

Capstone Project - Predicting Patient No-Shows Using Appointment Data

Derek Samsom

Missed medical appointments are a major problem in the medical industry, resulting in lost revenue. Medical providers can over-book appointments to try to minimize the lost revenue, but without any way to predict the probability of an appointment being missed, there will still be times where more or fewer patients show up at a given time than expected. The result will be that lost revenue will be reduced but not eliminated, as there will still be times that more appointments are missed than expected. There will also be times more appointments show up than expected, which can overwhelm staff and resources and affect the level of patient care.

This project is a classification problem that will explore the prediction of whether a medical appointment will be missed, and its probability of being kept or missed. The prediction error will result in times where there are too many or too few patients at a given time. The main goal in the prediction will be to minimize the error, as this will reduce instances of having more or fewer patients than desired.

There are countless reasons and circumstances that can lead someone to miss an appointment, such as a last minute work meeting or a family emergency, that aren't directly captured in the data and are impossible to know in advance. Missed appointments can only be predicted based on indirect factors that are known, such as past history and demographics. Because of this, there will be a level of error that cannot be eliminated, however, any reduction in error compared to having no predictive model at all is still beneficial.

Medical providers can use the missed appointment predictions by incorporating them into their booking methods and systems. The methods used in booking will have to consider the implications of the inherent prediction errors and balance the risk the errors represent: too many patients leading to staff/resource shortage, and too few patients leading to lost revenue. The methods of implementing the use of missed appointment predictions into an appointment booking system are client-specific and not included in the scope of this project, which is limited to minimizing the error while predicting the probability that an appointment will be missed or kept.

I will start off by loading the required packages and the data.

```
library(tidyverse)
library(lubridate)
library(caret)
library(randomForest)
library(GGally)
```

Read data and assign to `appointments`

```
appointments <- read_csv("Final_Data.csv")
```

```
## Parsed with column specification:
## cols(
##   kept_status = col_character(),
##   appt_date = col_character(),
##   appt_time = col_time(format = ""),
##   appt_length = col_integer(),
##   date_scheduled = col_character(),
##   patient_age = col_integer(),
##   patient_gender = col_character(),
##   billing_type = col_character(),
##   prior_missed = col_integer(),
```

```
## prior_kept = col_integer(),
## patient_distance = col_integer(),
## office_zip = col_character(),
## provider_specialty = col_character(),
## remind_call_result = col_character()
## )
```

```
appointments_original <- appointments
zipcodes <- read_csv("zipcodes.csv")
```

```
## Parsed with column specification:
## cols(
##   office_zip = col_character(),
##   county_code = col_character()
## )
```

The raw data, which has been named `appointments`, contains information on 342862 past appointments, pre-sorted by the date and time of appointment. The depended variable, `kept_status`, shows whether the appointment was kept or missed.

There is no field that can be used to identify a specific patients in the data set. A patient may have had more than one appointment during the time-period represented in the data, meaning that one individual patient may make up one or multiple observations. If there was a patient ID field, it would allow the data to be grouped by patient and give the option of organizing the data by patient rather than by appointment.

A secondary data set, `zipcodes`, has information about the county the offices are located in. This will be used to see if the location can help predict whether an appointment will be missed. The county names are converted to a 2-letter code for confidentiality.

Data Summary and Structure

```
summary(appointments)
```

```
## kept_status      appt_date      appt_time      appt_length
## Length:342862    Length:342862    Length:342862    Min.   : 10
## Class :character  Class :character  Class1:hms        1st Qu.: 60
## Mode  :character  Mode  :character  Class2:difftime   Median : 60
##                                     Mode   :numeric   Mean   : 57
##                                     3rd Qu.: 60
##                                     Max.   :600
##
## date_scheduled   patient_age      patient_gender    billing_type
## Length:342862     Min.   : 0.00    Length:342862     Length:342862
## Class :character  1st Qu.: 17.00   Class :character   Class :character
## Mode  :character  Median : 34.00   Mode  :character   Mode  :character
##                                     Mean   : 35.56
##                                     3rd Qu.: 54.00
##                                     Max.   :264.00
##
## prior_missed      prior_kept      patient_distance  office_zip
## Min.   : 0.000     Min.   : 0.00    Min.   : 0.0      Length:342862
## 1st Qu.: 1.000     1st Qu.: 2.00    1st Qu.: 0.0      Class :character
## Median : 2.000     Median : 6.00    Median : 3.0      Mode  :character
## Mean   : 2.451     Mean   : 8.02    Mean   : 10.8
## 3rd Qu.: 3.000     3rd Qu.: 11.00   3rd Qu.: 9.0
```

```
## Max.      :117.000   Max.      :676.00   Max.      :2688.0
##                                     NA's      :974
## provider_specialty remind_call_result
## Length:342862      Length:342862
## Class :character   Class :character
## Mode  :character   Mode  :character
##
##
##
##
```

```
str(appointments, give.attr = FALSE)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':   342862 obs. of  14 variables:
## $ kept_status      : chr  "Kept" "Kept" "Kept" "Kept" ...
## $ appt_date        : chr  "9/1/16" "9/1/16" "9/1/16" "9/1/16" ...
## $ appt_time        :Classes 'hms', 'difftime' atomic [1:342862] 19800 28800 28800 28800 28800 28800 ...
## $ appt_length      : int   90 60 120 60 60 60 60 60 60 90 ...
## $ date_scheduled   : chr  "8/1/16" "1/18/16" "2/3/16" "6/8/16" ...
## $ patient_age      : int   7 75 31 45 49 71 49 38 36 13 ...
## $ patient_gender    : chr  "Male" "Female" "Male" "Male" ...
## $ billing_type      : chr  "DMAP" "Commercial" "DMAP" "DMAP" ...
## $ prior_missed      : int   1 2 1 6 5 6 8 0 2 3 ...
## $ prior_kept        : int   3 5 5 15 6 6 20 0 5 12 ...
## $ patient_distance  : int   41 29 5 5 0 5 0 539 0 4 ...
## $ office_zip        : chr  "AP" "BL" "BL" "BL" ...
## $ provider_specialty: chr  "A" "A" "A" "B" ...
## $ remind_call_result: chr  "Left Message" "Answered - Confirmed" "Left Message" "Answered - No Response"
```

```
head(appointments[, 1:5])
```

```
## # A tibble: 6 x 5
##   kept_status appt_date appt_time appt_length date_scheduled
##   <chr>      <chr>      <time>      <int> <chr>
## 1 Kept      9/1/16      05:30         90 8/1/16
## 2 Kept      9/1/16      08:00         60 1/18/16
## 3 Kept      9/1/16      08:00        120 2/3/16
## 4 Kept      9/1/16      08:00         60 6/8/16
## 5 Missed    9/1/16      08:00         60 6/28/16
## 6 Kept      9/1/16      08:00         60 7/12/16
```

```
head(appointments[, 6:10])
```

```
## # A tibble: 6 x 5
##   patient_age patient_gender billing_type prior_missed prior_kept
##   <int> <chr>      <chr>      <int>      <int>
## 1      7 Male      DMAP         1         3
## 2     75 Female    Commercial    2         5
## 3     31 Male      DMAP         1         5
## 4     45 Male      DMAP         6        15
## 5     49 Male      Commercial    5         6
## 6     71 Male      DMAP         6         6
```

```
head(appointments[, 11:14])
```

```
## # A tibble: 6 x 4
##   patient_distance office_zip provider_specialty remind_call_result
```

##	<int>	<chr>	<chr>	<chr>
## 1	41	AP	A	Left Message
## 2	29	BL	A	Answered - Confirmed
## 3	5	BL	A	Left Message
## 4	5	BL	B	Answered - No Response
## 5	0	BL	B	Answered - No Response
## 6	5	BL	A	Answered - Confirmed

Data Dictionary

```
variable_descriptions <- c(
  "Dependent variable: kept or missed",
  "Appointment date",
  "Appointment time",
  "Appointment length in minutes",
  "Date appointment was scheduled",
  "Patient age",
  "Patient gender",
  "Billing type",
  "Number of prior missed appointments",
  "Number of prior kept appointments",
  "Patient distance from office in miles",
  "Office Zip Code - Anonymized",
  "Provider primary specialty code",
  "Reminder Call result")
variable <- colnames(appointments)

as_data_frame(cbind(c(1:length(variable)), variable, variable_descriptions))
```

```
## # A tibble: 14 x 3
##   V1    variable    variable_descriptions
##   <chr> <chr>         <chr>
## 1 1    kept_status    Dependent variable: kept or missed
## 2 2    appt_date      Appointment date
## 3 3    appt_time      Appointment time
## 4 4    appt_length    Appointment length in minutes
## 5 5    date_scheduled Date appointment was scheduled
## 6 6    patient_age    Patient age
## 7 7    patient_gender  Patient gender
## 8 8    billing_type    Billing type
## 9 9    prior_missed    Number of prior missed appointments
## 10 10   prior_kept      Number of prior kept appointments
## 11 11   patient_distance Patient distance from office in miles
## 12 12   office_zip     Office Zip Code - Anonymized
## 13 13   provider_specialty Provider primary specialty code
## 14 14   remind_call_result Reminder Call result
```

The `appt_date` and `appt_time` variables can be combined into one variable, `appt_datetime`.

```
appointments <- appointments %>%
  mutate(appt_datetime = lubridate::mdy_hms(paste(appt_date, appt_time)))

appointments$date_scheduled <- lubridate::as_date(
  appointments$date_scheduled, format = "%m/%d/%y", tz = "UTC")
```

Data Exploration

First I want to calculate the percent of missed appointments overall by creating a logical variable `missed`, where 1 represents a missed appointment and 0 represents a kept appointment. This will determine the degree of class imbalance.

```
appointments <- appointments %>%  
  mutate(missed = ifelse(appointments$kept_status == "Missed", 1, 0))  
missed_rate <- mean(appointments$missed)  
missed_rate
```

```
## [1] 0.1592944
```

15.93 % of the total appointments are missed. This is an imbalanced classification, which will have implications in the modeling. For example, the model could predict all of the appointments will be kept and be correct 84.07 % of the time. This results in a high accuracy without providing any useful prediction of which appointments will be missed.

Next I want to check the data to see if there are any missing values that could indicate reduced data integrity or adversely affect the modelling.

```
map_dbl(appointments, ~sum(is.na(.)))
```

```
##      kept_status      appt_date      appt_time  
##           0           0           0  
##      appt_length  date_scheduled  patient_age  
##           0           0           0  
##      patient_gender  billing_type  prior_missed  
##           0           0           0  
##      prior_kept  patient_distance  office_zip  
##           0           974           0  
## provider_specialty remind_call_result  appt_datetime  
##           0           0           0  
##           missed  
##           0
```

One variable, `patient_distance` has 974 missing value. This is fairly minor and will be evaluated later on when exploring the variable further.

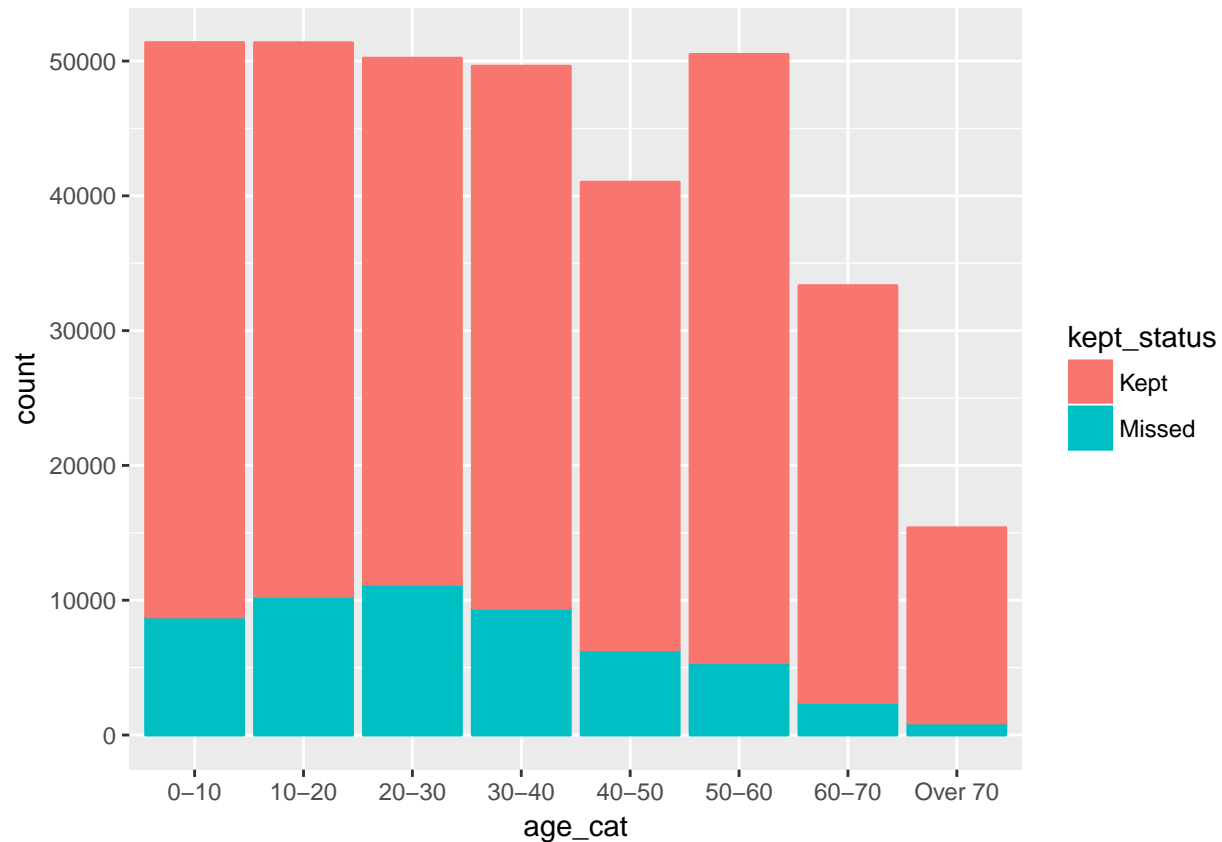
patient_age

I expected missed appointments to have to vary across age ranges. Perhaps older patients have fewer commitments with kids or work, and make their appointments more regularly, or perhaps younger adults might skip more appointments because they aren't as critical? I will break the data into age groups to make the plot simpler to evaluate.

There are a small number of observations where the age is higher than plausible. Therefore, the observations greater than age 110 will be removed from the data.

```
age_labels <- c("0-10", "10-20", "20-30", "30-40", "40-50", "50-60", "60-70",  
              "Over 70")  
age_breaks <- c(-1, 10, 20, 30, 40, 50, 60, 70, 111)  
  
appointments <- appointments %>%  
  filter(patient_age <= 110) %>%  
  mutate(  
    age_cat = cut(patient_age, breaks = age_breaks, labels = age_labels))
```

```
ggplot(
  appointments,
  aes(x = age_cat, color = kept_status, fill = kept_status)
) +
  stat_count()
```



Missed appointments are highest with young adults, and decrease with older and younger patients.

billing_type

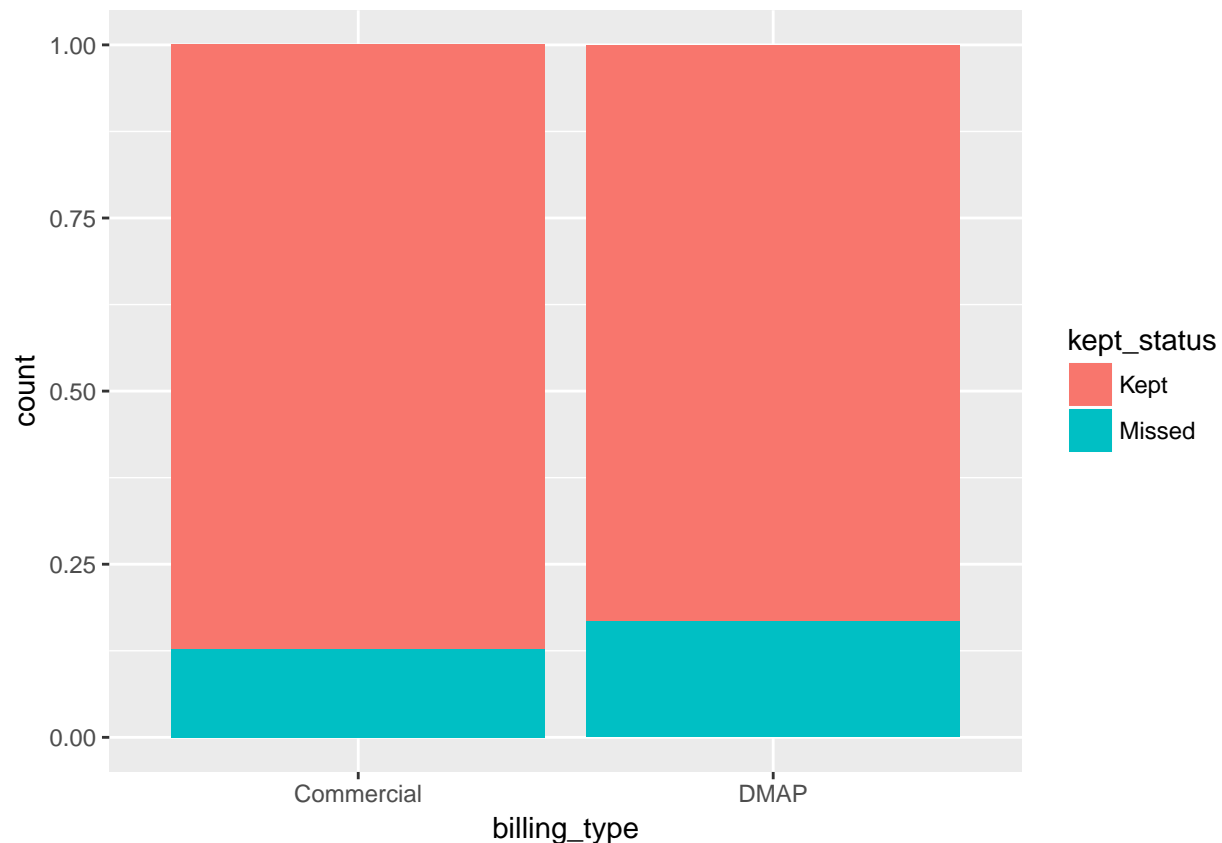
```
table(appointments$billing_type)
```

```
##
##      Commercial      DMAP To Be Assigned
##      78282          264500          1
```

There is only one observation of “To Be Assigned”, therefore it will be removed from the data.

```
appointments <- subset(appointments, billing_type != "To Be Assigned")
```

```
ggplot(
  appointments,
  aes(x = billing_type, fill = kept_status)
) +
  geom_bar(position = "fill")
```



There is a minor difference between billing types. DMAP has a higher proportion of missed appointments than commercial.

appt_datetime

For the variable `appt_datetime`, I will create an `hour` variable to see the variation in missed appointments by hour of day.

```
appointments <- appointments %>%
  mutate(hour = lubridate::hour(appointments$appt_datetime))

table(appointments$hour)
```

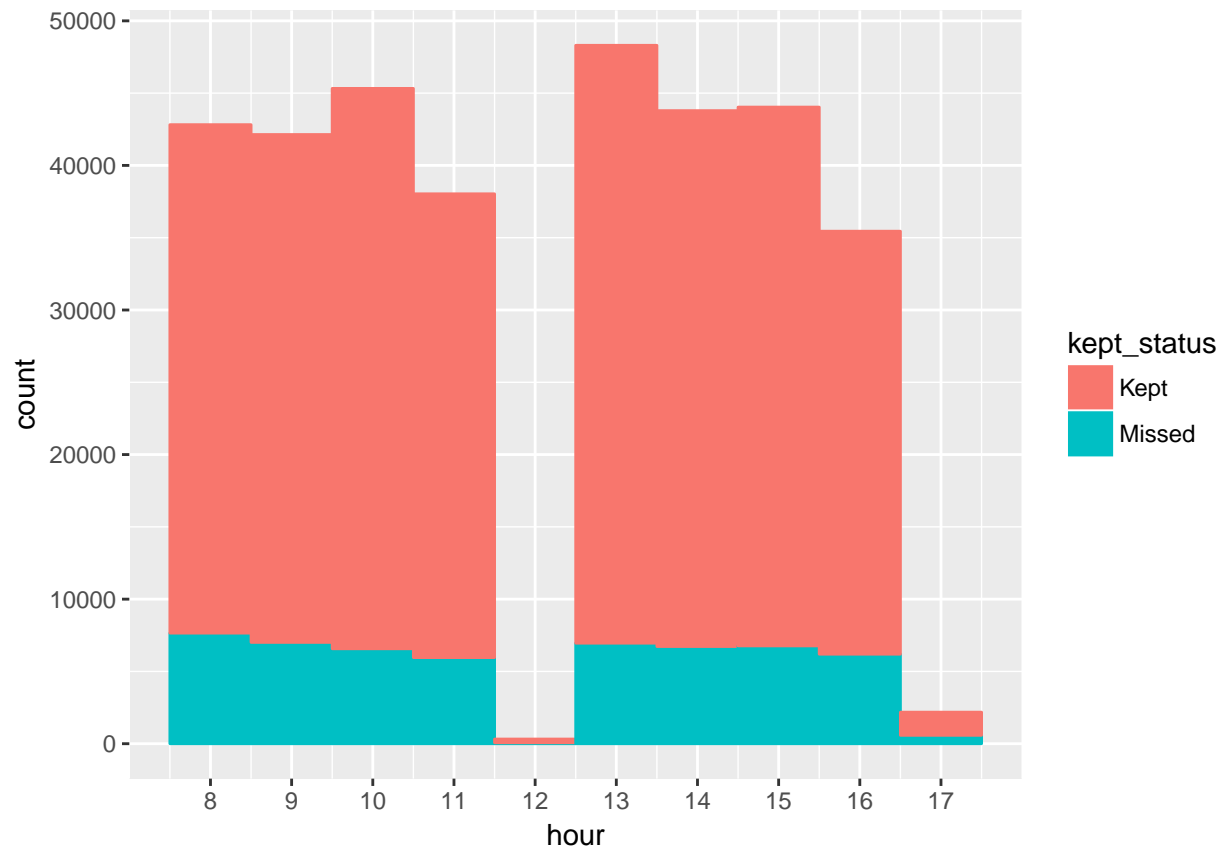
```
##
##      0      5      6      7      8      9     10     11     12     13     14     15
##      7     24     25    98 42816 42133 45326 38033   321 48307 43787 44033
##     16     17     18     19     20     21
## 35449 2180  205   33    3    2
```

Most appointments are scheduled between 8:00 AM and 5:00 PM, with an hour gap starting at 12:00.

```
appointments_hour <- appointments %>%
  select(kept_status, hour) %>%
  filter(hour >= 8 & hour <= 17)

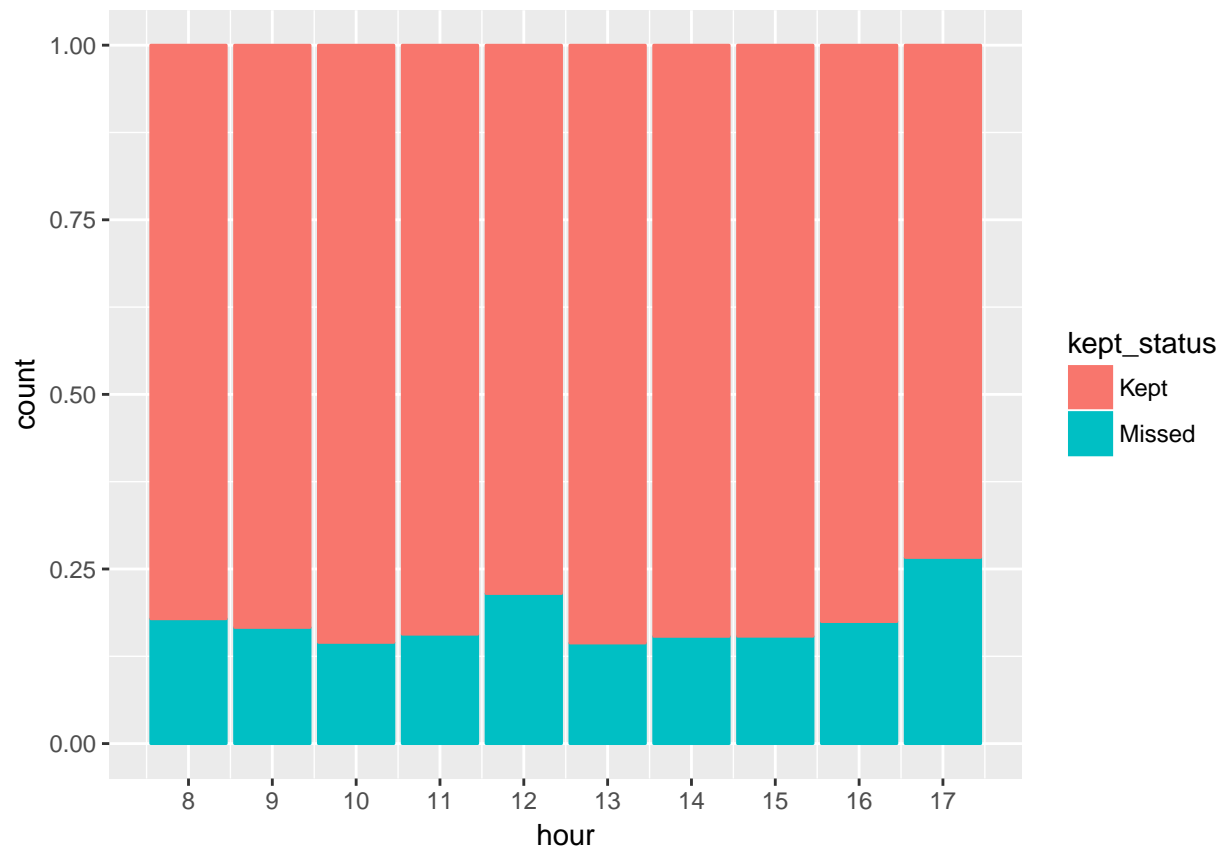
ggplot(
  appointments_hour,
  aes(x = hour, col = kept_status, fill = kept_status)
```

```
) +  
  geom_histogram(binwidth = 1) +  
  scale_x_continuous(breaks = seq(8, 17, 1))
```



There is a decline in the total number of missed appointments as both the morning and afternoon period progress, however, there are fewer appointments towards the end of the two periods.

```
ggplot(  
  appointments_hour,  
  aes(x = hour, col = kept_status, fill = kept_status)  
) +  
  geom_bar(position = "fill") +  
  scale_x_continuous(breaks = seq(8, 17, 1))
```

Proportionally more appointments are missed at the beginning and end of the typical scheduling hours, and during the few noon appointments.

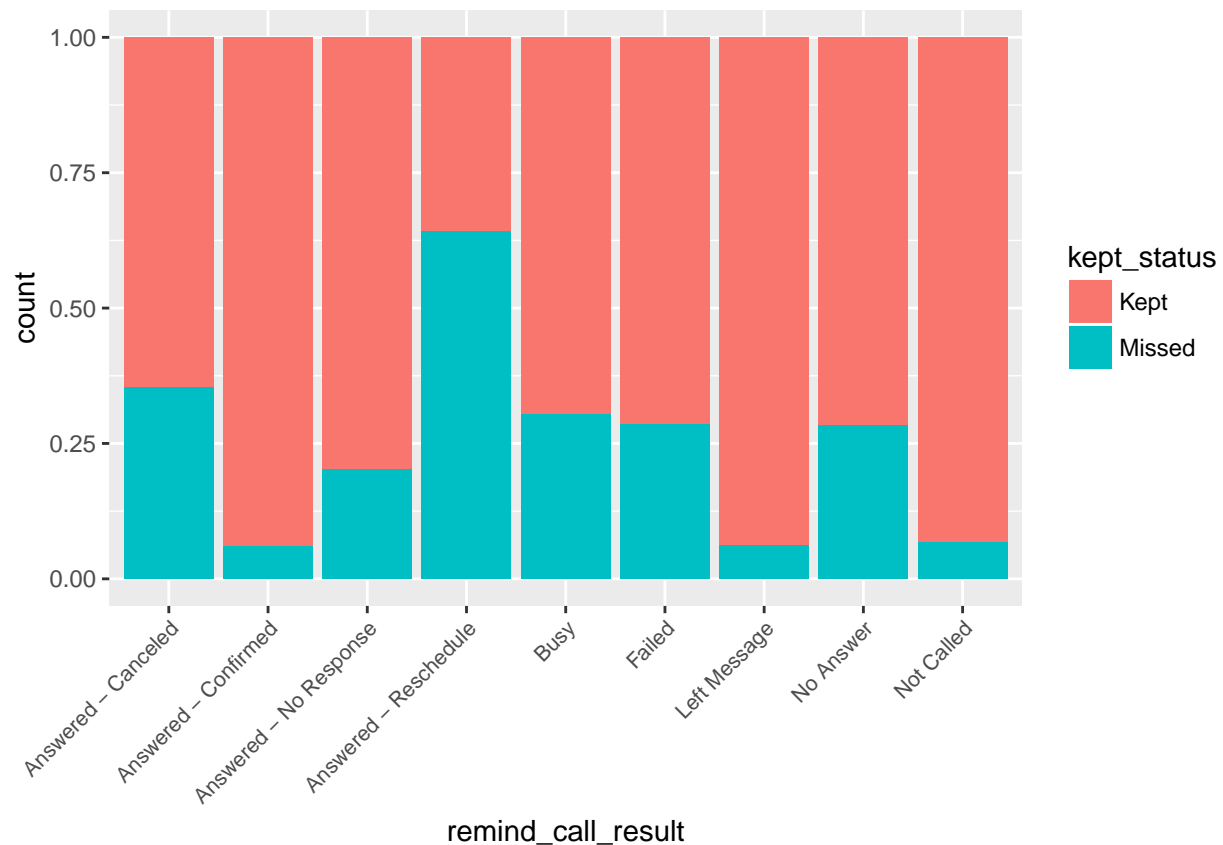
remind_call_result

```
table(appointments$remind_call_result)
```

```
##
##      Answered - Canceled      Answered - Confirmed Answered - No Response
##              152              49108              180869
## Answered - Reschedule              Busy              Failed
##              1369              1104              27944
##           Left Message              No Answer              Not Called
##              18430              377              63429
```

Low counts of “Answered - Cancelled”, “Answered - Reschedule”, “Busy”, and “No Answer”

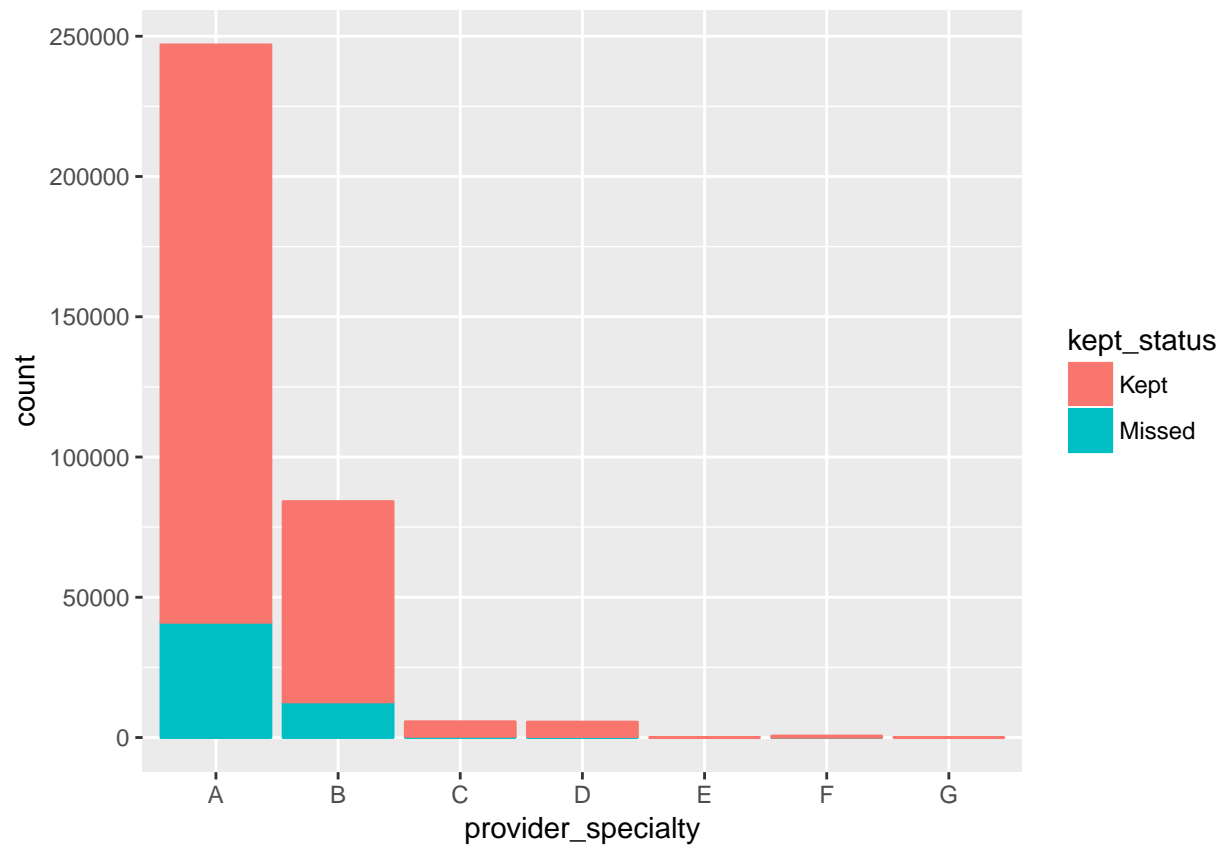
```
ggplot(
  appointments,
  aes(x = remind_call_result, fill = kept_status)
) +
  geom_bar(position = "fill") +
  theme(axis.text.x = element_text(size = 8, angle = 45, hjust = 1, vjust = 1))
```



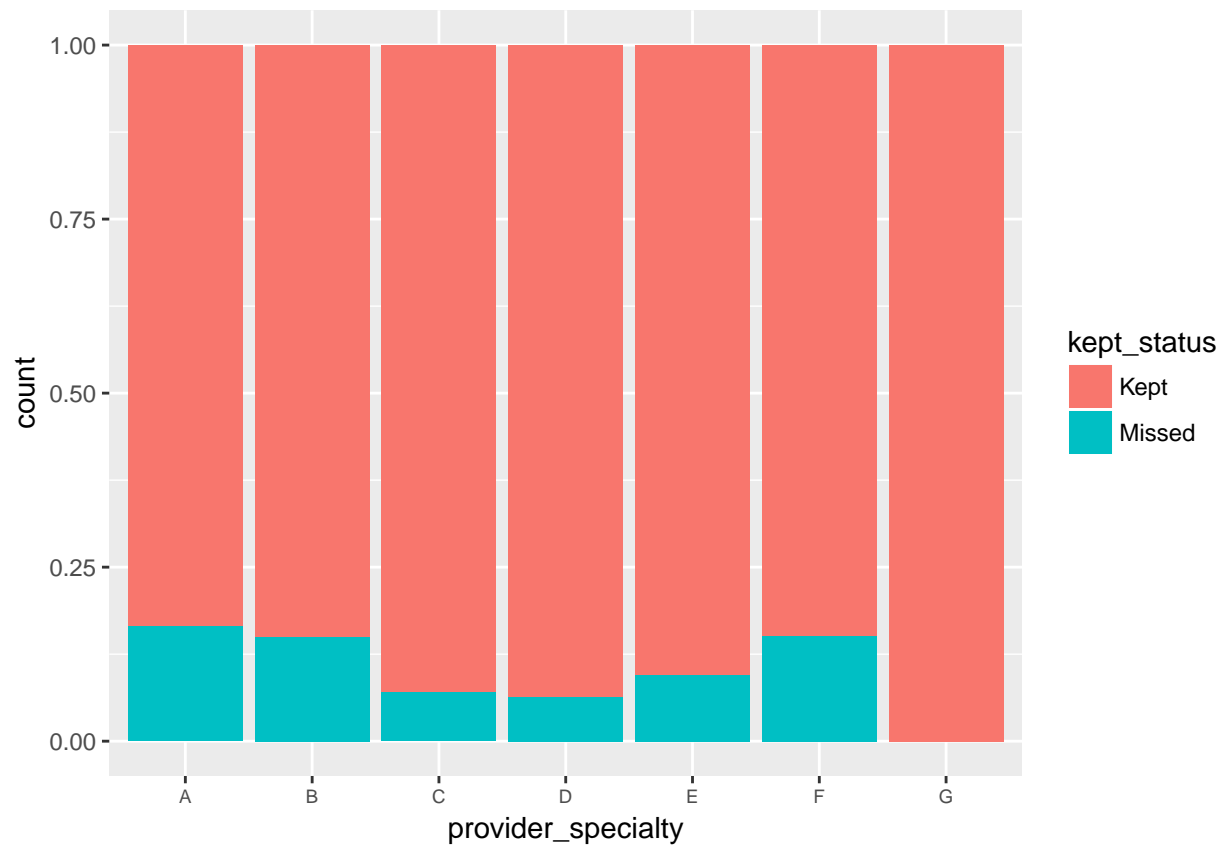
~65% of appointments with “Answered - Cancelled” and ~35% with “Answered-Reschedule” still kept their appointments, however, very few observations in these categories.

provider_specialty (Should I even include this in the report? not as interesting since I have the specialties encoded)

```
ggplot(
  appointments,
  aes(x = provider_specialty, col = kept_status, fill = kept_status)
) +
  stat_count()
```



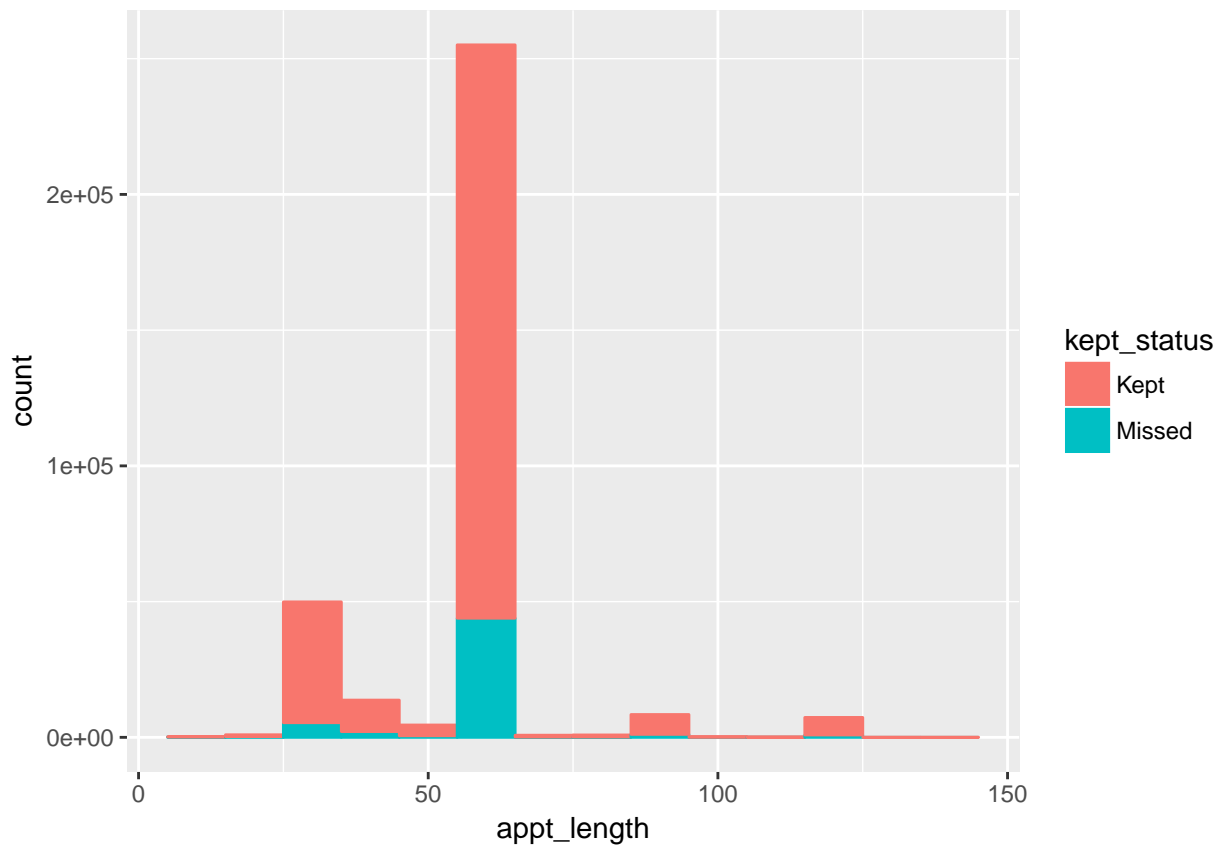
```
ggplot(  
  appointments,  
  aes(x = provider_specialty, fill = kept_status)  
) +  
  geom_bar(position = "fill") +  
  theme(axis.text.x = element_text(size = 7))
```



C, D, and E provider specialties have lower proportion of missed appointments.

appt_length

```
appointments %>%
  filter(appt_length < 150) %>%
  ggplot(
    aes(x = appt_length, color = kept_status, fill = kept_status)
  ) +
    geom_histogram(binwidth = 10)
```



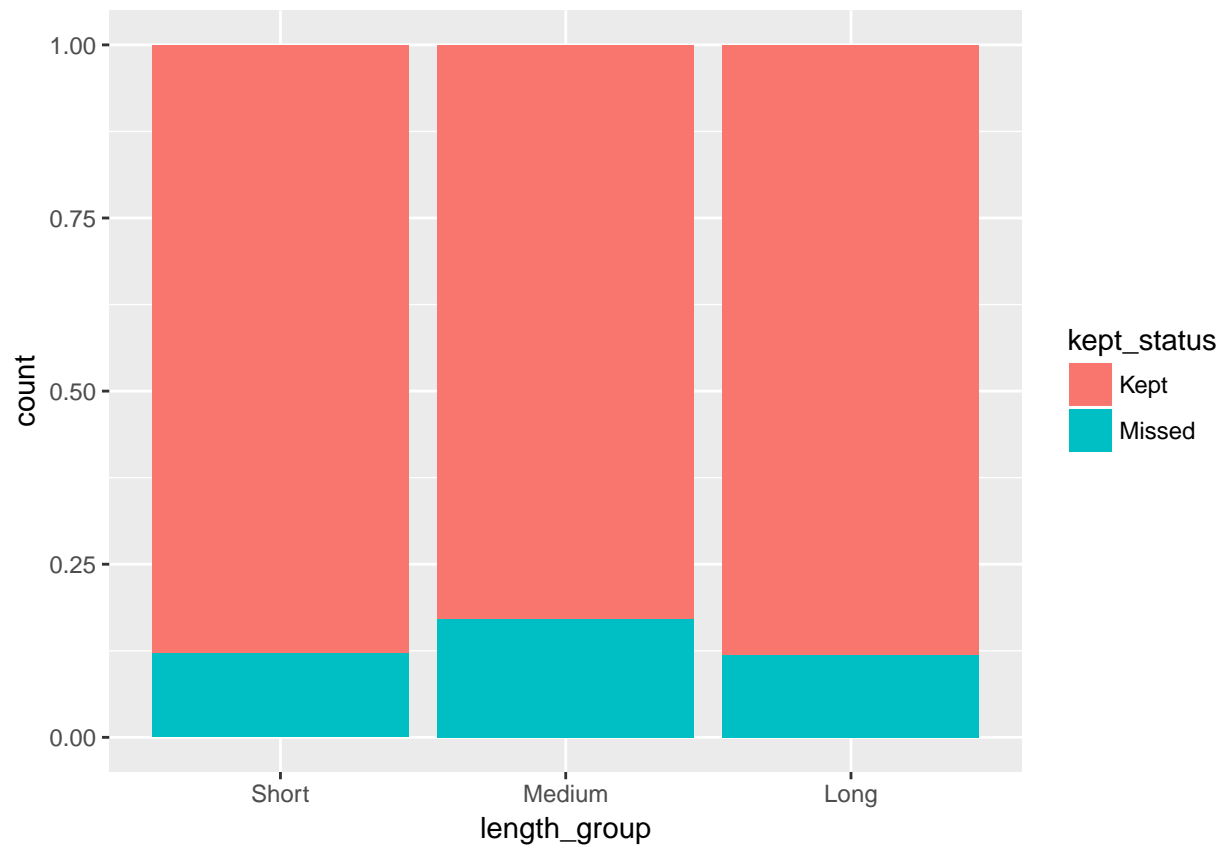
Most Appointments are 60 minutes long. 30-minute appointments are next most common.

```
length_breaks <- c(-1, 45, 75, 1000)

length_labels <- c("Short", "Medium", "Long")

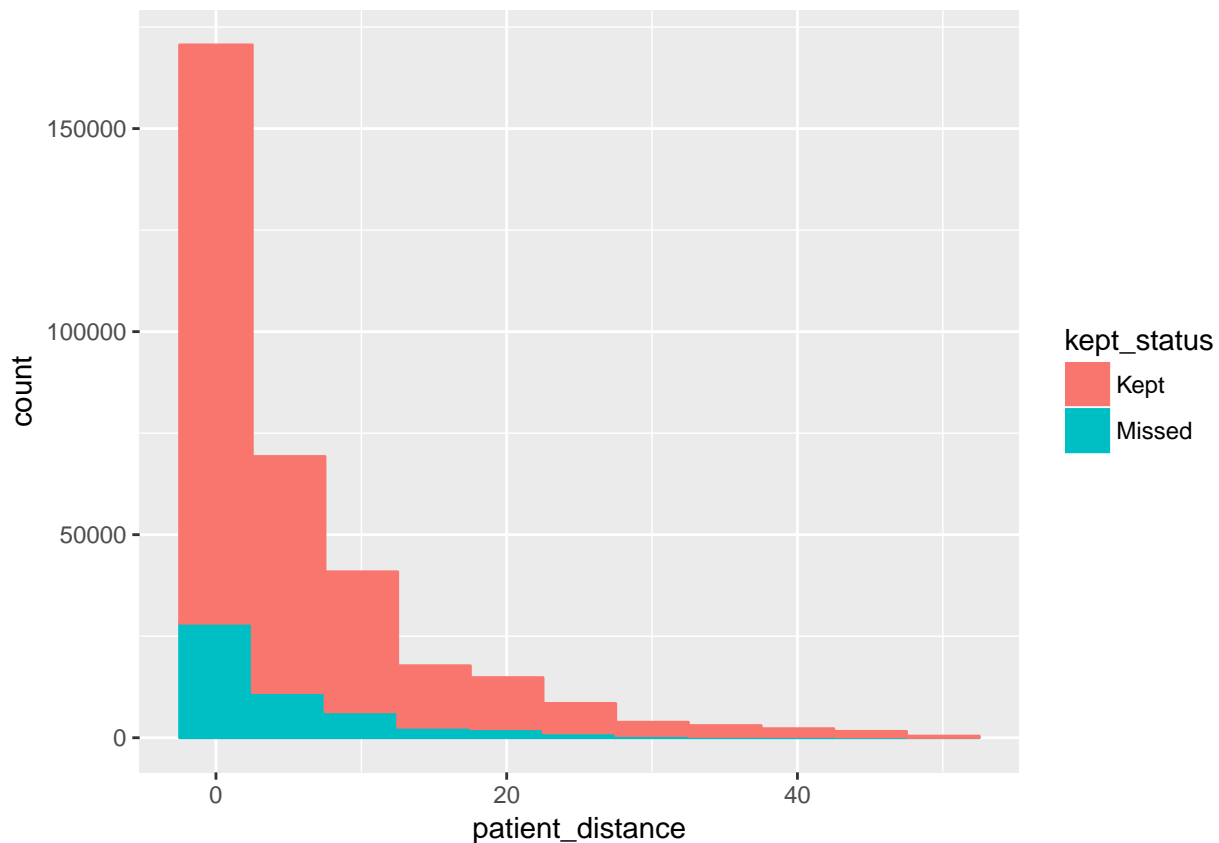
appointments <- appointments %>%
  mutate(
    length_group = cut(
      appt_length, breaks = length_breaks, labels = length_labels)
  )

ggplot(
  data = appointments,
  mapping = aes(x = length_group, fill = kept_status)
) +
  geom_bar(position = "fill")
```



patient_distance

```
appointments %>%  
  filter(patient_distance < 50) %>%  
  ggplot(  
    aes(x = patient_distance, color = kept_status, fill = kept_status)  
  ) +  
    geom_histogram(binwidth = 5)
```



patient_distance is very right-skewed, therefore NA values will be replaced with median rather than mean.

```
appointments$patient_distance <- appointments$patient_distance %>%
  replace_na(median(appointments$patient_distance, na.rm = TRUE))
```

New Variables

In addition to the original variables, there are several additional variables that can be calculated based on the originals.

The `percent_missed` variable is the percentage of prior appointments missed, calculated by dividing the prior missed appointments by the total number of prior appointments. For new patients, this calculation will result in an error because it will be attempting to divide by zero.

The variable `is_new_patient` will specify whether a patient is new, represented by a 1, or existing, represented by 0. My hypothesis is that new patients are more likely to keep their appointments, since I think it is human nature to try to give a good first impression. This is calculated by searching for appointments where `prior_missed` and `prior_kept` are both 0.

The variable `appt_lead_time` will calculate how far in advance an appointment was booked. This is calculated by taking the difference between `date_scheduled` and `appt_date`. If people are more likely to forget appointments booked farther in advance, or they are more likely to be for less urgent preventative care than last minute appointments, this will pick that up.

The variable `appt_weekday` is the day of the week the appointments occurs, and `weekday_scheduled` is the date the appointment was booked.

```
appointments <- appointments %>%
  mutate(percent_missed = prior_missed / (prior_missed + prior_kept)) %>%
```

```

mutate(
  is_new_patient = ifelse(prior_missed == 0 & prior_kept == 0, 1, 0)) %>%
mutate(appt_lead_time = date(appt_datetime) - date(date_scheduled)) %>%
mutate(appt_weekday = strftime(appt_datetime, "%A")) %>%
mutate(weekday_scheduled = strftime(date_scheduled, "%A"))

appointments$percent_missed <- as.integer(appointments$percent_missed * 100)
appointments$percent_missed <- appointments$percent_missed %>%
  tidyr::replace_na(0)

```

Add county_code from zipcode data.

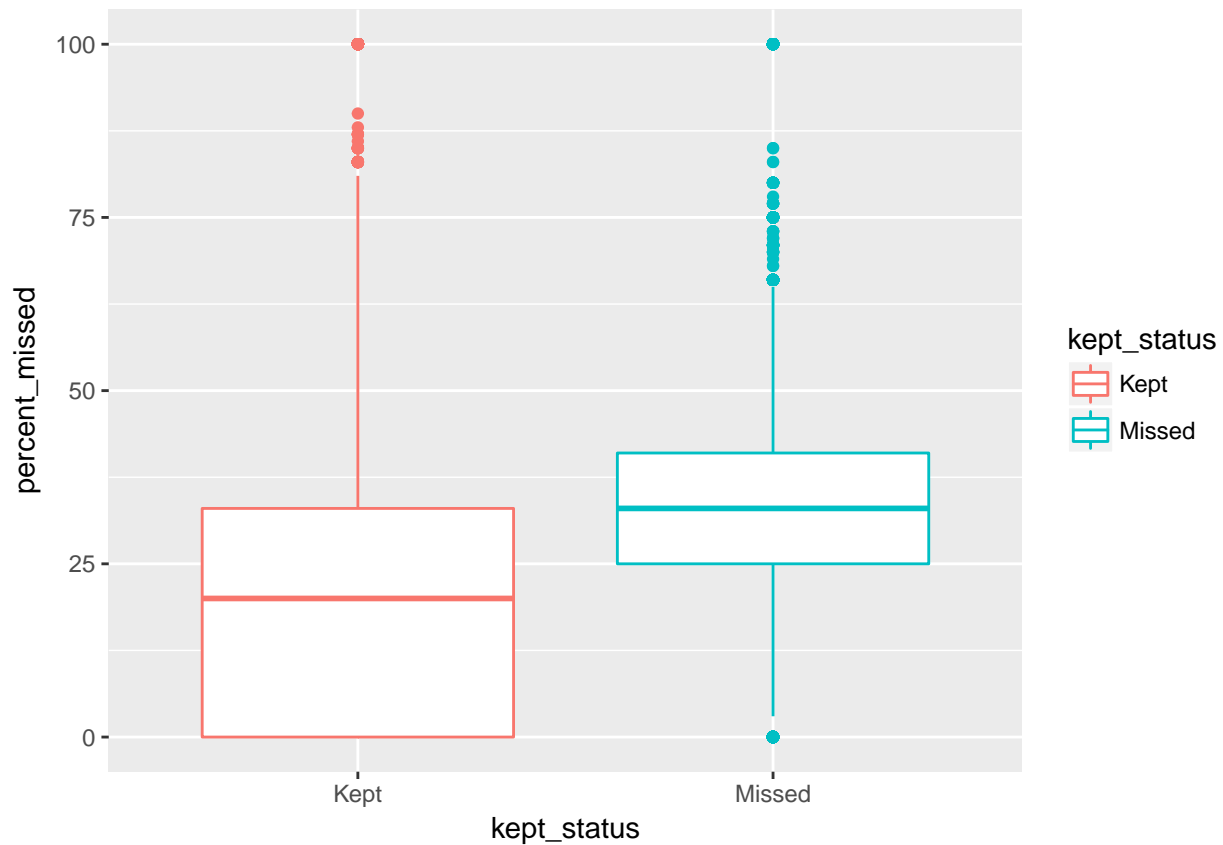
```
appointments <- dplyr::left_join(appointments, zipcodes, by = "office_zip")
```

percent_missed

```

ggplot(
  data = appointments,
  aes(x = kept_status, y = percent_missed, col = kept_status)
) +
  geom_boxplot()

```

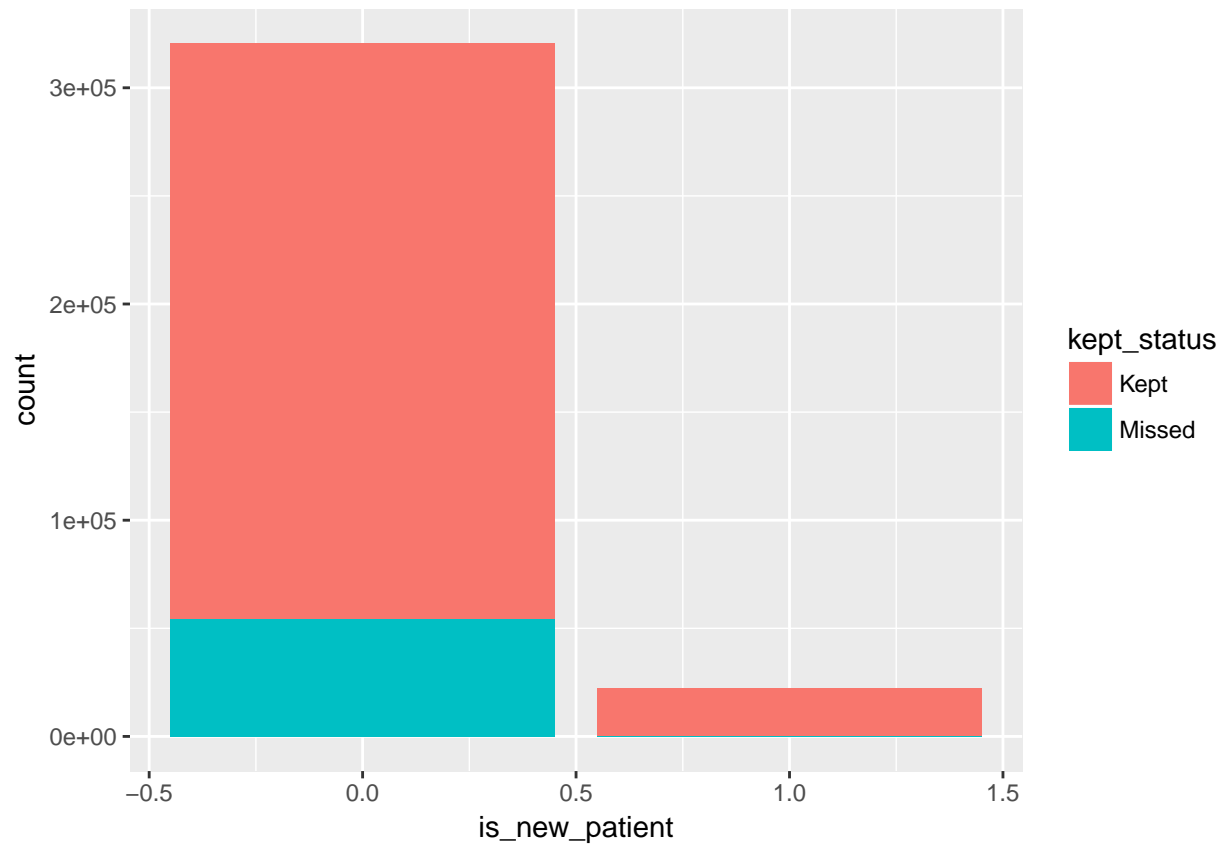


is_new_patient

```
table(appointments$is_new_patient)
```

```
##  
##      0      1  
## 320442 22340
```

```
ggplot(  
  appointments,  
  aes(x = is_new_patient, fill = kept_status)  
) +  
  geom_bar()
```

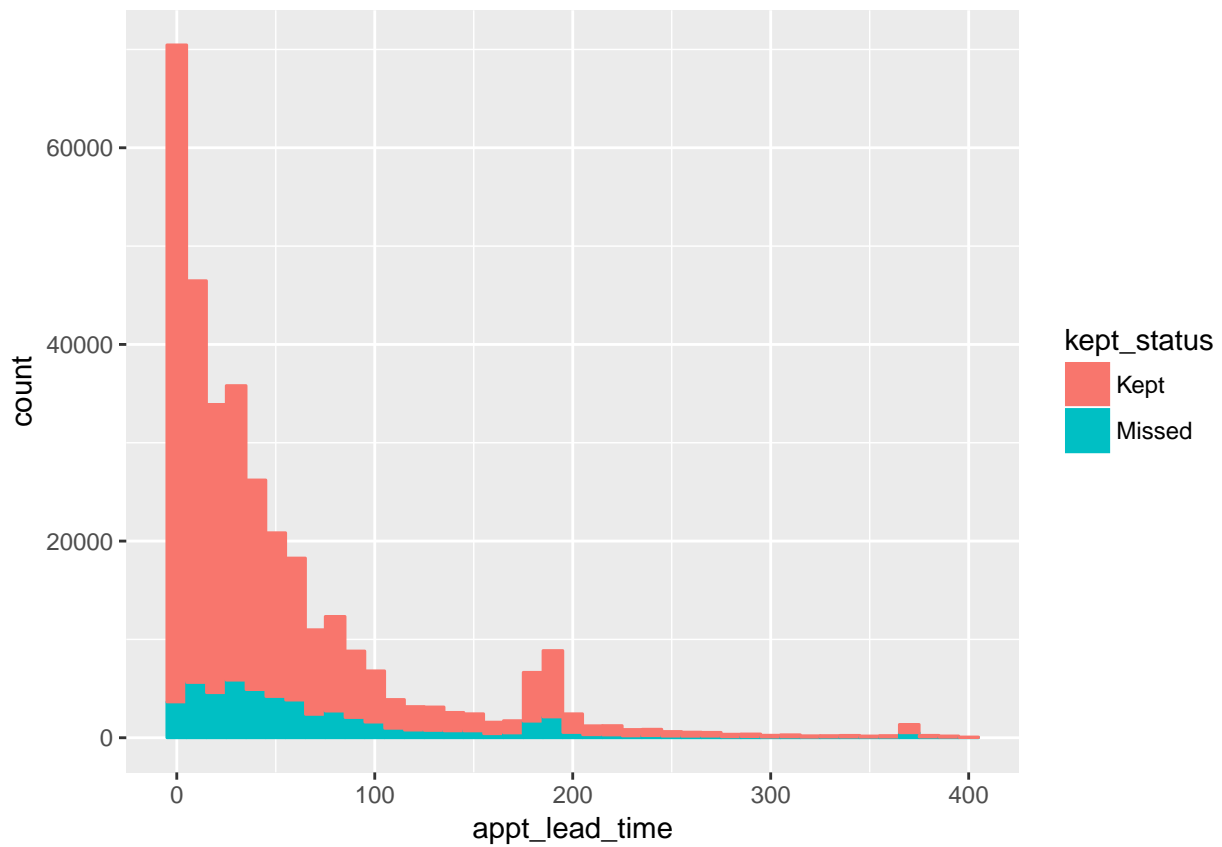


New patients have a very high percentage of kept appointments.

appt_lead_time

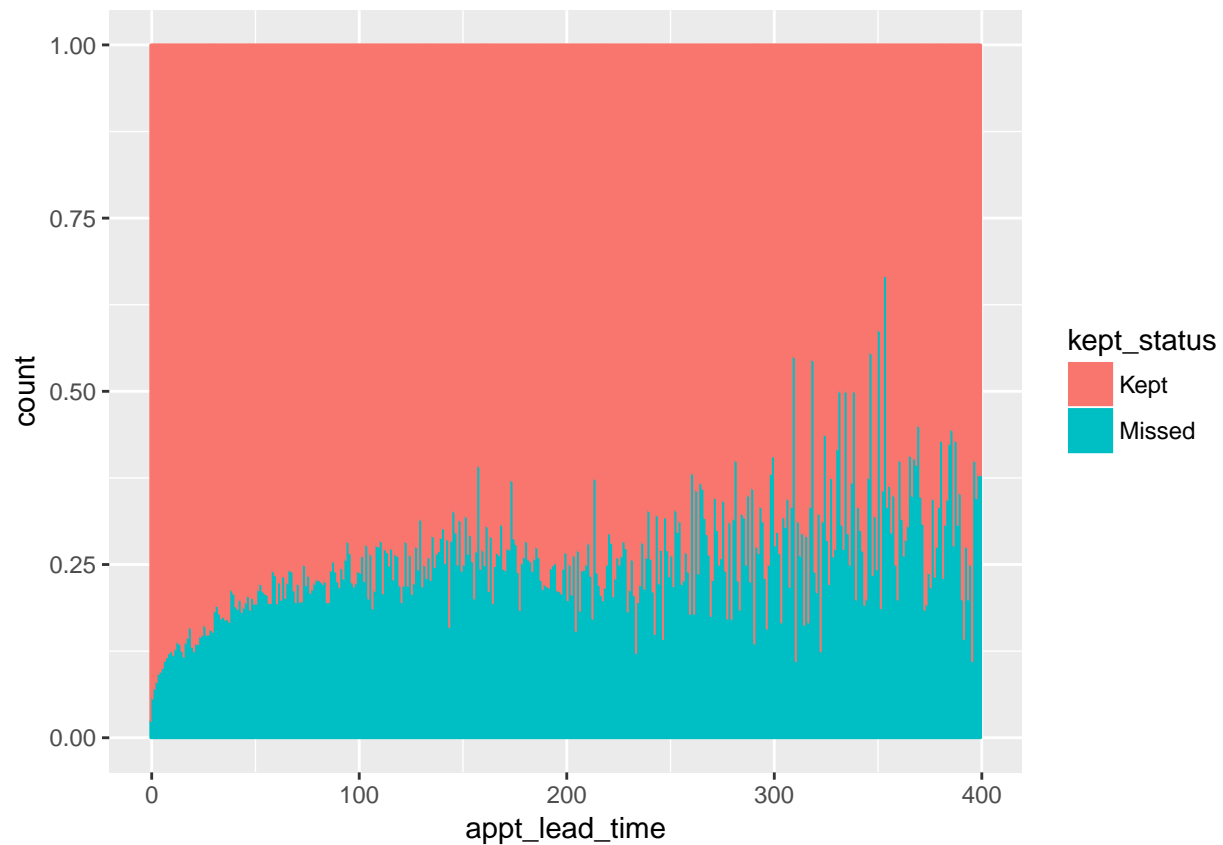
```
appointments %>%  
  filter(appt_lead_time >= 0 & appt_lead_time < 400) %>%  
  ggplot(  
    aes(x = appt_lead_time, color = kept_status, fill = kept_status)  
  ) +  
    geom_histogram(binwidth = 10)
```

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



```
appointments %>%  
  filter(appt_lead_time >= 0 & appt_lead_time < 400) %>%  
  ggplot(  
    aes(x = appt_lead_time, color = kept_status, fill = kept_status)  
  ) +  
    geom_bar(position = "fill")
```

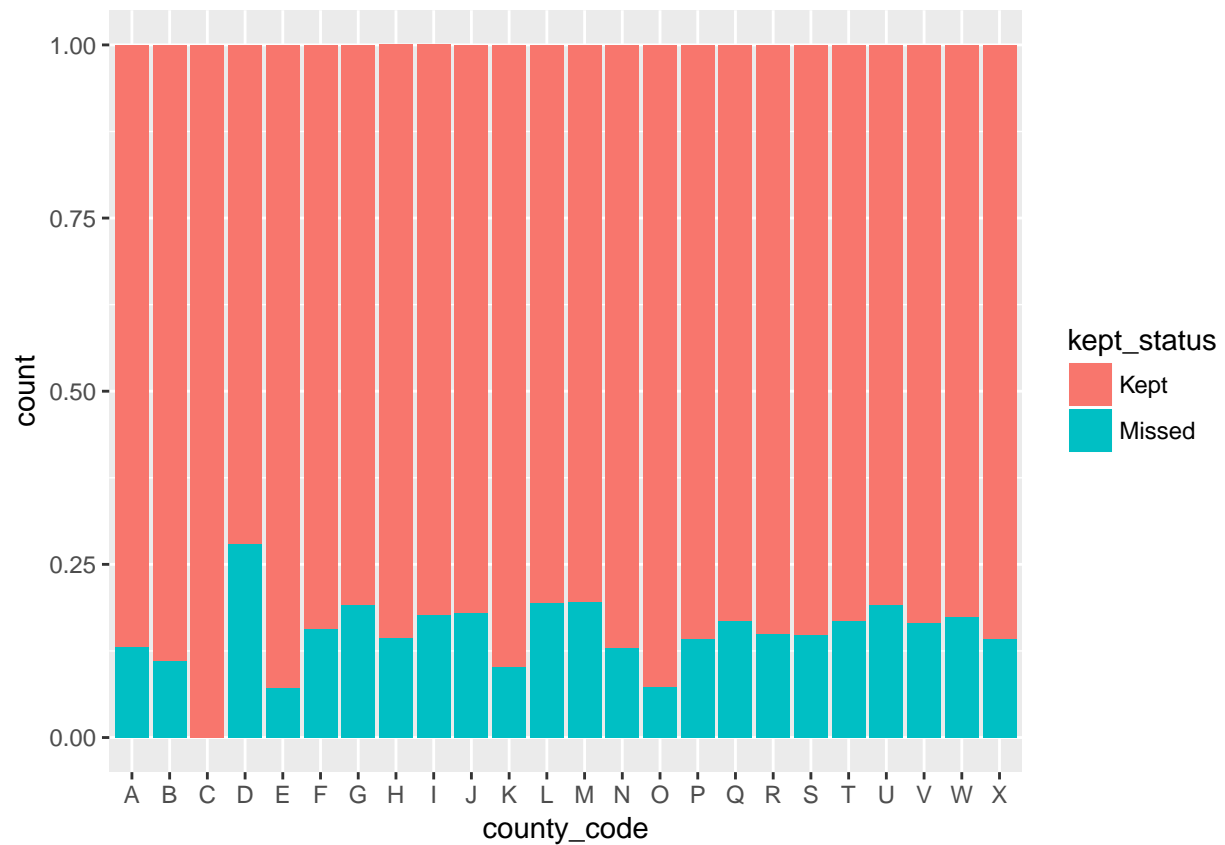
Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.



There is a lower proportion of missed appointments among those with the shortest lead times. This backs up my theory that shorter lead times could indicate more urgent health matters. Also, there are small bumps in the histogram around 180 and 360 days, which is probably indicative of regular 6-month and 12-month checkups.

county_code

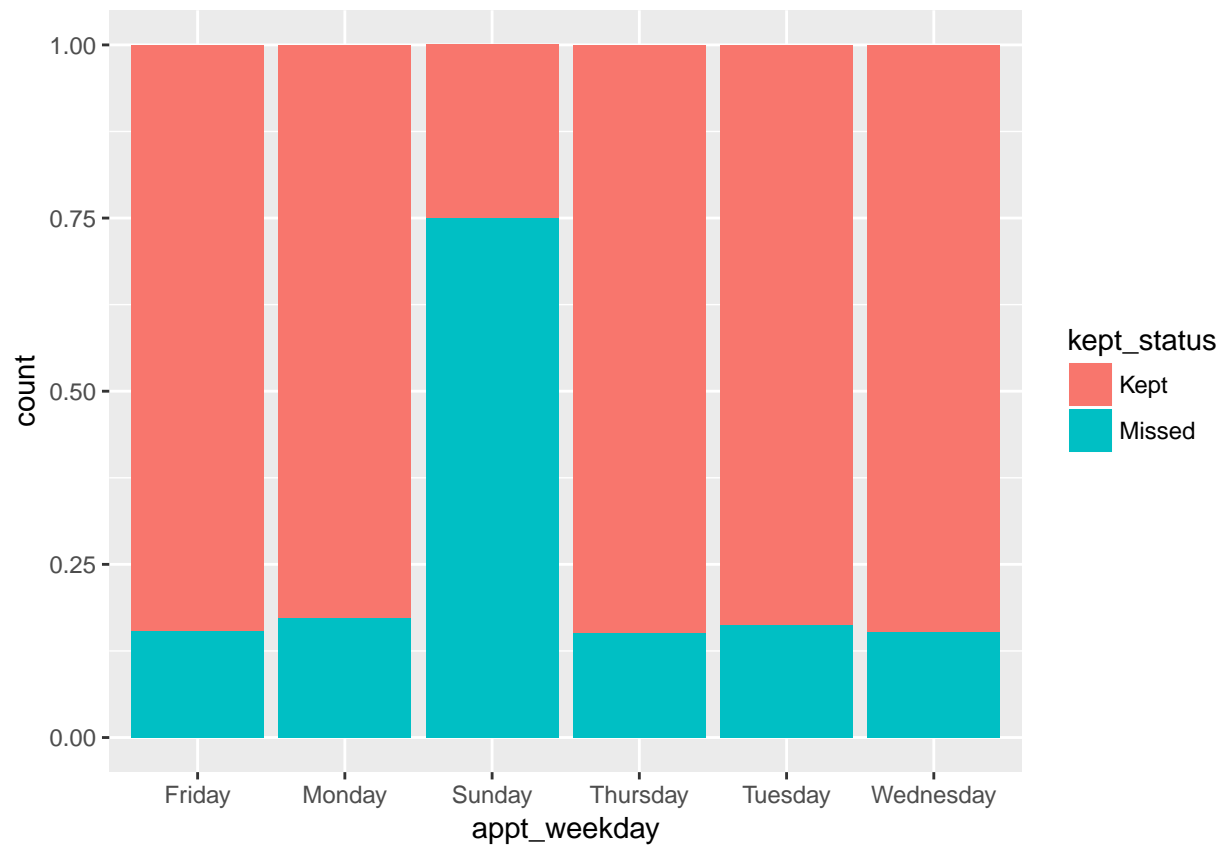
```
ggplot(  
  appointments,  
  aes(x = county_code, fill = kept_status)  
) +  
  geom_bar(position = "fill")
```



```
#appointments_sample_025 <- appointments_3 %>%
#   sample_frac(size = 0.025, replace = FALSE)
#ggpairs(data = appointments_sample_05[,20:24], cardinality_threshold = 50)
```

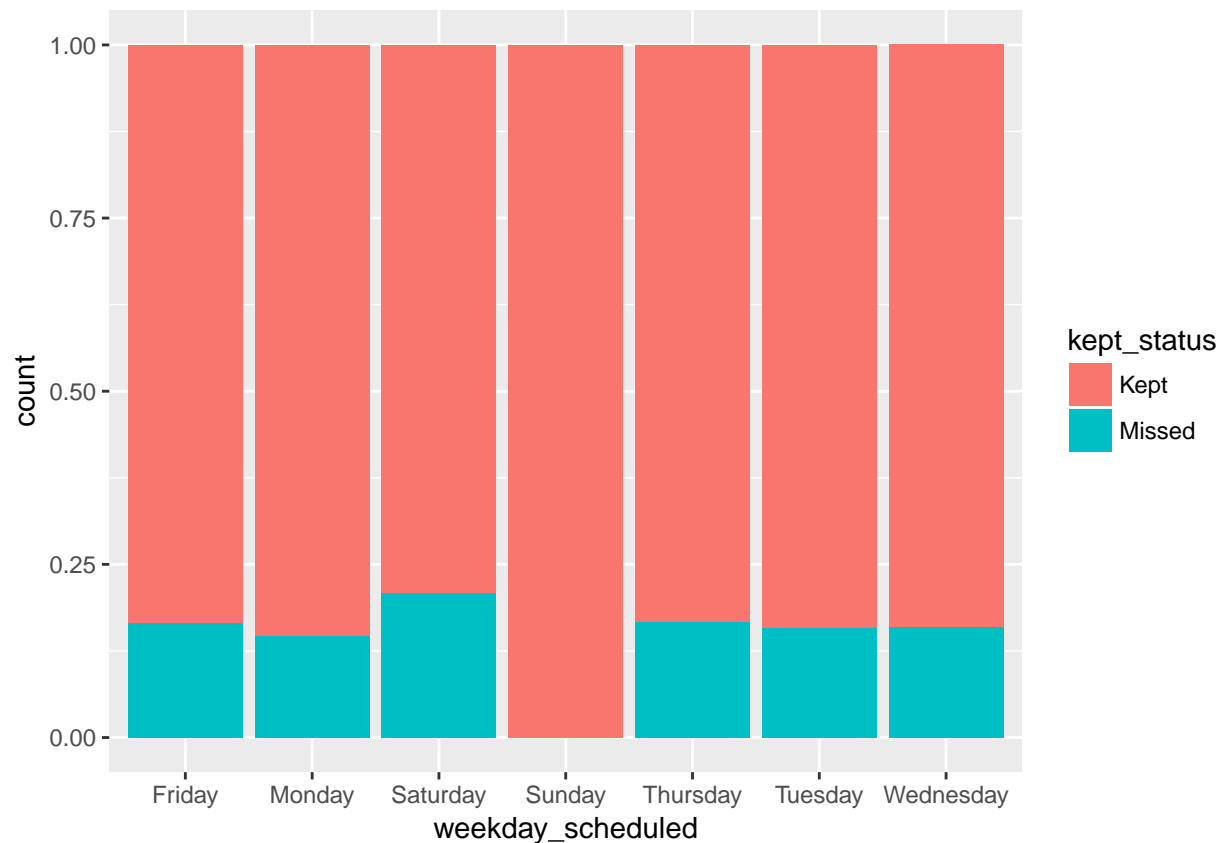
appt__weekday

```
ggplot(
  appointments,
  aes(x = appt_weekday, fill = kept_status)
) +
  geom_bar(position = "fill")
```



weekday_scheduled

```
ggplot(  
  appointments,  
  aes(x = weekday_scheduled, fill = kept_status)  
) +  
  geom_bar(position = "fill")
```



Modeling

Create Modeling Data

```
model_data <- appointments

factor_columns <- c(
  "kept_status", "patient_gender", "billing_type", "office_zip",
  "provider_specialty", "remind_call_result", "hour", "is_new_patient",
  "appt_weekday", "weekday_scheduled",
  "county_code")
model_data[factor_columns] <- lapply(model_data[factor_columns], factor)

model_data <- model_data %>%
  select(
    kept_status, appt_length, patient_age, patient_gender, billing_type,
    patient_distance, provider_specialty, remind_call_result, hour,
    percent_missed, is_new_patient, appt_lead_time, appt_weekday,
    weekday_scheduled, county_code)

table(model_data$provider_specialty)
```

```
##
##      A      B      C      D      E      F      G
```

```
## 246917 84115 5623 5512 42 525 48
model_data$provider_specialty <- model_data$provider_specialty %>%
  fct_collapse(Other = c("E", "F", "G"))

table(model_data$provider_specialty)

##
##      A      B      C      D Other
## 246917 84115 5623 5512 615
model_data$hour <- model_data$hour %>%
  fct_collapse(Other = c("0", "5", "6", "7", "12", "17", "18", "19", "20", "21"))

model_data <- model_data %>%
  filter(county_code != "D") %>%
  filter(weekday_scheduled != "Sunday")

model_data <- droplevels(model_data)
```

Divide model_data into train, validate, and test sets

The data will be divided into three sets: train, validate, and test. I will use 60% of the data for training, and 20% each for validation and test.

Because the data is sorted by date, I want to use the most recent data for the test data, the next oldest for validation, and the oldest for training. Since I am trying to predict future appointments, testing on the most recent data will result in the best measure of the model's performance.

```
# check out caret::downSample
train <- model_data[1:205660,]
validate <- model_data[205661:274200,]
test <- model_data[274201:nrow(model_data),]
table(train$kept_status)

##
##      Kept Missed
## 174599 31061
train_balance_subset <- train[168735:205660,]
table(train_balance_subset$kept_status)

##
##      Kept Missed
## 31061 5865
train_kept <- train_balance_subset[train_balance_subset$kept_status == "Kept",]
train_missed <- train[train$kept_status == "Missed",]
train_balanced <- rbind(train_kept, train_missed)
table(train_balanced$kept_status)

##
##      Kept Missed
## 31061 31061
```

Logistic Regression Model

```
glm_control <- caret::trainControl(method = "none")
```

```
glm_model <- caret::train(  
  kept_status ~ .,  
  data = train_balanced,  
  method = "glm",  
  trControl = glm_control  
)
```

```
summary(glm_model)
```

```
##  
## Call:  
## NULL  
##  
## Deviance Residuals:  
##      Min       1Q   Median       3Q      Max   
## -4.4014  -0.8828  -0.0011   0.8992   3.4588   
##  
## Coefficients:  
##              Estimate Std. Error z value  
## (Intercept)    -3.045e-01  3.612e-01  -0.843  
## appt_length      2.561e-03  5.887e-04   4.350  
## patient_age     -1.254e-02  5.248e-04 -23.887  
## patient_genderMale    7.431e-02  1.903e-02   3.905  
## patient_genderOther  -2.539e-01  1.095e+00  -0.232  
## patient_genderUnknown  8.512e-01  3.909e-01   2.178  
## billing_typeDMP      3.642e-02  2.444e-02   1.490  
## patient_distance  -5.283e-05  1.381e-04  -0.382  
## provider_specialtyB  -3.140e-01  2.288e-02 -13.727  
## provider_specialtyC  -6.137e-01  9.653e-02  -6.357  
## provider_specialtyD  -3.019e+00  1.230e-01 -24.543  
## provider_specialtyOther  1.074e+01  1.384e+02   0.078  
## `remind_call_resultAnswered - Confirmed` -1.296e+00  3.435e-01  -3.774  
## `remind_call_resultAnswered - No Response` -2.063e-01  3.423e-01  -0.603  
## `remind_call_resultAnswered - Reschedule`  2.299e+00  3.894e-01   5.905  
## remind_call_resultBusy    4.837e-01  6.835e-01   0.708  
## remind_call_resultFailed    3.557e-01  3.433e-01   1.036  
## `remind_call_resultLeft Message` -1.242e+00  3.459e-01  -3.591  
## `remind_call_resultNo Answer` -1.116e+00  3.922e-01  -2.845  
## `remind_call_resultNot Called` -1.473e+00  3.435e-01  -4.289  
## hour8            -7.838e-01  8.916e-02  -8.790  
## hour9            -8.468e-01  8.925e-02  -9.487  
## hour10           -9.497e-01  8.913e-02 -10.655  
## hour11           -8.057e-01  8.978e-02  -8.974  
## hour13           -9.064e-01  8.903e-02 -10.181  
## hour14           -9.300e-01  8.931e-02 -10.413  
## hour15           -9.823e-01  8.923e-02 -11.009  
## hour16           -9.068e-01  8.985e-02 -10.092  
## percent_missed     4.895e-02  6.836e-04  71.604  
## is_new_patient1    -2.064e+00  1.279e-01 -16.137  
## appt_lead_time     3.500e-03  1.152e-04  30.384
```


## appt_weekdayMonday	8.136e-03	3.567e-02	0.228
## appt_weekdaySunday	1.048e+01	6.187e+01	0.169
## appt_weekdayThursday	-1.228e-01	3.515e-02	-3.494
## appt_weekdayTuesday	-1.661e-01	3.403e-02	-4.882
## appt_weekdayWednesday	-2.884e-01	3.452e-02	-8.356
## weekday_scheduledMonday	-2.752e-02	3.338e-02	-0.824
## weekday_scheduledSaturday	1.051e+01	8.783e+01	0.120
## weekday_scheduledThursday	1.124e-01	3.323e-02	3.383
## weekday_scheduledTuesday	9.800e-02	3.266e-02	3.001
## weekday_scheduledWednesday	7.375e-02	3.306e-02	2.231
## county_codeB	1.275e-01	1.011e-01	1.261
## county_codeC	-1.161e+01	1.970e+02	-0.059
## county_codeE	-7.626e-01	6.330e-01	-1.205
## county_codeF	3.974e-02	6.687e-02	0.594
## county_codeG	8.259e-01	8.037e-02	10.276
## county_codeH	4.642e-01	7.481e-02	6.206
## county_codeI	6.871e-01	6.371e-02	10.786
## county_codeJ	3.996e-01	6.615e-02	6.040
## county_codeK	1.899e-01	1.009e-01	1.882
## county_codeL	1.923e-01	7.537e-02	2.551
## county_codeM	6.902e-01	7.958e-02	8.673
## county_codeN	3.830e-01	7.234e-02	5.295
## county_codeO	2.612e-01	1.234e-01	2.117
## county_codeP	4.752e-01	6.164e-02	7.709
## county_codeQ	4.112e-01	8.150e-02	5.045
## county_codeR	4.167e-01	7.476e-02	5.574
## county_codeS	1.287e-01	7.419e-02	1.735
## county_codeT	6.601e-01	8.127e-02	8.123
## county_codeU	7.275e-01	6.702e-02	10.855
## county_codeV	6.261e-01	8.505e-02	7.361
## county_codeW	5.045e-01	7.645e-02	6.599
## county_codeX	5.129e-01	1.050e-01	4.883
##	Pr(> z)		
## (Intercept)	0.399138		
## appt_length	1.36e-05	***	
## patient_age	< 2e-16	***	
## patient_genderMale	9.41e-05	***	
## patient_genderOther	0.816610		
## patient_genderUnknown	0.029437	*	
## billing_typeDMP	0.136117		
## patient_distance	0.702155		
## provider_specialtyB	< 2e-16	***	
## provider_specialtyC	2.06e-10	***	
## provider_specialtyD	< 2e-16	***	
## provider_specialtyOther	0.938111		
## `remind_call_resultAnswered - Confirmed`	0.000161	***	
## `remind_call_resultAnswered - No Response`	0.546840		
## `remind_call_resultAnswered - Reschedule`	3.53e-09	***	
## remind_call_resultBusy	0.479151		
## remind_call_resultFailed	0.300198		
## `remind_call_resultLeft Message`	0.000329	***	
## `remind_call_resultNo Answer`	0.004447	**	
## `remind_call_resultNot Called`	1.79e-05	***	
## hour8	< 2e-16	***	

```

## hour9 < 2e-16 ***
## hour10 < 2e-16 ***
## hour11 < 2e-16 ***
## hour13 < 2e-16 ***
## hour14 < 2e-16 ***
## hour15 < 2e-16 ***
## hour16 < 2e-16 ***
## percent_missed < 2e-16 ***
## is_new_patient1 < 2e-16 ***
## appt_lead_time < 2e-16 ***
## appt_weekdayMonday 0.819562
## appt_weekdaySunday 0.865442
## appt_weekdayThursday 0.000475 ***
## appt_weekdayTuesday 1.05e-06 ***
## appt_weekdayWednesday < 2e-16 ***
## weekday_scheduledMonday 0.409704
## weekday_scheduledSaturday 0.904714
## weekday_scheduledThursday 0.000716 ***
## weekday_scheduledTuesday 0.002690 **
## weekday_scheduledWednesday 0.025710 *
## county_codeB 0.207416
## county_codeC 0.952989
## county_codeE 0.228277
## county_codeF 0.552365
## county_codeG < 2e-16 ***
## county_codeH 5.45e-10 ***
## county_codeI < 2e-16 ***
## county_codeJ 1.54e-09 ***
## county_codeK 0.059777 .
## county_codeL 0.010733 *
## county_codeM < 2e-16 ***
## county_codeN 1.19e-07 ***
## county_codeO 0.034296 *
## county_codeP 1.27e-14 ***
## county_codeQ 4.54e-07 ***
## county_codeR 2.49e-08 ***
## county_codeS 0.082745 .
## county_codeT 4.56e-16 ***
## county_codeU < 2e-16 ***
## county_codeV 1.82e-13 ***
## county_codeW 4.13e-11 ***
## county_codeX 1.05e-06 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 86119 on 62121 degrees of freedom
## Residual deviance: 66915 on 62059 degrees of freedom
## AIC: 67041
##
## Number of Fisher Scoring iterations: 10

```

Linear Combos

```

dummy_vars <- caret::dummyVars(~ ., data = train_balanced)

train_balanced_dummy <- data.frame(predict(dummy_vars, newdata = train_balanced))

linear_combos <- caret::findLinearCombos(train_balanced_dummy)
linear_combos$remove

## [1] 8 10 16 25 34 37 44 50 73

lin_combo_index <- linear_combos$remove
colnames(train_balanced_dummy[,lin_combo_index])

```

```

## [1] "patient_gender.Unknown"      "billing_type.DMAP"
## [3] "provider_specialty.Other"    "remind_call_result.Not.Called"
## [5] "hour.16"                    "is_new_patient.1"
## [7] "appt_weekday.Wednesday"     "weekday_scheduled.Wednesday"
## [9] "county_code.X"

```

Correlation

```

cor_matrix <- cor(train_balanced_dummy)
high_cor <- as.data.frame(which(abs(cor_matrix) > 0.90, arr.ind = TRUE))
index <- high_cor %>% filter(row != col)

cor_matrix <- cor(train_balanced_dummy)
cor_matrix[index[,1], index[, 2]]

```

```

##                kept_status.Kept kept_status.Missed
## kept_status.Missed          -1.00000000          1.00000000
## kept_status.Kept             1.00000000          -1.00000000
## patient_gender.Male          -0.01749639          0.01749639
## patient_gender.Female         0.01801086          -0.01801086
## billing_type.DMAP             -0.05892705          0.05892705
## billing_type.Commercial        0.05892705          -0.05892705
## provider_specialty.B           0.01996238          -0.01996238
## provider_specialty.A          -0.05828428          0.05828428
## is_new_patient.1              0.18544792          -0.18544792
## is_new_patient.0             -0.18544792          0.18544792
##                patient_gender.Female patient_gender.Male
## kept_status.Missed          -0.01801086          0.01749639
## kept_status.Kept             0.01801086          -0.01749639
## patient_gender.Male          -0.99804586          1.00000000
## patient_gender.Female         1.00000000          -0.99804586
## billing_type.DMAP             0.05361193          -0.05302159
## billing_type.Commercial       -0.05361193          0.05302159
## provider_specialty.B          -0.01088265          0.01106922
## provider_specialty.A           0.01111819          -0.01126683
## is_new_patient.1             -0.01225583          0.01127324
## is_new_patient.0              0.01225583          -0.01127324
##                billing_type.Commercial billing_type.DMAP
## kept_status.Missed          -0.058927052          0.058927052
## kept_status.Kept             0.058927052          -0.058927052
## patient_gender.Male          0.053021586          -0.053021586
## patient_gender.Female       -0.053611925          0.053611925

```

```
## billing_type.DMAP          -1.000000000    1.000000000
## billing_type.Commercial    1.000000000   -1.000000000
## provider_specialty.B       0.014846253   -0.014846253
## provider_specialty.A       0.002134878   -0.002134878
## is_new_patient.1           0.050625263   -0.050625263
## is_new_patient.0          -0.050625263    0.050625263
##
##           provider_specialty.A provider_specialty.B
## kept_status.Missed         0.058284279   -0.01996238
## kept_status.Kept          -0.058284279    0.01996238
## patient_gender.Male       -0.011266829    0.01106922
## patient_gender.Female     0.011118195   -0.01088265
## billing_type.DMAP         -0.002134878   -0.01484625
## billing_type.Commercial    0.002134878    0.01484625
## provider_specialty.B      -0.933341273    1.000000000
## provider_specialty.A       1.000000000   -0.93334127
## is_new_patient.1           0.016077635   -0.02827018
## is_new_patient.0          -0.016077635    0.02827018
##
##           is_new_patient.0 is_new_patient.1
## kept_status.Missed         0.18544792   -0.18544792
## kept_status.Kept          -0.18544792    0.18544792
## patient_gender.Male       -0.01127324    0.01127324
## patient_gender.Female     0.01225583   -0.01225583
## billing_type.DMAP         0.05062526   -0.05062526
## billing_type.Commercial   -0.05062526    0.05062526
## provider_specialty.B       0.02827018   -0.02827018
## provider_specialty.A      -0.01607764    0.01607764
## is_new_patient.1          -1.00000000    1.00000000
## is_new_patient.0           1.00000000   -1.00000000
```

```
colnames(train_balanced_dummy[index[,1], index[, 2]])
```

```
## [1] "kept_status.Kept"      "kept_status.Missed"
## [3] "patient_gender.Female" "patient_gender.Male"
## [5] "billing_type.Commercial" "billing_type.DMAP"
## [7] "provider_specialty.A"   "provider_specialty.B"
## [9] "is_new_patient.0"      "is_new_patient.1"
```

Random Forest Model

Using caret Package

```
rf_control <- caret::trainControl(method = "cv", number = 2, classProbs = TRUE)
seed <- 7
metric <- "Accuracy"
set.seed(seed)
mtry <- 3
tuneGrid <- expand.grid(.mtry = mtry)

rf_model <- caret::train(
  kept_status ~ ., data = train_balanced, method = "rf", metric = metric,
  tuneGrid = tuneGrid, trControl = rf_control)
```

Model Comparison

```
pred_glm <- predict(glm_model, validate)

conf_mat_glm <- caret::confusionMatrix(
  pred_glm, validate$kept_status, positive = "Missed")

conf_mat_glm

## Confusion Matrix and Statistics
##
##           Reference
## Prediction Kept Missed
##      Kept   41757   3131
##      Missed 15327   8325
##
##              Accuracy : 0.7307
##              95% CI : (0.7274, 0.734)
##      No Information Rate : 0.8329
##      P-Value [Acc > NIR] : 1
##
##              Kappa : 0.3214
##  Mcnemar's Test P-Value : <2e-16
##
##              Sensitivity : 0.7267
##              Specificity : 0.7315
##              Pos Pred Value : 0.3520
##              Neg Pred Value : 0.9302
##              Prevalence : 0.1671
##              Detection Rate : 0.1215
##      Detection Prevalence : 0.3451
##              Balanced Accuracy : 0.7291
##
##              'Positive' Class : Missed
##
conf_mat_glm$byClass["F1"]

##           F1
## 0.4742509

pred_rf <- predict(rf_model, validate)

caret::confusionMatrix(rf_model)

## Cross-Validated (2 fold) Confusion Matrix
##
## (entries are percentual average cell counts across resamples)
##
##           Reference
## Prediction Kept Missed
##      Kept   33.6   7.9
##      Missed 16.4  42.1
##
##      Accuracy (average) : 0.7575
```

```
conf_mat_rf <- caret::confusionMatrix(
  pred_rf, validate$kept_status, positive = "Missed")
```

```
conf_mat_rf
```

```
## Confusion Matrix and Statistics
```

```
##
```

```
##           Reference
```

```
## Prediction  Kept Missed
```

```
##      Kept    39039    2216
```

```
##      Missed 18045    9240
```

```
##
```

```
##              Accuracy : 0.7044
```

```
##              95% CI : (0.701, 0.7078)
```

```
##      No Information Rate : 0.8329
```

```
##      P-Value [Acc > NIR] : 1
```

```
##
```

```
##              Kappa : 0.316
```

```
##      McNemar's Test P-Value : <2e-16
```

```
##
```

```
##              Sensitivity : 0.8066
```

```
##              Specificity : 0.6839
```

```
##      Pos Pred Value : 0.3386
```

```
##      Neg Pred Value : 0.9463
```

```
##              Prevalence : 0.1671
```

```
##      Detection Rate : 0.1348
```

```
##      Detection Prevalence : 0.3981
```

```
##      Balanced Accuracy : 0.7452
```

```
##
```

```
##      'Positive' Class : Missed
```

```
##
```

```
conf_mat_rf$byClass["F1"]
```

```
##           F1
```

```
## 0.477014
```