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 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 aslo be times more apppointments 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 goal of this project is to minimize the prediction error of missed appointments, as the error results in times where there are too many or too few patients at a given time. This is important to the client because it reduces the revenue loss caused by missed appontments in a way that reduces the undesired consequences of overbooking.

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 of future appointments. Missed appointments can only be predicted based on indirect factors that are known, such as patient 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 as it allows more overbooking to be done with fewer negative consequences.

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 appointentment predictions into a booking system are client-specific and not included in the scope of this project, which is limited to minimizing the error while predicting the classification and associated 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)

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

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

There is no field that can be used to identify a specific patient 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

```r
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
##
##
##
##
```

```r
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 288
##  $ 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 Resp
```

```r
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
```

```r
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
```

```r
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

```r
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)
variable_type <- unlist(map(appointments, class))
variable_type <- variable_type[-4]
as_data_frame(cbind(c(1:length(variable)), variable, variable_type, variable_descriptions))
```

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

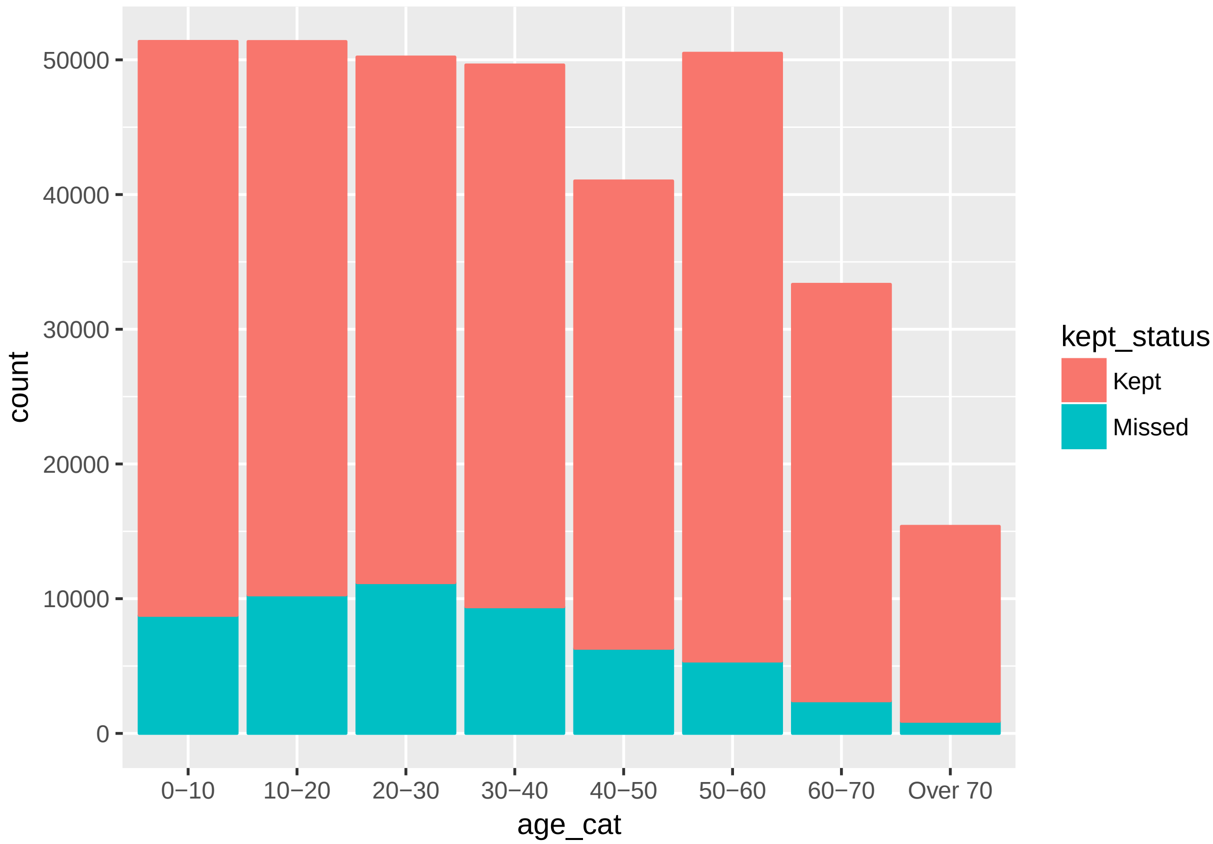
One variable, `patient_distance` has 974 missing value. This is fairly minor considering the size of the data set and will be evaluated later on when exploring the variable further.

patient_age

I expected missed appointments to vary across age ranges. Perhaps older patients have fewer commitments with children 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 patients, which follows a similar pattern to what I expcted.

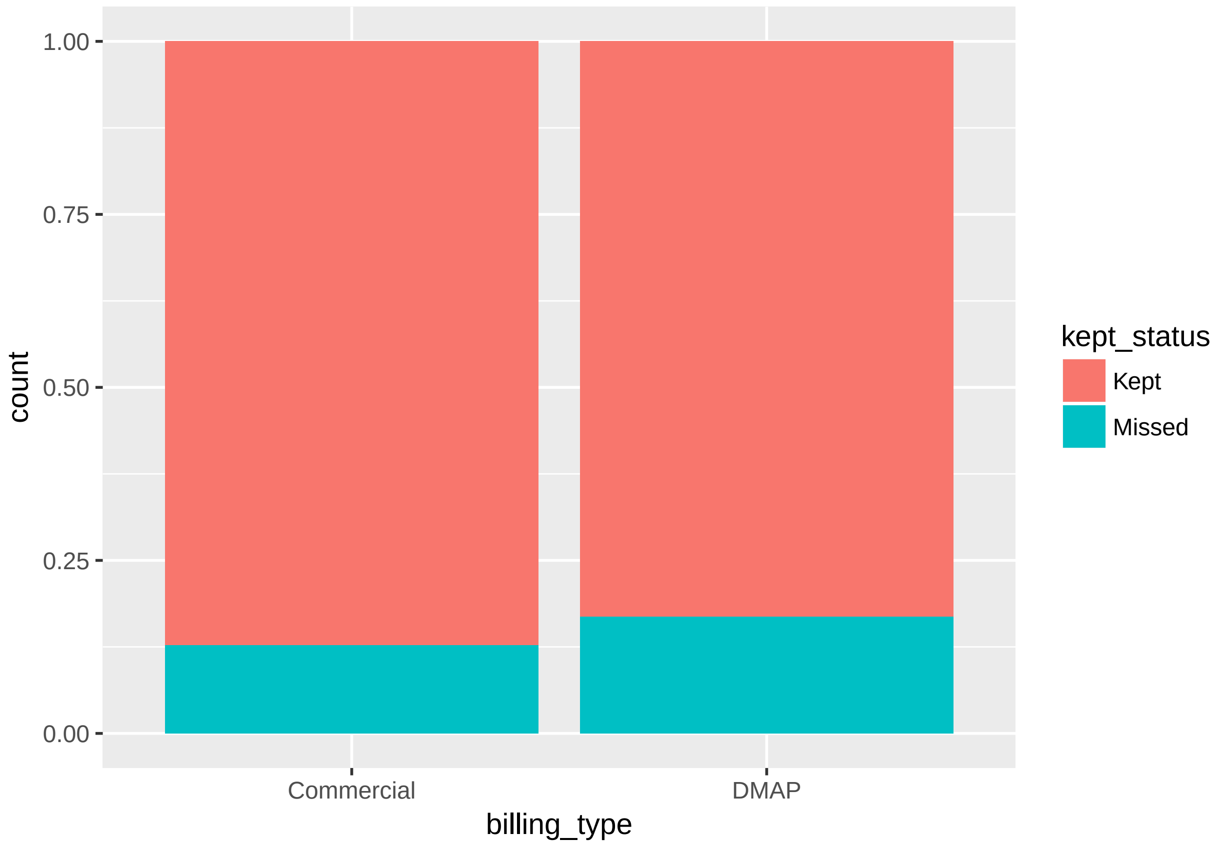
billing_type

```
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 fairly small yet significant 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. There are many ways the time of day can have an effect, such as rush hour traffic in the morning and afternoon causing more missed appointments, whereas mid-day appointments could be more likely to be missed by work factors.

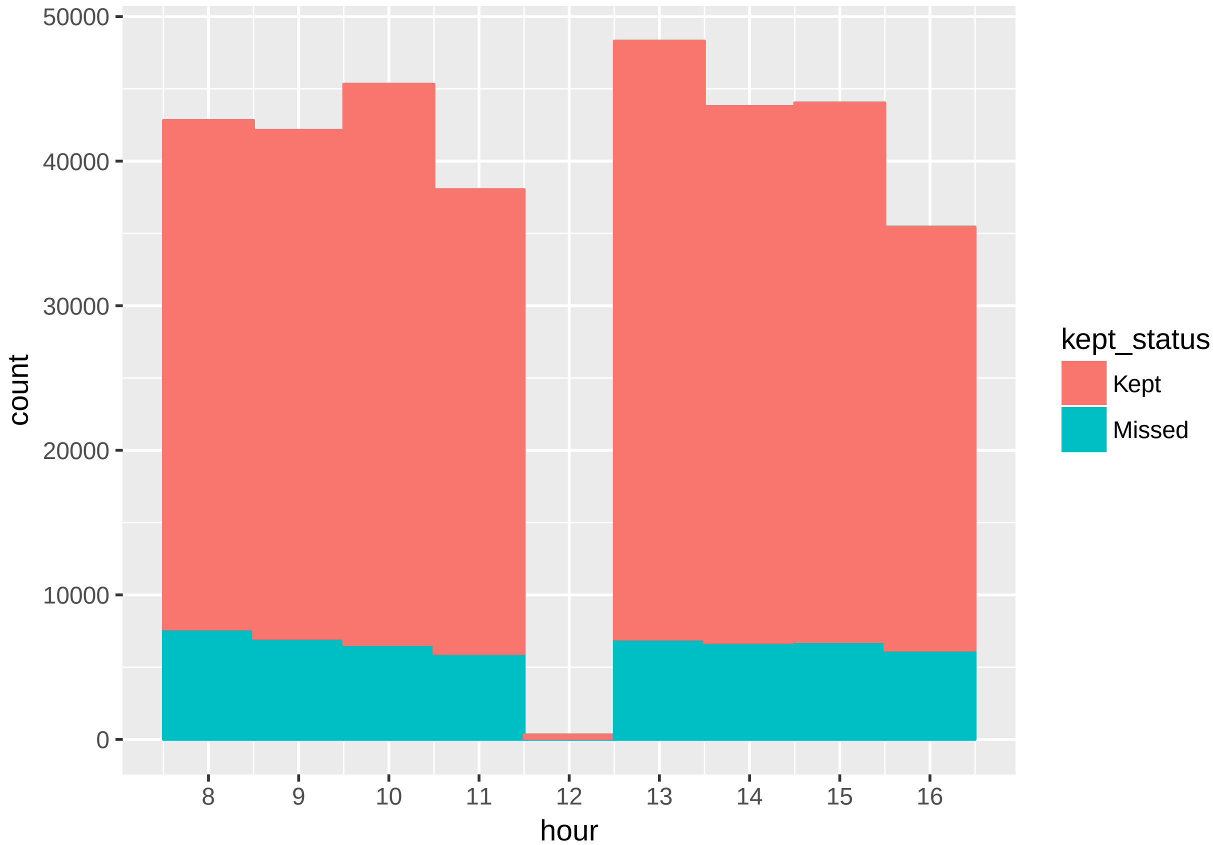
```
appointments <- appointments %>%
    mutate(hour = lubridate::hour(appointments$appt_datetime))

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 a one hour gap starting at 12:00.

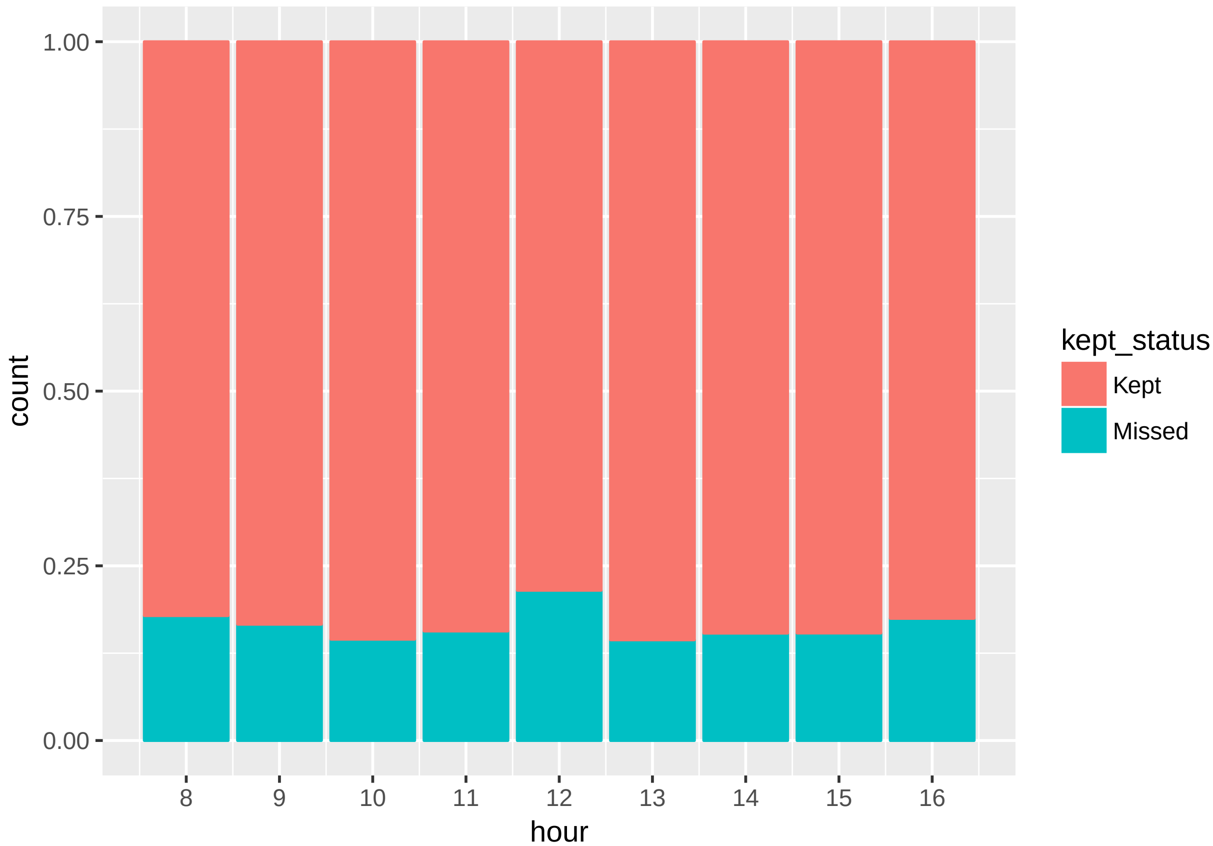
```
appointments_hour <- appointments %>%
    select(kept_status, hour) %>%
    filter(hour >= 8 & hour <= 16)

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. The ratio of missed appointments is hard to read, so I will create a proportional plot to see if it shows any trends.

```
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.

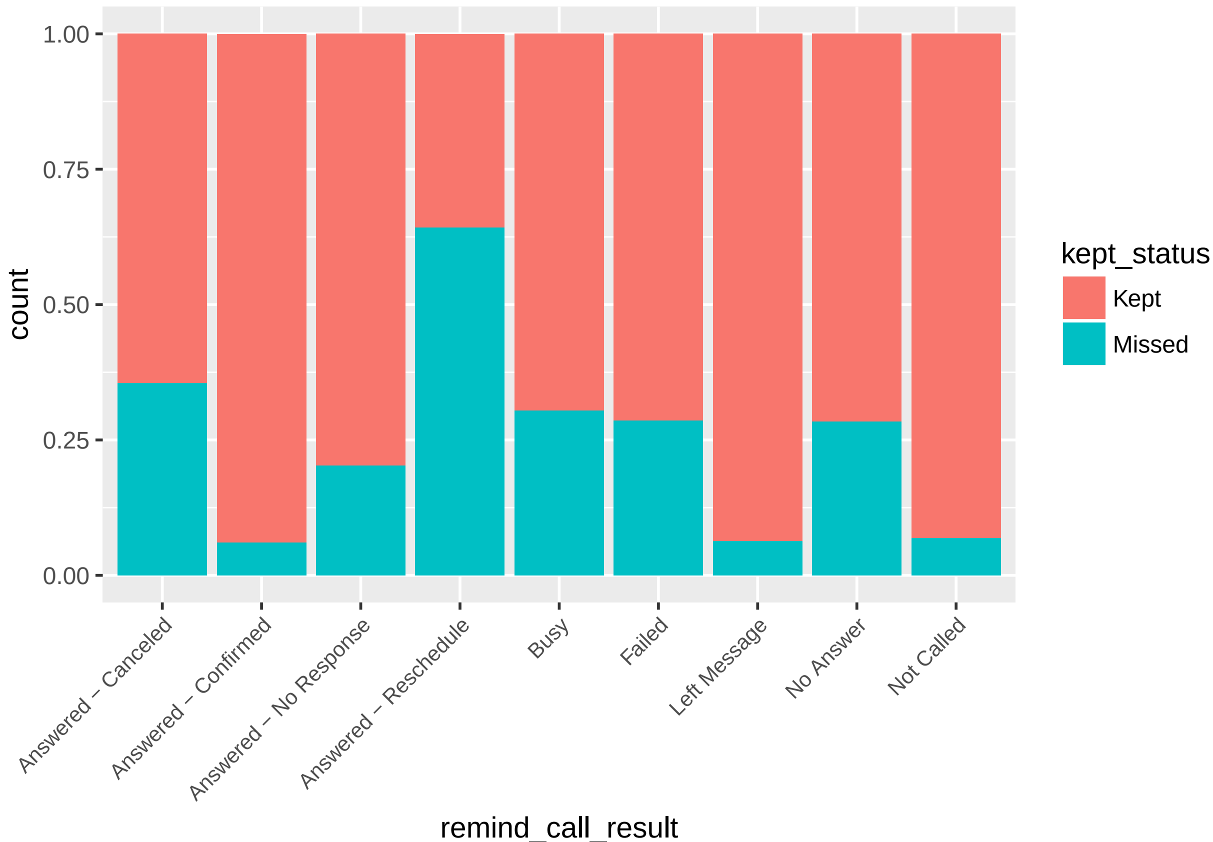
remind_call_result

```r
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
```

There are relatively few instances of "Answered - Cancelled", "Answered - Reschedule", "Busy", and "No Answer"

```r
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))
```



Reminder call responses of "Answered - Confirmed", "Left Message", and "Not Called" have the lowest ratios of missed appointments. This seems logical although I would not expect "Not Called" to have as low of a rate, since it doesn't seem like it is indicating one way or another. The best explanation for the low miss rate is a bias in choosing who doesn't need a reminder call.

Responses of "Answered - No Response", "Busy", "Failed", and "No Answer" have average or higher rates of missed appointments, which makes sense as I would rate these responses as neutral to slightly negative.

Responses "Answered - Cancelled" and "Answered - Reschedule" have the highest rates of about 35% and 65%, respectively. It's not surprising that these have the highest missed rates, but I would expect them to be higher. Perhaps the patients wanted to cancel or reschdule, but circumstances changed and they ultimately decided to come.

provider_specialty

THe variable `provider_specialty` indicates what the medical staff assigned to the appointment specializes in, but is encoded for confidentiality.

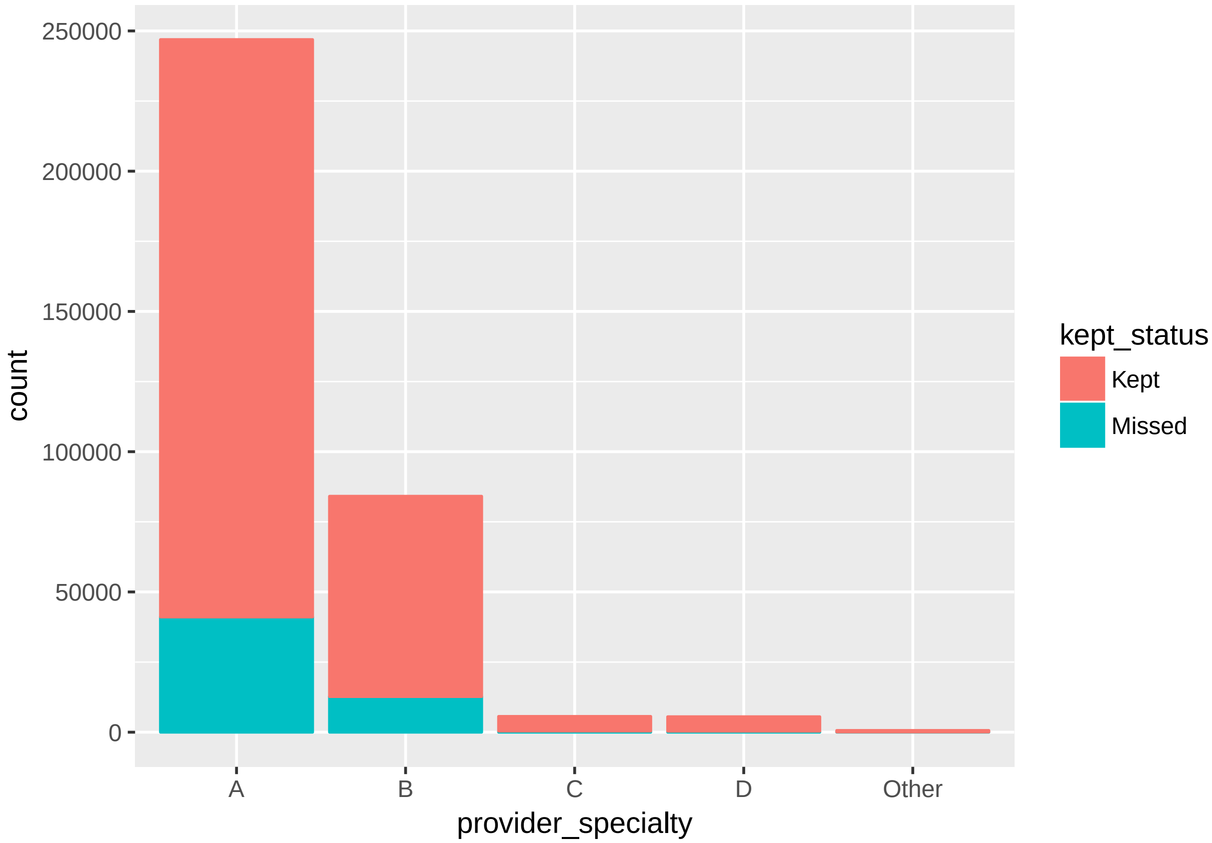
```
table(appointments$provider_specialty)
```

```
##
##      A      B      C      D      E      F      G
## 246917  84115   5623   5512     42    525     48
```

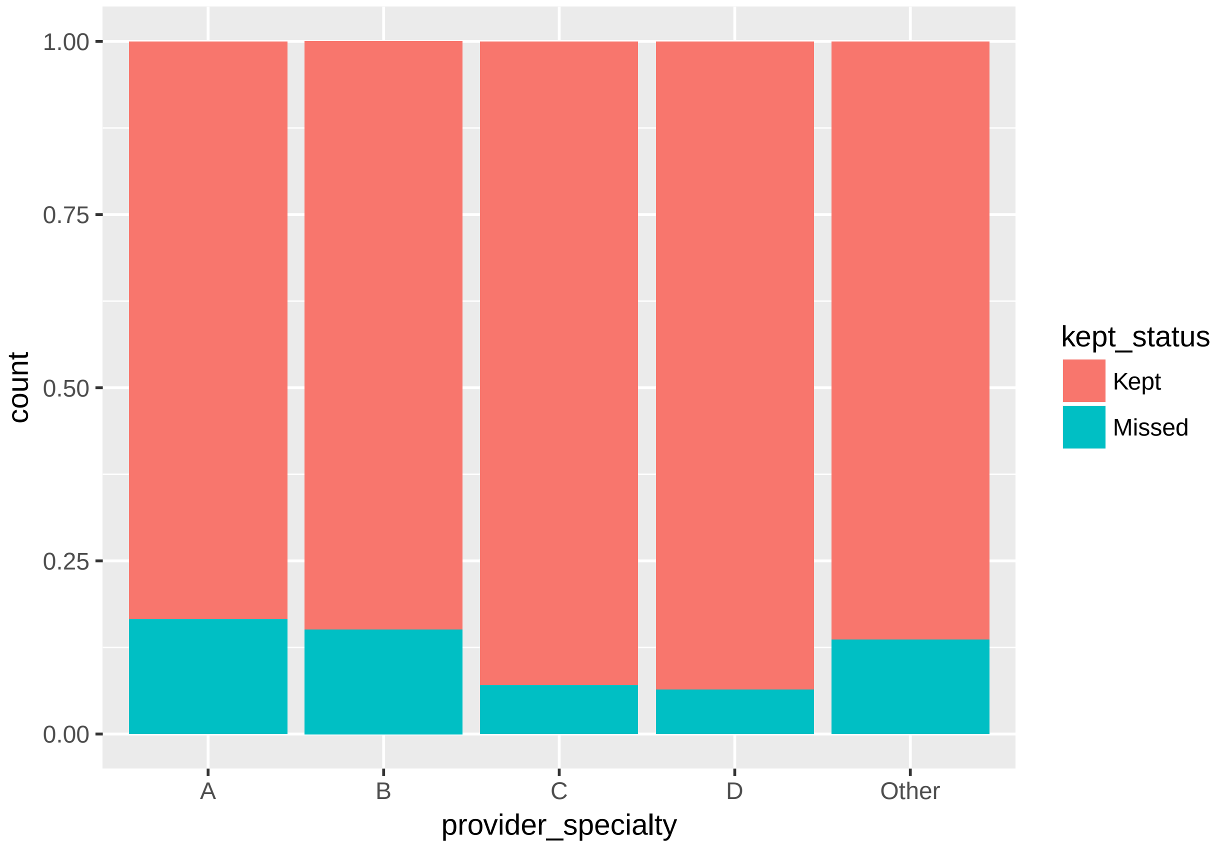
Most observations are specialty A and B. Specialties E, F, and G have very few observations and will be grouped as "Other".

```
appointments$provider_specialty <- appointments$provider_specialty %>%
    fct_collapse(Other = c("E", "F", "G"))
```

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



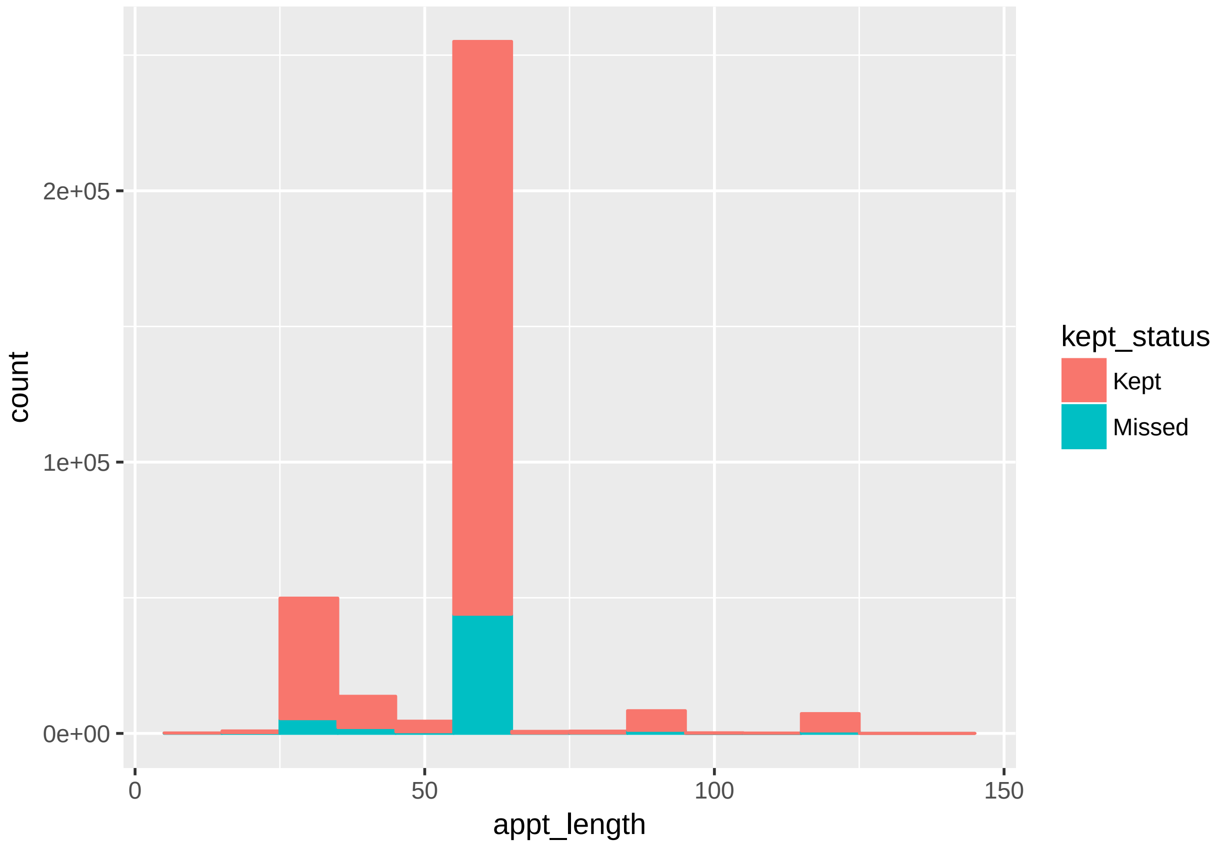

```r
ggplot(
    appointments,
    aes(x = provider_specialty, fill = kept_status)
) +
    geom_bar(position = "fill")
```



The first plot shows that most appointmetns have a provider specialty of "A" or "B". The second plot shows that specialties "C" and "D" have the lowest rates of missed appointments. Without knowing the details of the specialties, my best hypothesis is that specialties "C" and "D" could be for more critical but less common types of appointments, leading to fewer misses.

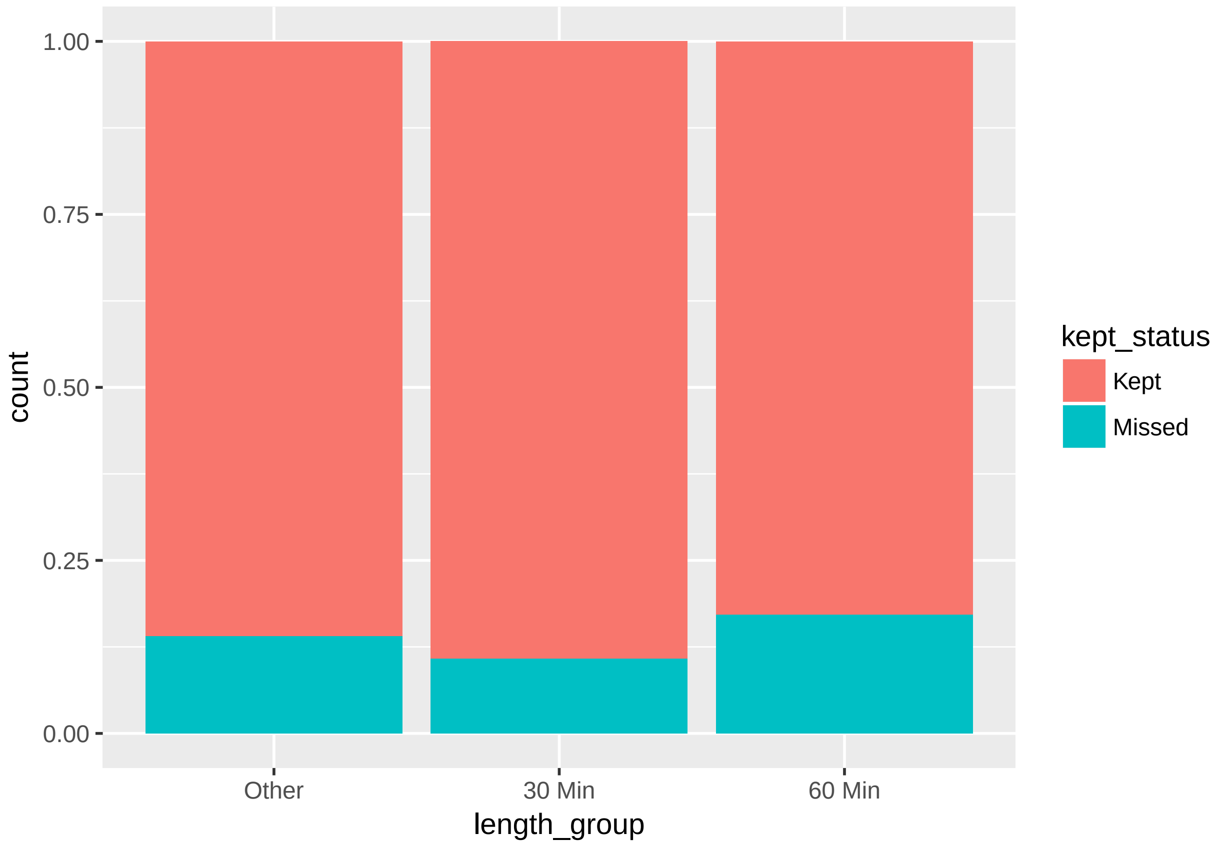
appt_length

```
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 the next most common. I'll group them as "30 Min", "60 Min", and the rest as "Other", and take a look at the differences.

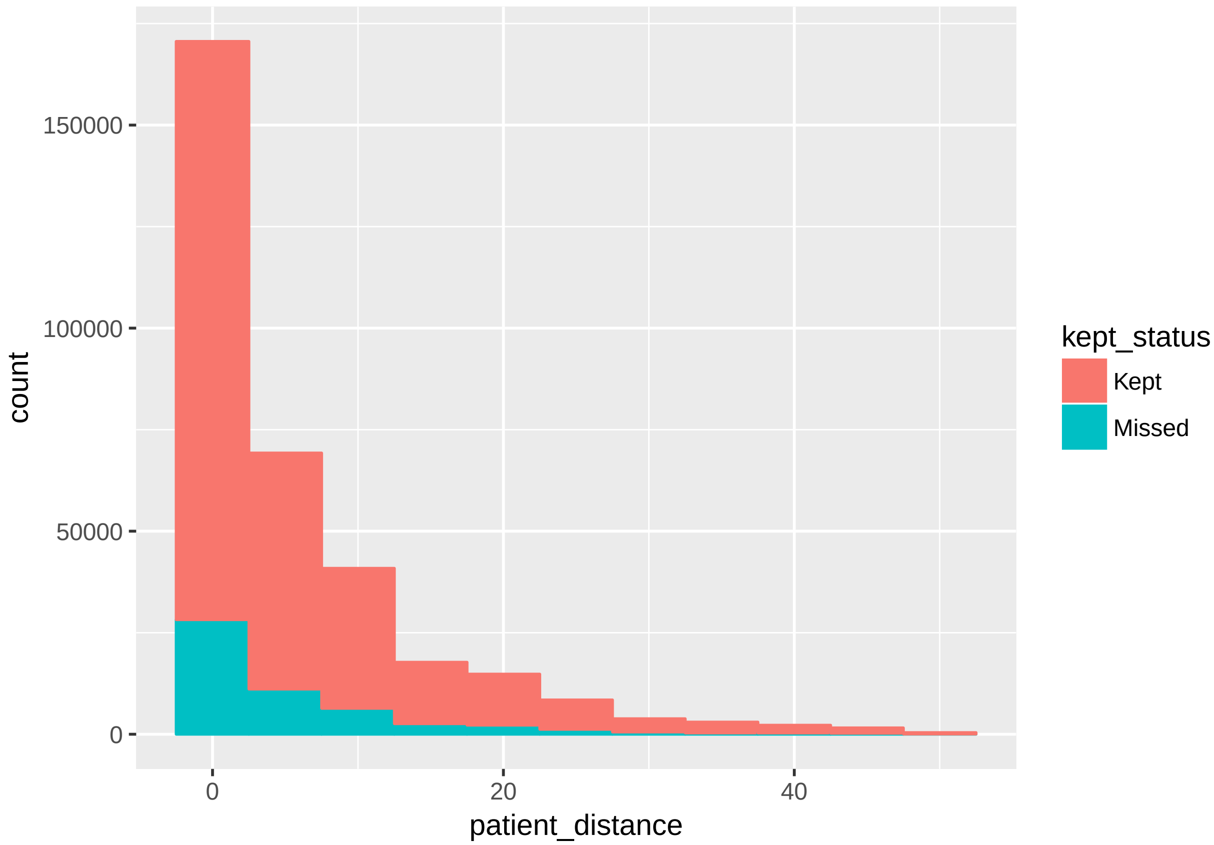
```
length_breaks <- c(-1, 29, 30, 59, 60, 1000)

length_labels <- c("Other1", "30 Min", "Other2", "60 Min", "Other3")

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

appointments$length_group <- appointments$length_group %>%
    fct_collapse(Other = c("Other1", "Other2", "Other3"))

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



Longer appointments are more likely to be missed. This could be because a patient is more inclined to go to a shorter appointment, and shorter appointments are less likely to be impacted by scheduling conflicts.

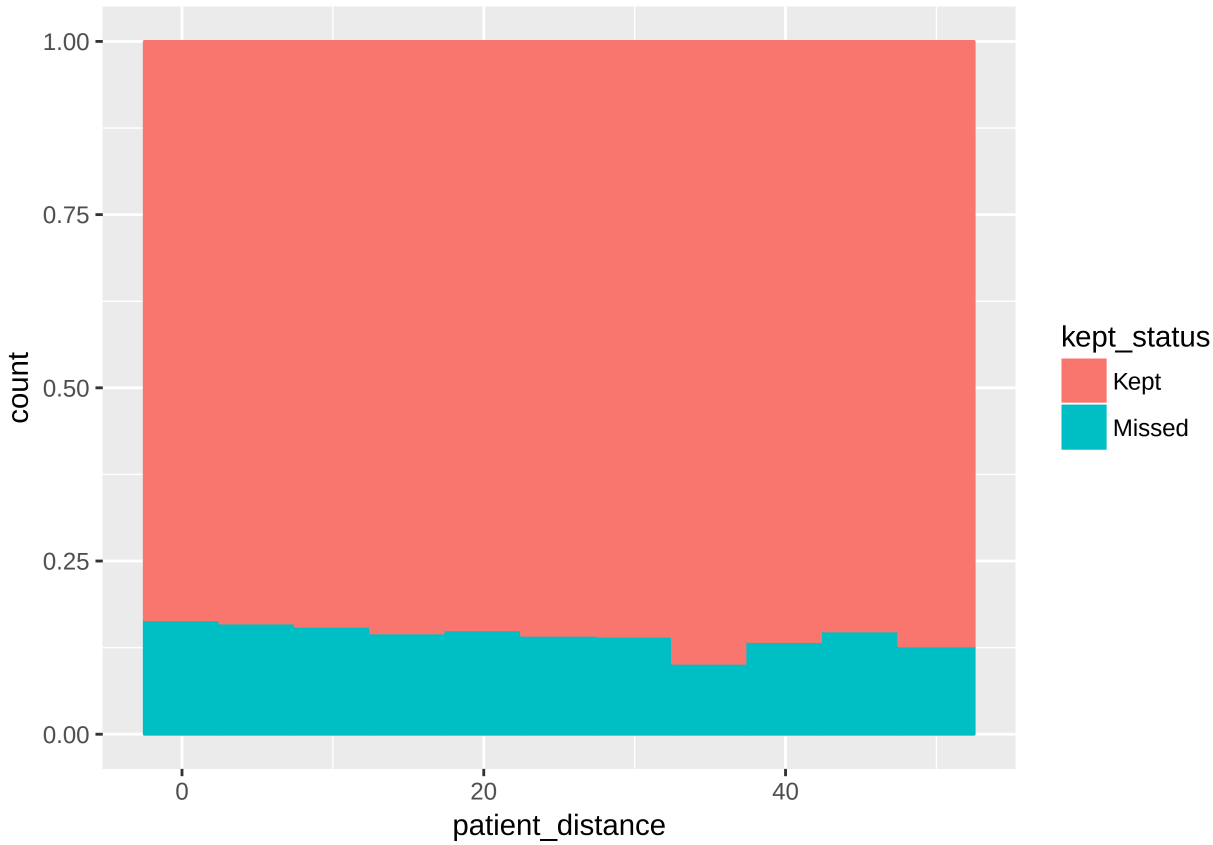
patient_distance

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




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

```
## Warning: `geom_bar()` no longer has a `binwidth` parameter. Please use
## `geom_histogram()` instead.
```



The variable `patient_distance` is the only one with missing values, which will be replaced with median rather than mean since it is a right-skewed distribution.

```
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. Five new variables will be created, summarized below:

The `prior_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 errors will be replaced with 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.

```r
appointments <- appointments %>%
    mutate(
        prior_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$prior_percent_missed <- appointments$prior_percent_missed %>%
    tidyr::replace_na(0)
```

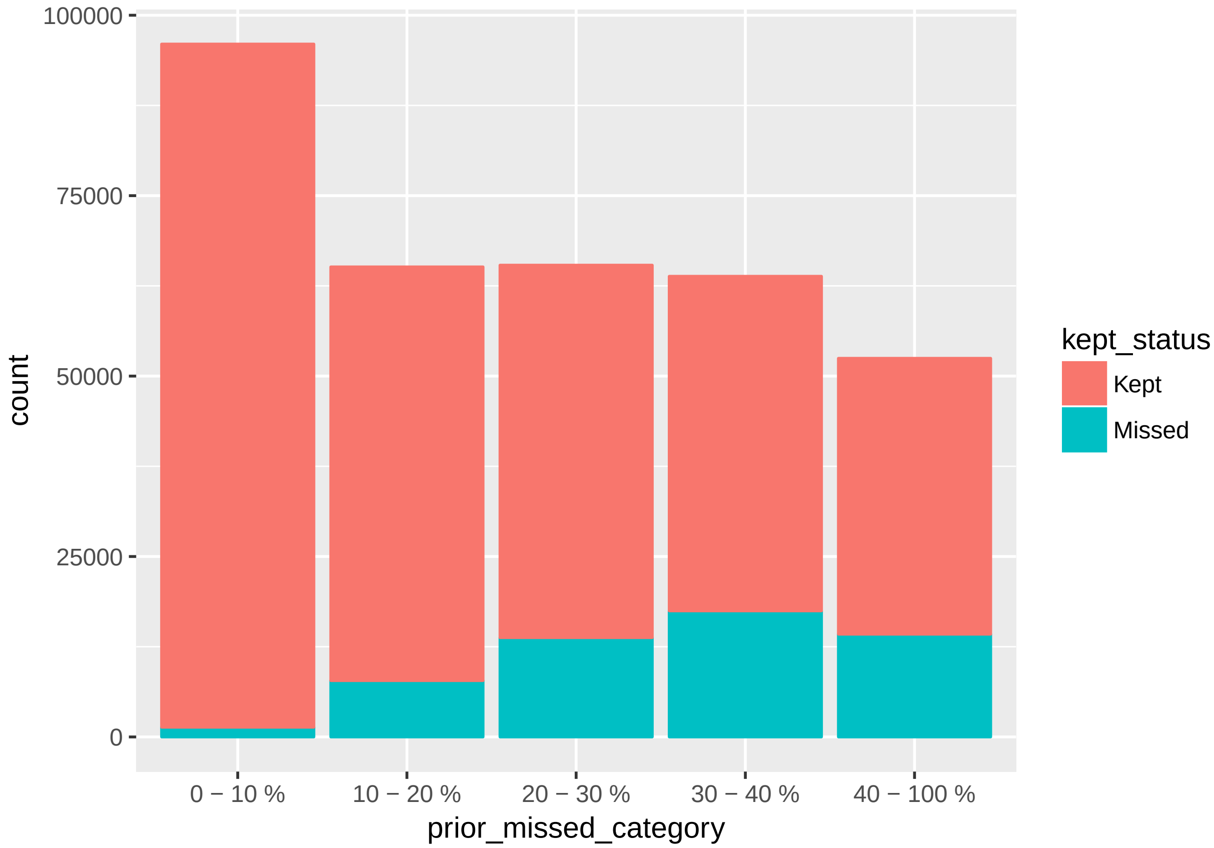
In addition to the claculated variables, the variable `county_code` will be brought in from the zipcode dataset. This will allow differences between geographical areas to be modeled. For example, a more populous county might have more traffic related missed appointments, and a county more prone to inclement weather could have more weather related missed appointments.

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

prior_percent_missed

This is expected to be an important variable, because regardless of the reason a patient missed appointments in the past, those same reasons are probably more likely to occur on future appointments. For example, a patient that has missed appointments in the past because they have a very hectic schedule, will be more likely to miss future appointments due to the same hectic schedule.

```r
prior_missed_labels <- c("0 - 10 %", "10 - 20 %", "20 - 30 %", "30 - 40 %", "40 - 100 %")
prior_missed_breaks <- c(-.01, 0.10, 0.20, 0.30, 0.40, 1.01)

appointments <- appointments %>%
    mutate(
        prior_missed_category = cut(
            prior_percent_missed,
            breaks = prior_missed_breaks,
            labels = prior_missed_labels))

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



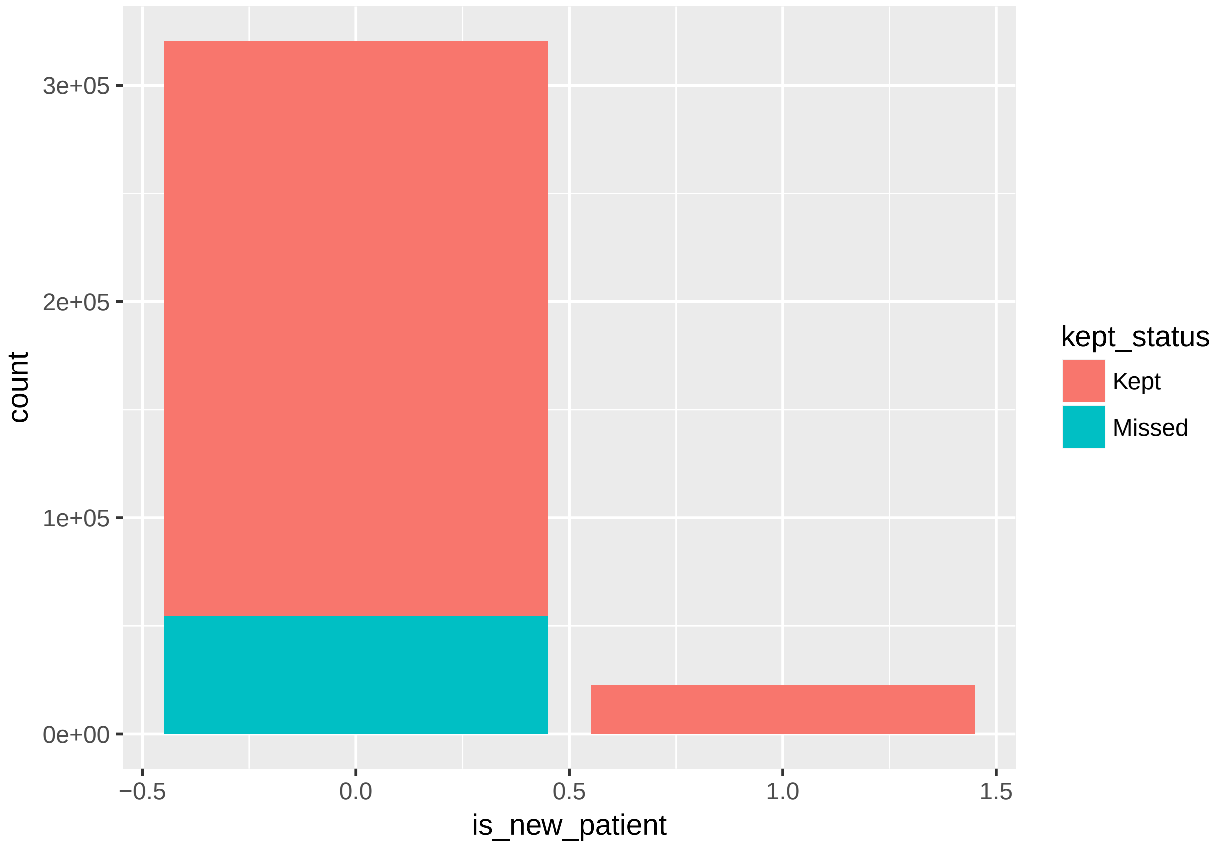
As expected, past ratio of missed appointments is a strong predictor of future missed appointments. Appointments where the prior rate is 0 - 10% overwhelmingly kept their appointments.

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, but make up a small percentage of the total appointments.

appt_lead_time

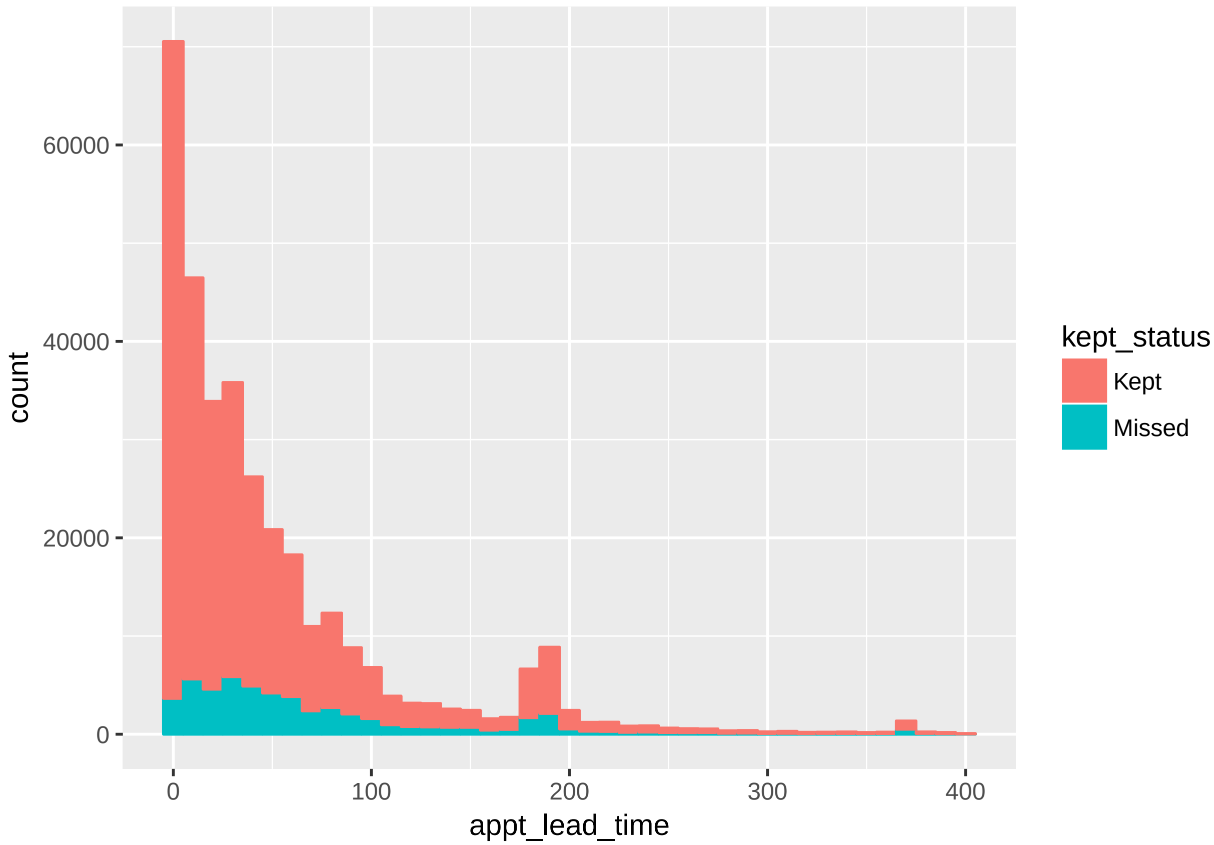
First I will check for negative values. Negative values suggest the appointment occured before it was booked, which wouldn't be possible and must represent an entry error.

```
length(which(appointments$appt_lead_time < 0))
```

```
## [1] 82
```

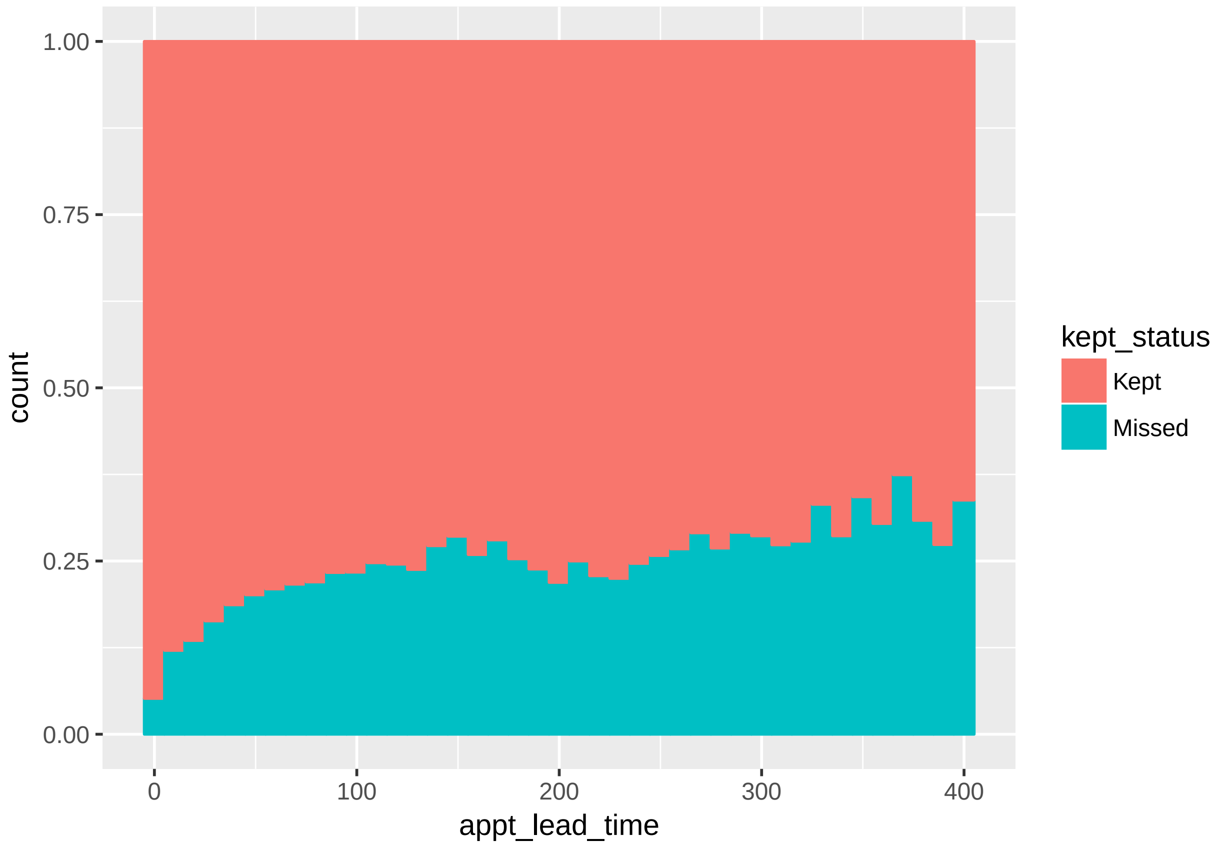
There are 82 negative values, which will be replaced with zero.

```
appointments$appt_lead_time <- ifelse(
    appointments$appt_lead_time < 0, 0, 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)
```



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 which have a higher incentive to keep. Also, there are small bumps in the histogram around 180 and 360 days, which is probably indicitive of regular 6-month and 12-month checkups.

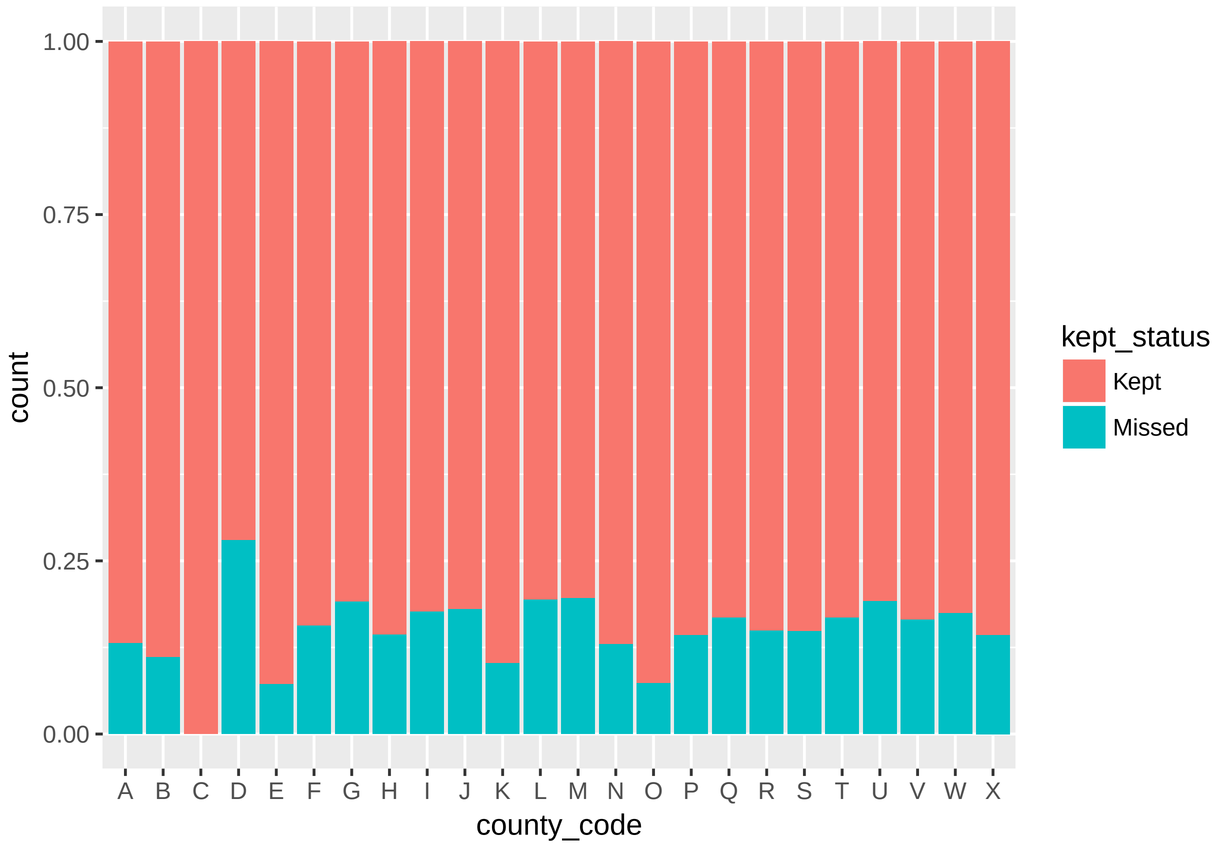
```
appointments %>%
    filter(appt_lead_time >= 0 & appt_lead_time < 400) %>%
    ggplot(
        aes(x = appt_lead_time, color = kept_status, fill = kept_status)
    ) +
        geom_histogram(binwidth = 10, position = "fill")
```



Looking at `appt_lead_time` proportionally, the drop in missed appointments among those with the shortest lead times is easier to see.

county_code

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



**

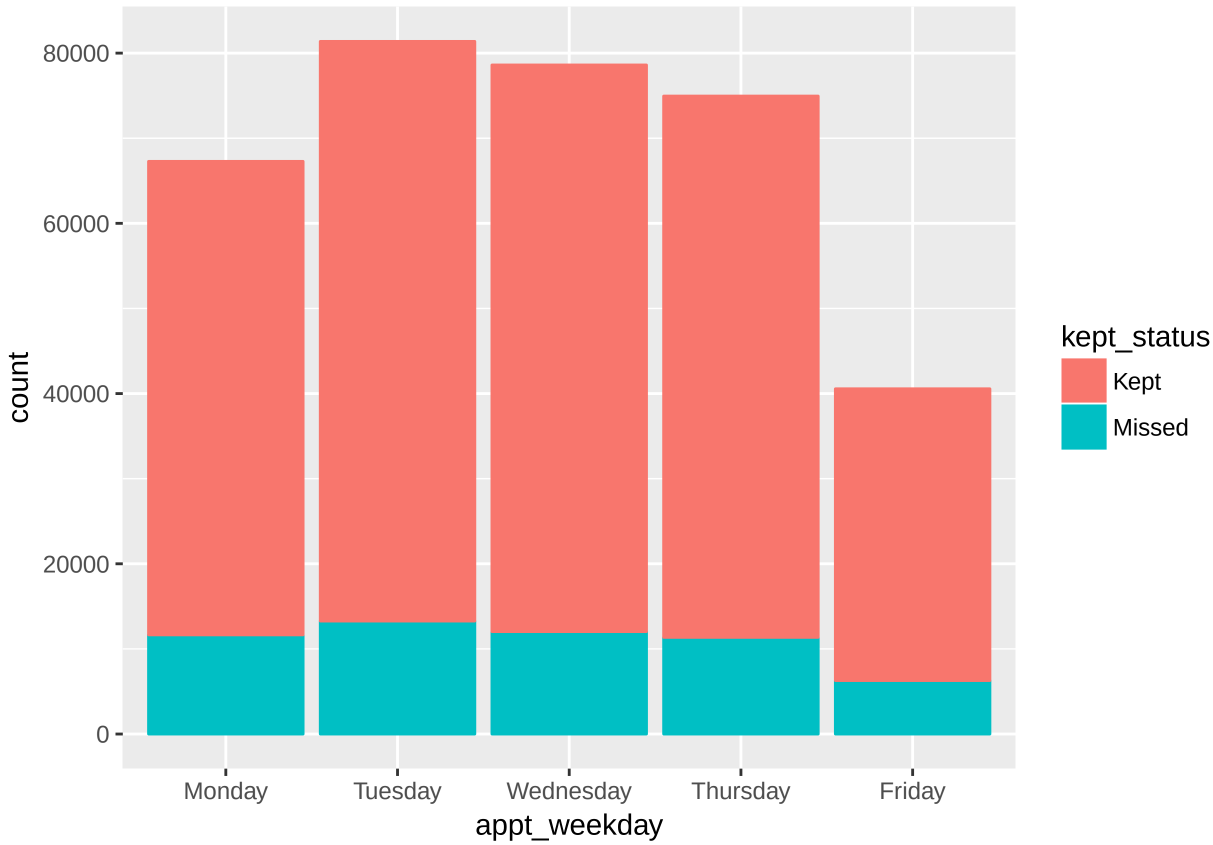
appt_weekday

```
table(appointments$appt_weekday)
```

```
##
##    Friday    Monday    Sunday  Thursday   Tuesday Wednesday
##     40558     67280        12     74952     81376     78604
```

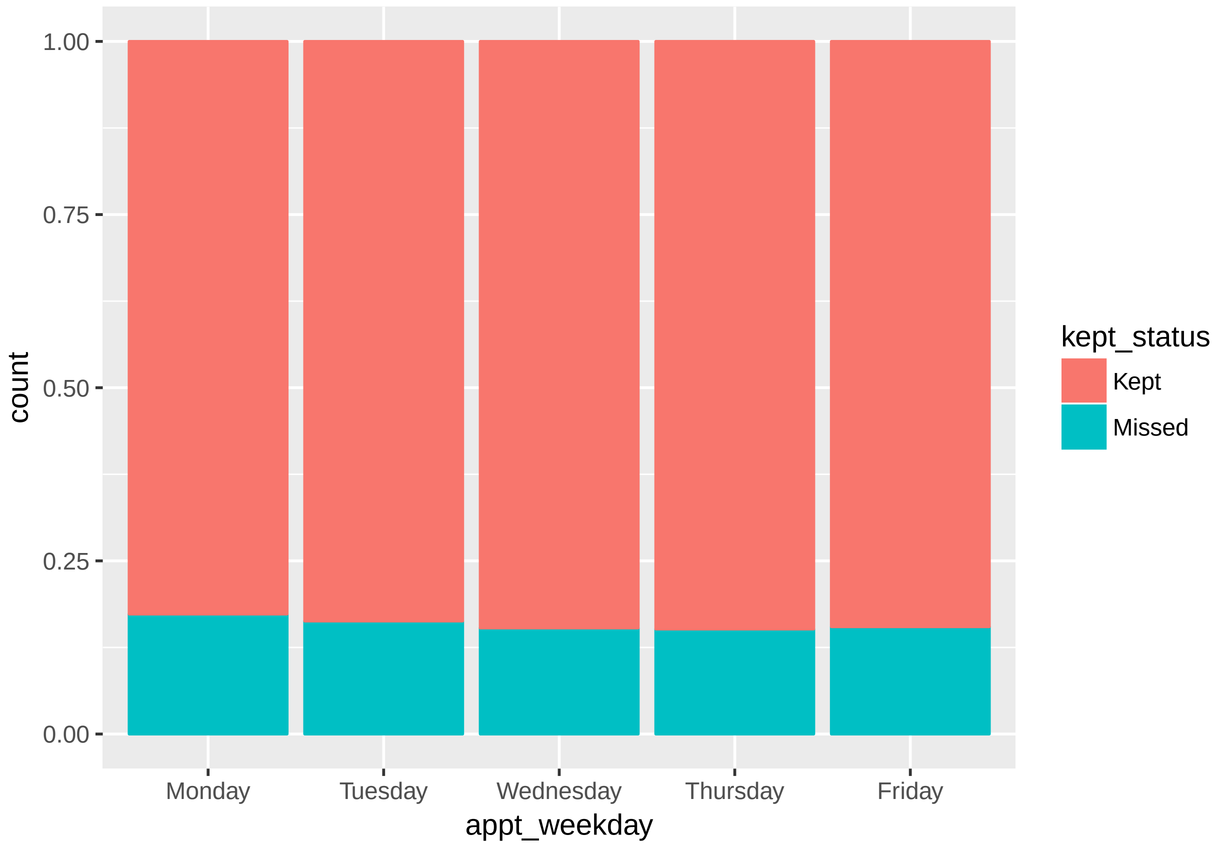
There are only 12 Sunday appointments, so I will remove the observations. I will also convert the character variable to an ordered factor to see the days of the week in the correct order.

```
appointments <- appointments %>%
    filter(appt_weekday != "Sunday")


appointments$appt_weekday <- factor(
    appointments$appt_weekday,
    levels = c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday"))

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



Most appointments occur on Tuesdays, and trail off later in the week with Friday having the fewest. Due to the difference between the number of appointments each day, it is difficult to see the ratio of missed appoiinments each day, so I will create a proportional plot.

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



There is a small variance in the ratio of missed appointments across the days of the week. The highest ratio of missed appointments occurs on Monday, and drops off slightly through Wednesday before leveling off.

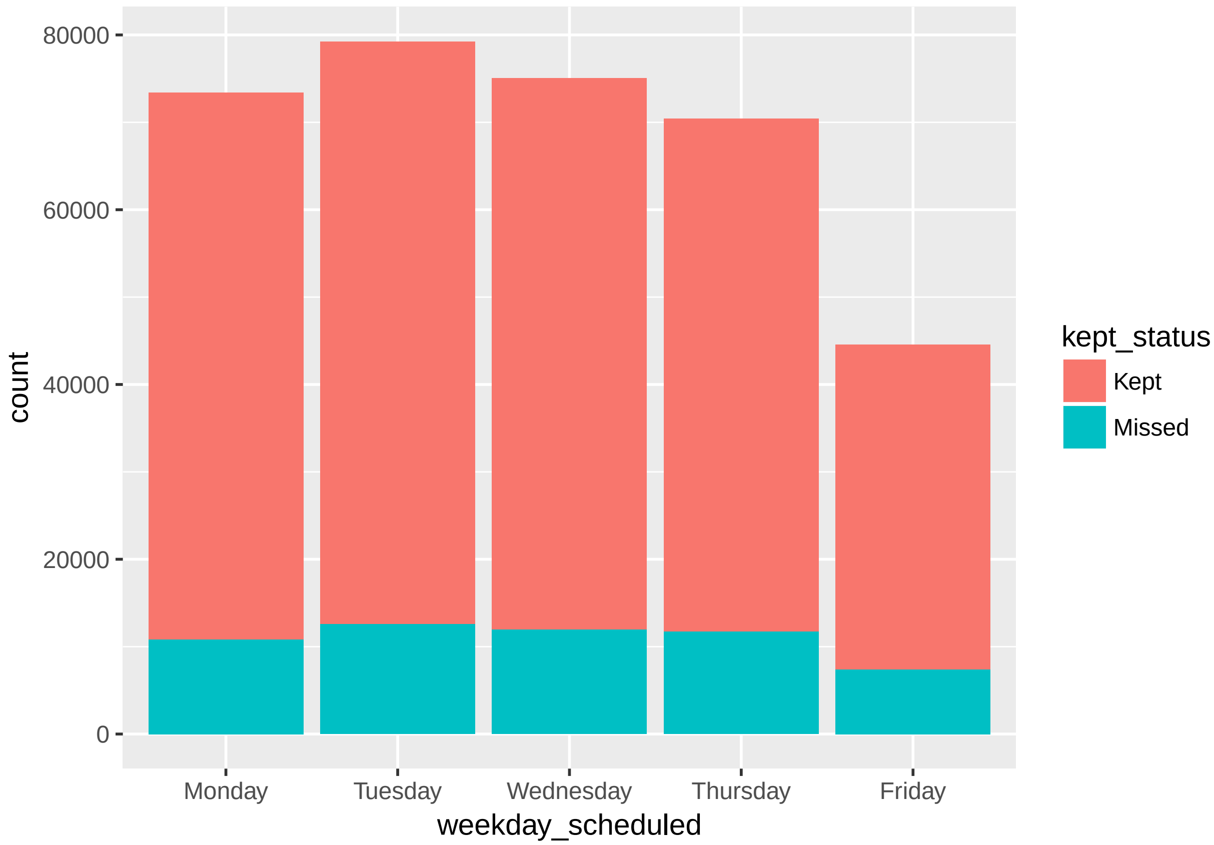
weekday_scheduled

```
table(appointments$weekday_scheduled)
```

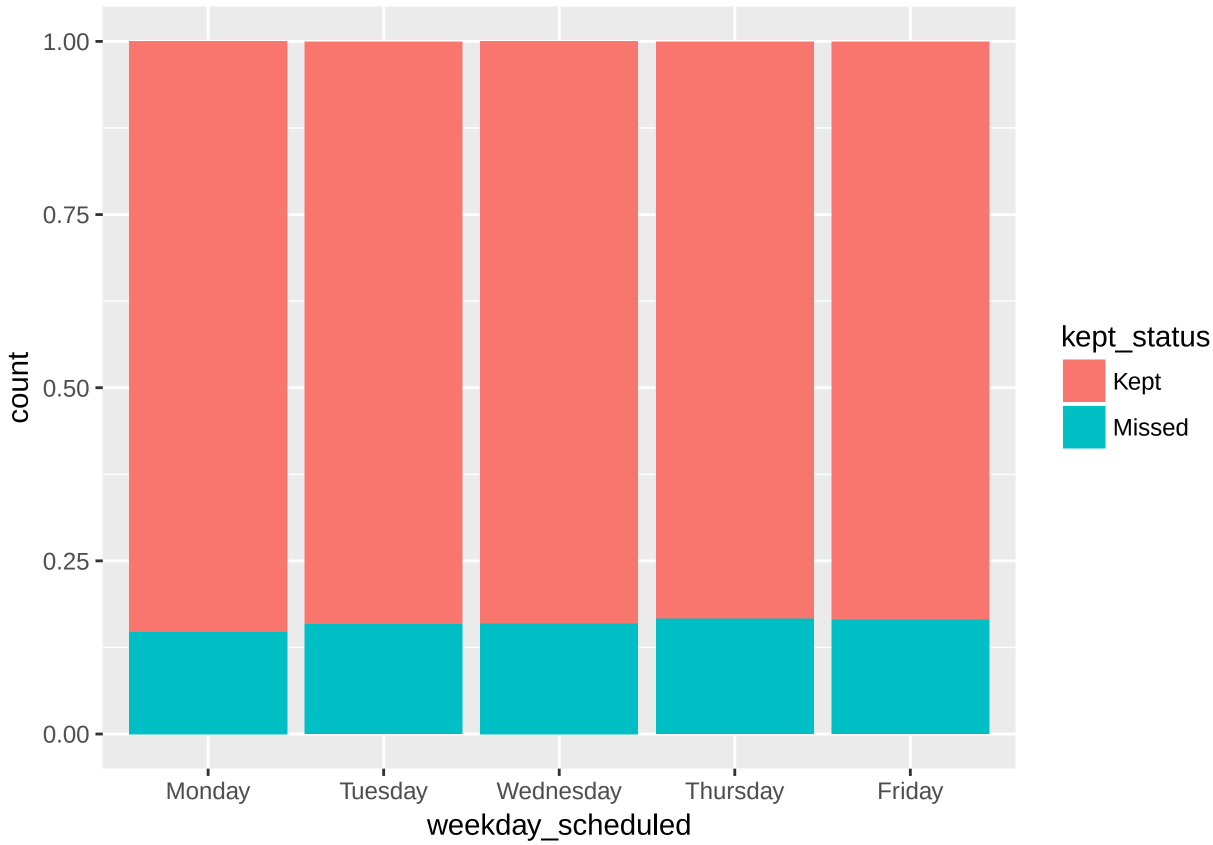
```
##
##    Friday    Monday  Saturday    Sunday  Thursday   Tuesday Wednesday
##     44553     73413        43         7     70441     79261     75052
```

A very small percentage of the observations occur on Saturday or Sunday, so I will remove them.

```
appointments <- appointments %>%
    filter(weekday_scheduled != "Sunday") %>%
    filter(weekday_scheduled != "Saturday")

appointments$weekday_scheduled <- factor(
    appointments$weekday_scheduled,
    levels = c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday"))

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




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



The weekday the appointments are scheduled follows a similar pattern as the weekday the appointments occur, peaking on Tuesday and trailing off as the week progresses. However, the weekday the appointment is initially booked has the opposite effect as the weekday the appointment occurs, with the ratio of missed appointments gradually rising throughout the week. I'm not sure what could cause this, but perhaps people who call in early in the week after a weekend o

Modeling

Create Modeling Data

I will select the data to be used in modeling and assign to `model_data`, then convert the categorical information to dummy variables to help with the modelling.

```
model_data <- appointments %>%
    select(
        length_group, patient_age, patient_gender, billing_type,
        patient_distance, provider_specialty, remind_call_result, hour,
        prior_percent_missed, appt_lead_time, appt_weekday,
        weekday_scheduled, county_code)


factor_columns <- c(
    "length_group", "patient_gender", "billing_type",
    "provider_specialty", "remind_call_result", "hour",
    "appt_weekday", "weekday_scheduled",
    "county_code")
model_data[factor_columns] <- map(model_data[factor_columns], factor)
```

```r
dummy_vars <- caret::dummyVars(~ ., data = model_data)

model_data_dummy <- data.frame(predict(dummy_vars, newdata = model_data))
```

The next step is to look for linear combinations, highly correlated variables, and variables with near zero variance. These variables add complexity to the model without providing any significant information, so I will remove them to help the model work better and more efficiently.

First I will look for linear combinations.

```r
linear_combos <- caret::findLinearCombos(model_data_dummy)

colnames(model_data_dummy[, linear_combos$remove])
```

```
## [1] "patient_gender.Unknown"        "billing_type.DMAP"
## [3] "provider_specialty.Other"      "remind_call_result.Not.Called"
## [5] "hour.21"                       "appt_weekday.Friday"
## [7] "weekday_scheduled.Friday"      "county_code.X"
```

```r
model_data_dummy <- model_data_dummy[, -linear_combos$remove]
```

Next I will check for highly correlated variables.

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

cor_matrix[cm_index[, 1], cm_index[, 2]]
```

```
##                        patient_gender.Female patient_gender.Male
## patient_gender.Male             -0.997632962         1.000000000
## patient_gender.Female            1.000000000        -0.997632962
## provider_specialty.B            -0.004807503         0.004841524
## provider_specialty.A             0.008441594        -0.008524185
##                        provider_specialty.A provider_specialty.B
## patient_gender.Male            -0.008524185          0.004841524
## patient_gender.Female           0.008441594         -0.004807503
## provider_specialty.B           -0.915215490          1.000000000
## provider_specialty.A            1.000000000         -0.915215490
```

```r
cbind(
    cm_index[, 1],
    colnames(model_data_dummy[cm_index[,1]]),
    cm_index[,2],
    colnames(model_data_dummy[cm_index[,2]]))
```

```
##      [,1] [,2]                    [,3] [,4]
## [1,] "6"  "patient_gender.Male"   "5"  "patient_gender.Female"
## [2,] "5"  "patient_gender.Female" "6"  "patient_gender.Male"
## [3,] "11" "provider_specialty.B"  "10" "provider_specialty.A"
## [4,] "10" "provider_specialty.A"  "11" "provider_specialty.B"
```

Most cases of `patient_gender` are `male` or `female`, with few values of `other` and `unknown`, leading to a high negative correlation between male and female. Similary, most cases of `provider_specialty` are `A` or `B`. Due to the dominance of two possible values, there is a high correlation between them and one of each pair will be eliminated. I will eliminate the dummy variables `patient_gender.Male` and `provider_specialty.B`.

```r
model_data_dummy <- model_data_dummy[,-c(6, 11)]
```

Finally, I will check for variables with a variance near zero, and remove them.

```r
near_zero_var <- caret::nearZeroVar(model_data_dummy)

colnames(model_data_dummy[, near_zero_var])
```

```
##  [1] "patient_gender.Other"
##  [2] "provider_specialty.C"
##  [3] "provider_specialty.D"
##  [4] "remind_call_result.Answered...Canceled"
##  [5] "remind_call_result.Answered...Reschedule"
##  [6] "remind_call_result.Busy"
##  [7] "remind_call_result.No.Answer"
##  [8] "hour.0"
##  [9] "hour.5"
## [10] "hour.6"
## [11] "hour.7"
## [12] "hour.12"
## [13] "hour.17"
## [14] "hour.18"
## [15] "hour.19"
## [16] "hour.20"
## [17] "county_code.A"
## [18] "county_code.B"
## [19] "county_code.C"
## [20] "county_code.D"
## [21] "county_code.E"
## [22] "county_code.G"
## [23] "county_code.H"
## [24] "county_code.K"
## [25] "county_code.L"
## [26] "county_code.M"
## [27] "county_code.N"
## [28] "county_code.O"
## [29] "county_code.Q"
## [30] "county_code.R"
## [31] "county_code.S"
## [32] "county_code.T"
## [33] "county_code.V"
## [34] "county_code.W"
```

```r
model_data_dummy <- model_data_dummy[, -near_zero_var]
```

Now that the dummy variables are created and removed when necassary, I will add back the dependent variable `kept_status`.

```r
model_data_dummy <- cbind(appointments[, 1], model_data_dummy)

model_data_dummy$kept_status <- as.factor(model_data_dummy$kept_status)
```

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 tata 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 on future appointments.

```
train <- model_data_dummy[1:205660,]
validate <- model_data_dummy[205661:274200,]
test <- model_data_dummy[274201:nrow(model_data_dummy),]
table(train$kept_status)
```

```
##
##   Kept Missed
## 174610  31050
```

```
train_balance_subset <- train[168750:205660,]
table(train_balance_subset$kept_status)
```

```
##
##   Kept Missed
##  31050   5861
```

```
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
##  31050  31050
```

The models chosen for the missed appointment predictions are glm and random forest, which will both be used with the caret package. The glm model is chosen because this is a binary classification problem, where it is also useful to predict the probabilities of each outcome, and the glm will is well suited to this task. The probabilities of each class will be useful for a client's implementation of the prediction because it allows the individual probabilities to be combined in a given time slot to give an the expected probability for each possible number of mis

**Explain why random forest is being used

Set Up Model Parameters

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

Logistic Regression Model

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

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```

```
summary(glm_model)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.6549  -0.8904  -0.0916   0.9228   2.6492
##
## Coefficients: (1 not defined because of singularities)
##                                          Estimate Std. Error z value
## (Intercept)                            -1.1872445  0.1006652 -11.794
## length_group.Other                     -0.4327036  0.0309163 -13.996
## length_group.30.Min                    -0.5177630  0.0300340 -17.239
## length_group.60.Min                            NA         NA      NA
## patient_age                            -0.0107111  0.0005059 -21.171
## patient_gender.Female                  -0.0727063  0.0187210  -3.884
## billing_type.Commercial                -0.0251882  0.0238528  -1.056
## patient_distance                       -0.0001383  0.0001371  -1.009
## provider_specialty.A                    0.4185288  0.0216471  19.334
## remind_call_result.Answered...Confirmed   -0.1412512  0.0410870  -3.438
## remind_call_result.Answered...No.Response  0.9341118  0.0300142  31.122
## remind_call_result.Failed               1.4894795  0.0393891  37.815
## remind_call_result.Left.Message        -0.1073545  0.0579059  -1.854
## hour.8                                 -0.8348102  0.0875714  -9.533
## hour.9                                 -0.9081427  0.0878155 -10.341
## hour.10                                -1.0100098  0.0876417 -11.524
## hour.11                                -0.9046058  0.0884389 -10.229
## hour.13                                -0.9605157  0.0875406 -10.972
## hour.14                                -0.9842929  0.0878596 -11.203
## hour.15                                -1.0354549  0.0877416 -11.801
## hour.16                                -0.9879241  0.0885063 -11.162
```

```
## prior_percent_missed                      5.0904148  0.0647682  78.594
## appt_lead_time                            0.0040572  0.0001165  34.832
## appt_weekday.Monday                       0.0227201  0.0350346   0.649
## appt_weekday.Tuesday                     -0.2141510  0.0333828  -6.415
## appt_weekday.Wednesday                   -0.3276031  0.0338726  -9.672
## appt_weekday.Thursday                    -0.1710173  0.0344781  -4.960
## weekday_scheduled.Monday                 -0.0260882  0.0328475  -0.794
## weekday_scheduled.Tuesday                 0.0896933  0.0321042   2.794
## weekday_scheduled.Wednesday               0.0658365  0.0325136   2.025
## weekday_scheduled.Thursday                0.1149014  0.0326767   3.516
## county_code.F                            -0.3994385  0.0366336 -10.904
## county_code.I                             0.2251877  0.0301834   7.461
## county_code.J                             0.0487022  0.0348062   1.399
## county_code.P                            -0.0285786  0.0255264  -1.120
## county_code.U                             0.3439278  0.0366015   9.397
##                                          Pr(>|z|)
## (Intercept)                               < 2e-16 ***
## length_group.Other                        < 2e-16 ***
## length_group.30.Min                       < 2e-16 ***
## length_group.60.Min                            NA
## patient_age                               < 2e-16 ***
## patient_gender.Female                    0.000103 ***
## billing_type.Commercial                  0.290975
## patient_distance                         0.312868
## provider_specialty.A                      < 2e-16 ***
## remind_call_result.Answered...Confirmed   0.000586 ***
## remind_call_result.Answered...No.Response  < 2e-16 ***
## remind_call_result.Failed                 < 2e-16 ***
## remind_call_result.Left.Message          0.063747 .
## hour.8                                    < 2e-16 ***
## hour.9                                    < 2e-16 ***
## hour.10                                   < 2e-16 ***
## hour.11                                   < 2e-16 ***
## hour.13                                   < 2e-16 ***
## hour.14                                   < 2e-16 ***
## hour.15                                   < 2e-16 ***
## hour.16                                   < 2e-16 ***
## prior_percent_missed                      < 2e-16 ***
## appt_lead_time                            < 2e-16 ***
## appt_weekday.Monday                      0.516658
## appt_weekday.Tuesday                     1.41e-10 ***
## appt_weekday.Wednesday                    < 2e-16 ***
## appt_weekday.Thursday                    7.04e-07 ***
## weekday_scheduled.Monday                 0.427067
## weekday_scheduled.Tuesday                0.005209 **
## weekday_scheduled.Wednesday              0.042879 *
## weekday_scheduled.Thursday               0.000438 ***
## county_code.F                             < 2e-16 ***
## county_code.I                            8.61e-14 ***
## county_code.J                            0.161741
## county_code.P                            0.262897
## county_code.U                             < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 86089  on 62099  degrees of freedom
## Residual deviance: 68818  on 62065  degrees of freedom
## AIC: 68888
##
## Number of Fisher Scoring iterations: 5
```

Random Forest Model

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

Model Selection

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

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```

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

conf_mat_glm
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  Kept Missed
##     Kept   42262   3460
##     Missed 14822   7996
##
##                Accuracy : 0.7333
##                  95% CI : (0.7299, 0.7366)
##     No Information Rate : 0.8329
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.3139
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.6980
##             Specificity : 0.7403
##          Pos Pred Value : 0.3504
##          Neg Pred Value : 0.9243
##              Prevalence : 0.1671
##          Detection Rate : 0.1167
##    Detection Prevalence : 0.3329
##       Balanced Accuracy : 0.7192
##
##        'Positive' Class : Missed
```

```
##
```

```r
conf_mat_glm$byClass["F1"]
```

```
##        F1
## 0.4665928
```

```r
pred_rf <- predict(rf_model, validate)

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

conf_mat_rf
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  Kept Missed
##     Kept   40174   2459
##     Missed 16910   8997
##
##                Accuracy : 0.7174
##                  95% CI : (0.714, 0.7208)
##     No Information Rate : 0.8329
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.3252
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.7854
##             Specificity : 0.7038
##          Pos Pred Value : 0.3473
##          Neg Pred Value : 0.9423
##              Prevalence : 0.1671
##          Detection Rate : 0.1313
##    Detection Prevalence : 0.3780
##       Balanced Accuracy : 0.7446
##
##        'Positive' Class : Missed
##
```

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

```
##        F1
## 0.4815994
```

I will use the positive predictive value (where positive is missed appointments) and the F1 scores to select the best model. Predictiing the missed appointments is more difficult since it is the minority class, and this is where I am most interested in reducing the error. The missed appointments were designated the positive class, therefore the positive predictive value is the accuracy of the model in predicting the missed appointments. The F1 score will be looked at rather than accuracy, since the F1 is a harmonic average that is a better metric due to the imbalance.

Looking at the F1 scores, the random forest model outperforms the glm model, with a score of 0.4815994 vs the glm model with a score of 0.4665928.

The glm model has a positive predictive value of 0.3475, and the random forest model has a positive predictive value of …

The random forest outperforms the glm model in both the F1 and positive predictive value metric.

Testing Expected Model Performance

add code and desciption here

- Paragraph about the results
- speak about the performance overall and how i view it when considering the nature of the problem. eg the model is still wrong over half the time when it predicts a missed appointment, yet with no model, predicting a missed appointment would be incorrect 86 percent of the time.

Recommendations

*3 concrete recommendations.

*Speak about additional improvements that could be made. further analysis can be done on combining the probabilities of multiple appointments and determining the error rate of the combined probability. could be interesting since the error goes both ways, multiple errors could somewhat offset each other.

*paragraph about the class predictions

```
rf_pred_probs <- predict(rf_model, newdata = validate, type = "prob")
rf_probs <- cbind(validate$kept_status, rf_pred_probs)
head(rf_probs)
```

Conclusion

final summary - client-benefit centric

control <- caret::trainControl(method = "none")