Assignment 1

Rashik Mahmud Orchi-B00968298

26/05/2024

```
rm(list = ls())
```

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.

Variable	Description	Data Type
PatientID	Unique identifier for each patient	Integer
AppointmentID	Unique identifier for each appointment	Integer
Gender	Patient Gender (Male/Female)	Categorical
ScheduledDate	Date on which the appointment was scheduled	Date
AppointmentDate	Date of the actual appointment	Date
Age	Patient age	Integer
Neighbourhood	District of Vitória in which the appointment was scheduled	Categorical
SocialWelfare	Patient is a recipient of Bolsa Família welfare payments(0/1)	Binary
Hypertension	Patient previously diagnosed with hypertension (0/1)	Binary
Diabetes	Patient previously diagnosed with diabetes (0/1)	Binary
AlcoholUseDisorder	Patient previously diagnosed with alcohol use disorder (0/1)	Binary
Disability	Patient previously diagnosed with a disability (severity rated 0-4)	Ordinal

Variable	Description	Data Type
SMSReceived	At least 1 reminder text sent before appointment (0/1)	Binary
NoShow	Patient did not attend scheduled appointment (Yes/No)	Boolean

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

1. Patients with alcohol use disorder are more likely to miss medical appointments.
2. Patients with chronic health conditions, such as hypertension or diabetes, are more likely to miss medical appointments due to the cumulative burden of managing multiple health issues.
3. Patients who did not receive an SMS reminder are more likely to miss their medical appointments compared to those who received at least one reminder text.

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. The dataset includes the variable Disability on a Likert scale (0-4) but does not provide definitions for each level. This ambiguity makes it difficult to interpret the severity of the disability and its potential impact on missed appointments.
2. Variables such as SocialWelfare, Hypertension, Diabetes, AlcoholUseDisorder, and SMSReceived are coded as 0 and 1 without contextual definitions. This lack of context makes it challenging to understand what each code represents (e.g., whether 0 means absence or non-receipt and 1 means presence or receipt) and how these factors might influence appointment attendance.
3. The dataset does not specify the type of medical appointment associated with each AppointmentID. This missing information is critical as different types of appointments (e.g., routine check-up vs. urgent care) may have varying levels of importance to patients, affecting the likelihood of no-shows.
4. The dataset lacks critical information regarding the socioeconomic status of patients, such as income level, education level, employment status, and household composition.

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("https://raw.githubusercontent.com/maguire-lab/health_data_science_research/master
```

```
## ‘curl‘ package not installed, falling back to using ‘url()‘
## Rows: 110527 Columns: 14
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (3): Gender, Neighbourhood, NoShow
## dbl  (9): PatientID, AppointmentID, Age, SocialWelfare, Hypertension, Diabet...
## dttm (2): ScheduledDate, AppointmentDate
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```
#raw.data <- readr::read_csv('https://raw.githubusercontent.com/maguire-lab/health_data_science_researc
```

```
head(raw.data)
```

```
## # A tibble: 6 x 14
##   PatientID AppointmentID Gender ScheduledDate       AppointmentDate       Age
##       <dbl>         <dbl> <chr>  <dttm>              <dttm>              <dbl>
## 1   2.99e13       5642903 F      2016-04-29 18:38:08 2016-04-29 00:00:00    62
## 2   5.59e14       5642503 M      2016-04-29 16:08:27 2016-04-29 00:00:00    56
## 3   4.26e12       5642549 F      2016-04-29 16:19:04 2016-04-29 00:00:00    62
## 4   8.68e11       5642828 F      2016-04-29 17:29:31 2016-04-29 00:00:00     8
## 5   8.84e12       5642494 F      2016-04-29 16:07:23 2016-04-29 00:00:00    56
## 6   9.60e13       5626772 F      2016-04-27 08:36:51 2016-04-29 00:00:00    76
## # i 8 more variables: Neighbourhood <chr>, SocialWelfare <dbl>,
## #   Hypertension <dbl>, Diabetes <dbl>, AlcoholUseDisorder <dbl>,
## #   Disability <dbl>, SMSReceived <dbl>, NoShow <chr>
```

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
##       <dbl>         <dbl> <chr>  <dttm>              <dttm>              <dbl>
## 1   3.20e13       5700278 F      2016-05-16 09:17:44 2016-05-19 00:00:00   115
## 2   3.20e13       5700279 F      2016-05-16 09:17:44 2016-05-19 00:00:00   115
## 3   3.20e13       5562812 F      2016-04-08 14:29:17 2016-05-16 00:00:00   115
## 4   3.20e13       5744037 F      2016-05-30 09:44:51 2016-05-30 00:00:00   115
## 5   7.48e14       5717451 F      2016-05-19 07:57:56 2016-06-03 00:00:00   115
## # i 8 more variables: Neighbourhood <chr>, SocialWelfare <dbl>,
## #   Hypertension <dbl>, Diabetes <dbl>, AlcoholUseDisorder <dbl>,
## #   Disability <dbl>, SMSReceived <dbl>, NoShow <chr>
```

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

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

```
summary(raw.data$Age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   -1.00   18.00   37.00   37.09   55.00  115.00
```

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

```
## # A tibble: 1 x 14
##   PatientID AppointmentID Gender ScheduledDate       AppointmentDate       Age
##       <dbl>         <dbl> <chr>  <dttm>              <dttm>              <dbl>
## 1   4.66e14       5775010 F      2016-06-06 08:58:13 2016-06-06 00:00:00    -1
## # i 8 more variables: Neighbourhood <chr>, SocialWelfare <dbl>,
## #   Hypertension <dbl>, Diabetes <dbl>, AlcoholUseDisorder <dbl>,
## #   Disability <dbl>, SMSReceived <dbl>, NoShow <chr>
```

Since age cannot be negative, values lower than 0 are invalid data points. Therefore, there is only one individual with impossible age and we will remove these entries from our dataset.

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

```r
summary(raw.data$Age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    0.00   18.00   37.00   37.09   55.00  115.00
```

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:

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

```
## # A tibble: 1 x 3
##   Hypertension Diabetes AlcoholUseDisorder
##          <dbl>    <dbl>              <dbl>
## 1            0        0                  0
```

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

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

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

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

```r
appointments_per_patient <- raw.data %>%
  group_by(PatientID) %>%
  summarise(AppointmentCount = n()) %>%
  arrange(desc(AppointmentCount))
appointments_per_patient
```

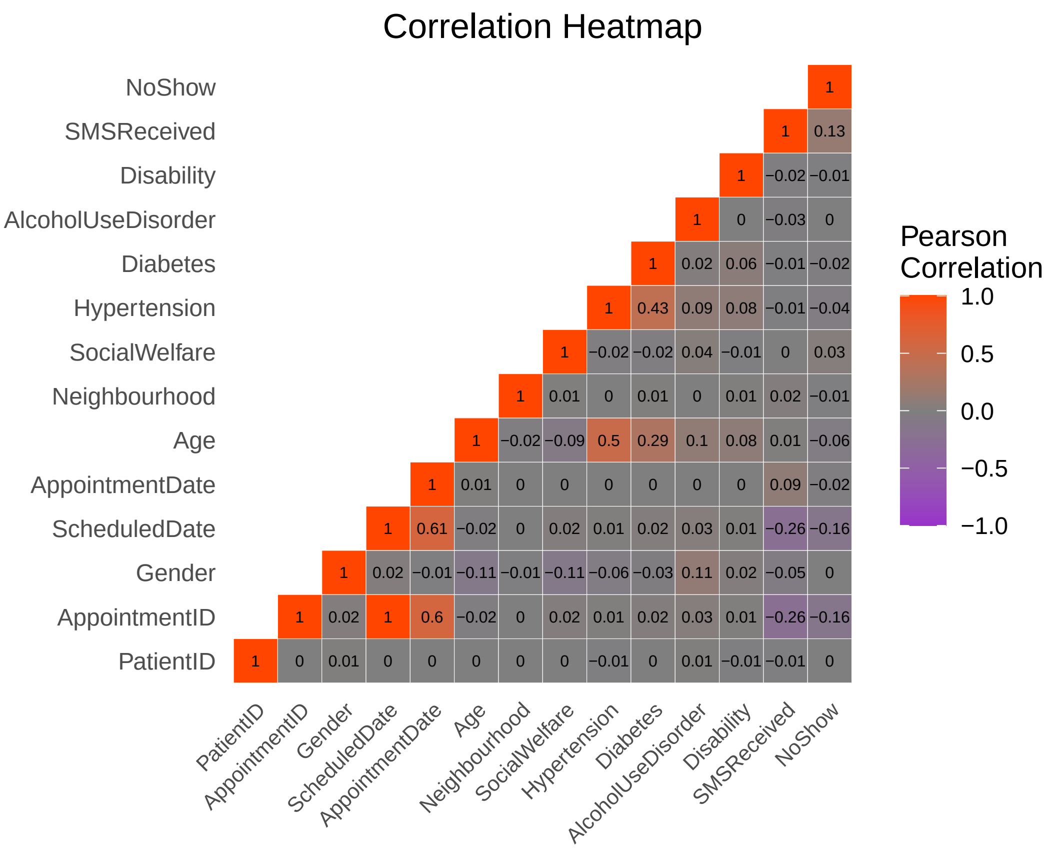
```
## # A tibble: 62,298 x 2
##    PatientID AppointmentCount
##        <dbl>            <int>
##  1   8.22e14               88
##  2   9.96e10               84
##  3   2.69e13               70
##  4   3.35e13               65
##  5   2.58e11               62
##  6   6.26e12               62
##  7   7.58e13               62
##  8   8.71e14               62
##  9   6.68e13               57
## 10   8.72e11               55
## # i 62,288 more rows
```

So the highest appoinment count is 88 by the pateint ID 8.221459e+14.

Let's explore the correlation between variables:

```r
raw.data$Gender <- as.factor(raw.data$Gender)
raw.data$Neighbourhood <- as.factor(raw.data$Neighbourhood)
raw.data$NoShow <- as.factor(raw.data$NoShow)
```

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

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


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

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

  upper_tri <- get_upper_tri(cor_matrix_raw)

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

  cor_graph
}

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

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



	PatientID	AppointmentID	Gender	ScheduledDate	AppointmentDate	Age	Neighbourhood	SocialWelfare	Hypertension	Diabetes	AlcoholUseDisorder	Disability	SMSReceived	NoShow
NoShow														1
SMSReceived													1	0.13
Disability												1	−0.02	−0.01
AlcoholUseDisorder											1	0	−0.03	0
Diabetes										1	0.02	0.06	−0.01	−0.02
Hypertension									1	0.43	0.09	0.08	−0.01	−0.04
SocialWelfare								1	−0.02	−0.02	0.04	−0.01	0	0.03
Neighbourhood							1	0.01	0	0.01	0	0.01	0.02	−0.01
Age						1	−0.02	−0.09	0.5	0.29	0.1	0.08	0.01	−0.06
AppointmentDate					1	0.01	0	0	0	0	0	0	0.09	−0.02
ScheduledDate				1	0.61	−0.02	0	0.02	0.01	0.02	0.03	0.01	−0.26	−0.16
Gender			1	0.02	−0.01	−0.11	−0.01	−0.11	−0.06	−0.03	0.11	0.02	−0.05	0
AppointmentID		1	0.02	1	0.6	−0.02	0	0.02	0.01	0.02	0.03	0.01	−0.26	−0.16
PatientID	1	0	0.01	0	0	0	0	0	−0.01	0	0.01	−0.01	−0.01	0

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

7 Which parameters most strongly correlate with missing appointments (`NoShow`)?
The absolute correlation values of varaibles with target variable(Noshow) is provided below:

```
selected_cols <- c("PatientID", "AppointmentID", "Gender", "ScheduledDate", "AppointmentDate",
                   "Age", "Neighbourhood", "SocialWelfare", "Hypertension", "Diabetes",
                   "AlcoholUseDisorder", "Disability", "SMSReceived", "NoShow")

selected_data <- raw.data %>%
  select(all_of(selected_cols))


numeric_data <- selected_data %>%
  mutate(across(everything(), as.numeric))

cor_matrix <- cor(numeric_data, use = "complete.obs")
noshow_correlations <- cor_matrix["NoShow", ]
noshow_correlations_df <- data.frame(
  Feature = as.character(names(noshow_correlations)),
  Correlation = as.numeric(abs(noshow_correlations)))


noshow_correlations_df <- noshow_correlations_df %>%
  filter(Feature != "NoShow")
```

```r
noshow_correlations_df <- noshow_correlations_df %>%
  arrange(desc(Correlation))

print(as.data.frame(noshow_correlations_df))
```

```
##                Feature  Correlation
## 1        AppointmentID 0.1625973850
## 2        ScheduledDate 0.1623391055
## 3          SMSReceived 0.1264279433
## 4                  Age 0.0603268227
## 5         Hypertension 0.0357035139
## 6        SocialWelfare 0.0291335766
## 7      AppointmentDate 0.0224019644
## 8             Diabetes 0.0151812329
## 9        Neighbourhood 0.0091197565
## 10          Disability 0.0060768466
## 11              Gender 0.0041219877
## 12           PatientID 0.0014556525
## 13 AlcoholUseDisorder 0.0001968498
```

So, AppointmentID, ScheduledDate, SMSReceived, Age has relatively strong correlation with NoShow.

8 Are there any other variables which strongly correlate with one another?
The absolute value of pairwise correlation of top ten variables is provided below:

```r
upper_triangle <- cor_matrix[upper.tri(cor_matrix, diag = FALSE)]


indices <- which(upper.tri(cor_matrix, diag = FALSE), arr.ind = TRUE)

cor_df <- data.frame(
  Variable1 = rownames(cor_matrix)[indices[, 1]],
  Variable2 = rownames(cor_matrix)[indices[, 2]],
  Correlation = upper_triangle
)

cor_df <- cor_df %>%
  filter(abs(Correlation) < 1)

cor_df$Correlation <- abs(cor_df$Correlation)


cor_df <- cor_df %>%
  arrange(desc(Correlation))

##Top 10 absolute values
print(head(cor_df,n = 10))
```

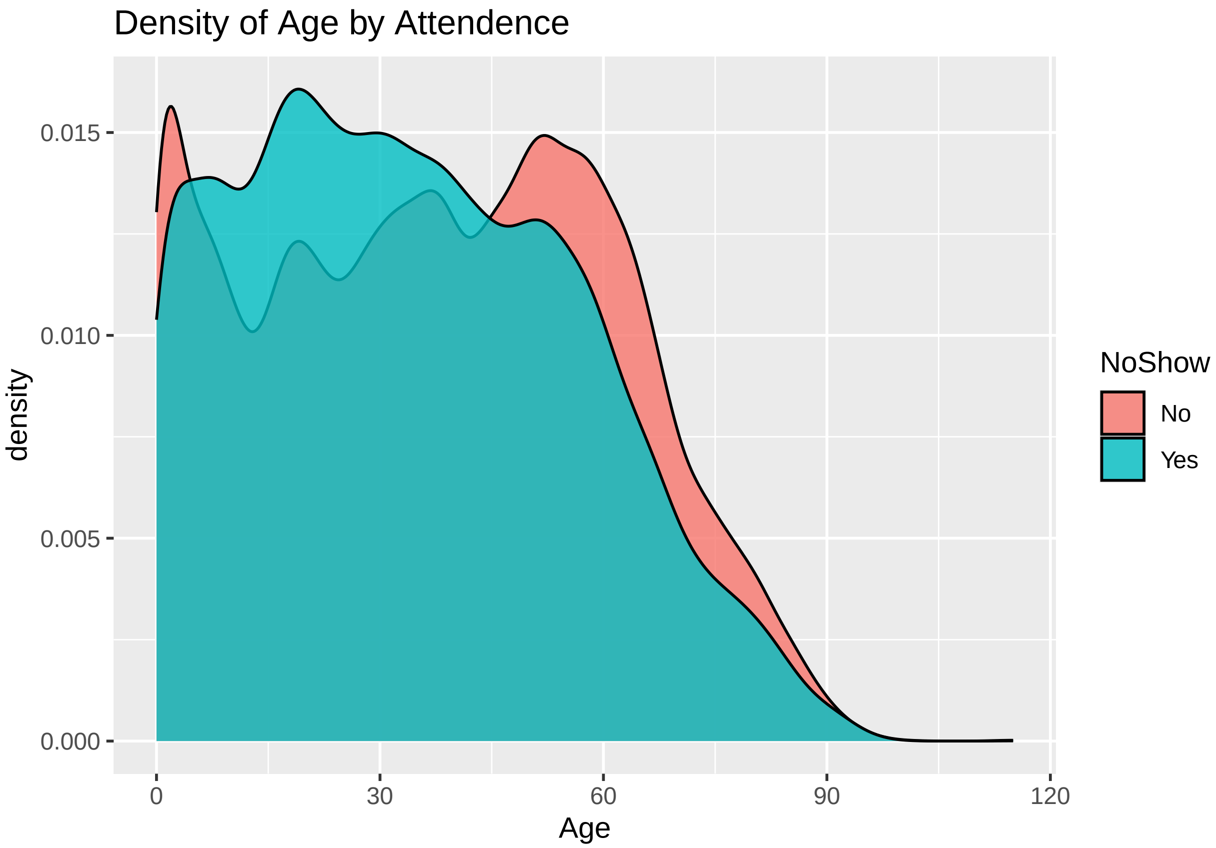
```
##        Variable1       Variable2 Correlation
## 1  AppointmentID   ScheduledDate   0.9983081
## 2  ScheduledDate AppointmentDate   0.6050459
## 3  AppointmentID AppointmentDate   0.6014186
## 4            Age    Hypertension   0.5045856
## 5   Hypertension        Diabetes   0.4330850
## 6            Age        Diabetes   0.2923910
## 7  ScheduledDate     SMSReceived   0.2572403
## 8  AppointmentID     SMSReceived   0.2566126
## 9  AppointmentID          NoShow   0.1625974
## 10 ScheduledDate          NoShow   0.1623391
```

Correlation between AppointmendID and ScheduledDate is very strong and almost close 1 which is an issue. So to avoid multcollinearity we should drop the AppointmendID variable. However there are other varibles which also aslo have realtivly strong correlation among themselves and free from multicollinearity threshold for example (ScheduledDate & AppoinmentDate),(Age& Hypertension), (Hypertension& Diabetes), (Age & Diabetes).

9 Do you see any issues with PatientID/AppointmentID being included in this plot?
Including PatientID/AppointmentID in the correlation plot can be problematic because these identifiers are unique to each patient or appointment and do not have a meaningful statistical relationship with other variables. They are more for record-keeping than analysis and could introduce noise into our analysis.

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

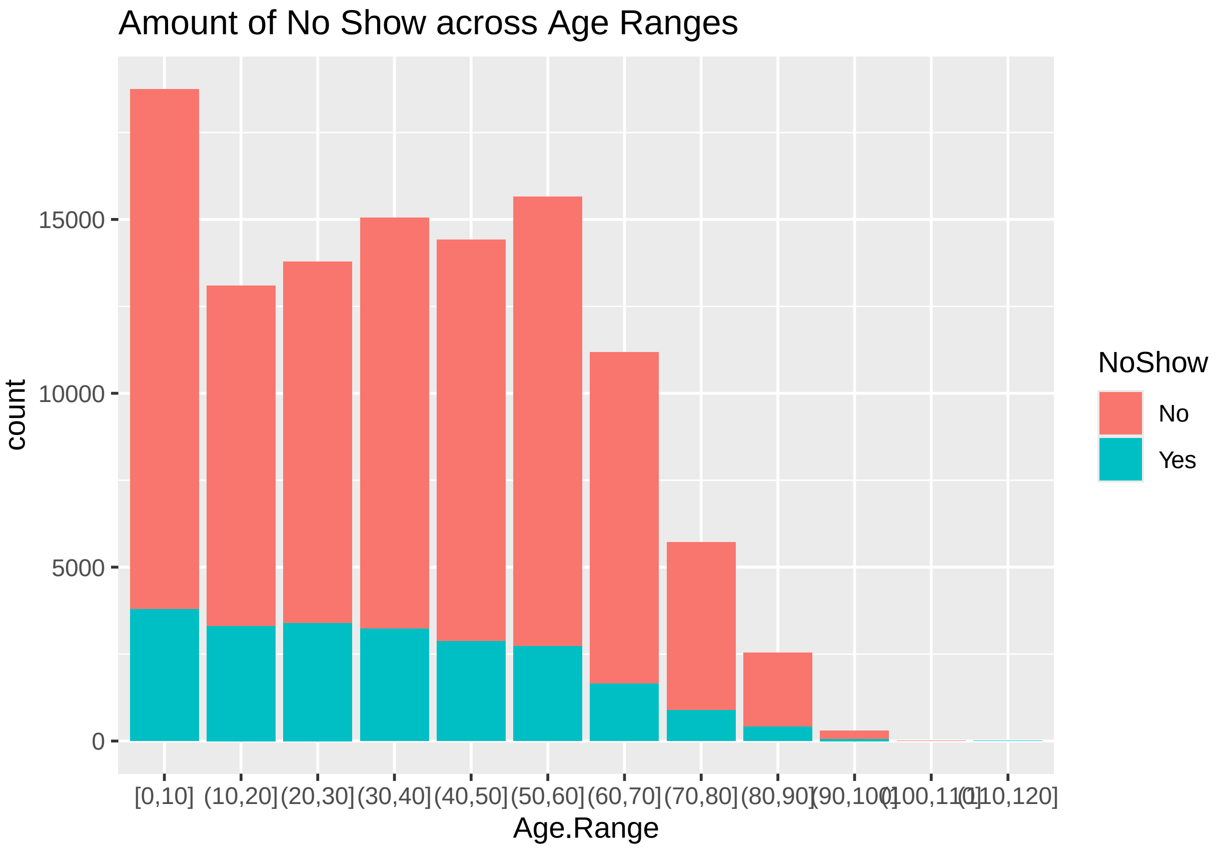
```
ggplot(raw.data) +
  geom_density(aes(x=Age, fill=NoShow), alpha=0.8) +
  ggtitle("Density of Age by Attendence")
```



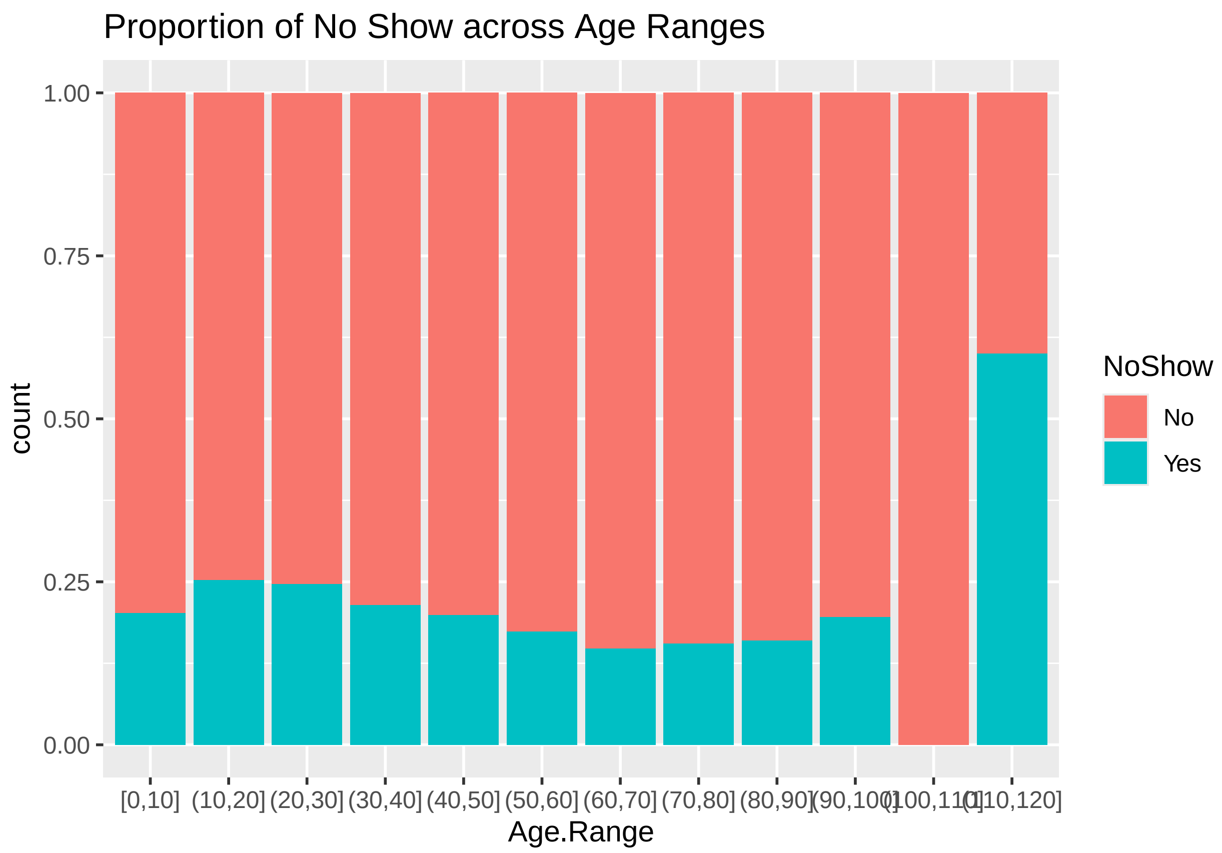
There does seem to be a difference in the distribution of ages of people that miss and don't miss appointments. However, the shape of this distribution means the actual correlation is near 0 in the heatmap above. This highlights the need to look at individual variables.

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?

First, if we only look at Figure 1, which shows the number of missed appointments by age group, we might conclude that younger individuals (especially those aged 0-10) miss the most appointments. This could lead us to believe that the likelihood of missing appointments decreases as age increases, due to the downward-sloping trend in the plot. However, this plot doesn't account for the relative sizes of the age groups, so the high number of missed appointments in the younger age group could simply reflect that there are more individuals in that group.

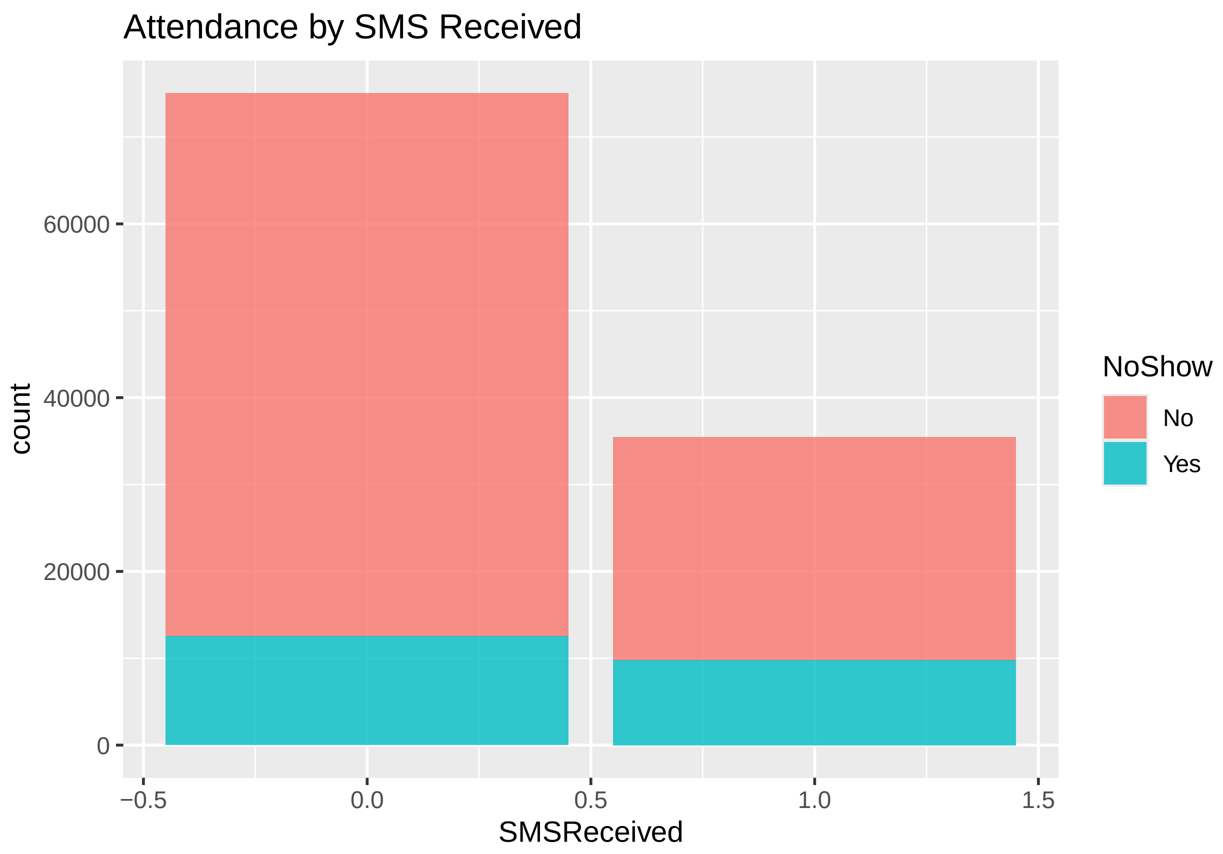
On the other hand, if we only look at Figure 2, which shows the proportion of missed appointments by age group, we might observe that individuals aged between 110 and 120 miss appointments most proportionally. This could suggest that this age group is at the highest risk of missing appointments. However, without the context provided by the actual number of individuals in each age group (as seen in Figure 1), you might not realize that there are very few people in the 110-120 age group, making this proportion less impactful on the overall pattern.

To fully understand the attendance patterns across different age groups, it is crucial to consider both plots together. The first plot shows the raw number of missed appointments, highlighting the absolute scale, while the second plot provides the proportion of missed appointments, offering insight into the relative likelihood within each age group. Only by examining both plots can we can get a complete picture of how age affects appointment attendance.

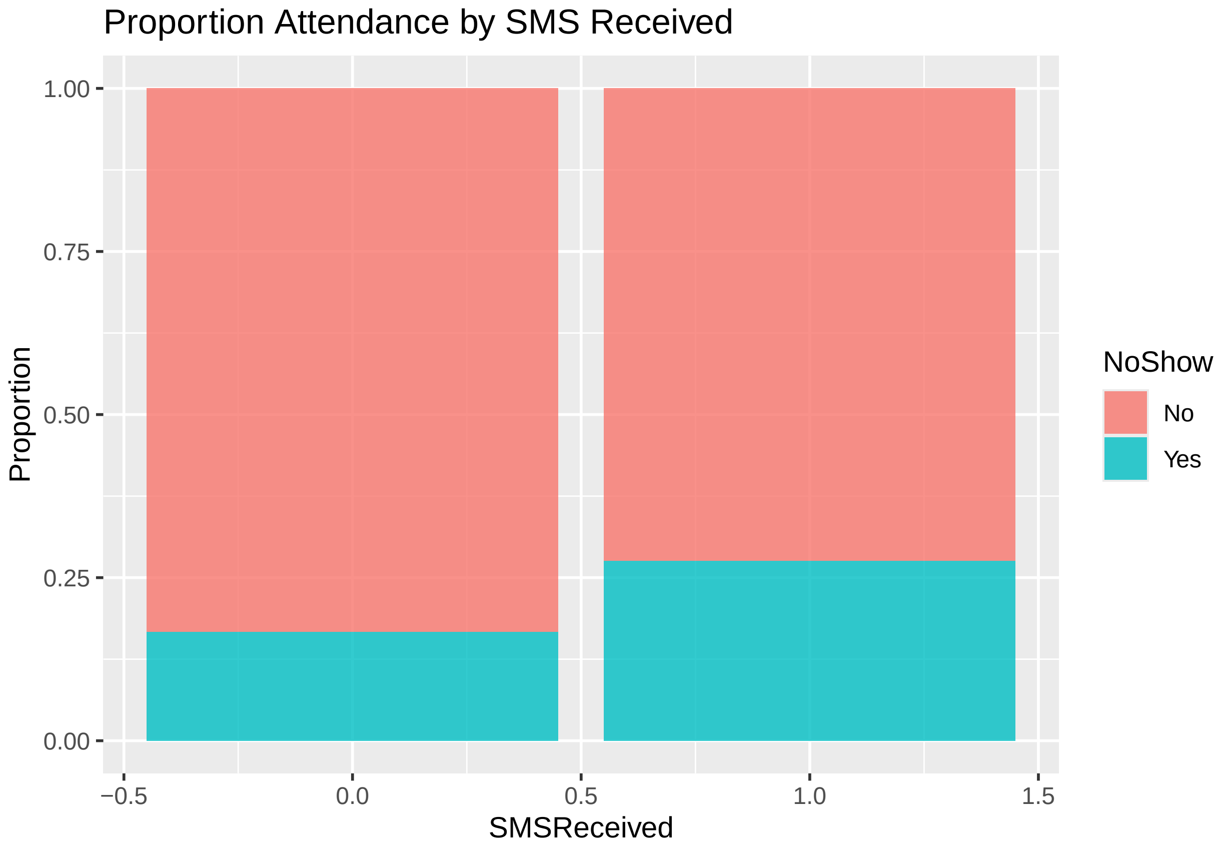
The key takeaway from this is that number of individuals > 90 are very few from plot 1 so probably are very small so unlikely to make much of an impact on the overall distributions. However, other patterns do emerge such as 10-20 age group is nearly twice as likely to miss appointments as the 60-70 years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

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

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



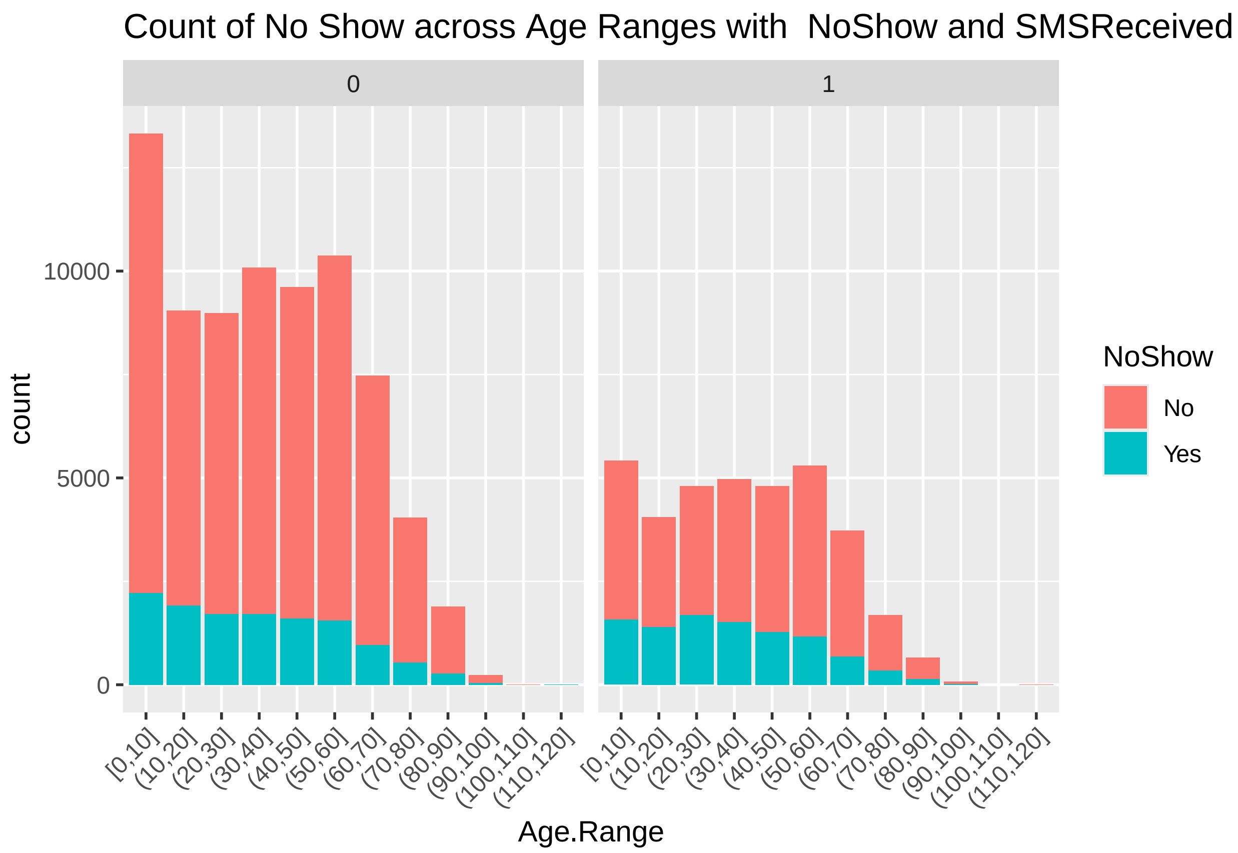
```
ggplot(raw.data) +
  geom_bar(aes(x=SMSReceived, fill=NoShow), position='fill', alpha=0.8) +
  ggtitle("Proportion Attendance by SMS Received")+
  ylab("Proportion")
```



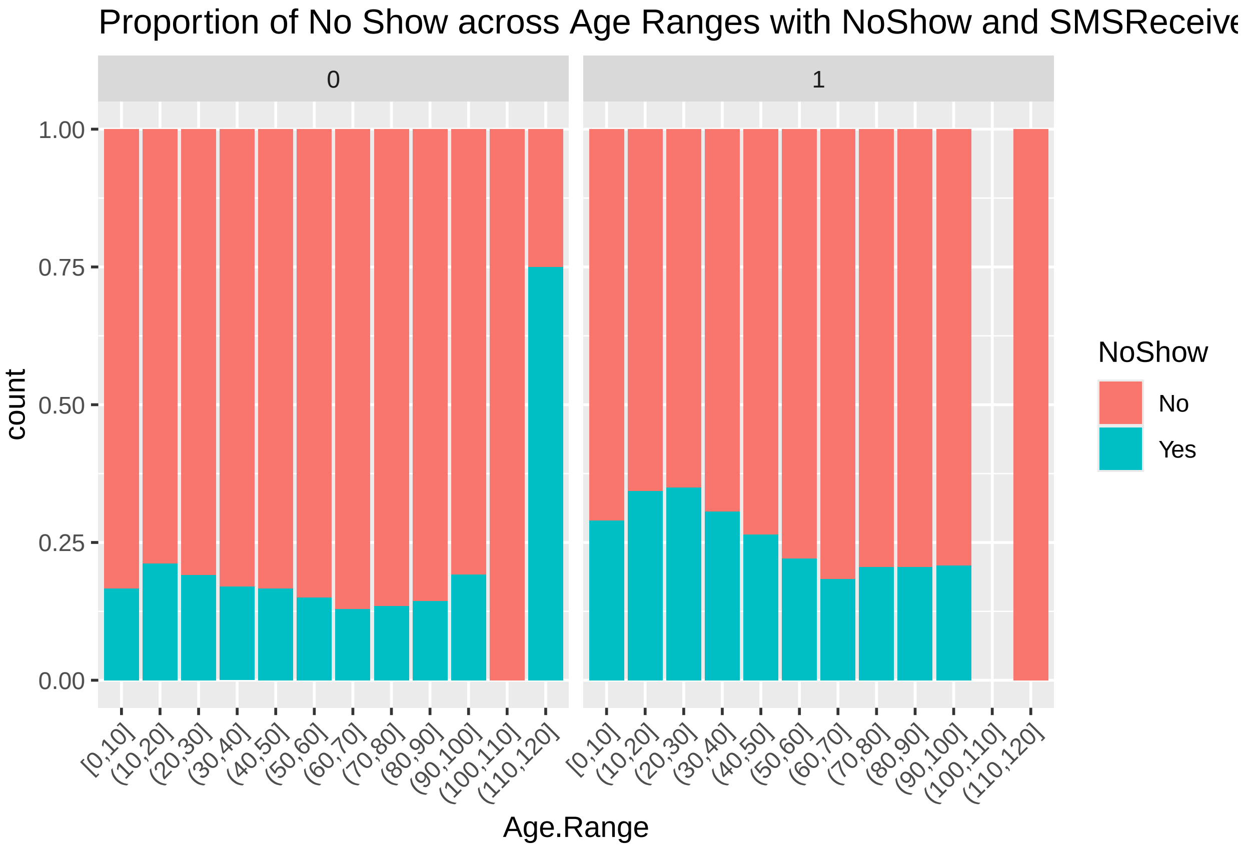
11 From this plot does it look like SMS reminders increase or decrease the chance of someone not attending an appointment? Why might the opposite actually be true (hint: think about biases)?

From the above the plot it *doesn't* look like that SMS reminders increase or decrease the chance of someone not attending an appointment. Actually, Since the data set didn't provide any context about what 0 and 1 means explicitly, I am assuming that 0 represents who didn't receive SMS reminders and 1 who received. Though the number of patients missing appointments who didn't receive SMS reminder are slightly higher but if we look the proportion in plot 2, the appointments missed by the patients who received SMS reminder are higher .However, The difference is not significant enough to conclude that SMS reminders have a substantial impact on appointment attendance or non-attendance.

```r
ggplot(raw.data) +
  geom_bar(aes(x = Age.Range, fill = NoShow)) +
  facet_wrap(~ SMSReceived) +
  ggtitle("Count of No Show across Age Ranges with  NoShow and SMSReceived")+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```r
# Second plot
ggplot(raw.data) +
  geom_bar(aes(x = Age.Range, fill = NoShow), position = 'fill') +
  facet_wrap(~ SMSReceived) +
  ggtitle("Proportion of No Show across Age Ranges with NoShow and SMSReceived ")+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



The opposite might actually be true for following reason:

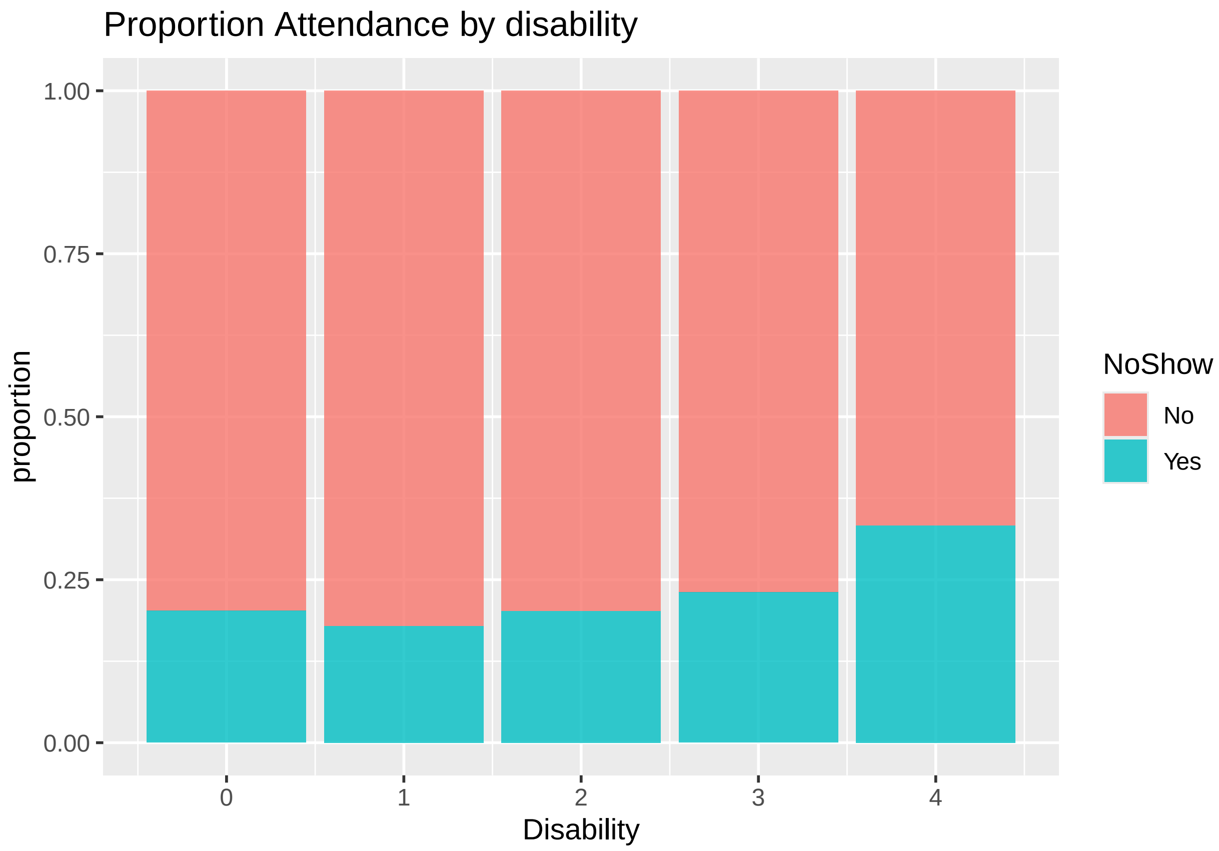
1. Patients might not have a cell phone to receive SMS reminders due to age or socioeconomic conditions. Additionally, patients who are not tech-savvy might not check their texts regularly, diminishing the effectiveness of the reminders.
2. SMS reminders might not be sent uniformly across all patient groups. For example, reminders could be more frequently sent to patients with a history of missing appointments or to younger, more tech-savvy demographics. These groups might have different inherent attendance behaviors regardless of receiving an SMS reminder.
3. The data might not accurately reflect all missed appointments. There could be instances where patients do not report whether they received an SMS reminder, or there could be inaccuracies in record-keeping. This can lead to misleading conclusions about the effectiveness of SMS reminders.

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

```
table(raw.data$Disability,raw.data$NoShow)
```

```
##
##          No   Yes
##   0 86373 21912
##   1  1676   366
##   2   146    37
##   3    10     3
##   4     2     1
```

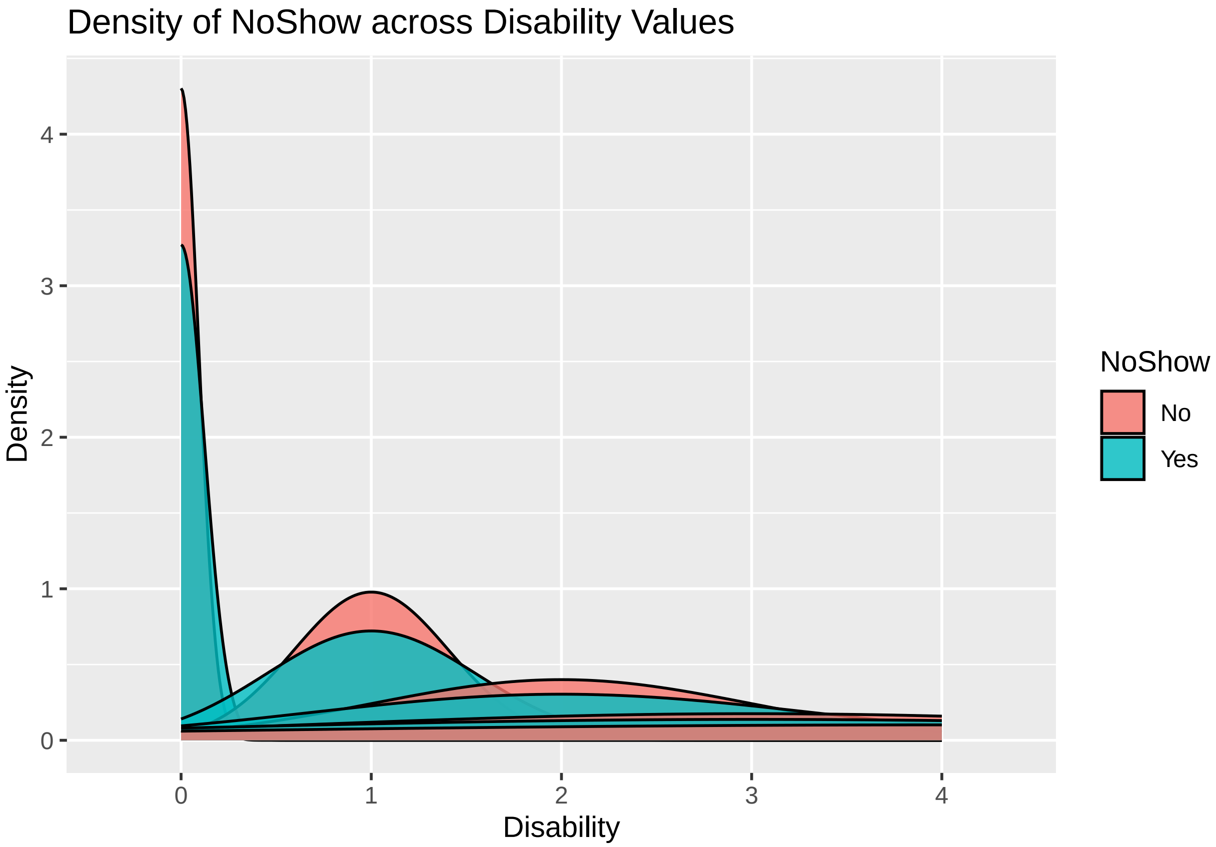
```
ggplot(raw.data) +
  geom_bar(aes(x=Disability, fill=NoShow), position='fill', alpha=0.8) +
  ggtitle("Proportion Attendance by disability") +
  ylab("proportion")
```



```
raw.data$NoShow <- factor(raw.data$NoShow)
raw.data$Disability <- factor(raw.data$Disability)


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

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

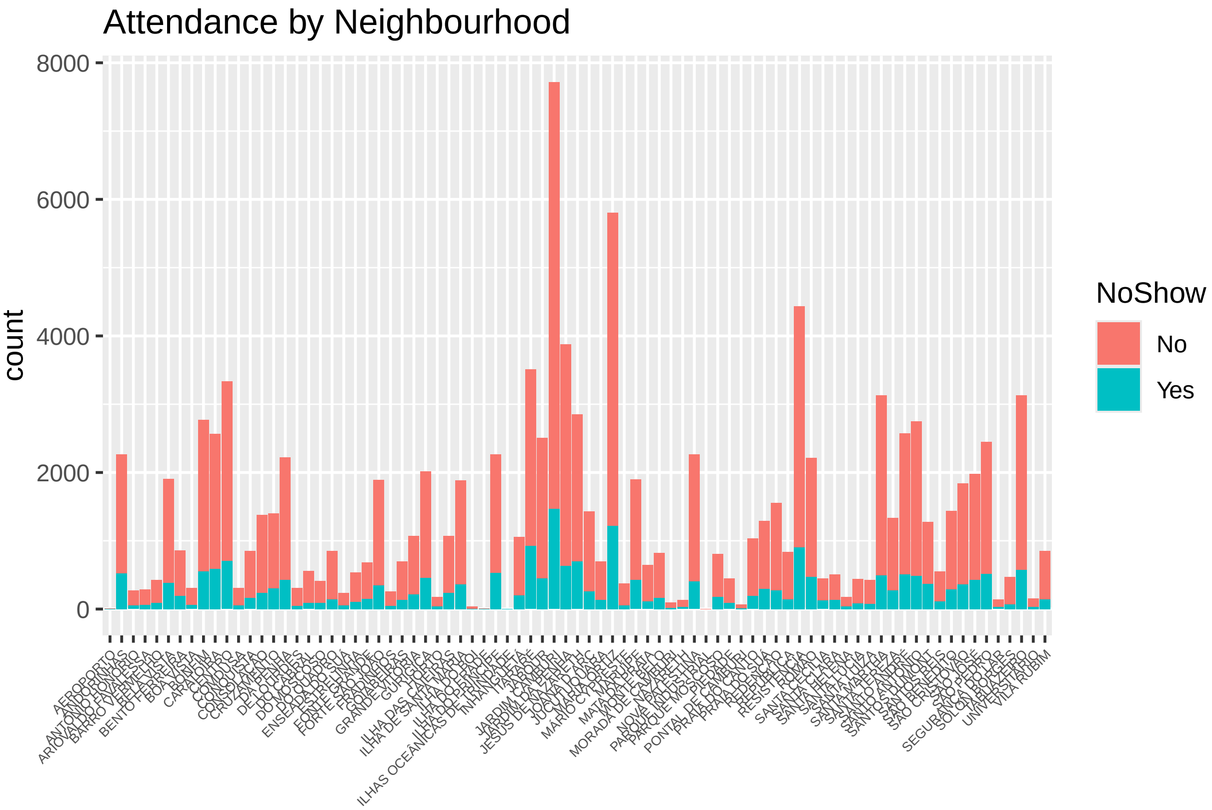
```
## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning
## -Inf
```



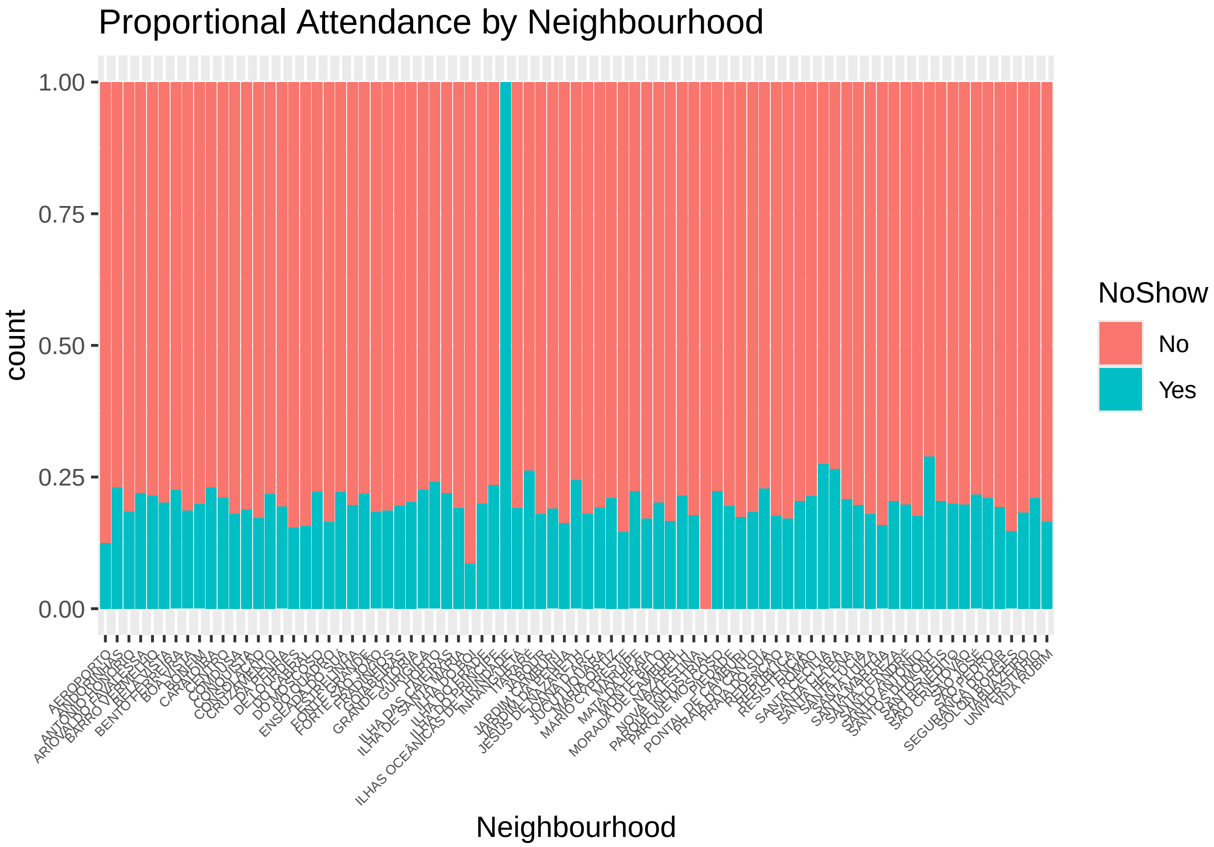
From the density plot we can see can that there is high peak around (Disability==0)

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')
```



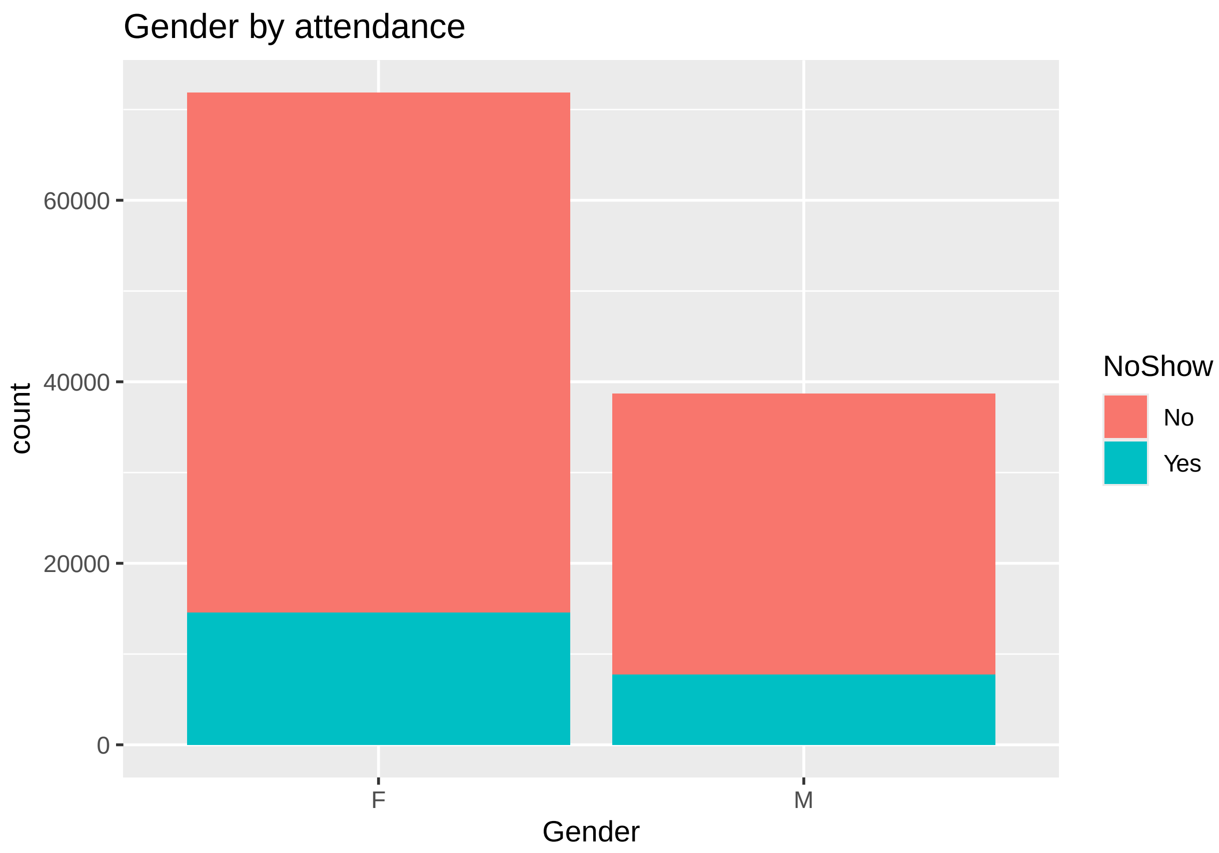
Most neighborhoods have similar proportions of no-show but some have much higher and lower rates.

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

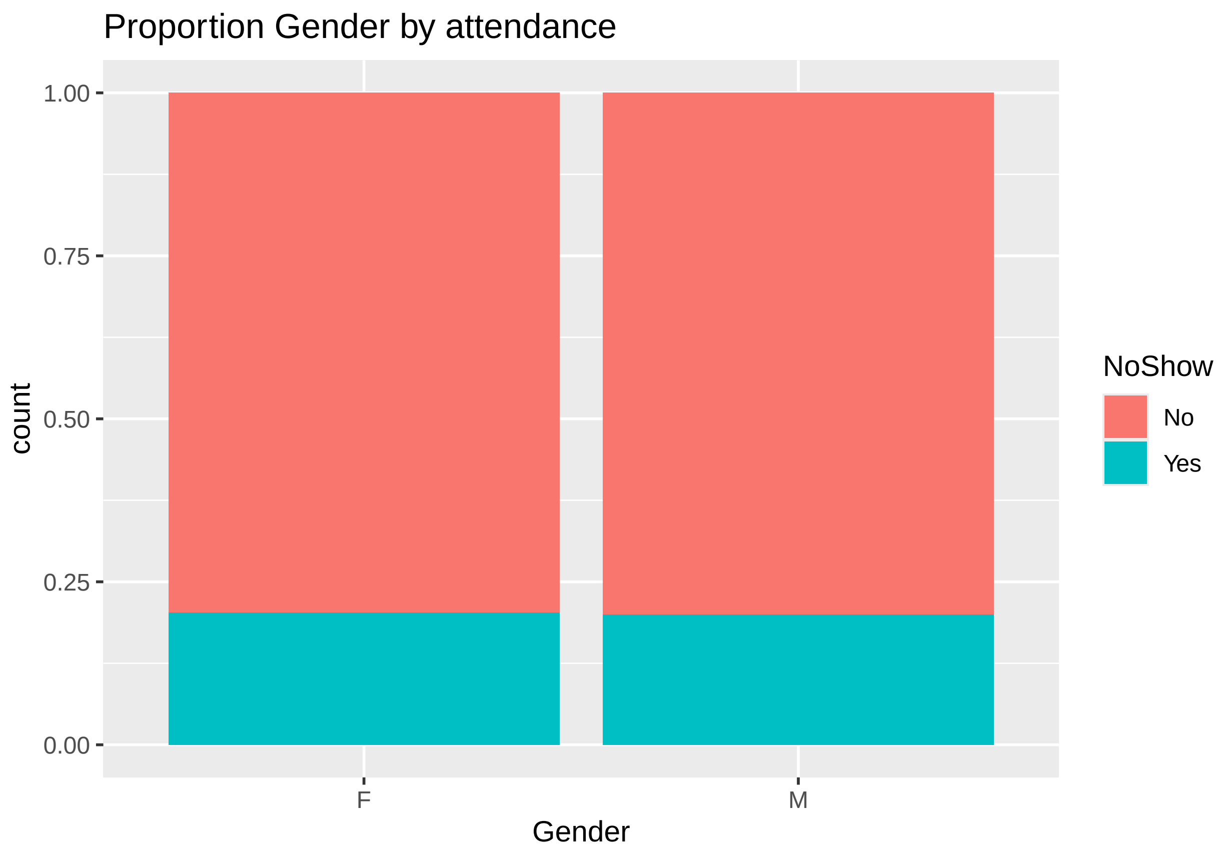
The socio-economic status can be reason for for differences in attendance rates across neighbourhoods.Neighborhoods with higher levels of poverty may have lower attendance rates due to financial barriers. Residents might struggle with transportation costs, childcare, or taking time off work to attend appointments. There could be other factors such as the quality of healthcare facility and the location of healthcare facility across neighbourhoods.

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

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

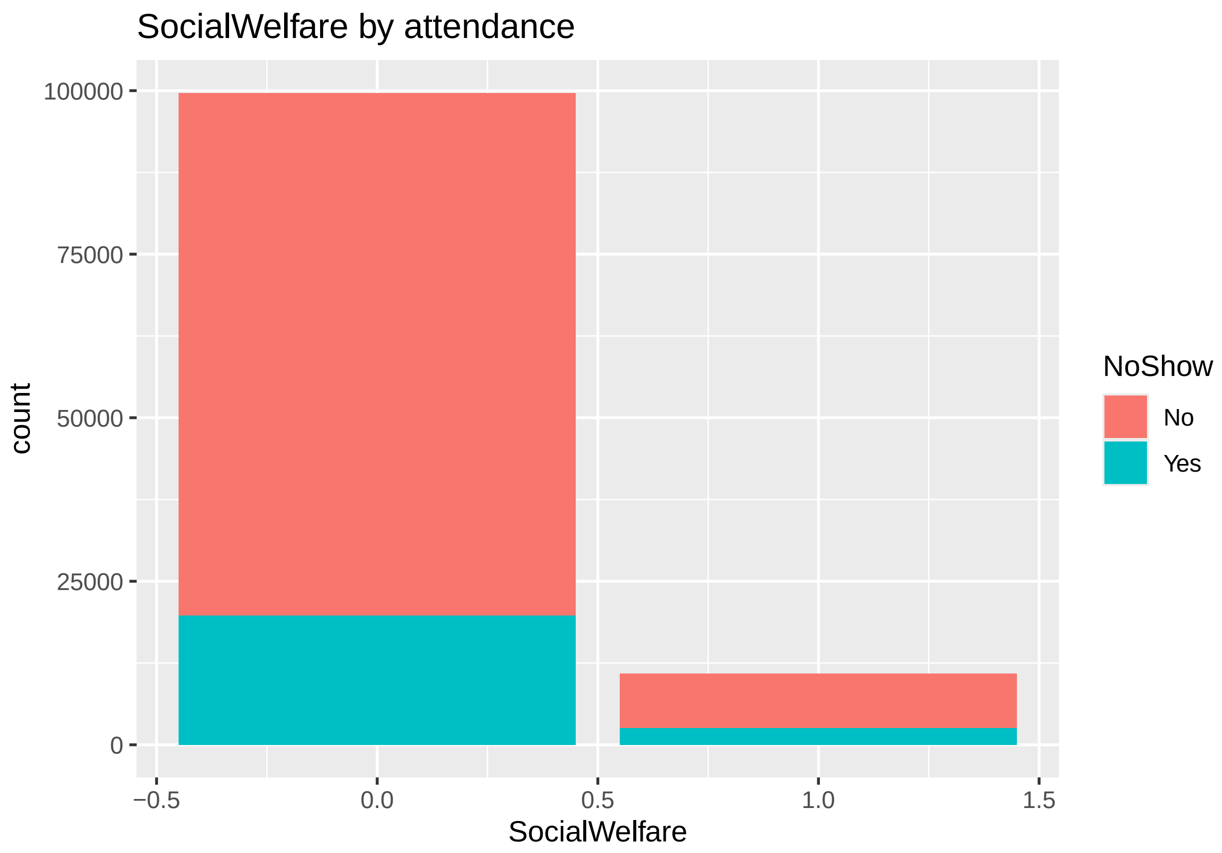


```
ggplot(raw.data) +
  geom_bar(aes(x=Gender, fill=NoShow), position='fill')+
  ggtitle("Proportion Gender by attendance")
```

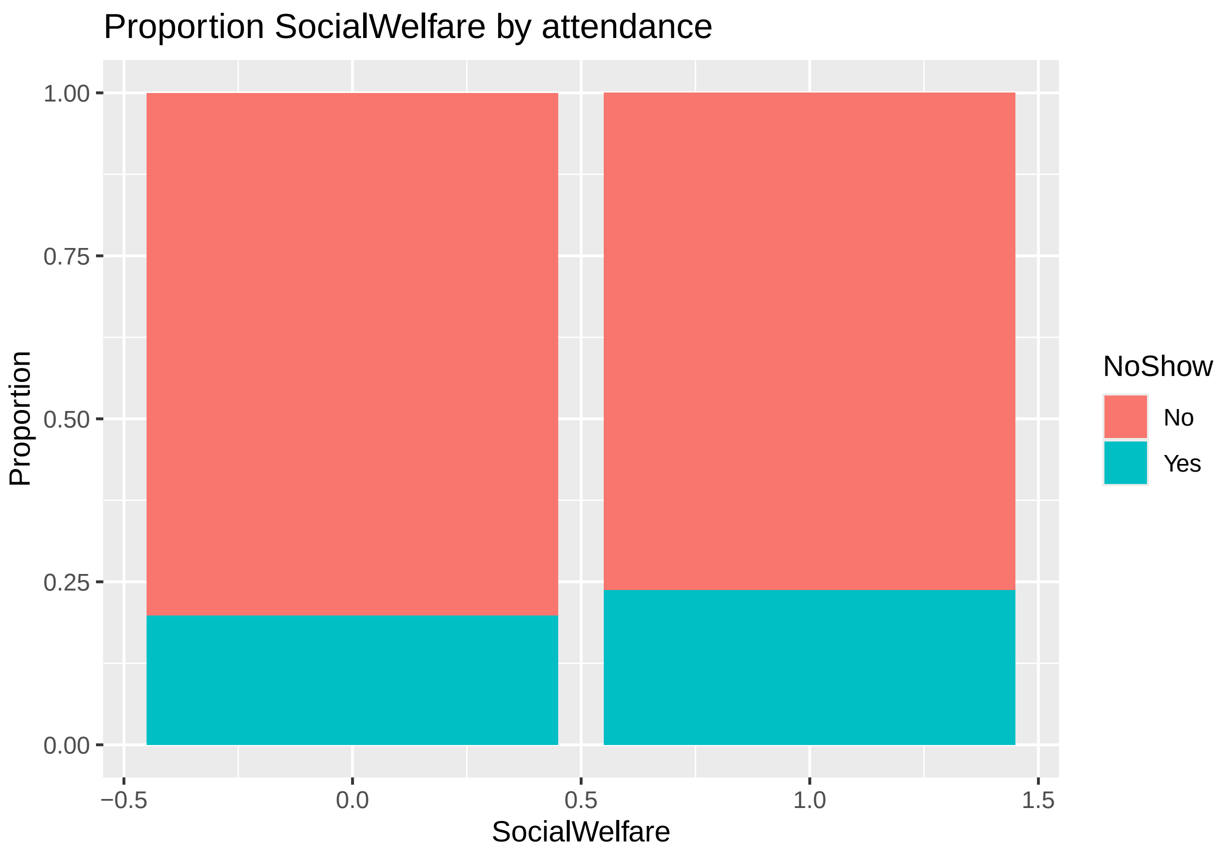


14 Create a similar plot using `SocialWelfare`

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



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



So, Social welfare doesn't show any significant change in attendance proportionately Despite being a huge gap between the number of people who enjoy social welfare and who doesn't.

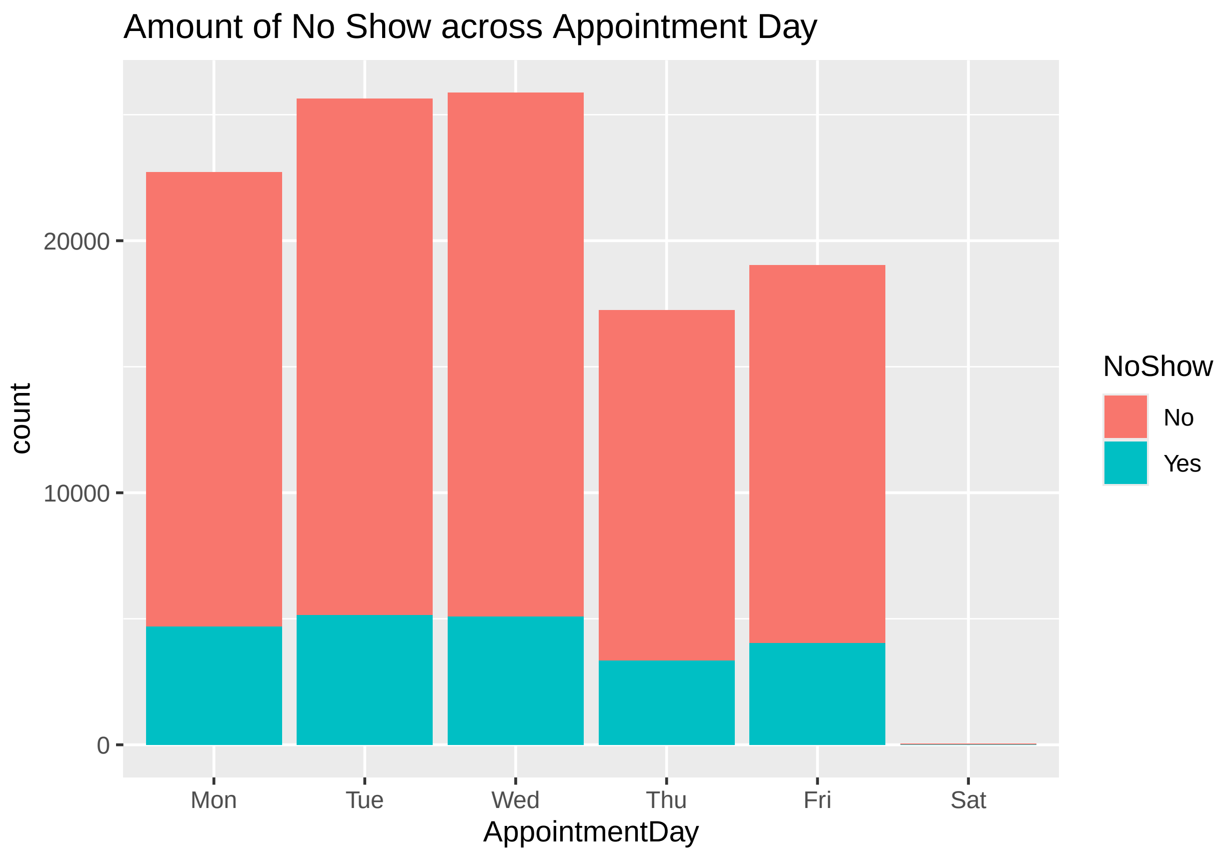
Far more exploration could still be done, including dimensionality reduction approaches but although we have found some patterns there is no major/striking patterns on the data as it currently stands.

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

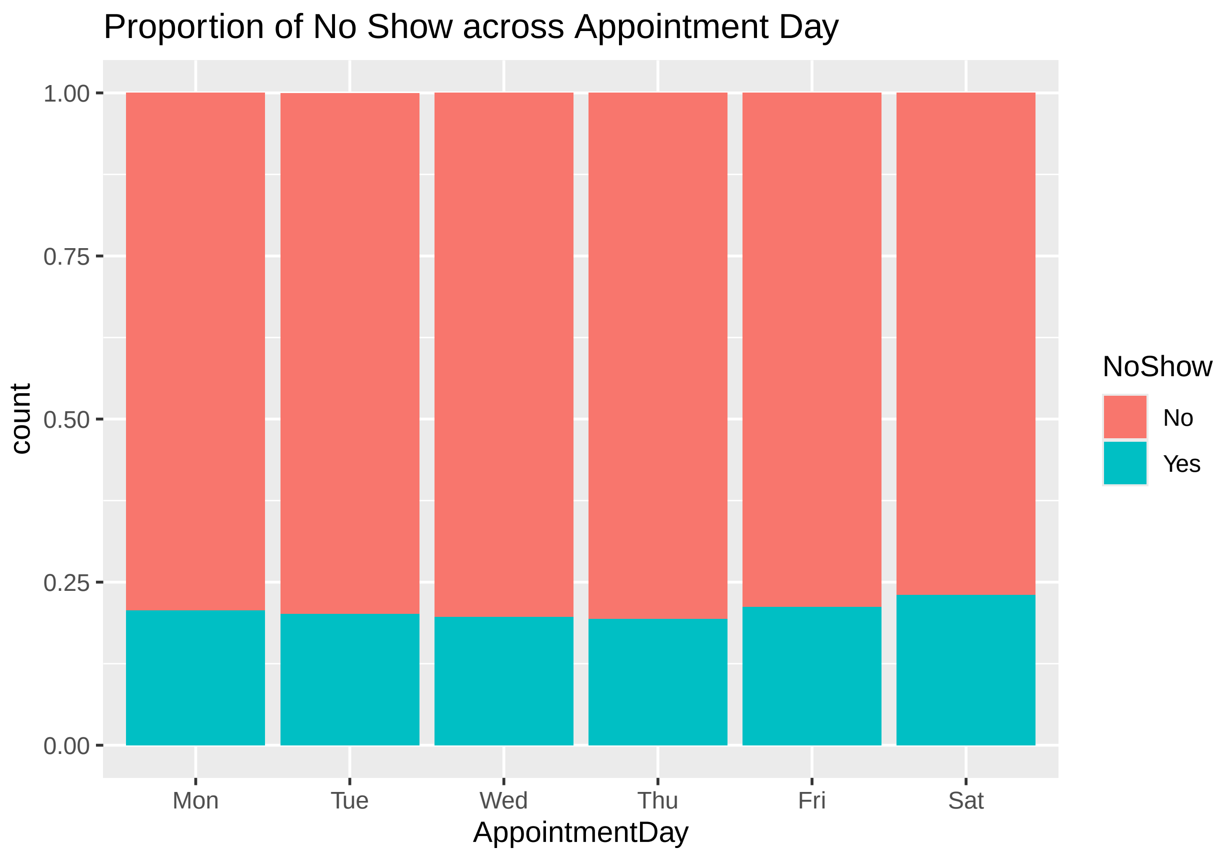
Feature Engineering

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")
```



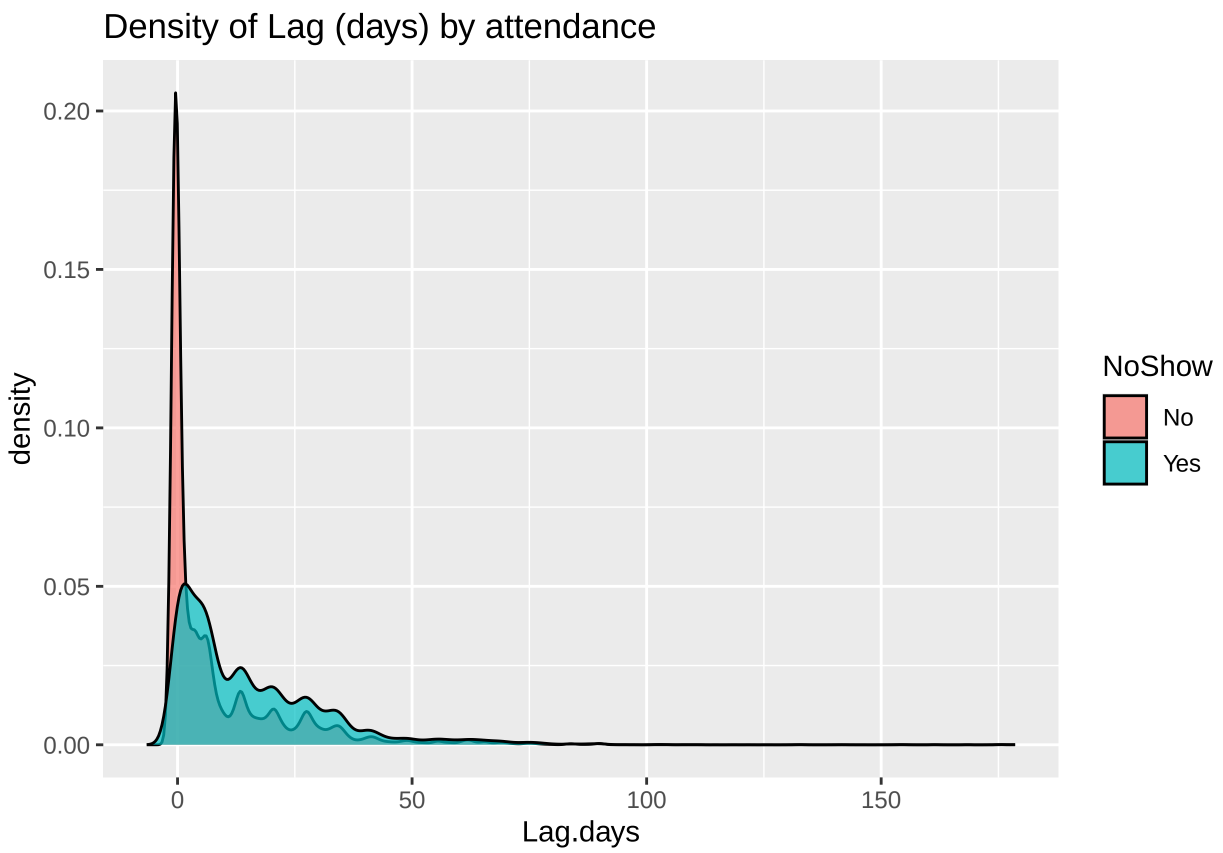
```
ggplot(raw.data) +
  geom_bar(aes(x=AppointmentDay, fill=NoShow), position = 'fill') +
  ggtitle("Proportion of No Show across Appointment Day")
```



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

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

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

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



15 Have a look at the values in lag variable, does anything seem odd? Long Tail: There is a long tail extending beyond 50 days, even reaching over 150 days. Typically, medical appointments are scheduled within a shorter timeframe (a few days to a few weeks). Appointments scheduled months in advance are unusual and may indicate data entry errors or special cases that need further investigation.

High Density Near Zero: There is a significant peak at 0 days, indicating that many appointments are scheduled and attended (or missed) on the same day. This could be normal for walk-in appointments, but it's worth verifying if these entries are accurate.

Multiple Small Peaks: There are several small peaks in the density for both attended and missed appointments. These may indicate specific patterns or periodic scheduling practices, but could also be due to irregular data entry or specific scenarios that need contextual understanding.

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?

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  26430  6425
##        Yes   115   188
##
##                Accuracy : 0.8028
##                  95% CI : (0.7984, 0.807)
##     No Information Rate : 0.8006
##     P-Value [Acc > NIR] : 0.1595
##
##                   Kappa : 0.0375
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.028429
##             Specificity : 0.995668
##          Pos Pred Value : 0.620462
##          Neg Pred Value : 0.804444
##              Prevalence : 0.199439
##          Detection Rate : 0.005670
##    Detection Prevalence : 0.009138
##       Balanced Accuracy : 0.512048
##
##        'Positive' Class : Yes
##
```

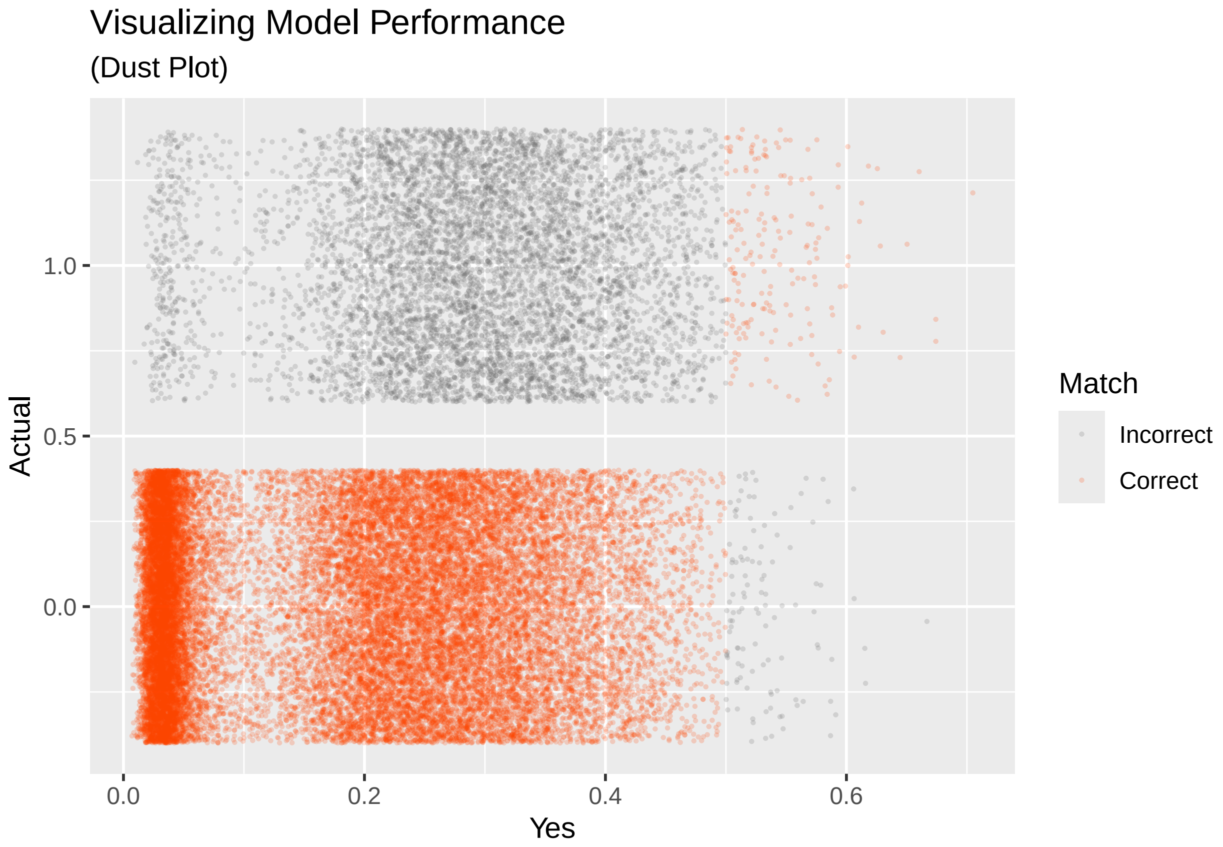
```
paste("XGBoost Area under ROC Curve: ", round(auc(test$NoShow.numerical, xgb.probs[,2]),3), sep="")
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```
## [1] "XGBoost Area under ROC Curve: 0.743"
```

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

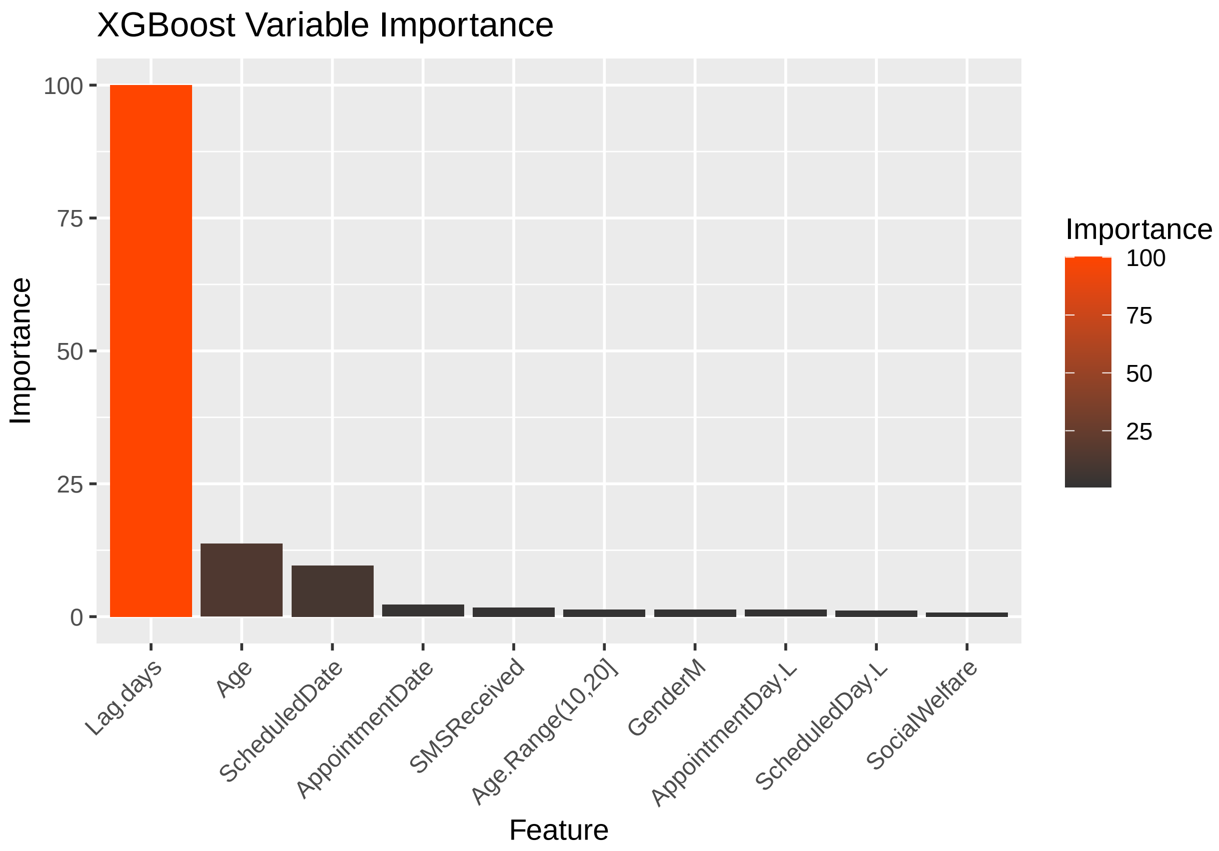
```
xgb.probs$Actual = test$NoShow.numerical
xgb.probs$ActualClass = test$NoShow
xgb.probs$PredictedClass = xgb.pred
xgb.probs$Match = ifelse(xgb.probs$ActualClass == xgb.probs$PredictedClass,
                         "Correct","Incorrect")
# [4.8] Plot Accuracy
xgb.probs$Match = factor(xgb.probs$Match,levels=c("Incorrect","Correct"))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

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


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



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

```r
library(randomForest)
```

```
## randomForest 4.7-1.1

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':
##
##     margin
## The following object is masked from ’package:dplyr’:
##
##     combine
```

```r
library(doParallel)
```

```
## Loading required package: foreach
```

```
##
## Attaching package: ’foreach’
```

```
## The following objects are masked from ’package:purrr’:
##
##     accumulate, when
```

```
## Loading required package: iterators
```

```
## Loading required package: parallel
```

```r
library(foreach)

cl <- makeCluster(detectCores())
registerDoParallel(cl)
# grid for randomForest
rf.grid <- expand.grid(mtry = c(2, 3, 4))

rf.model <- train(NoShow ~ .,
                  data = train,
                  method = "rf",
                  metric = "ROC",
                  tuneGrid = rf.grid,
                  trControl = fit.control,
                  ntree = 10)
```

```r
stopCluster(cl)
registerDoSEQ()
print(rf.model)
```

```
## Random Forest
##
## 77368 samples
##    16 predictor
##     2 classes: ’No’, ’Yes’
##
## No pre-processing
## Resampling: Cross-Validated (3 fold)
## Summary of sample sizes: 51578, 51579, 51579
## Resampling results across tuning parameters:
##
##   mtry  ROC        Sens       Spec
##   2     0.5464005  1.0000000  0.0000000000
##   3     0.6129030  0.9996594  0.0008276767
##   4     0.6502689  0.9966430  0.0119062455
##
## ROC was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 4.
```

```
# Make predictions
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")
```

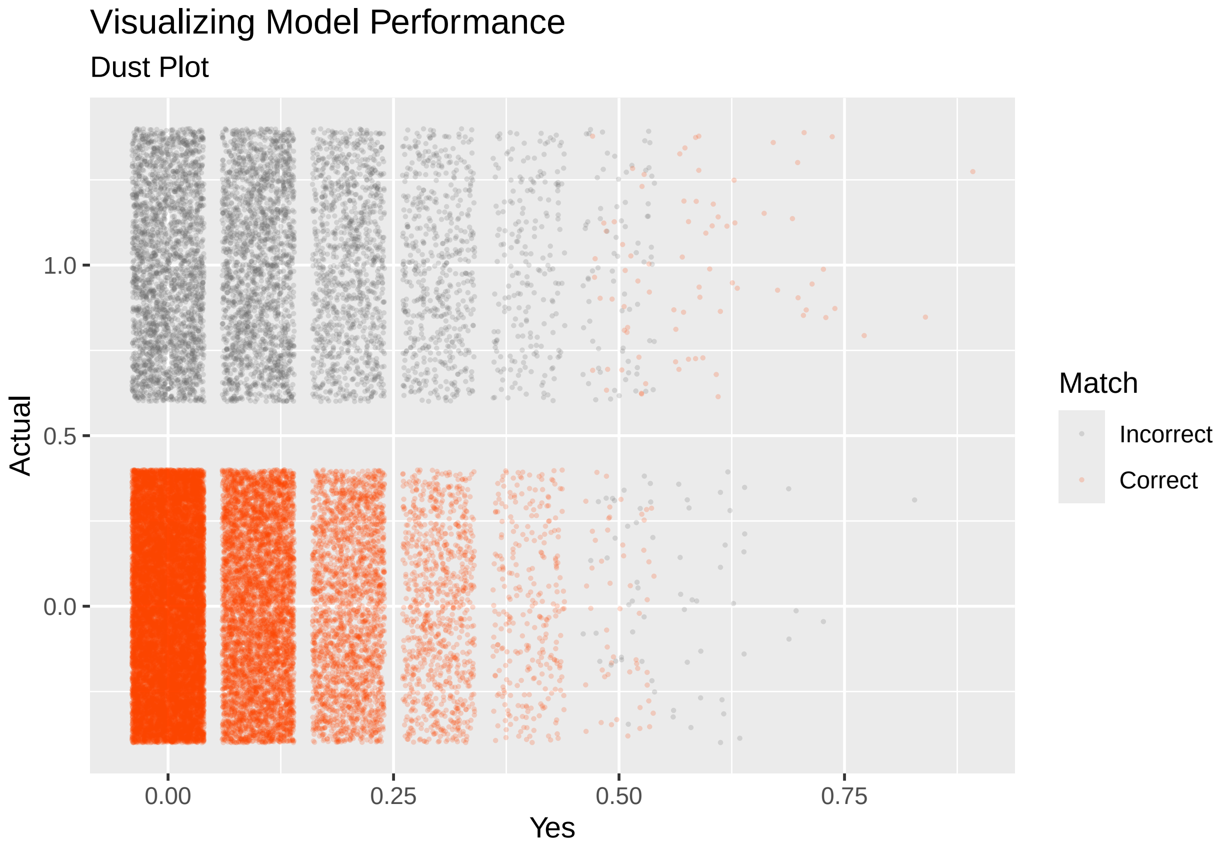
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    No   Yes
##        No  26480  6535
##        Yes    65    78
##
##                Accuracy : 0.801
##                  95% CI : (0.7966, 0.8052)
##     No Information Rate : 0.8006
##     P-Value [Acc > NIR] : 0.4323
##
##                   Kappa : 0.0148
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.011795
##             Specificity : 0.997551
##          Pos Pred Value : 0.545455
##          Neg Pred Value : 0.802060
##              Prevalence : 0.199439
##          Detection Rate : 0.002352
##    Detection Prevalence : 0.004313
##       Balanced Accuracy : 0.504673
##
##        'Positive' Class : Yes
##
```

```
paste("Radom 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] "Radom Forest Area under ROC Curve: 0.646"
```

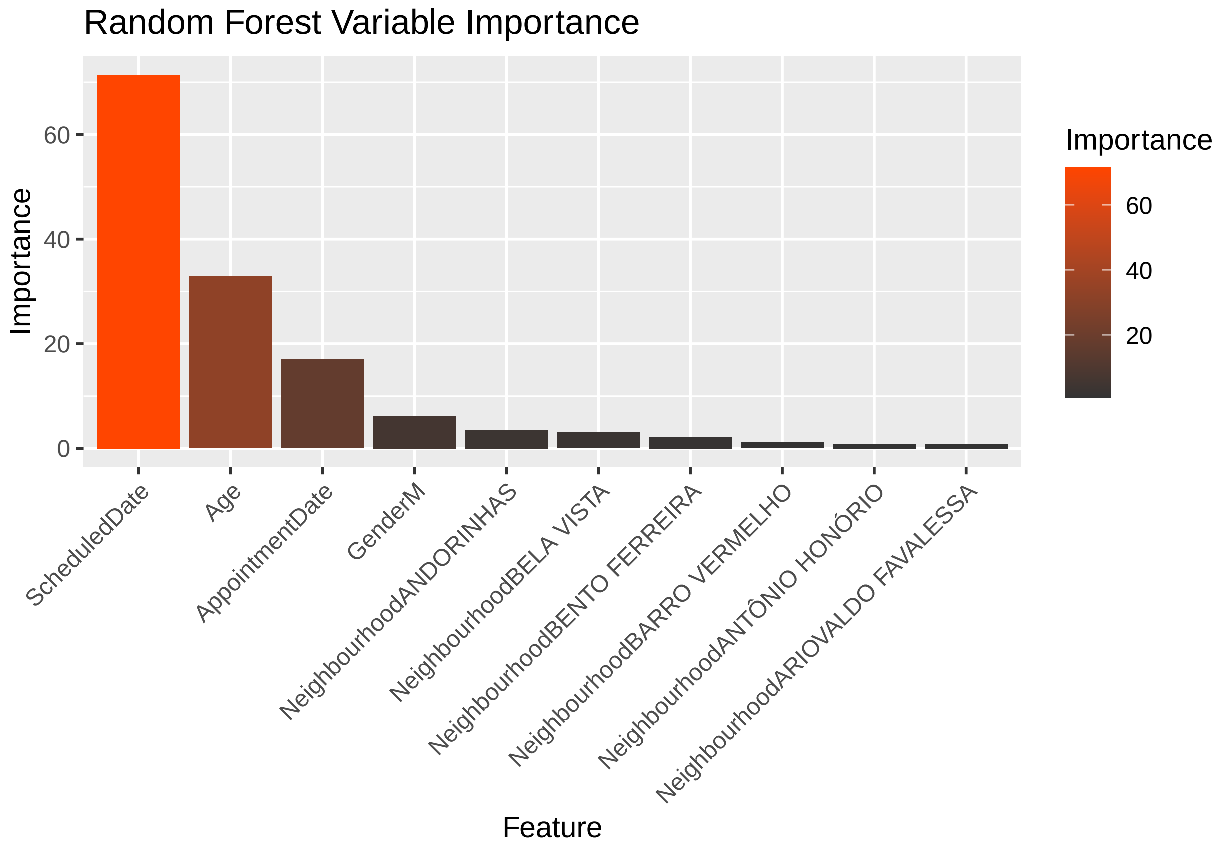
```
rf.probs$Actual = test$NoShow.numerical
rf.probs$ActualClass = test$NoShow
rf.probs$PredictedClass = rf.pred
rf.probs$Match = ifelse(rf.probs$ActualClass == rf.probs$PredictedClass, "Correct", "Incorrect")
```

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



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

```r
results <- data.frame(Feature = rownames(varImp(rf.model)$importance)[1:10],
                      Importance = varImp(rf.model)$importance[1:10, ])
results <- results %>%
  arrange(desc(Importance)) %>%
  mutate(Feature = factor(Feature, levels = Feature))

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



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

At first we conducted exploratory data analysis to understand the dataset and address potential issues. Howeverm there were some challenges due to limited context provided about the variables, which compromised interpretability and to some extent conclusions had to be drawn regarding the values of the variables based on reasonable assumption.

Moving on the modeling phase, we employed two machine learning algorithms, XGBoost (boosting) and Random Forest (bagging).Based on analysis results from using both XGBoost and Random Forest models on the dataset, there are several factors to consider regarding the trustworthiness of these analyses.

Area Under the ROC Curve (AUC): The AUC scores are 0.646 for Random Forest and 0.743 for XGBoost. While these scores are above 0.5, indicating better-than-random performance, they are not particularly high, suggesting room for improvement in distinguishing between classes.

Accuracy: Both models achieved high accuracy rates, with XGBoost at 80.13% and Random Forest at 80.28%. High accuracy suggests that the models are generally performing well in predicting the outcomes.

Kappa Statistic: -The low Kappa values (0.0163 for XGBoost and 0.0375 for Random Forest) suggest that the models' performance is not much better than random guessing. This is a significant concern.

Precision and Recall: Precision (Pos Pred Value) is relatively low (58.45% for XGBoost and 62.05% for Random Forest), indicating that many positive predictions are false positives. Recall (Neg Pred Value) is quite high (>80%), meaning the models are good at identifying negative cases correctly.

Prevalence and Imbalance: The prevalence of the positive class (Yes) is 19.94%, indicating a highly imbalanced dataset. This imbalance can affect model performance, especially in terms of precision and recall.

Model Performance Metrics: The low Kappa values and moderate AUC scores suggest that while the models are accurate overall, their ability to reliably predict the positive class (Yes) is limited. This is particularly concerning given the importance of accurately identifying no-shows.

Potential ways to increase the reliability of the analysis based on this dataset:

Dataset Imbalance: The high prevalence of the negative class (No) exacerbates the challenge of accurately predicting the positive class. Techniques such as oversampling the minority class, undersampling the majority class, or using SMOTE can potentially improve model performance.

Feature Importance: Features with low correlations to the target variable might not contribute significantly to prediction accuracy.Hence, removing less important features from the model can be a solution to increase the reliability of the analysis.

Increasing Feature space: If possible adding more features to the dataset could potentially enhance the reliability of the analysis. For instance, incorporating socio-economic factors such as income levels, education, and employment status could provide valuable context about the patients' backgrounds. Additionally, including geographic information such as the distance between patients' neighborhoods and healthcare facilities, as well as the availability of transportation means, could offer insights into accessibility and potential barriers to healthcare access.

In conclusion, while our analysis with both the models demonstrate reasonable accuracy, the low Kappa scores and moderate AUC values raise concerns about their reliability in real-world applications, especially considering the critical nature of accurately predicting no-shows. Addressing the dataset imbalance and further exploring feature engineering and model tuning strategies could enhance the models' predictive power and trustworthiness. If possible incorporating more features to this dataset can be put under consideration.

Credits

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