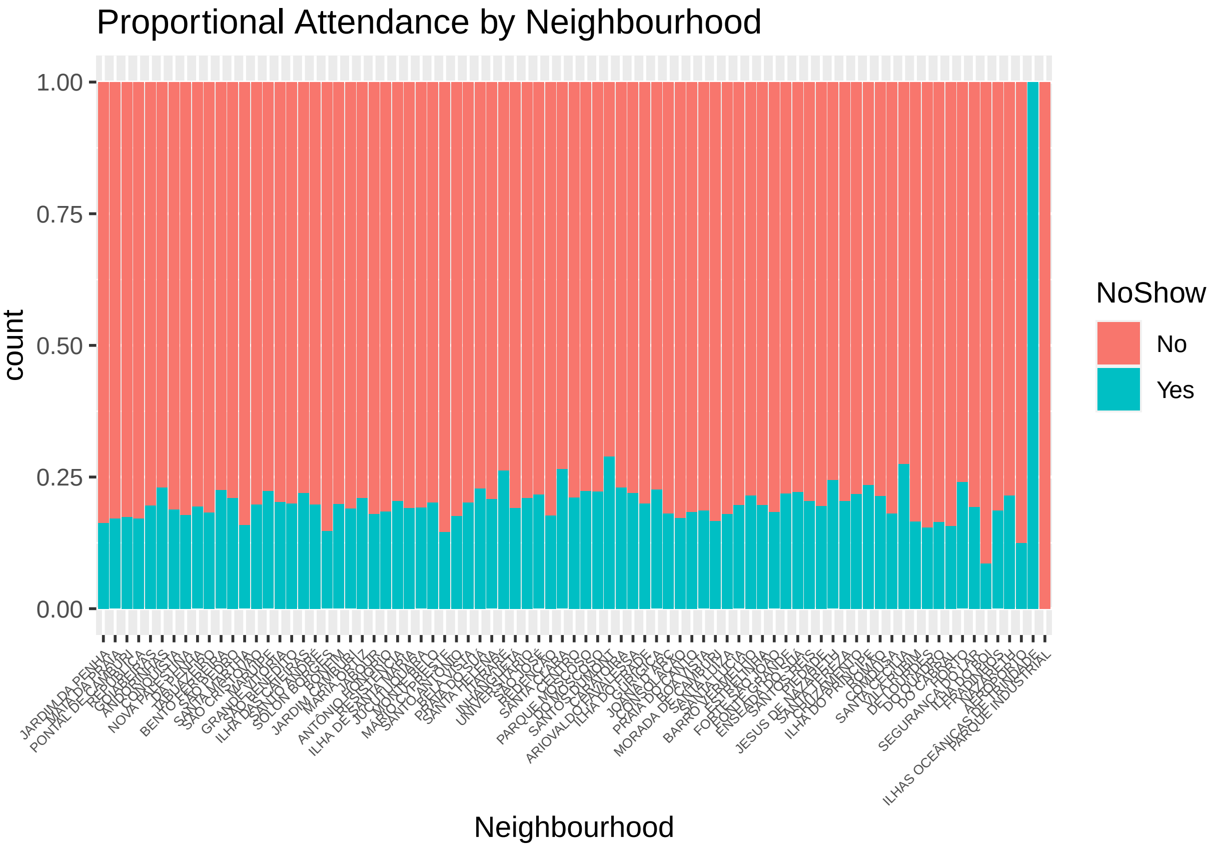


Now let's look at the neighbourhood data as location can correlate highly with many social determinants of health.

```
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')
```



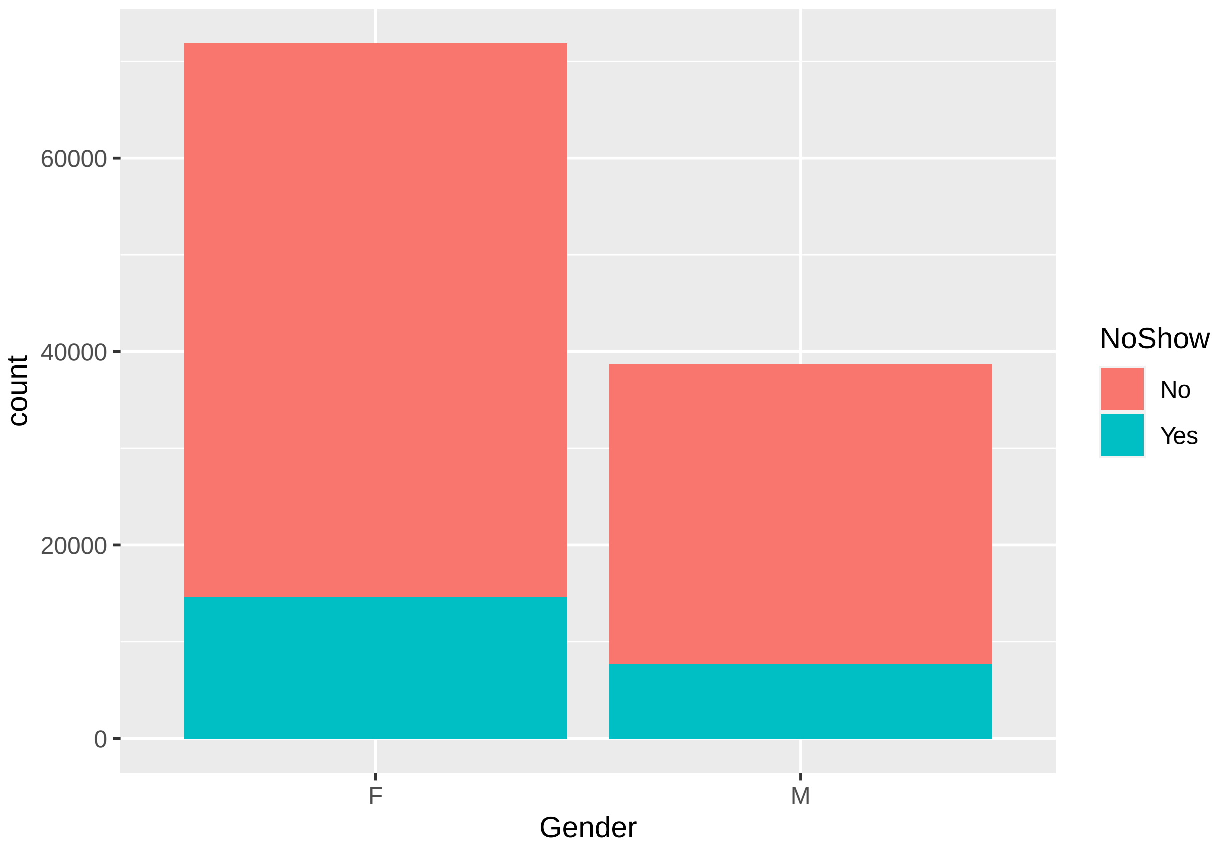
Most neighborhoods have similar proportions of no-show but some have much higher and lower rates.

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

A reason that one neighborhood has the highest attendance compared to the others could be do to higher population of patient in that area. They also have the highest yes and no's for NoShow, another reason could be that they are closer to the appointment locations compared to the other neighbourhoods.

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

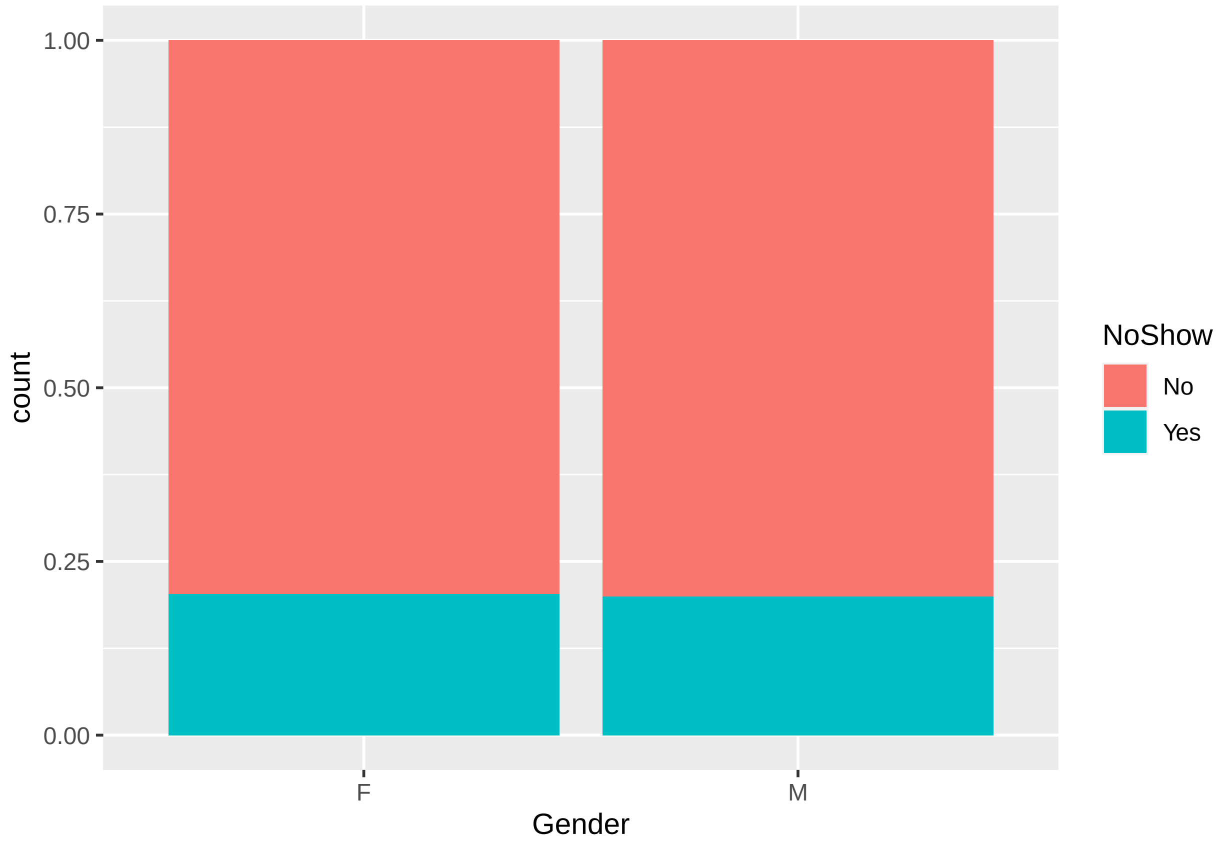
```
ggplot(raw.data) +
  geom_bar(aes(x=Gender, fill=NoShow))
```



```
  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')
```

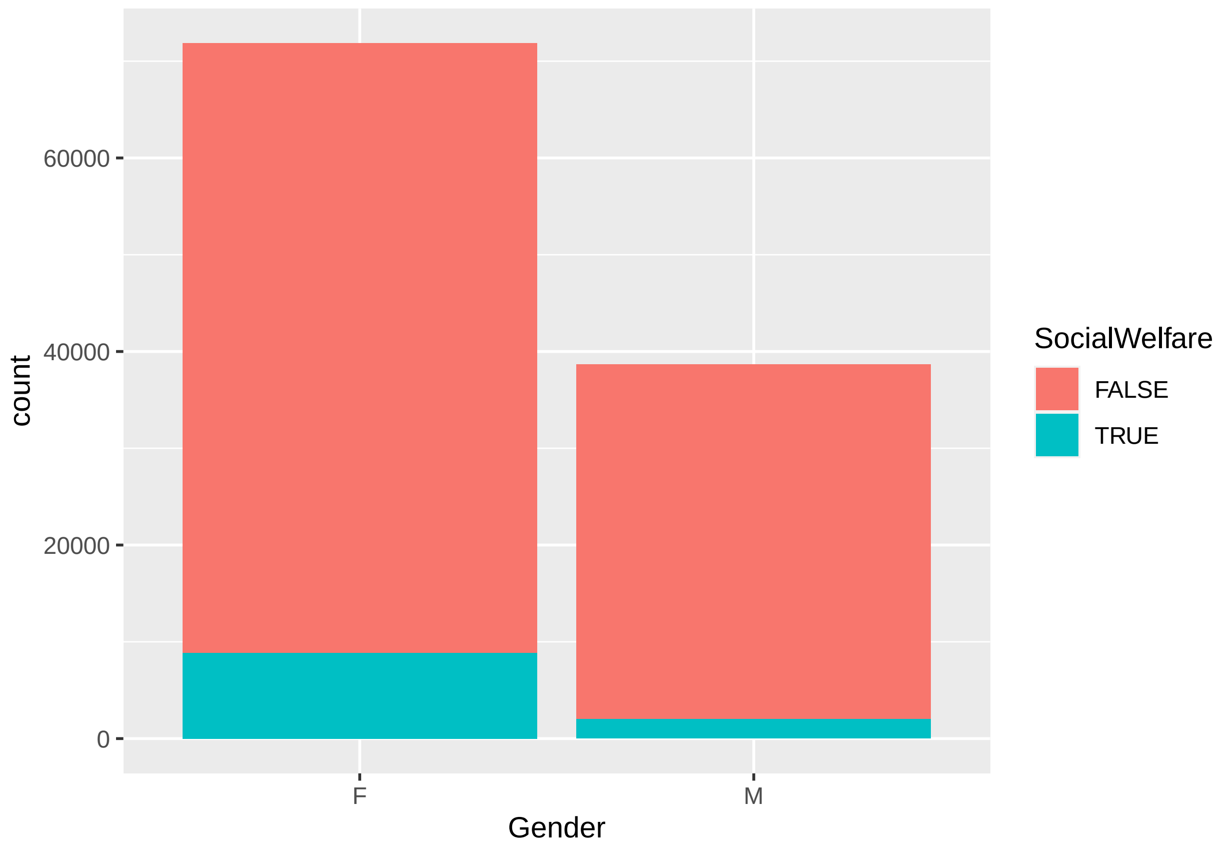


```
  ggtitle("Gender by attendance")
```

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

14 Create a similar plot using `SocialWelfare`

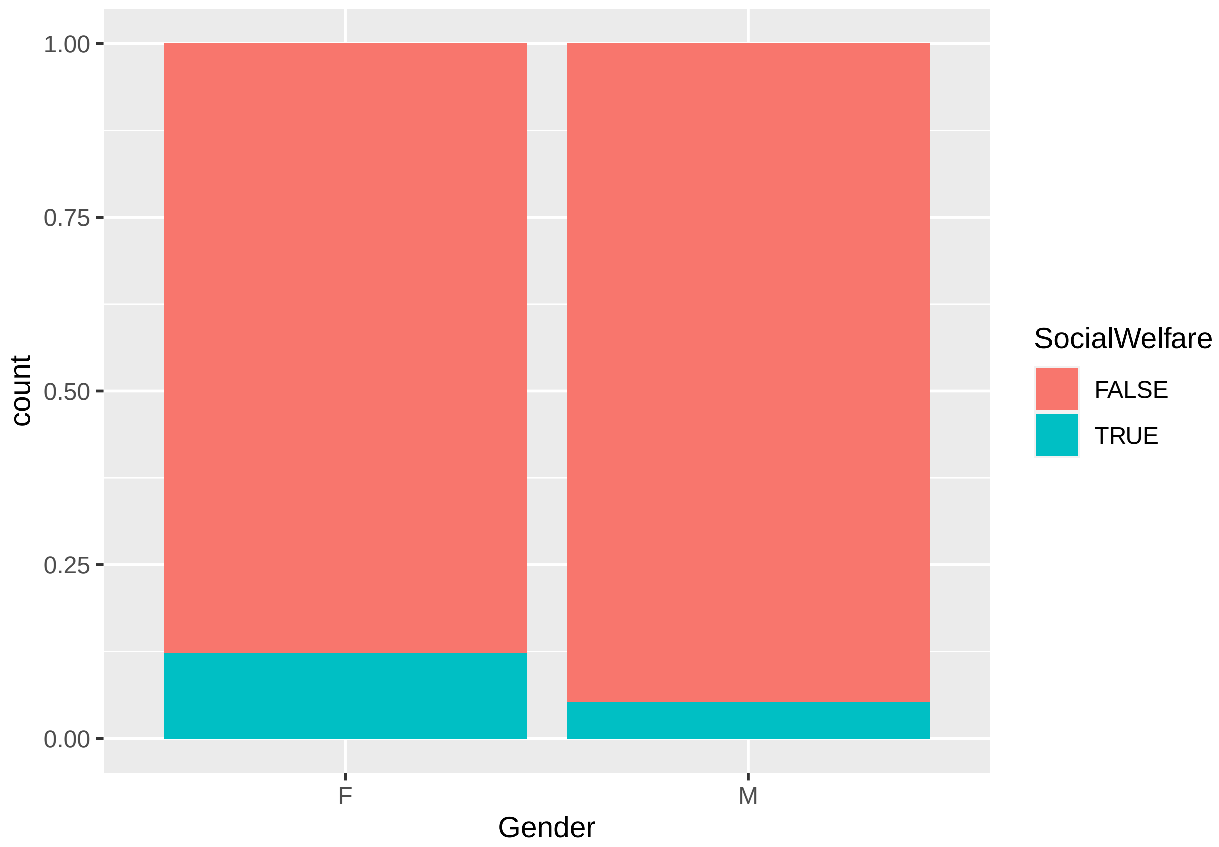
```
#Insert plot
ggplot(raw.data) +
  geom_bar(aes(x=Gender, fill=SocialWelfare))
```



```
  ggtitle("Gender by socialwelfare")
```

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

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



```
  ggtitle("Gender by socialwelfare")
```

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

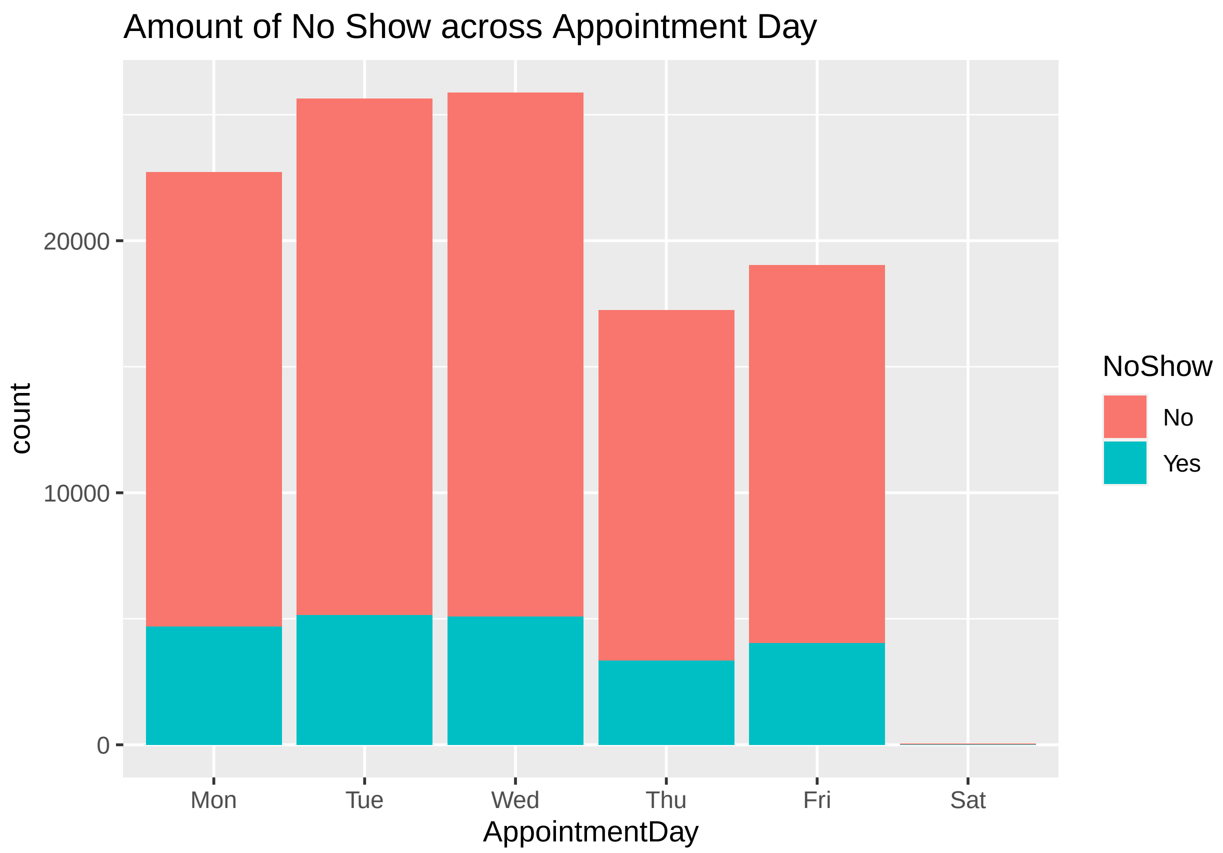
Far more exploration could still be done, including dimensionality reduction approaches but although we have found some patterns there is no major/striking patterns on the data as it currently stands.

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

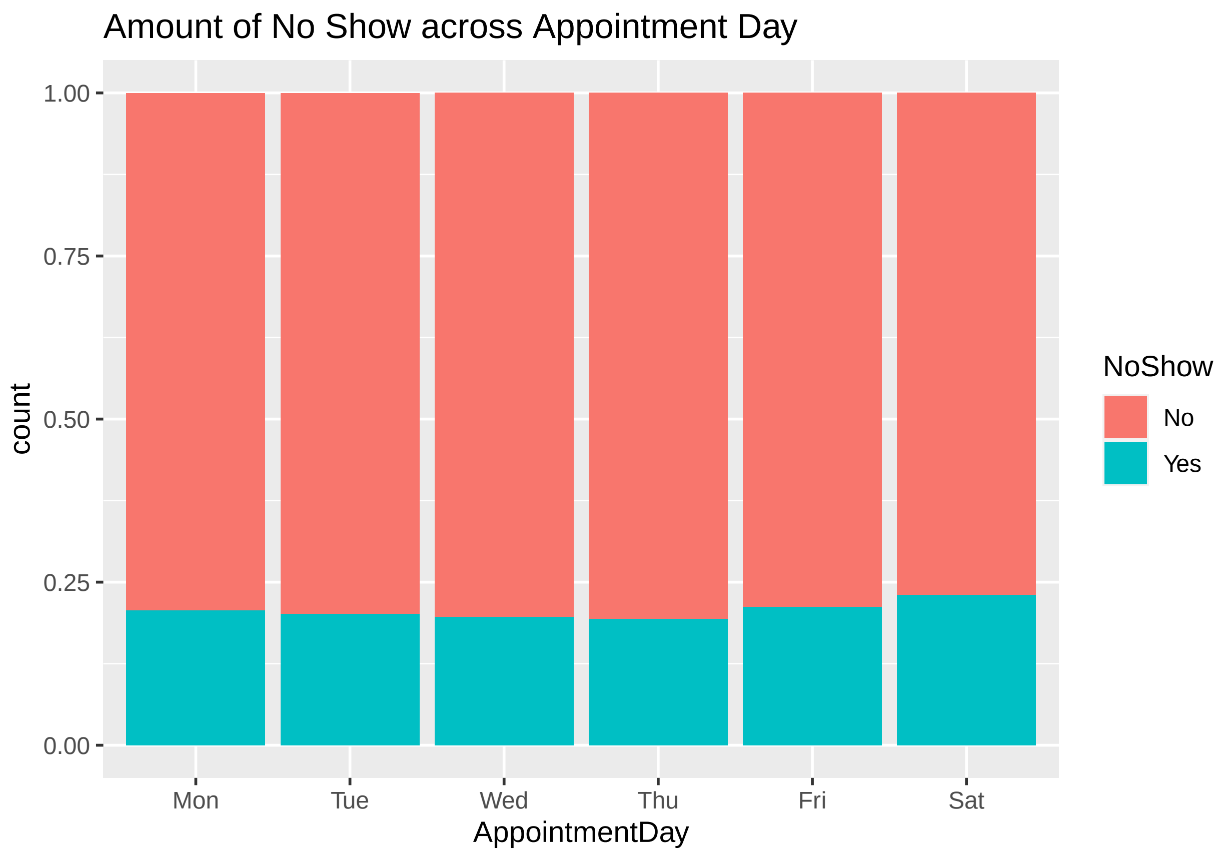
Feature Engineering

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")
```



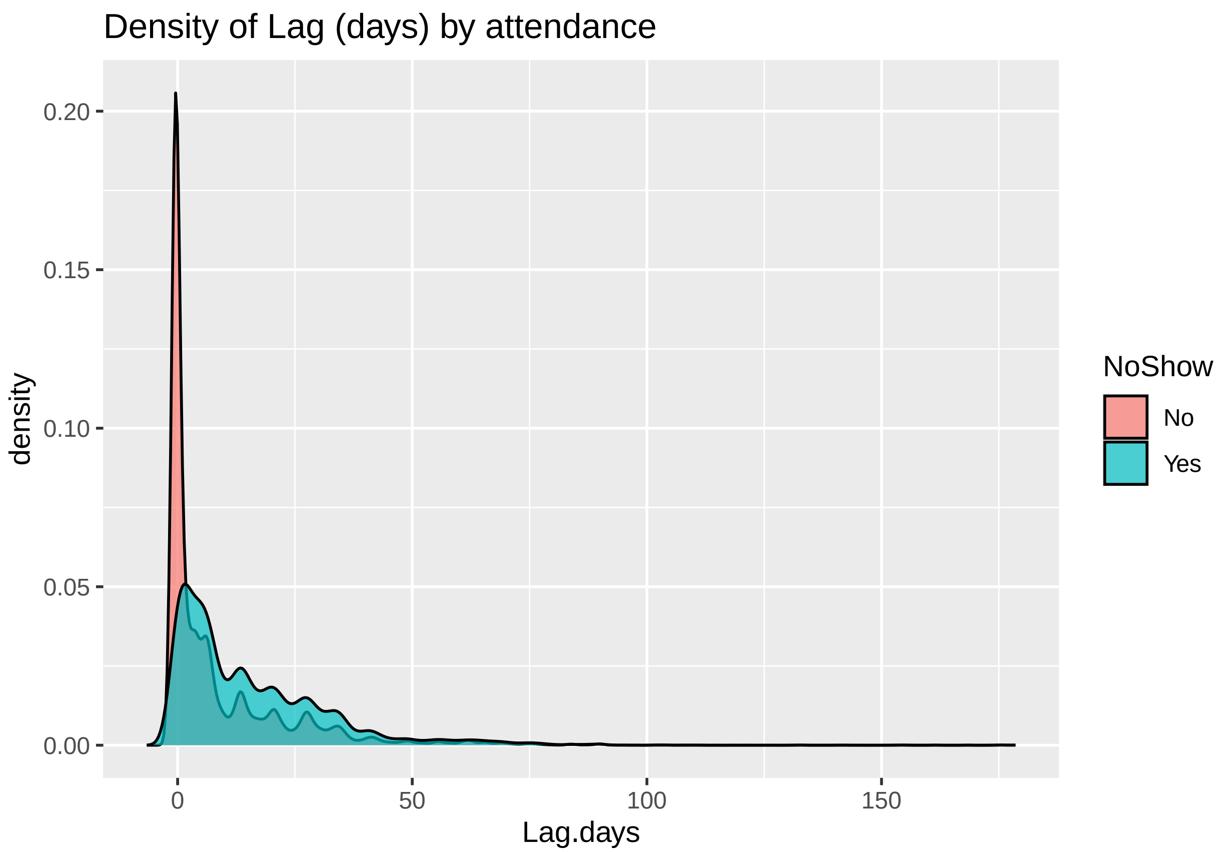
```
ggplot(raw.data) +
  geom_bar(aes(x=AppointmentDay, fill=NoShow), position = 'fill') +
  ggtitle("Amount of No Show across Appointment Day")
```



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?

there is a spike in the amount of no around 0 days and thats due to the fact that it was the day that appointments were scheduled and since it was scheduled that day there will be a higher spike, in no's, compared to yes some maybe waiting before agreeing to those dates of when they should be due for their actual appointment.

The only thing odd here is the over powering of no on the first couple of days.

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?

I think that it will work well, based on the EDA we already understand the datasets so creating a prediction for NoShow I believe it will work. We already know the corrlation with NoShow and the other variables.

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")
```

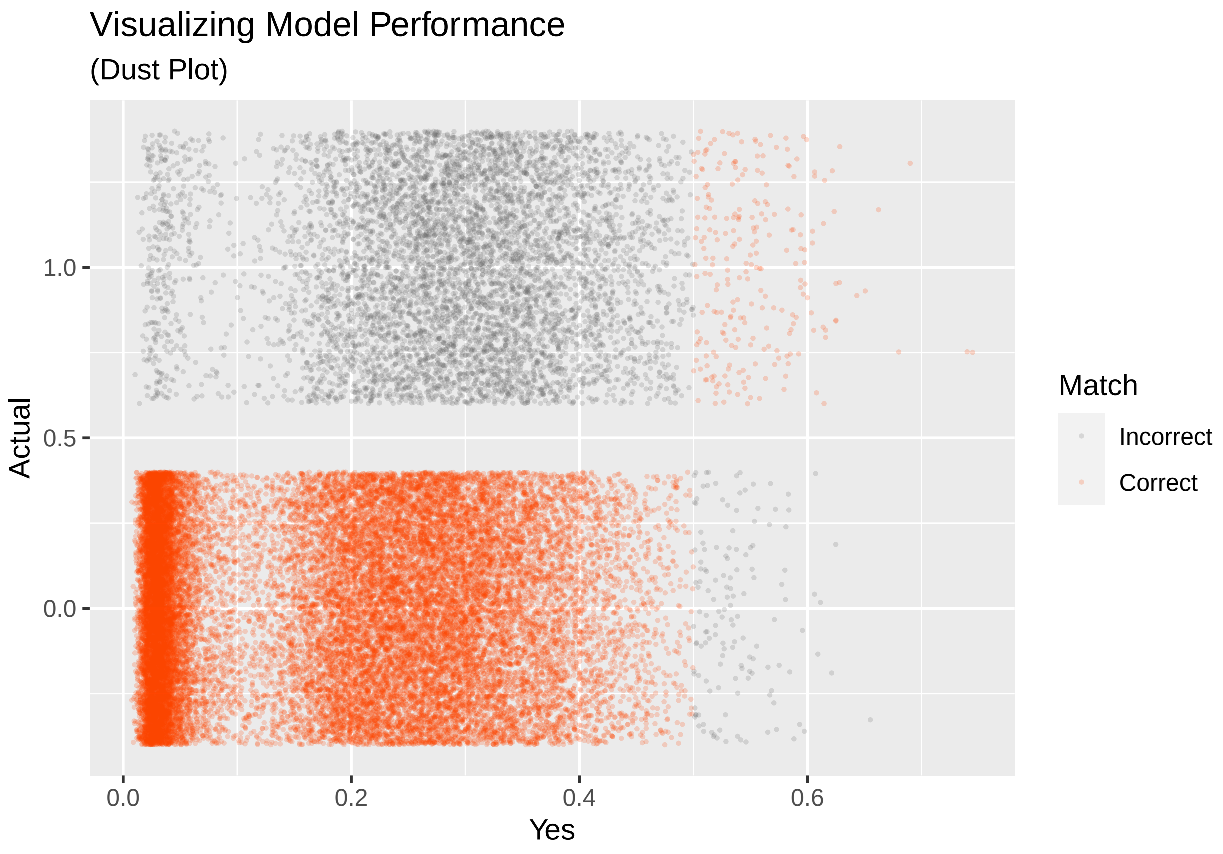
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    No   Yes
##        No  26385  6390
##        Yes   142   242
##
##                Accuracy : 0.803
##                  95% CI : (0.7987, 0.8073)
##     No Information Rate : 0.8
##     P-Value [Acc > NIR] : 0.08578
##
##                   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"
```

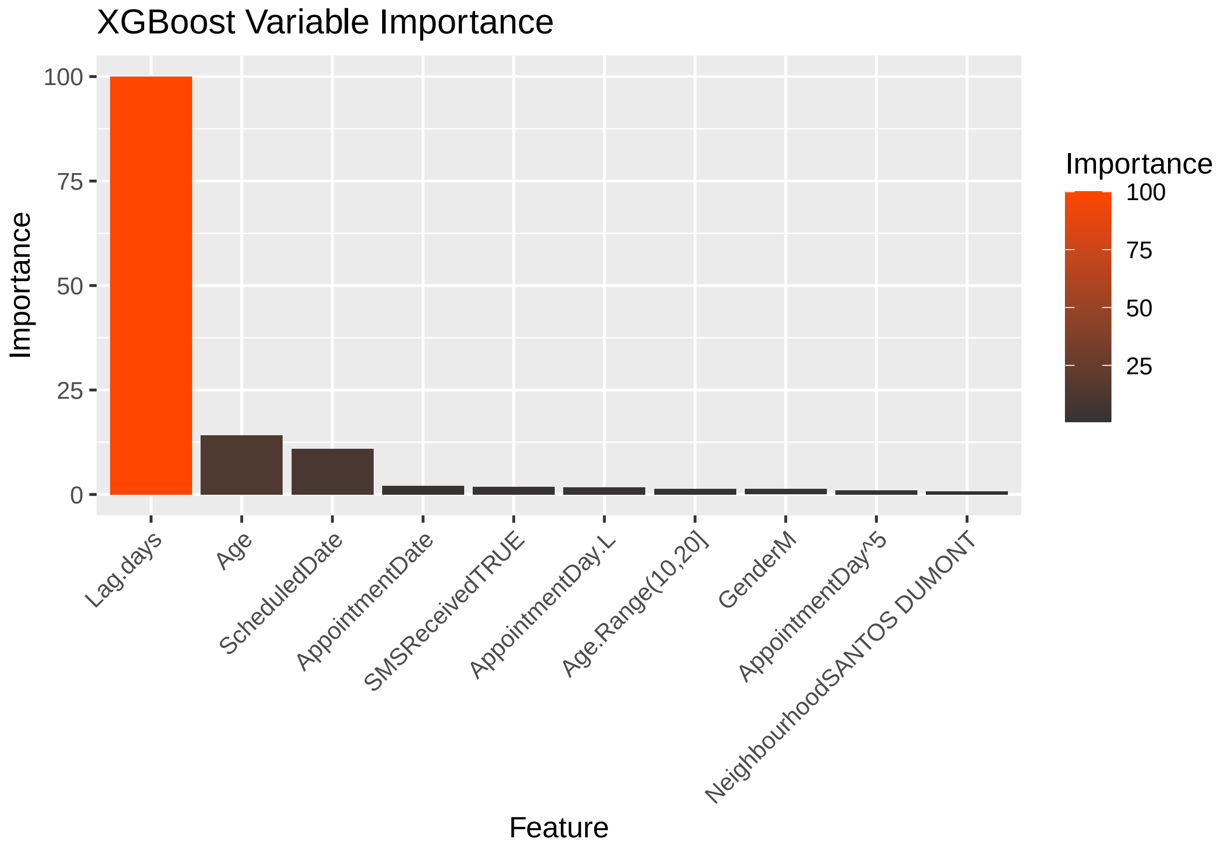
This isn't an unreasonable performance, but let's look a bit more carefully at the correct and incorrect predictions,

```
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.

(MY VISUAL DATA WAS NOT WORKING OR SHOWING I HAD TO REFERENCE THE CLASS HTML FILE VERSION) We can run fit to evaluate the ML model on this data for recall this can look at the "total actual positive" referencing article. This method would look at the extra positive that the data must have missed.

Reference: https://towardsdatascience.com/various-ways-to-evaluate-a-machine-learning-models-performance-230449055f15

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

Yes and no, I think based on these dataset, having seen the visuals and understanding the meaning of each variables/features, and the correlation between them, does make it more understandable. I can say I do trust the data analyses, however I am also not going to say a 100% that everything about these datasets are fully accurate. Overall seeing the backend of it helps compared to maybe just looking at the visuals.

Credits

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