Predicting Non-Attendance of Healthcare Appointments CMM535

KAREN JEWELL, *1415410@rgu.ac.uk*

May 7, 2020

Introduction

The Problem

When patients do not attend scheduled appointments, it is a waste to healthcare providers' time and resources. The ability to predict if a patient will not attend an appointment could allow healthcare providers to provide more targeted and specific reminders to those who are at greater risk of not attending, encouraging increased attendance overall.

Project Objectives

The intention of the project is to identify if a classification model can predict if a patient will attend their appointment.

Data

The dataset used for this project is downloaded from `https://www.kaggle.com/joniarroba/noshowappointments` and is a collection of patient-appointment attendance history in Brazil over 5 weeks between 29 April 2016 and 8 June 2016

The dataset has 110,527 instances and 13 attributes. The 13 attributes are a mix of numerical, datetime, string and boolean values. There are some issues with data formats which are treated in the cleaning stage of this project

Data Dictionary

- **PatientId** A unique identification number for a patient
- **AppointmentID** A unique identification number for each appointment
- **Gender** The gender of the patient being Female or Male
- **ScheduledDay** The day and time a patient requested the appoinment.
- **AppointmentDay** The day and time of the actual appointment.
- **Age** The age of the patient.
- **Neighbourhood** Where the appointment takes place.

- **Scholarship** If the patient is receiving social welfare aid. Further reading https://en.wikipedia.org/wiki/Bolsa_Fam%C3%ADlia (1 if True)
- **Hipertension** If the patient is known to be diagnosed with Hypertension (1 if True).
- **Diabetes** If the patient is known to be diagnosed with Diabetes (1 if True).
- **Alcoholism** If the patient is known to be diagnosed with Alcoholism (1 if True).
- **Handcap** If the patient is known to have a handicap (1 if True).
- **SMS_received** If the patient was sent an SMS reminder of the appointment (1 if True).
- **No-show** If the patient attended their appointment (no.show is no) or did not attend the appointment (no.show is yes)

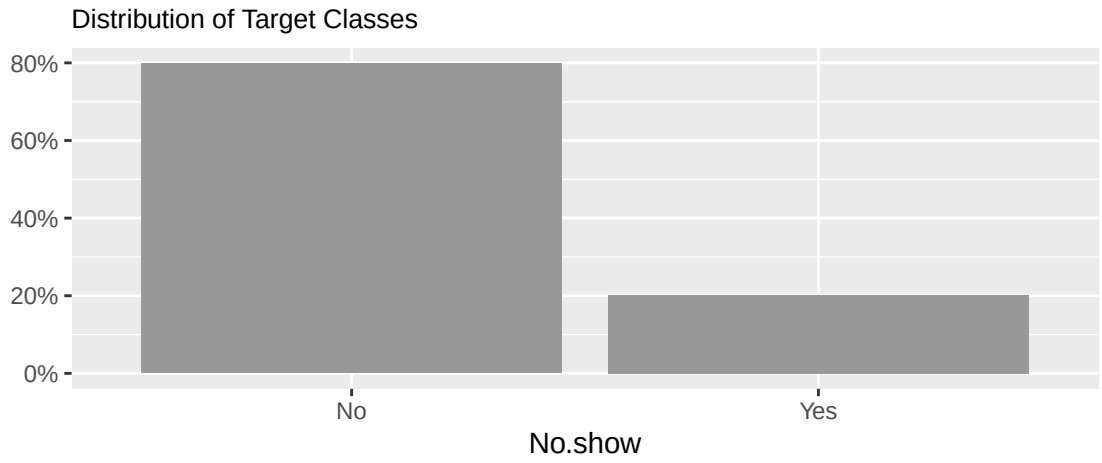
```
# Loading the data
data <- read.csv('KaggleV2-May-2016-NoShowAppointments.csv')
str(data)

## 'data.frame': 110527 obs. of  14 variables:
##  $ PatientId     : num  2.99e+13 5.59e+14 4.26e+12 8.68e+11 8.84e+12 ...
##  $ AppointmentID : int  5642903 5642503 5642549 5642828 5642494 5626772 5630279 5630575 5638447 5629123 ...
##  $ Gender        : Factor w/ 2 levels "F","M": 1 2 1 1 1 1 1 1 1 1 ...
##  $ ScheduledDay  : Factor w/ 103549 levels "2015-11-10T07:13:56Z",..: 27742 27504 27539 27709 27498 20074 21386
##  $ AppointmentDay: Factor w/ 27 levels "2016-04-29T00:00:00Z",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Age           : int  62 56 62 8 56 76 23 39 21 19 ...
##  $ Neighbourhood : Factor w/ 81 levels "AEROPORTO","ANDORINHAS",..: 40 40 47 55 40 59 26 26 2 13 ...
##  $ Scholarship   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Hipertension  : int  1 0 0 0 1 1 0 0 0 0 ...
##  $ Diabetes      : int  0 0 0 0 1 0 0 0 0 0 ...
##  $ Alcoholism    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Handcap       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ SMS_received  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ No.show       : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 2 2 1 1 ...
```

Classification Target

The target variable is “No.show” which is a binary classification representing whether a patient attended their appointment (“No”) or not (“Yes”). It is an imbalanced dataset over 2 classes with 20% of patients not attending their appointments.

The use of a double-negative "No.show is No" actually meaning the patient did attend their appointment is unintuitive and confusing. So to aid interpretation of the results, the target class is renamed to "Attended" (Yes/No) in the data cleaning stage of this project and refered to as such in the rest of this document.

```
# Distribution of classes
table(data$No.show)

##
##    No   Yes
## 88208 22319

ggplot(data, aes(No.show)) +
  geom_bar(aes(y = (..count..)/sum(..count..)), fill="#999999") +
  scale_y_continuous(labels=scales::percent, name="") +
  ggtitle("Distribution of Target Classes") +
  theme(plot.title = element_text(size=10))
```



Exploratory Analysis

Initial Exploration

```
# Checking for any missing data
sum(is.na(data))

## [1] 0

# Checking if Appointment IDs are unique
duplicated(data$AppointmentId)

## logical(0)
```

For an initial investigation, the dataset is checked for any missing data, which none are found; and if the AppointmentID is unique to ensure no duplication of instances, which none are found, suggesting all instances are indeed unique.

Cleaning Data

```
# Creating new repo for cleaned data, retaining original load for sense checking
cleandata <- data

# Correcting column names
names(cleandata)[names(cleandata)=="Handcap"] <- "Handicap"
names(cleandata)[names(cleandata)=="Hipertension"] <- "Hypertension"
names(cleandata)[names(cleandata)=="Scholarship"] <- "WelfareAid"

# Converting int to factors
boolcols <- c('WelfareAid','Hypertension','Diabetes','Alcoholism','Handicap','SMS_received')

cleandata[boolcols] <-
        lapply(cleandata[boolcols], factor,
        levels=c(0, 1),
        labels = c("FALSE","TRUE"))

# Separating time and date from Appointment and Schedule
cleandata$AppointmentDate <-
```

```
  format(as.POSIXct(cleandata$AppointmentDay,format='%Y-%m-%dT%H:%M:%SZ'),format='%Y-%m-%d')

cleandata$AppointmentDayofWeek <-
  format(as.POSIXct(cleandata$AppointmentDay,format='%Y-%m-%dT%H:%M:%SZ'),format='%A')

cleandata$AppointmentTime <-
  format(as.POSIXct(cleandata$AppointmentDay,format='%Y-%m-%dT%H:%M:%SZ'),format='%H:%M:%S')

cleandata$RequestDate <-
  format(as.POSIXct(cleandata$ScheduledDay,format='%Y-%m-%dT%H:%M:%SZ'),format='%Y-%m-%d')

cleandata$RequestDayofWeek <-
  format(as.POSIXct(cleandata$ScheduledDay,format='%Y-%m-%dT%H:%M:%SZ'),format='%A')

cleandata$RequestTime <-
  format(as.POSIXct(cleandata$ScheduledDay,format='%Y-%m-%dT%H:%M:%SZ'),format='%H:%M:%S')

# Converting day strings to factors
daysofweek = c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday")
cleandata$AppointmentDayofWeek <- factor(cleandata$AppointmentDayofWeek, levels = daysofweek)
cleandata$RequestDayofWeek <- factor(cleandata$RequestDayofWeek, levels = daysofweek)

# Creating new data
cleandata$DaysBetween <-
  as.integer(difftime(cleandata$AppointmentDate,cleandata$RequestDate, units='days'))

# Inversing the target variable to be Attended Y/N
# (No.show=No >>> Attended=Yes; No.show=Yes >>> Attended = No)
cleandata$Attended <-
  factor(cleandata$No.show, levels = c("Yes", "No"), labels = c("No", "Yes"))
```

In this stage, the data is cleaned to prepare it for modelling.

3 column names of "Handcap" "Hipertension" "Scholarship" are renamed to "Handicap", "Hypertension" and "WelfareAid" respectively to correct spelling errors or to improve clarity given its translation from the original Portuguese language.

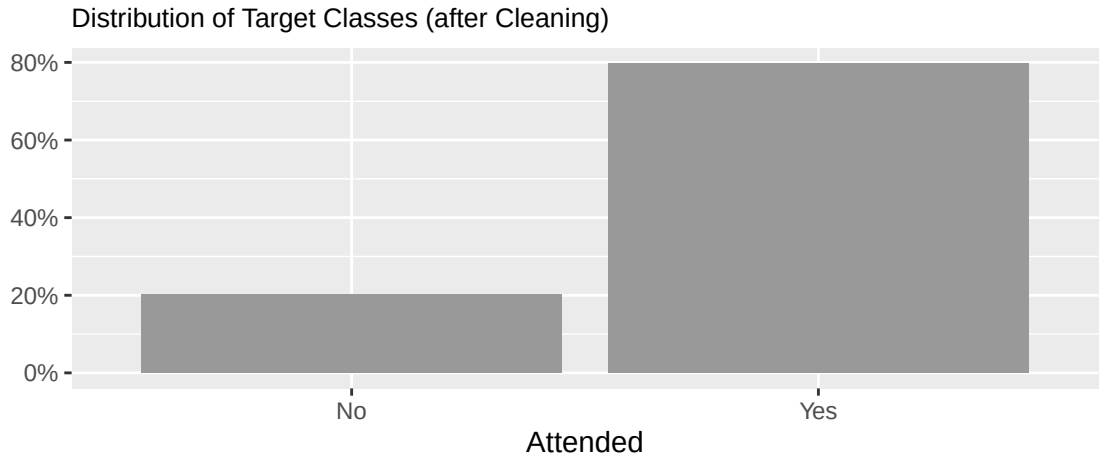
For the columns 'WelfareAid','Hypertension','Diabetes','Alcoholism','Handicap','SMS_received' which are originally integer values, these are converted to a factor data type with 2 levels of TRUE or FALSE. Even though the data is boolean in nature, it is best practice to convert it so the model recognises there is no order between the values as would have been implied if left as an integer.

The 2 datetime columns are formatted and split into 6 columns (3 each of the original) to retain only the date, the time and the name of the day. The names of the days are then converted to a 7-level factor 1 for each day of the week.

A new variable "DaysBetween" was calculated to represent the difference in days between when the appointment was made "RequestDate" and when the appointment was to occur "AppointmentDate".

Last and perhaps most crucially, the target variable was renamed and inversed to add clarity and become more intuitive to interpret later. The target was renamed from "No.show" to "Attended", and thus where a patient was a no-show (No.show=Yes) is now a Attended=No and likewise where they weren't a no-show (No.show=No) they are now a Attended=Yes. The target class "No" now becomes the minority class and the class of particular interest. After this inversion, the target is represented as below:

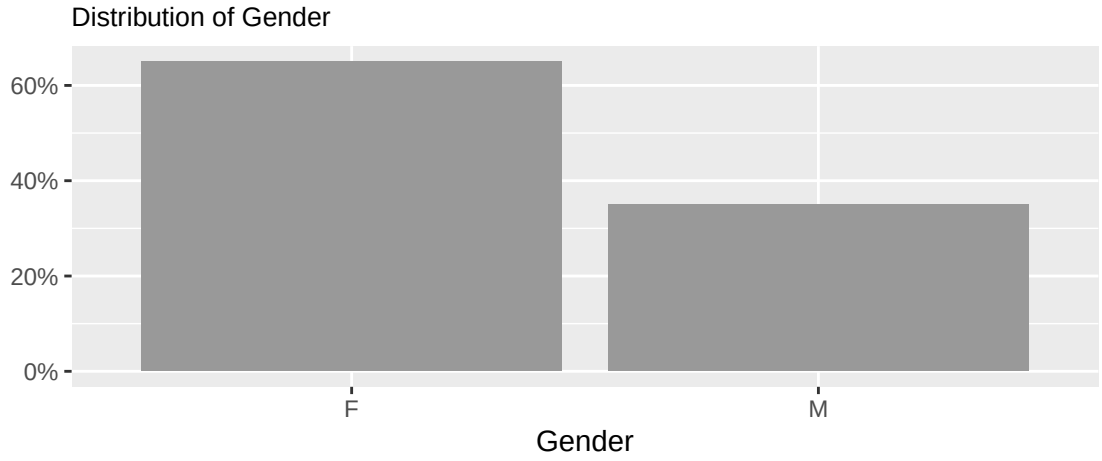
```r
# Distribution of classes
table(cleandata$Attended)

##
##    No   Yes
## 22319 88208

ggplot(cleandata, aes(Attended)) +
  geom_bar(aes(y = (..count..)/sum(..count..)), fill="#999999") +
  scale_y_continuous(labels=scales::percent, name="") +
  ggtitle("Distribution of Target Classes (after Cleaning)") +
  theme(plot.title = element_text(size=10))
```



Attribute Analysis

Gender Distribution

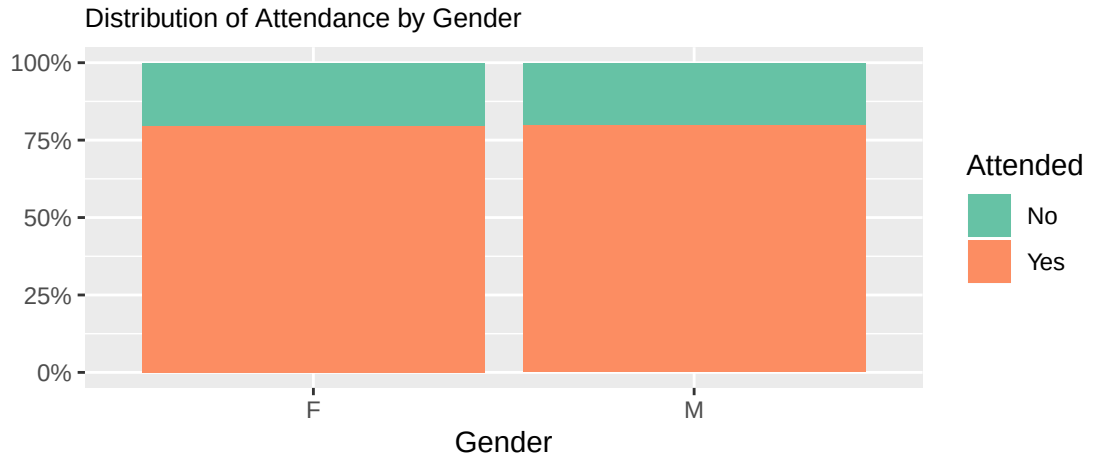
```
# Distribution of gender
with(cleandata, table(Gender)) %>%
  prop.table()

## Gender
##         F         M
## 0.6499769 0.3500231

ggplot(cleandata, aes(Gender)) +
  geom_bar(aes(y = (..count..)/sum(..count..)), fill="#999999") +
  scale_y_continuous(labels=scales::percent, name="") +
  ggtitle("Distribution of Gender") +
  theme(plot.title = element_text(size=10))
```



In the dataset, approximately 65% of the appointments were for Female patients and the remaining 35% for Male patients. Which is in itself an interesting observation that Females are nearly twice as likely as Males to require an appointment. However, looking deeper as in the figures below, gender does not appear to impact appointment attendance as 20% of both Female and Male patients did not attend their appointments (remembering that the total propotion of non-attendance is also 20%).

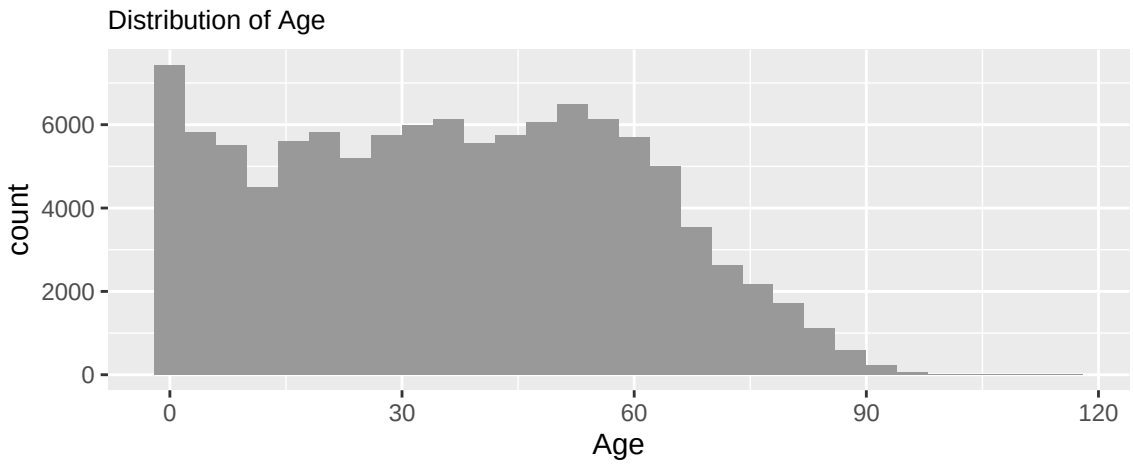
```
# Proportion of Females not attending appointments
with(subset(cleandata, Gender=="F"), table(Attended)) %>%
  prop.table()

## Attended
##        No       Yes
## 0.2031459 0.7968541

# Proportion of Males not attending appointments
with(subset(cleandata, Gender=="M"), table(Attended)) %>%
  prop.table()

## Attended
##        No       Yes
## 0.1996795 0.8003205

ggplot(cleandata, aes(x=Gender, fill=Attended)) +
    scale_y_continuous(labels=scales::percent, name="") +
```

```
    geom_bar(position="fill", stat="count") +
    scale_fill_brewer(palette="Set2") +
    ggtitle("Distribution of Attendance by Gender") +
    theme(plot.title = element_text(size=10))
```

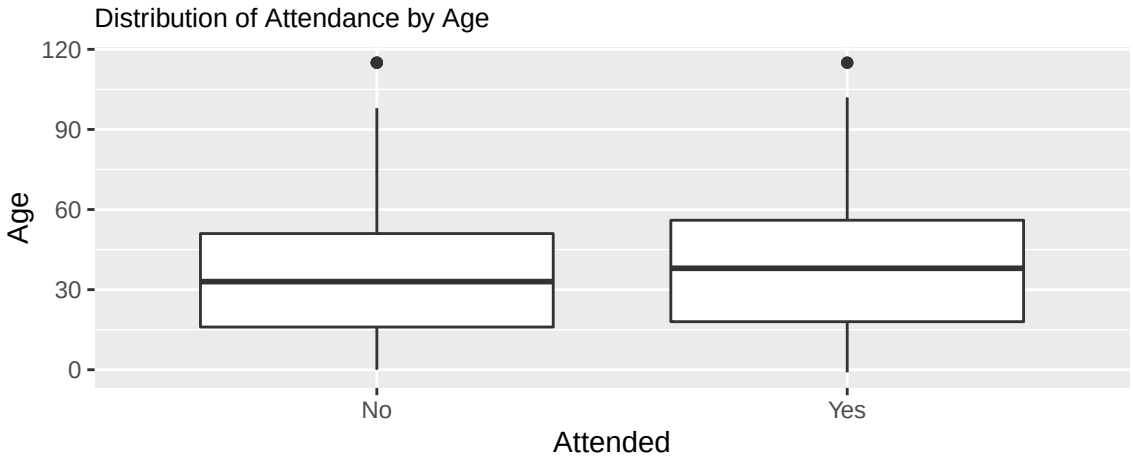


Age Distribution

```
# Distribution of ages
summary(cleandata$Age)

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

# Looking into the -1 age
length(cleandata$Age[cleandata$Age<0])

## [1] 1

ggplot(cleandata, aes(Age)) +
  geom_histogram(fill="#999999") +
  ggtitle("Distribution of Age") +
  theme(plot.title = element_text(size=10))
```



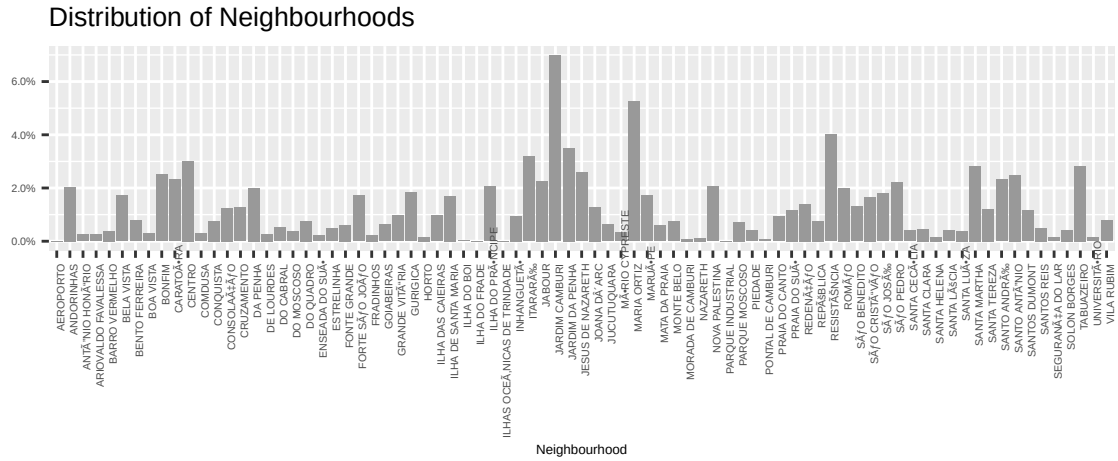
```
ggplot(cleandata, aes(x=Attended, y=Age)) +
  geom_boxplot() +
  ggtitle("Distribution of Attendance by Age") +
  theme(plot.title = element_text(size=10))
```



Looking at the distribution of patients' ages in the dataset, the youngest age is -1 and further investigation reveals this is a single instance which seems to indicate is an error; the oldest age is 115. However, there also appears to be little significant relationship between the age of the patient and attendance of the appointment. Although not explored in this project, it could be potentially interesting to investigate further if groups of ages (infants, children, teenagers, adults, seniors) tend to impact appointment attendance.

Neighbourhood Distribution

```
# Distribution of neighbourhoods
length(unique(cleandata$Neighbourhood))

## [1] 81

ggplot(cleandata, aes(Neighbourhood)) +
  geom_bar(aes(y = (..count..)/sum(..count..)), fill="#999999") +
  scale_y_continuous(labels=scales::percent, name="") +
  ggtitle("Distribution of Neighbourhoods") +
  theme(text = element_text(size=5),
        axis.text.x=element_text(angle=90, hjust=1),
        plot.title = element_text(size=10))
```



There are 81 distinct neighbourhoods identified in this dataset.

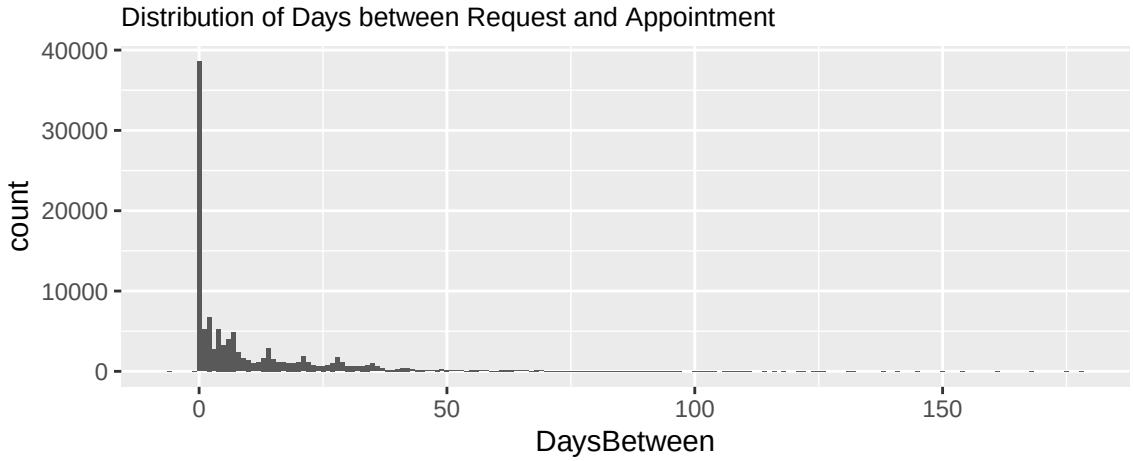
Distribution of Appointment Time

```
# Checking post-processing
unique(cleandata$AppointmentTime)

## [1] "00:00:00"
```

A theory which could have been explored, is if the time of an appointment affected appointment attendance. For example, if appointments were early in the morning and a patient could not travel across to the place of their appointment in time. Unfortunately, after extracting the Appointment Time information from the original AppointmentDay column, it turns out the appointment time is either not recorded, or had been omitted for potential privacy reasons (although unlikely if patientID is retained). This unfortunately renders the variable redundant.

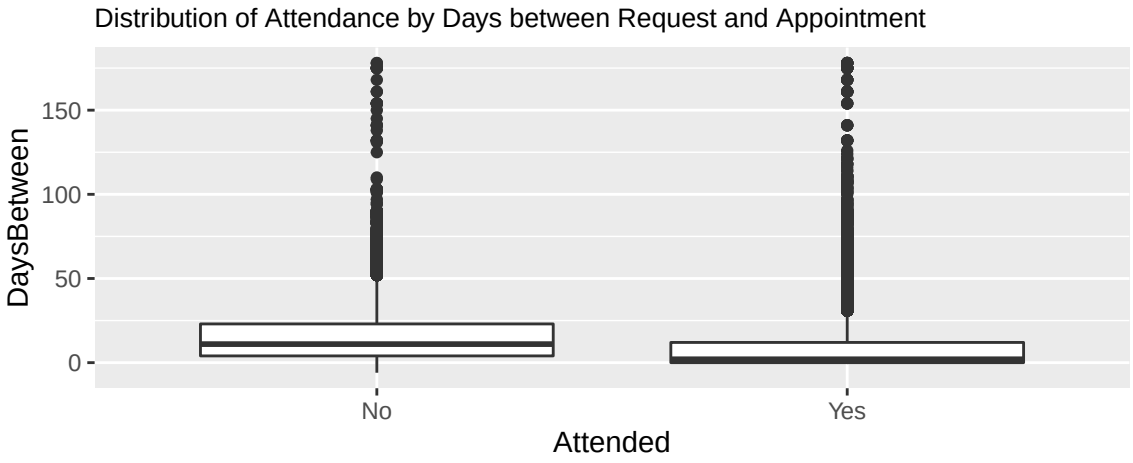
Elapsed Time between Appointment and When Requested

```
# Proportion of entries where appointments are made on the same day
length(cleandata$DaysBetween[cleandata$DaysBetween==0])/length(cleandata$DaysBetween)

## [1] 0.3489012

summary(cleandata$DaysBetween)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   -6.00    0.00    4.00   10.16   15.00  178.00

# Distribution of days between appointment
ggplot(cleandata, aes(DaysBetween)) +
  geom_histogram(stat="count") +
  ggtitle("Distribution of Days between Request and Appointment") +
  theme(plot.title = element_text(size=10))
```



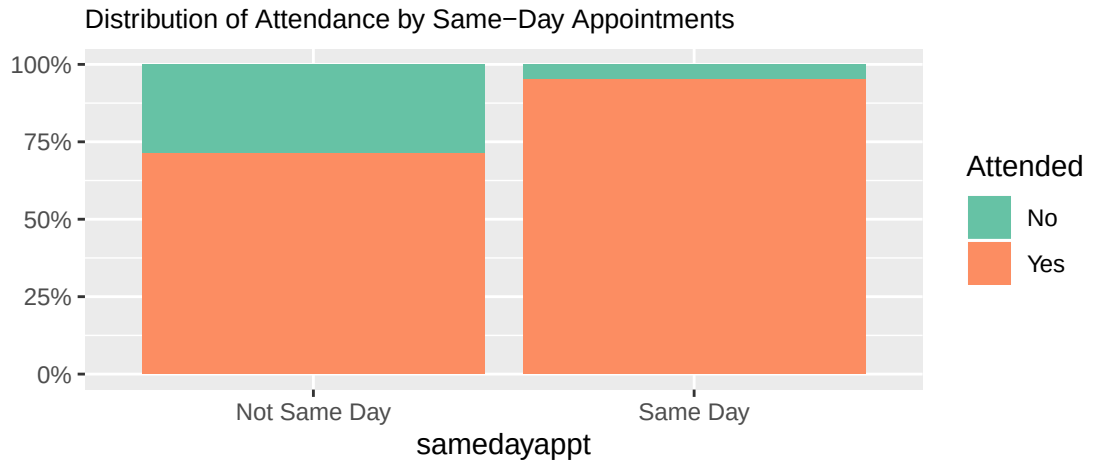
In looking at the days between when an appointment is requested and when it is due to happen, 35% of appointments occur on the same day it is requested. The remaining 65% of appointments occur between 1 and 178 days and the median is a 4 day wait. It is also apparent that there are some negative waiting days. It should be impossible to schedule an appointment for a date in history, therefore these are possibly added in retrospect or potentially erroneously transposed.

```
ggplot(cleandata, aes(x=Attended, y=DaysBetween)) +
  geom_boxplot() +
  ggtitle("Distribution of Attendance by Days between Request and Appointment") +
  theme(plot.title = element_text(size=10))
```




```
cleandata$samedayappt <- as.factor(ifelse(cleandata$DaysBetween == 0,"Same Day", "Not Same Day"))

# Proportion of same-day appointments attended
with(subset(cleandata, samedayappt=="Same Day"), table(Attended)) %>%
  prop.table()

## Attended
##         No        Yes
## 0.04646941 0.95353059

# Proportion of non-same-day appointments attended
with(subset(cleandata, samedayappt=="Not Same Day"), table(Attended)) %>%
  prop.table()
```

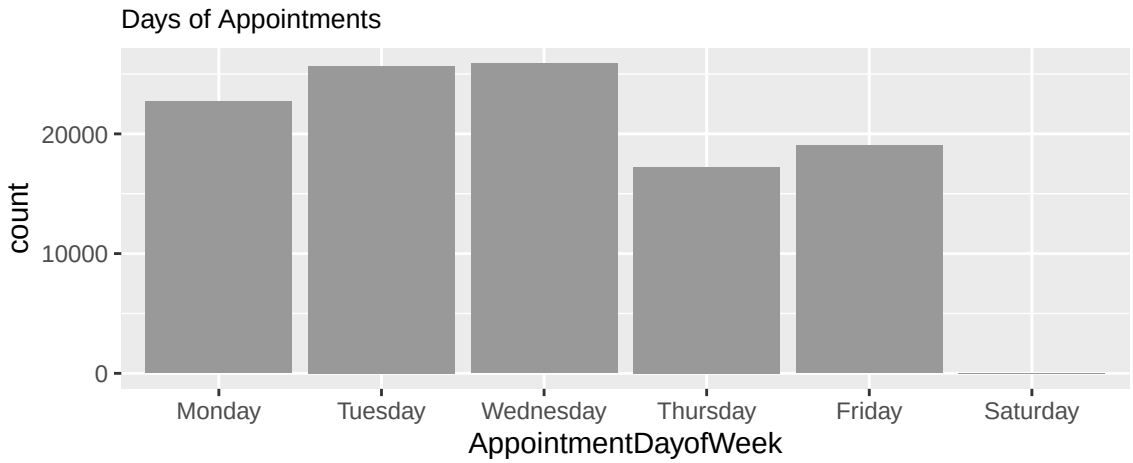
```
## Attended
##        No       Yes
## 0.2852398 0.7147602

ggplot(cleandata, aes(x=samedayappt, fill=Attended)) +
    scale_y_continuous(labels=scales::percent, name="") +
    geom_bar(position="fill", stat="count") +
    scale_fill_brewer(palette="Set2") +
    ggtitle("Distribution of Attendance by Same-Day Appointments") +
    theme(plot.title = element_text(size=10))
```



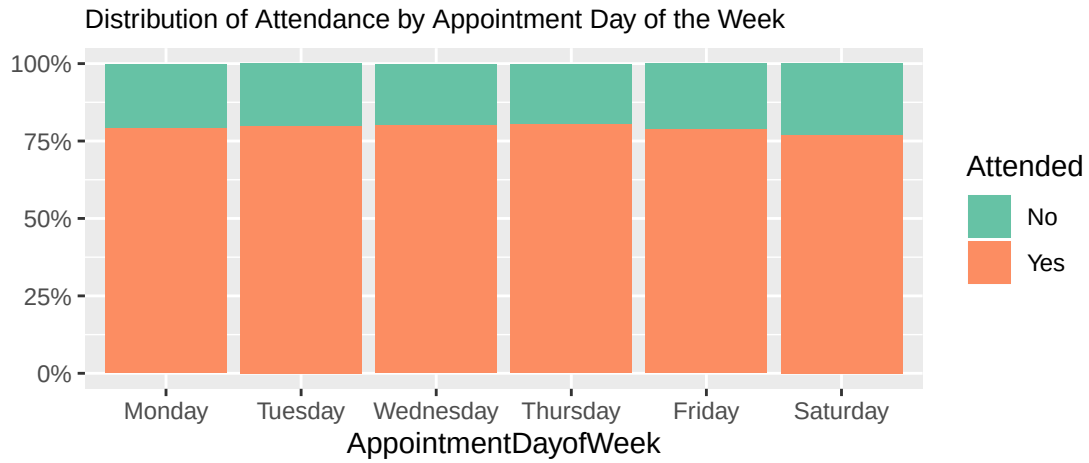
When considering the length of waiting days and attendance of appointments, same-day appointments see better attendance rates then appointments made in advance.

Appointment Day Distribution

```
# Distribution of appointment day
table(cleandata$AppointmentDayofWeek, cleandata$Attended)

##
##                No   Yes
##   Monday     4690 18025
##   Tuesday    5152 20488
##   Wednesday  5093 20774
##   Thursday   3338 13909
##   Friday     4037 14982
##   Saturday      9    30
##   Sunday        0     0

ggplot(cleandata, aes(AppointmentDayofWeek)) +
  geom_histogram(stat="count", fill="#999999") +
  ggtitle("Days of Appointments") +
  theme(plot.title = element_text(size=10))
```




```
ggplot(cleandata, aes(fill=Attended, x=AppointmentDayofWeek)) +
   scale_y_continuous(labels=scales::percent, name="") +
    geom_bar(position="fill", stat="count") +
    scale_fill_brewer(palette="Set2") +
    ggtitle("Distribution of Attendance by Appointment Day of the Week") +
    theme(plot.title = element_text(size=10))
```

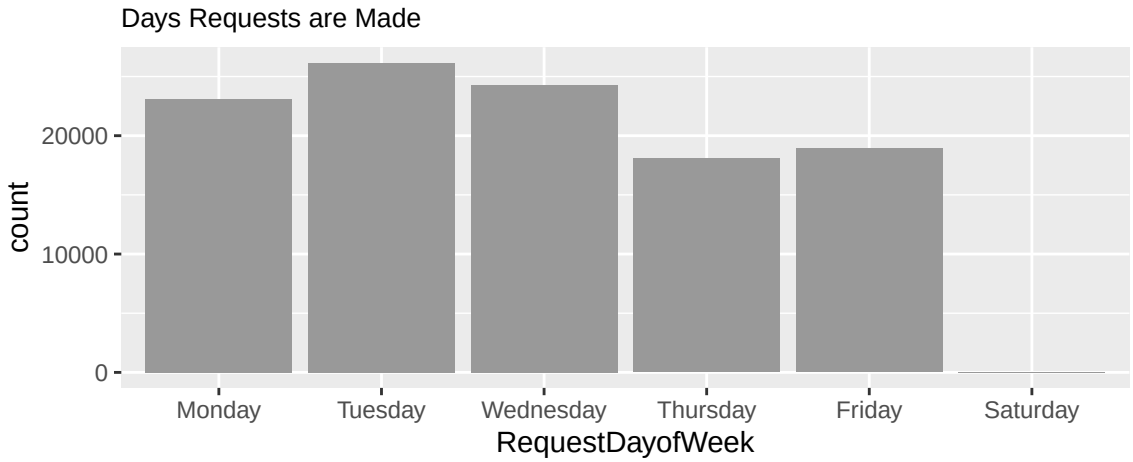


There appears to be little variation in attendance given the day of the week, although it is evident that appointments occur Monday to Friday with very few on Saturdays and none on Sundays.

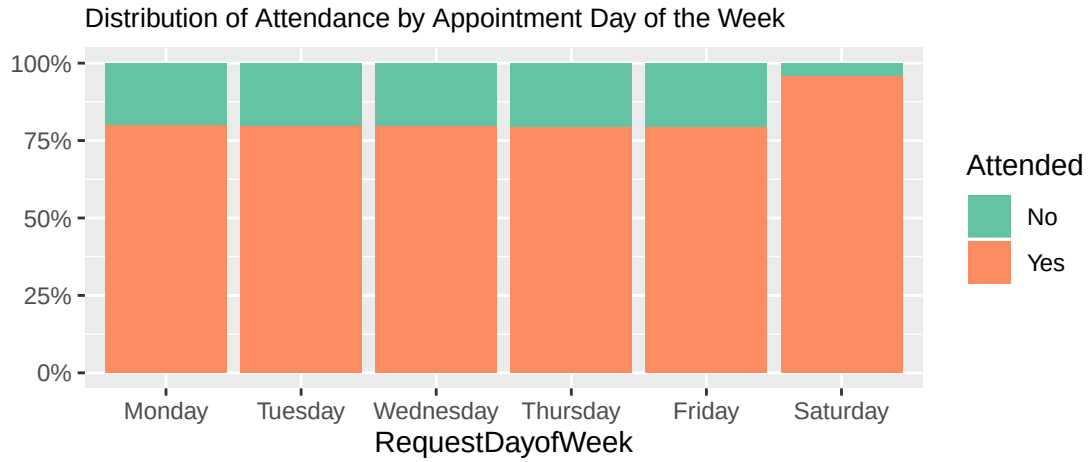
Day of Appointment Request Distribution

```
# Distribution of request day
table(cleandata$RequestDayofWeek, cleandata$Attended)

##
##                No   Yes
##   Monday     4561 18524
##   Tuesday    5291 20877
##   Wednesday  4879 19383
##   Thursday   3700 14373
##   Friday     3887 15028
##   Saturday      1    23
##   Sunday        0     0
```

```
ggplot(cleandata, aes(RequestDayofWeek)) +
  geom_histogram(stat="count", fill="#999999") +
  ggtitle("Days Requests are Made") +
  theme(plot.title = element_text(size=10))
```

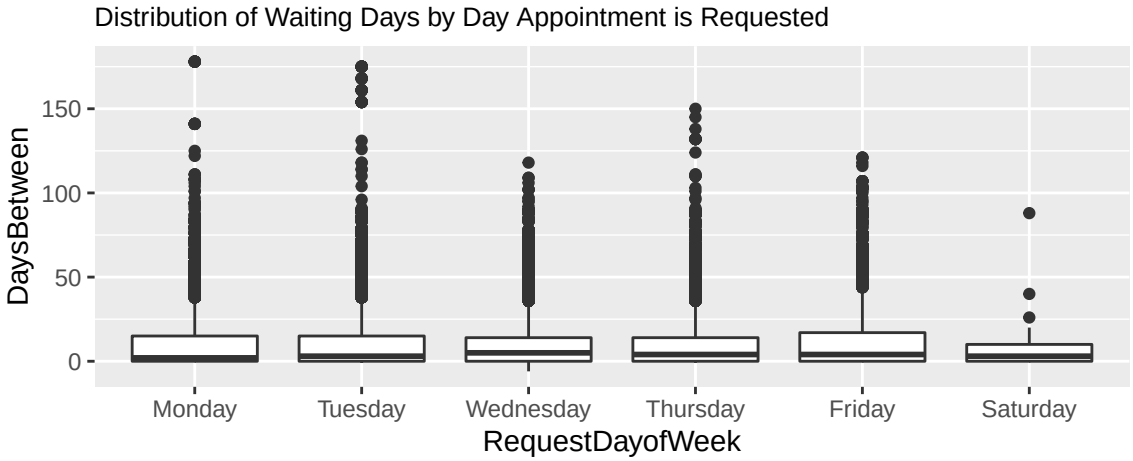



```
ggplot(cleandata, aes(fill=Attended, x=RequestDayofWeek)) +
  scale_y_continuous(labels=scales::percent, name="") +
  geom_bar(position="fill", stat="count") +
  scale_fill_brewer(palette="Set2") +
  ggtitle("Distribution of Attendance by Appointment Day of the Week") +
  theme(plot.title = element_text(size=10))
```



Similar to which day an appointment occurs, the day when an appointment is requested appears to not relate to attendance, except for appointments made on Saturdays where there is less non-attendance. Again here we also we see far fewer appointments made on Saturdays, and none on Sundays presumably because healthcare providers do not operate on those days.

```
ggplot(cleandata, aes(x=RequestDayofWeek, y=DaysBetween)) +
  geom_boxplot() +
  ggtitle("Distribution of Waiting Days by Day Appointment is Requested") +
  theme(plot.title = element_text(size=10))
```



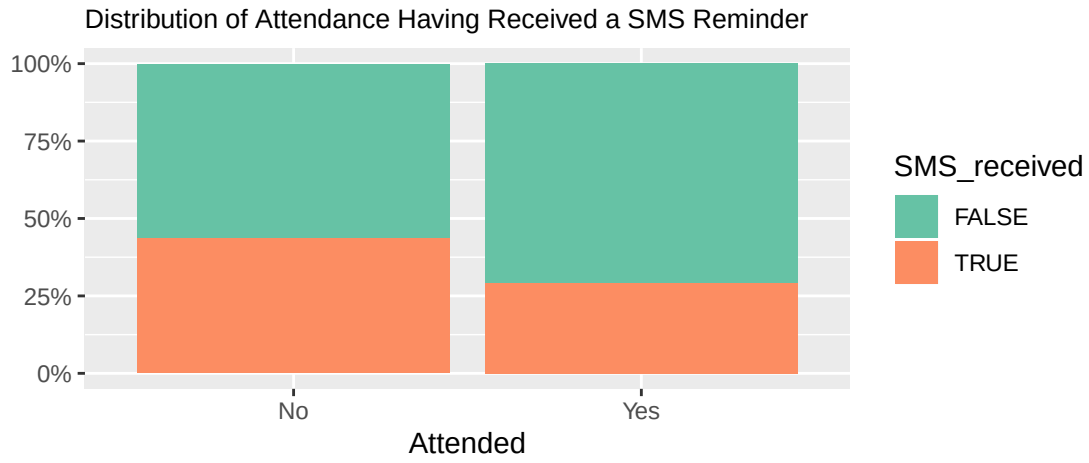
Distribution of SMS Reminders

```
# Distribution of sms_received
with(cleandata, table(SMS_received)) %>%
  prop.table()

## SMS_received
##     FALSE      TRUE
## 0.6789744 0.3210256

# Proportion of attendance having received a sms reminder
with(subset(cleandata, SMS_received=="TRUE"), table(Attended)) %>%
  prop.table()

## Attended
##        No       Yes
## 0.2757454 0.7242546

# Proportion of attendance not having received a sms reminder
with(subset(cleandata, SMS_received=="FALSE"), table(Attended)) %>%
  prop.table()

## Attended
##        No       Yes
## 0.1670331 0.8329669
```

Of the entire dataset, 68% of patients did not receive an SMS reminder. 16.7% of those patients who had not received an SMS reminder did not attend their appointments.

This compared to the 32% of patients who did receive an SMS reminder, 27.5% of those patients did not attend their appointments.

```
ggplot(cleandata, aes(x=Attended, fill=SMS_received)) +
    scale_y_continuous(labels=scales::percent, name="") +
    geom_bar(position="fill", stat="count") +
    scale_fill_brewer(palette="Set2") +
    ggtitle("Distribution of Attendance Having Received a SMS Reminder") +
    theme(plot.title = element_text(size=10))
```



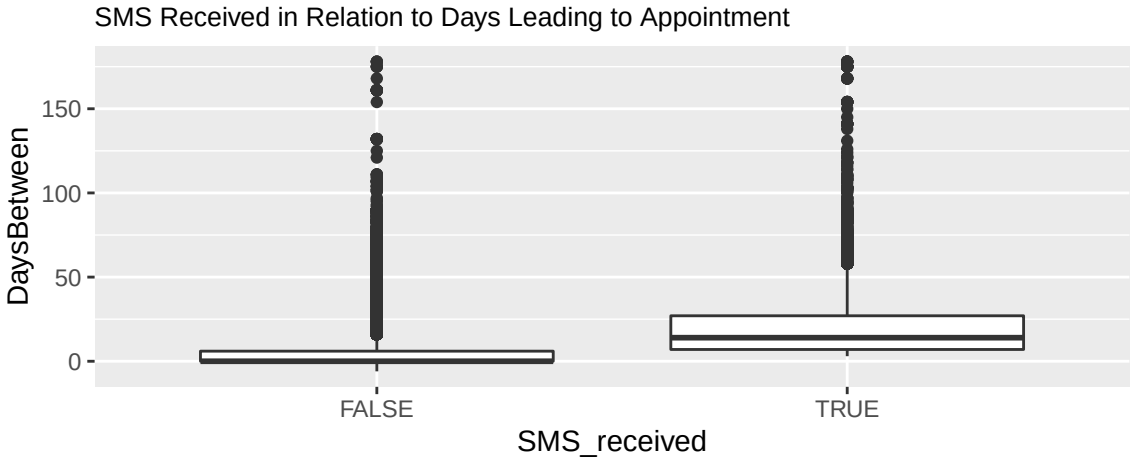

```
# Proportion of non-attendants having received a sms reminder
with(subset(cleandata, Attended=="No"), table(SMS_received)) %>%
  prop.table()

## SMS_received
##     FALSE      TRUE
## 0.5616291 0.4383709

# Proportion of postive attendants having received a sms reminder
with(subset(cleandata, Attended=="Yes"), table(SMS_received)) %>%
  prop.table()

## SMS_received
##     FALSE      TRUE
## 0.7086659 0.2913341
```

Considering a different perspective, of the 80% of attended appointments, 70% of those attended appointments, did not receive an SMS reminder, 30% had received one. Of the 20% of non-attended appointments, 56% of those patients had not received an SMS reminder, yet 43.8% had received one and still not attended.

It would appear then that there is little relationship between appointment attendance and the SMS reminders intended to encourage attending these appointments.

Looking further into the use of SMS reminders, it is not evident what the conditions are for an SMS reminder to be sent. This could have been an opt-in method at the time of making the appointment, if a patient requested to be reminded, or if it is an automated service. One potential theory to explore was if the length of time between making an appoinment and attending the appointment had an impact on whether an SMS reminder was sent.

```
ggplot(cleandata, aes(x=SMS_received, y=DaysBetween)) +
  geom_boxplot() +
  ggtitle("SMS Received in Relation to Days Leading to Appointment") +
  theme(plot.title = element_text(size=10))
```




```
summary(subset(cleandata, SMS_received=="FALSE")$DaysBetween)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -6.000   0.000   0.000   5.992   6.000 178.000

summary(subset(cleandata, SMS_received=="TRUE")$DaysBetween)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    3.00    7.00   14.00   18.97   27.00  178.00
```

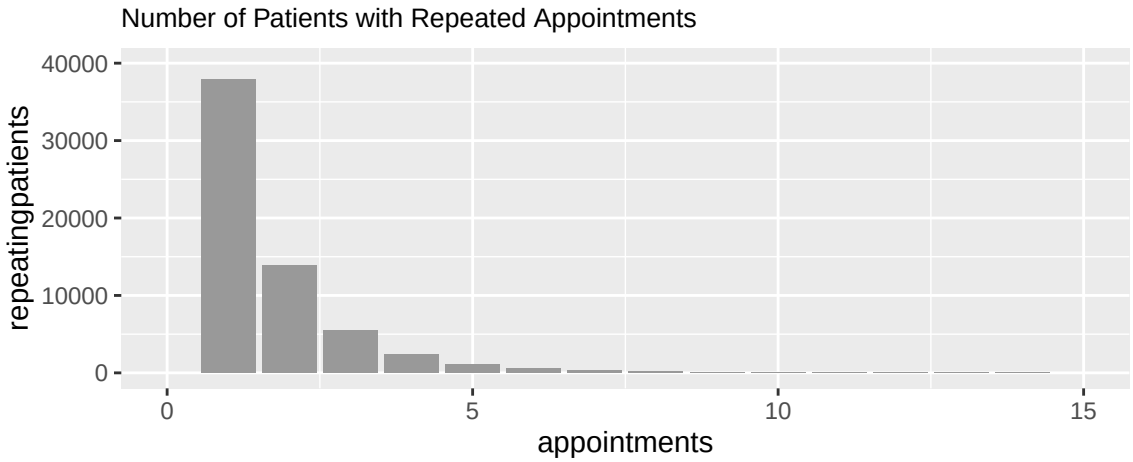
From this figure, we see there is some significant relationship where approximately 75% (up to Q3) of the SMS reminders not sent to patients had 6 or less waiting days in between, and similarly 75% (from Q1) of the SMS reminders which were sent had waiting days of 7 and above. Although as in the box plot, we see some notable exceptions and therefore suggests this is not a hard rule or an automated feature.

Repeating Visits

```
repeatedvisits <- cleandata %>%
      group_by(PatientId) %>%
      summarise(appointments = n()) %>%
      group_by(appointments) %>%
      summarise(repeatingpatients = n())
repeatedvisits

## # A tibble: 44 x 2
##    appointments repeatingpatients
##           <int>             <int>
##  1            1             37920
##  2            2             13895
##  3            3              5500
##  4            4              2367
##  5            5              1119
##  6            6               553
##  7            7               306
##  8            8               202
##  9            9               104
## 10           10                85
## # ... with 34 more rows

summary(repeatedvisits$appointments)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00   11.75   22.50   29.55   43.00   88.00
```

```
ggplot(repeatedvisits, aes(x=appointments, y =repeatingpatients )) +
  geom_bar(stat='identity',fill="#999999") +
  ylim(0,40000) +
  xlim(0,15) +
  ggtitle("Number of Patients with Repeated Appointments") +
  theme(plot.title = element_text(size=10))
```



Another theory to explore is one of repeating visits, and more importantly, repeating non-attendance. Considering the dataset contains appointments only over 5 continous weeks which makes approximately 30 possible appointment days, what is the likelihood of a patient having repeated visits? According to this dataset, as many as 88, which perhaps is suspicious and warrants further investigation.

Predictive Modelling

Pre-modelling data preparation

```
moddata <- cleandata

# Keeping instances with 0 or more days between request and appointment date
moddata <- subset(moddata, DaysBetween>=0)

# Keeping instances with an Age equal or larger than 0
moddata <- subset(moddata, Age>=0)

# Dropping redundant data
moddata$AppointmentID <- NULL
moddata$PatientId <- NULL
moddata$AppointmentDay <- NULL
moddata$ScheduledDay <- NULL
moddata$AppointmentDate <-NULL
moddata$RequestDate <-NULL
moddata$AppointmentTime <- NULL
moddata$RequestTime <-NULL
moddata$samedayappt <- NULL
moddata$No.show <- NULL


# Removing missing data (errors in model)
moddata <- na.omit(moddata)

# Removal of Neighbourhood variable because >53 levels
moddata$Neighbourhood <- NULL
```

The following modifications were made to reduce noise in the model:

- Removed **DaysBetween** and **Age** where instances are less than 0. These are likely errors in recording.
- Removed **AppointmentID** and **PatientID** as these are unique values pertaining to the individual instances.
- Removed **AppointmentDay** and **ScheduledDay** as these are replaced by the 6 new split date and time variables introduced in the cleaning stage.
- Removed **AppointmentDate** and **RequestDate** as these are fixed date values in history and not useful for prediction.
- Removed **AppointmentTime** as these are unique values pertaining to the individual instances.
- Removed **RequestTime** as these are highly unique and potentially not very useful.
- Removed **samedayappt** as it is a gouping of **DaysBetween** which is being retained.
- Removed **No.show** as it is replaced by the new target **Attended**.

Reluctantly, the Neighbourhood variable was also removed as it contained 81 factor levels and the following models have a limit of 53 levels to process. In a future project, this could be explored by potentially one-hot encoding the Neighbourhood variable, or grouping the instances into higher level regional classes.

Sampling the Dataset

```
# Random sampling without addressing class imbalance
nbsampledset <- sample_n(moddata,5000)

# Over and Under sampling to address class imbalance
sampledset <-
  ovun.sample(Attended ~ ., data = moddata, method = "both", p=0.5, N=5000, seed = 1)$data
table(sampledset$Attended)

##
##  Yes   No
## 2564 2436

# Training Control
ctrl <- trainControl(method="repeatedcv", number=5, repeats=2)
```

As the dataset is highly imbalanced, over and undersampling was performed to create a more balanced dataset for prediction. At 110,032 instances even after removing instances with errors, it is a very large dataset to process and for the purposes of being able to run the models, a smaller sample of 5000 instances are used for the following predictive models.

All models in this project are trained with repeating cross-validation.

```
# J48
set.seed(1)
J48nb.mod <- train(Attended ~., data=nbsampledset, method = "J48", trControl = ctrl)
#Check the accuracy
confusionMatrix(J48nb.mod)

## Cross-Validated (5 fold, repeated 2 times) Confusion Matrix
##
## (entries are percentual average cell counts across resamples)
##
##           Reference
## Prediction   No  Yes
##        No   0.0  0.0
##        Yes 19.9 80.1
##
##  Accuracy (average) : 0.8008
```

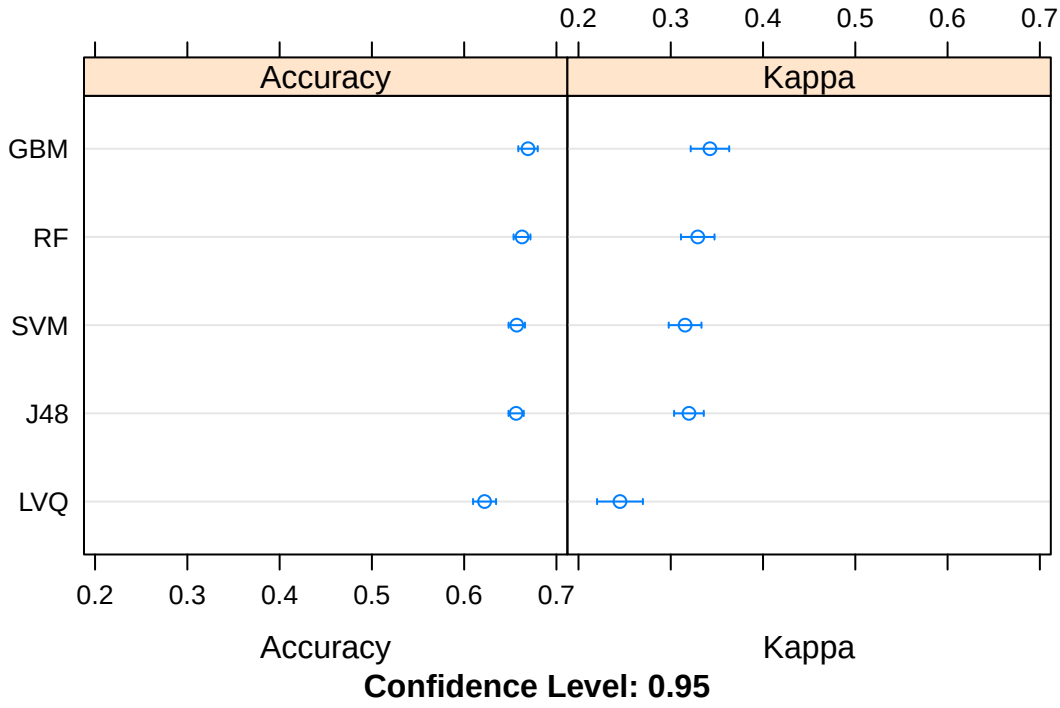
As we see in this example, where the non-balanced set is used with a J48 classifier, the accuracy of the model is high at 80%, but when looking at the confusion matrix, it has done no better than to predict every appointment as being attended (Attended=Yes) to maximise accuracy. While it is the best accuracy out of all the models to follow, it is not any better than taking a simple proportion of the classes and assuming an error every 1 of 5 appointments.

Models

```
# LVQ
set.seed(1)
lvq.mod <- train(Attended~., data=sampledset, method="lvq", trControl=ctrl)
# GBM
set.seed(1)
gbm.mod <- train(Attended~., data=sampledset, method="gbm", trControl=ctrl, verbose=FALSE)
# SVM
set.seed(1)
svm.mod <- train(Attended~., data=sampledset, method="svmRadial", trControl=ctrl)
# J48
```

```r
set.seed(1)
J48.mod <- train(Attended ~., data=sampledset, method = "J48", trControl = ctrl)
# Random Forest
set.seed(1)
rf.mod <- train(Attended~., data=sampledset, method="rf", trControl=ctrl)

results <- resamples(list(LVQ=lvq.mod, GBM=gbm.mod, SVM=svm.mod, J48=J48.mod, RF=rf.mod))
summary(results)

##
## Call:
## summary.resamples(object = results)
##
## Models: LVQ, GBM, SVM, J48, RF
## Number of resamples: 10
##
## Accuracy
##          Min.   1st Qu.    Median      Mean   3rd Qu.  Max. NA's
## LVQ 0.5930000 0.6158046 0.6243128 0.6220987 0.6292780 0.647    0
## GBM 0.6530000 0.6575774 0.6664992 0.6691995 0.6777500 0.698    0
## SVM 0.6413586 0.6473974 0.6530000 0.6569987 0.6674988 0.677    0
## J48 0.6400000 0.6483377 0.6538273 0.6563027 0.6654992 0.675    0
## RF  0.6490000 0.6524239 0.6611683 0.6627013 0.6707500 0.687    0
##
## Kappa
##          Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## LVQ 0.1860846 0.2329980 0.2487253 0.2449819 0.2595981 0.2955385    0
## GBM 0.3111580 0.3187399 0.3369751 0.3424764 0.3599343 0.3993421    0
## SVM 0.2841640 0.2959679 0.3074044 0.3155449 0.3365830 0.3561428    0
## J48 0.2906600 0.3019845 0.3136923 0.3197124 0.3369349 0.3577481    0
## RF  0.3022091 0.3080684 0.3260503 0.3292416 0.3452744 0.3779463    0
```

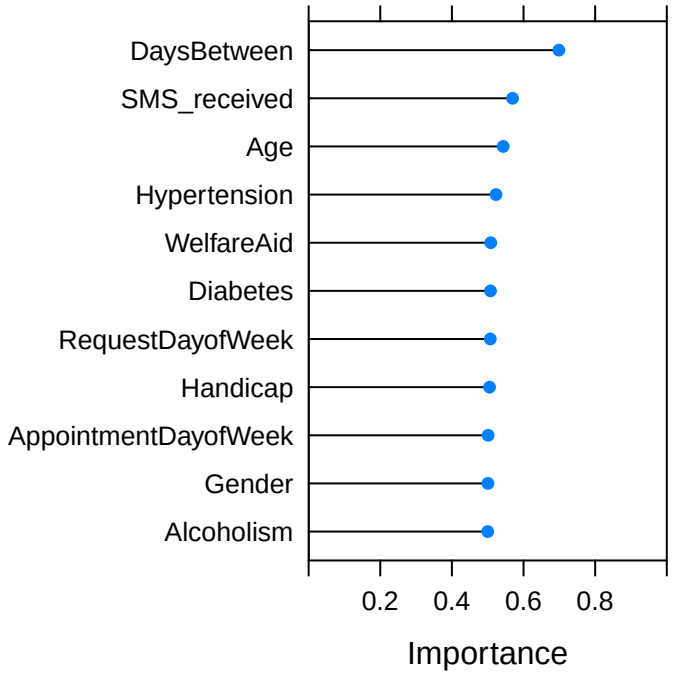
```r
dotplot(results)
```




```
#Checking the model
confusionMatrix(gbm.mod)

## Cross-Validated (5 fold, repeated 2 times) Confusion Matrix
##
## (entries are percentual average cell counts across resamples)
##
##           Reference
## Prediction  Yes   No
##        Yes 28.1  9.8
##        No  23.2 38.9
##
##  Accuracy (average) : 0.6692
```

The Gradient Boosting Model (GBM) performed the best of the 5 models, although only statstically significant when compared to the worst performing model Learning Vector Quantization (LVQ).

```
# Variable importance
plot(varImp(svm.mod, scale=FALSE),xlim=c(0,1))
```




```
print(varImp(svm.mod, scale=FALSE))

## ROC curve variable importance
##
##                       Importance
## DaysBetween               0.6988
## SMS_received              0.5693
## Age                       0.5431
## Hypertension              0.5231
## WelfareAid                0.5086
## Diabetes                  0.5078
## RequestDayofWeek          0.5072
## Handicap                  0.5052
## AppointmentDayofWeek      0.5012
## Gender                    0.5007
## Alcoholism                0.5000
```

When inspecting the importance of variables, it's no surprise given the analysis before that the length of time between when the appointment was requested and when it occured is deemed the most important feature, and whether an SMS reminder is sent is second.

Results and Conclusion

Having passed a balanced sample of 5,000 instances through 5 different models, none of the models have performed exceptionally well. The model with best accuracy was actually the J48 using an imbalanced dataset, but it also had the worst Recall and Precision (as 0) if considering the "No" class was the intended target since it never predicted a patient as failing to attend.

Of all the other models performed on the over and under sampled balanced dataset, GBM returned the highest Accuracy and Kappa statistic, but it is perhaps being overly pessimistic in predicting more non-attending

patients than there really are. The dimensionality of the features is a factor in the models' performance, but perhaps most of all, as a result of all the exploratory analysis performed, I am not convinced the variables are relevant to predicting appointment attendance to begin with.

Ultimately, the predictive models have not been a great success, more could have been done to tune the models but given the data, I am not sure given the dataset and relationships between the variables, that it would have produced a much better effect. From this project, perhaps the best value can be gained from having performed the exploratory analysis. From this, we know that the waiting time between making and attending an appoinment is the best indicator of attendance, specifically same-day appoinments having the best attendance rates. It can also be determined that age, gender, and some types of medical conditions do not impact attendance, and perhaps this may aid future research in at least eliminating what is not relevant.