Capstone Project - Predicting Patient No-Shows Using Appointment 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()
## )
```

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

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

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)
```

```
## 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 2880
##   .. ..- attr(*, "units")= chr "secs"
##  $ 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 Respo
##  - attr(*, "spec")=List of 2
##   ..$ cols   :List of 14
##   .. ..$ kept_status       : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ appt_date         : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ appt_time         :List of 1
##   .. .. ..$ format: chr ""
##   .. .. ..- attr(*, "class")= chr  "collector_time" "collector"
##   .. ..$ appt_length       : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ date_scheduled    : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ patient_age       : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ patient_gender    : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ billing_type      : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ prior_missed      : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ prior_kept        : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ patient_distance  : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ office_zip        : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ provider_specialty: list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ remind_call_result: list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   ..$ default: list()
##   .. ..- attr(*, "class")= chr  "collector_guess" "collector"
##   ..- attr(*, "class")= chr "col_spec"
```

```
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
```

```
# Check for NAs
sapply(appointments, function(x) sum(is.na(x)))
```

```
##        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
##                  0                  0
```

patient_distance variable has 972 NA values.

Data Dictionary

```
variable_description <- 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_description))
```

```
## # A tibble: 14 x 3
##    V1    variable           variable_description
##    <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
```

Will combine the appointment time and date into one variable, appt_datetime.

```
appointments_2 <- appointments %>%
    mutate(appt_datetime = lubridate::mdy_hms(paste(appt_date, appt_time)))
appointments_2$date_scheduled <- as.POSIXct(appointments_2$date_scheduled,
                                            format = "%m/%d/%y")
```

Calculating percent of missed appointments overall. Will first create a logical variable *missed*, where 1 represents a missed appointment and 0 represents a kept appointment.

```
appointments_2 <- appointments_2 %>%
    mutate(missed = ifelse(appointments_2$kept_status == "Missed", 1,0))
missed_rate <- mean(appointments_2$missed)
missed_rate
```

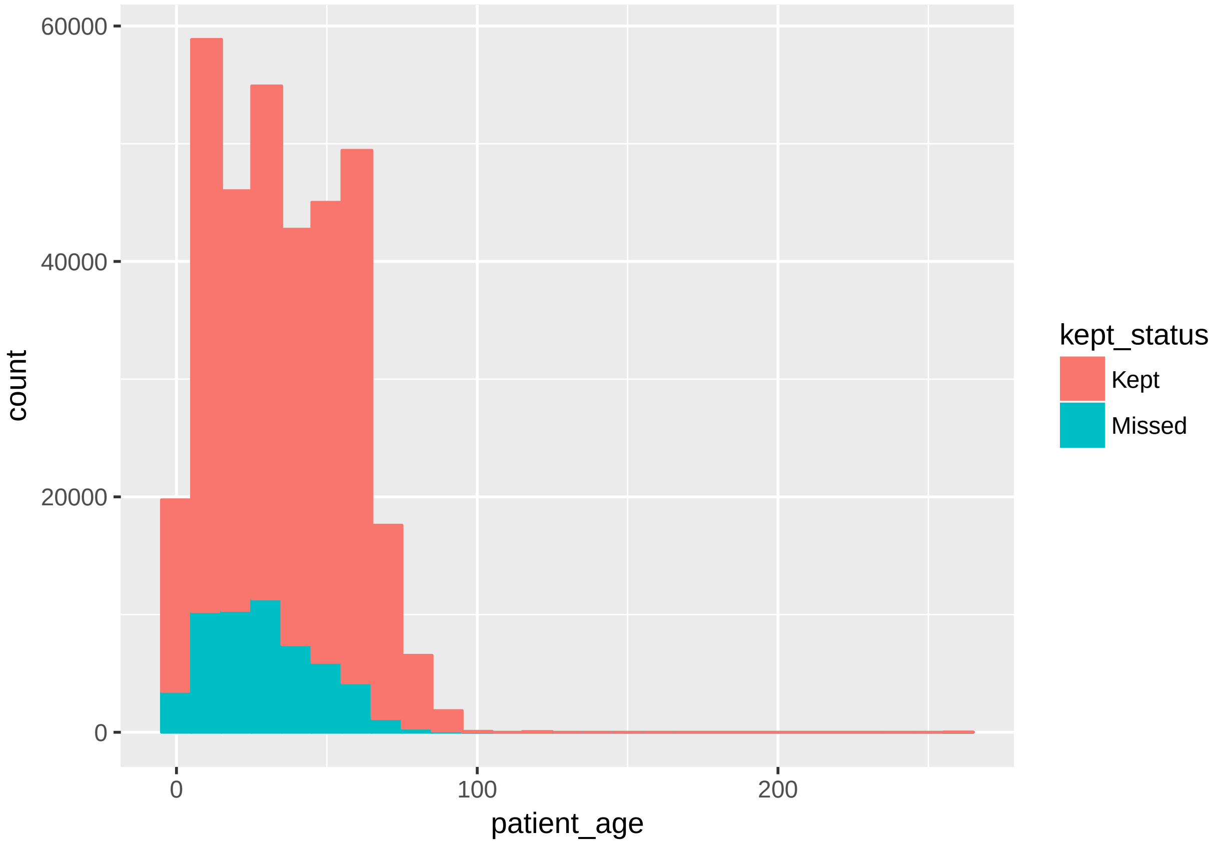
```
## [1] 0.1592944
```

About 16% of the total appointments are missed.

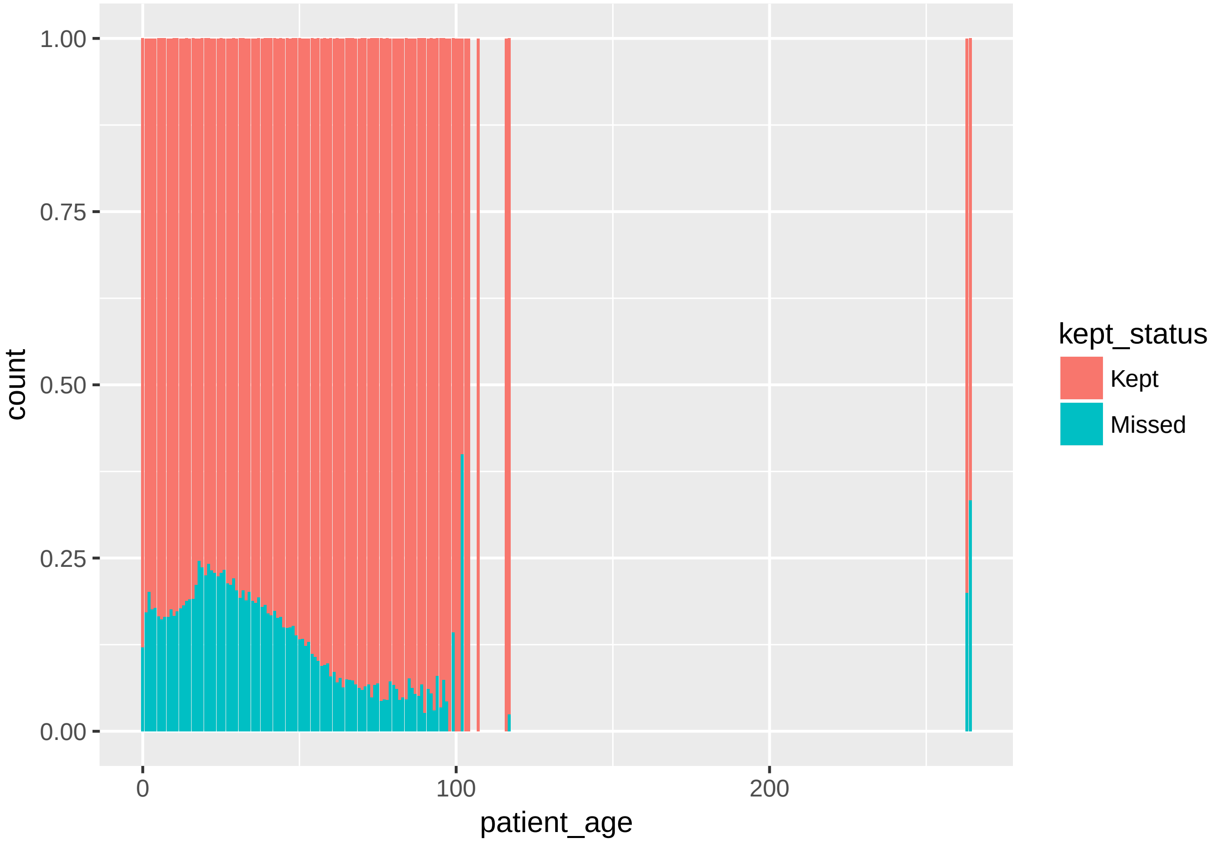
Data Exploration

patient_age

```
ggplot(
  data = appointments_2,
  mappng = aes(x = patient_age)
) +
  geom_histogram(
    mapping = aes(x = patient_age, col = kept_status, fill = kept_status),
    binwidth = 10)
```



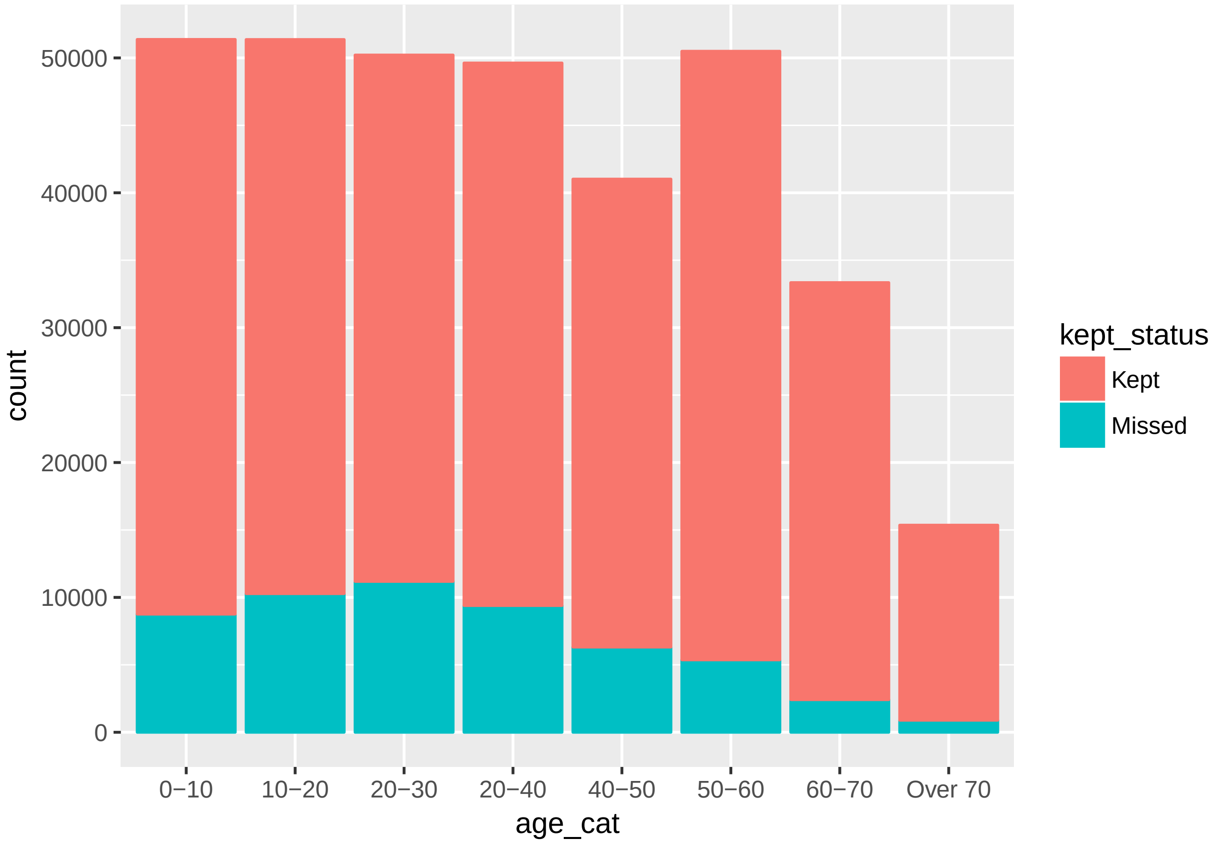
```
ggplot(data = appointments_2) +
    geom_bar(
        mapping = aes(x = patient_age, fill = kept_status),
        position = "fill")
```



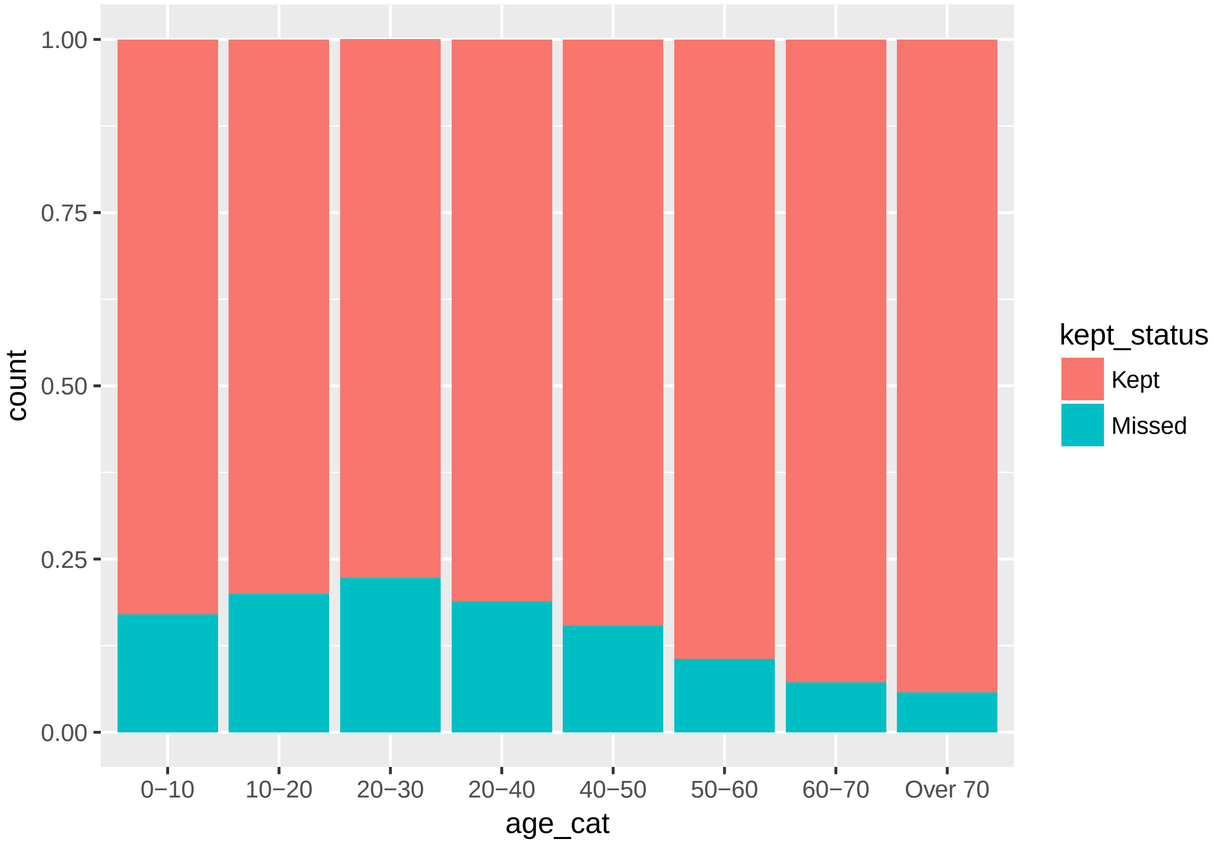
Ranges from 0-264, so there are obviously a few impossible values. Ratio of missed appointments decreases with age in general.

Removing obervations of ages greater than 100, creating categorical age groups and replotting.

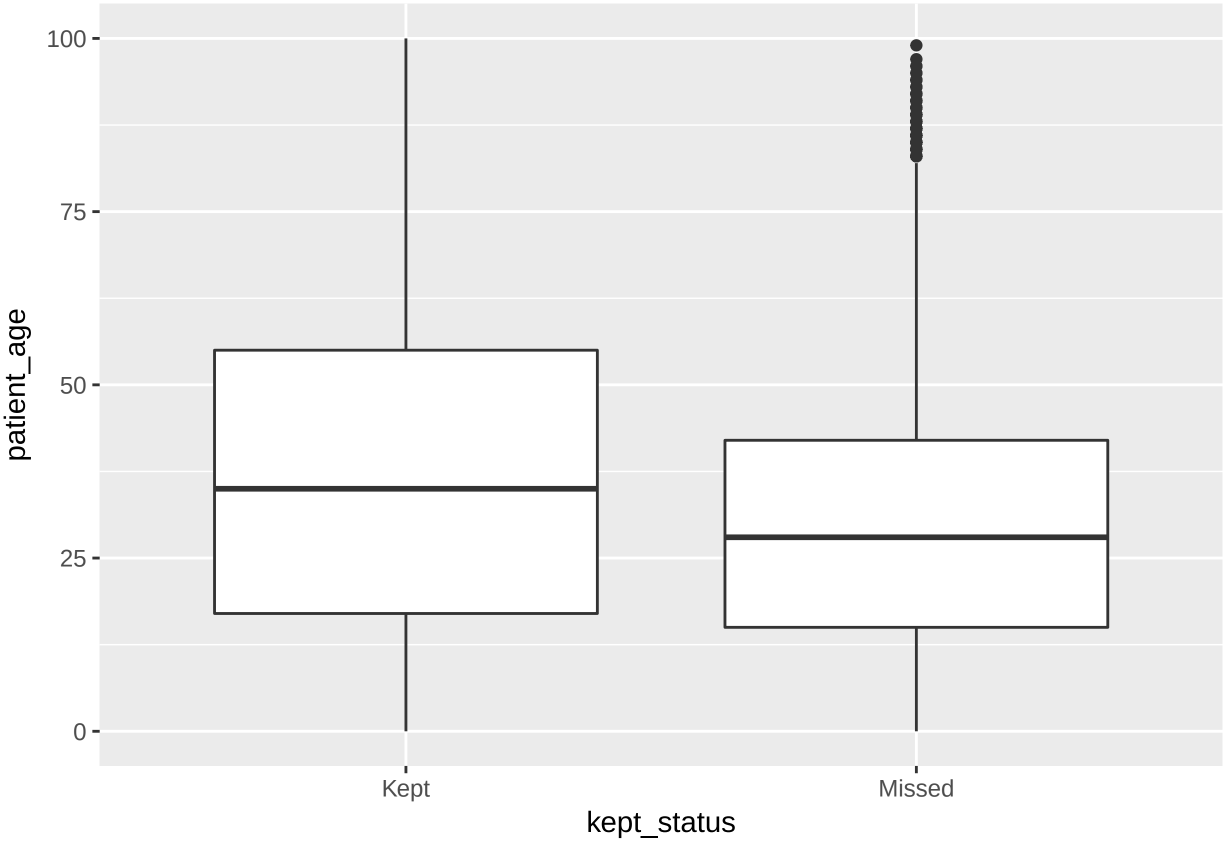
```
appointments_2 <- appointments_2 %>%
    filter(patient_age <= 100)

appointments_2 <- appointments_2 %>%
    mutate(age_cat = cut(patient_age, breaks = c(-1, 10, 20, 30, 40, 50, 60, 70, 101),
                         labels = c("0-10", "10-20", "20-30", "20-40", "40-50",
                                    "50-60", "60-70", "Over 70")))


ggplot(appointments_2, aes(x = age_cat, group = kept_status, col = kept_status,
                           fill = kept_status)) +
    stat_count()
```



```
ggplot(data = appointments_2) +
    geom_bar(mapping = aes(x = age_cat, fill = kept_status), position = "fill")
```



```
ggplot(data = appointments_2, aes(x = kept_status, y = patient_age)) +
    geom_boxplot()
```

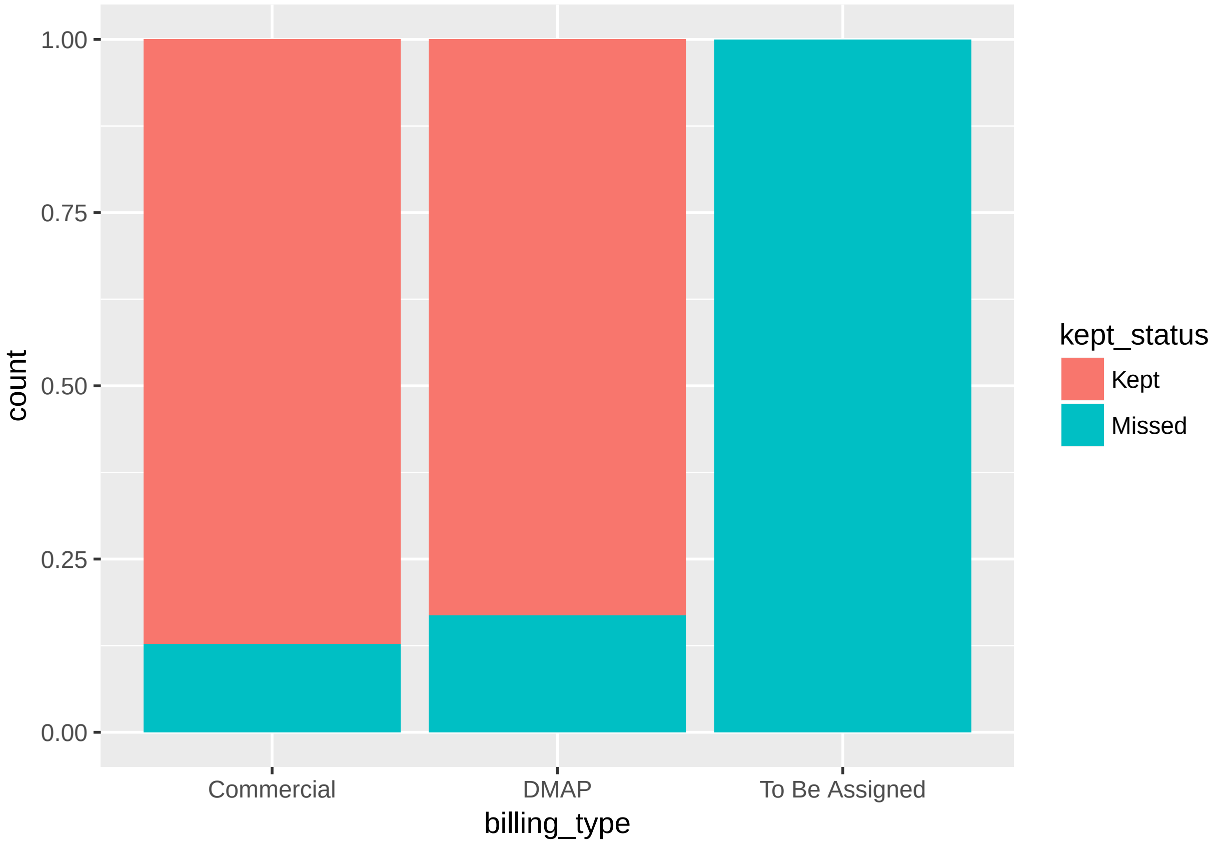


billing_type

```
table(appointments_2$billing_type)
```

```
##
##     Commercial           DMAP To Be Assigned
##          78278         264486              1
```

```
ggplot(data = appointments_2) +
    geom_bar(mapping = aes(x = billing_type, fill = kept_status), position = "fill")
```



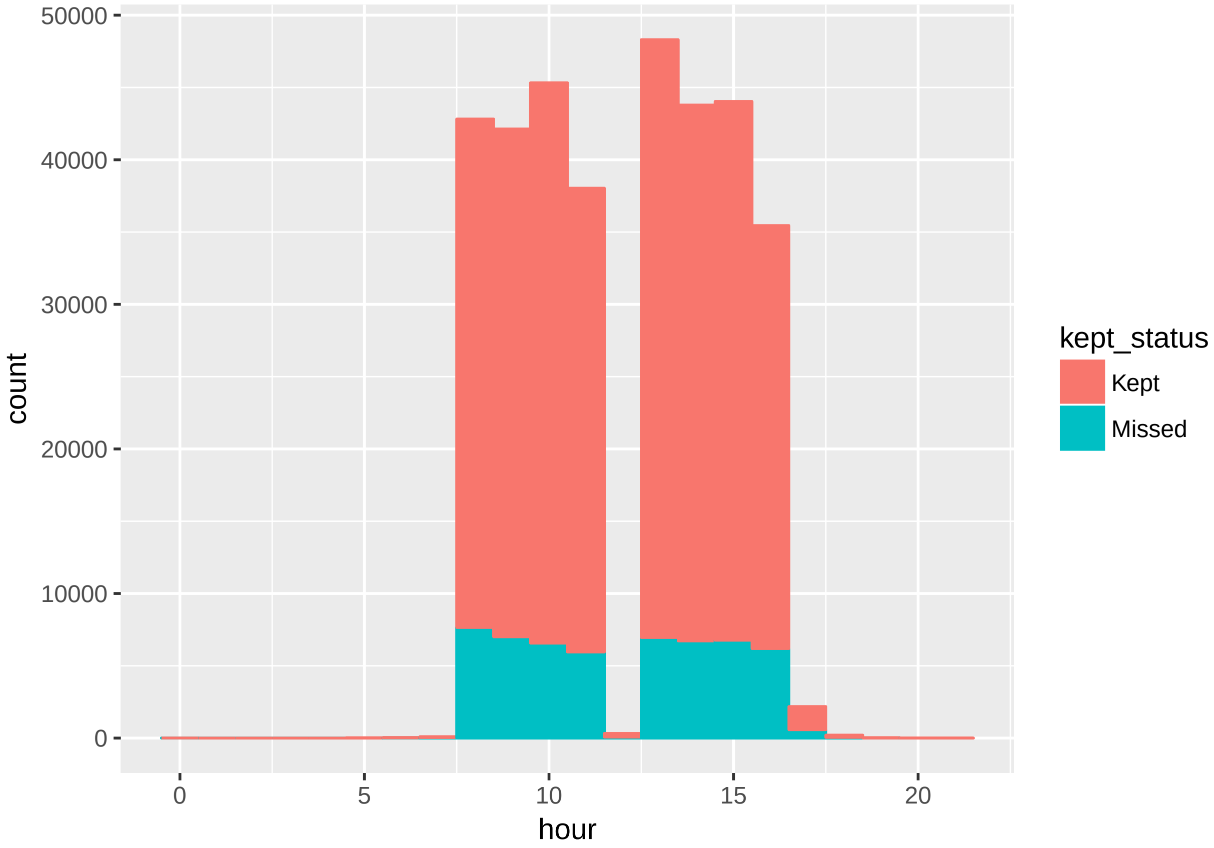
Only one row has *To Be Assigned* value and will just be removed There is a minor difference between billing types. DMAP has a higher proportion of missed appointments

```
appointments_2 <- subset(appointments_2,
                         appointments_2$billing_type != "To Be Assigned")
```

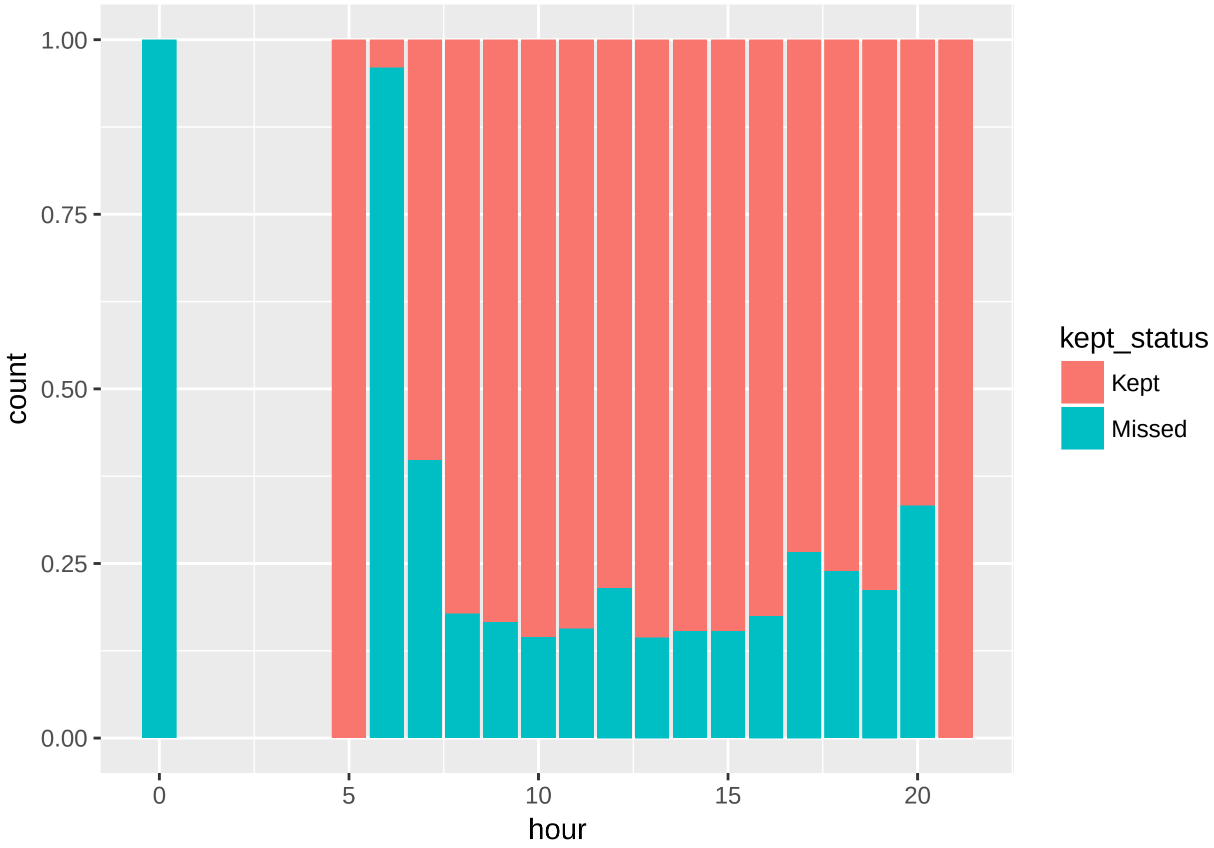
appt_datetime

Creating new *hour* variable and plot by hour

```
appointments_2 <- appointments_2 %>%
    mutate(hour = lubridate::hour(appointments_2$appt_datetime))

ggplot(data = appointments_2,
       aes(x = hour, group = kept_status, col = kept_status, fill = kept_status)) +
    geom_histogram(binwidth = 1)
```



```
ggplot(data = appointments_2) +
    geom_bar(mapping = aes(x = hour, fill = kept_status), position = "fill")
```



Ranges from 00:00:00 to 21:00:00. More appointments are missed in the early morning, late afternoon and early evening, and around lunchtime, however, there are very few appointments at these times. During main scheduling periods, the variation is less significant.

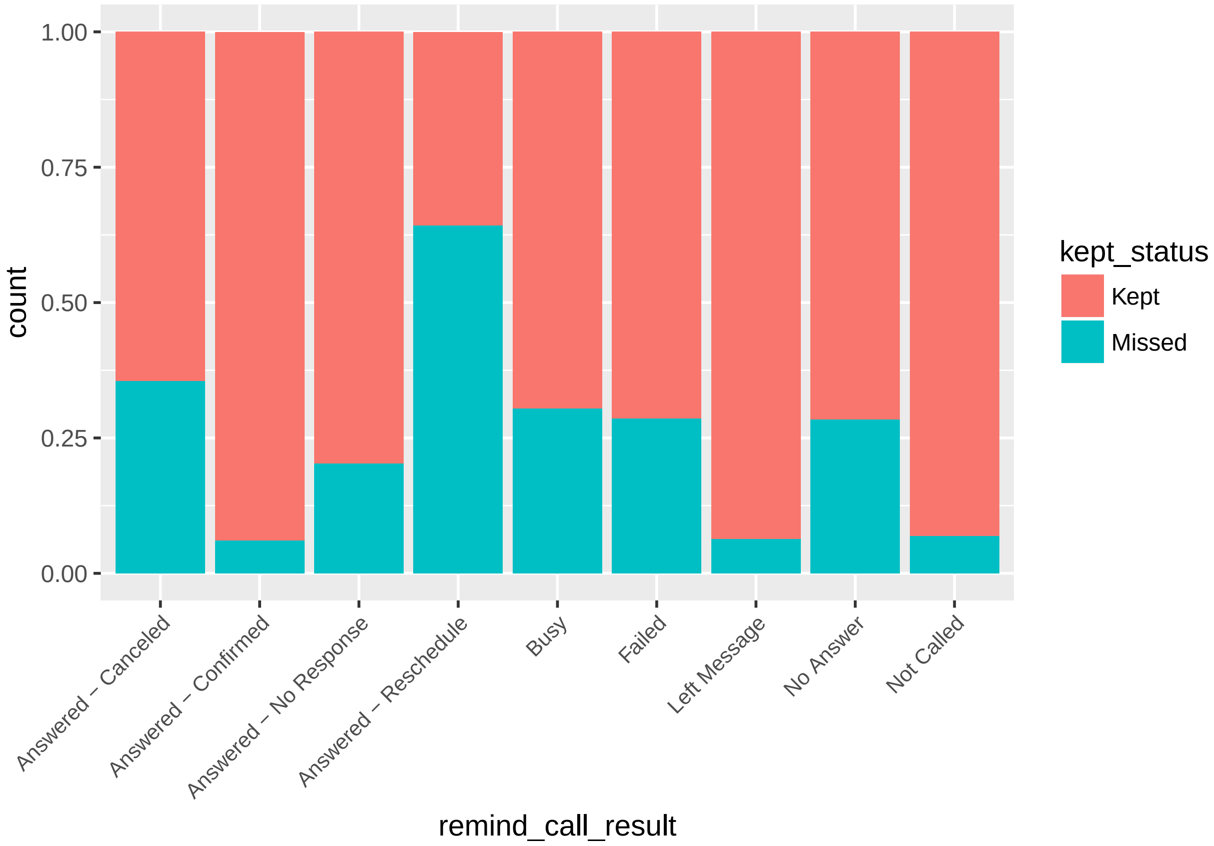
remind_call_result

```r
table(appointments_2$remind_call_result)
```

```
##
##    Answered - Canceled   Answered - Confirmed Answered - No Response
##                    152                  49108                 180860
##  Answered - Reschedule                   Busy                 Failed
##                   1369                   1104                  27943
##           Left Message              No Answer             Not Called
##                  18429                    377                  63422
```

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

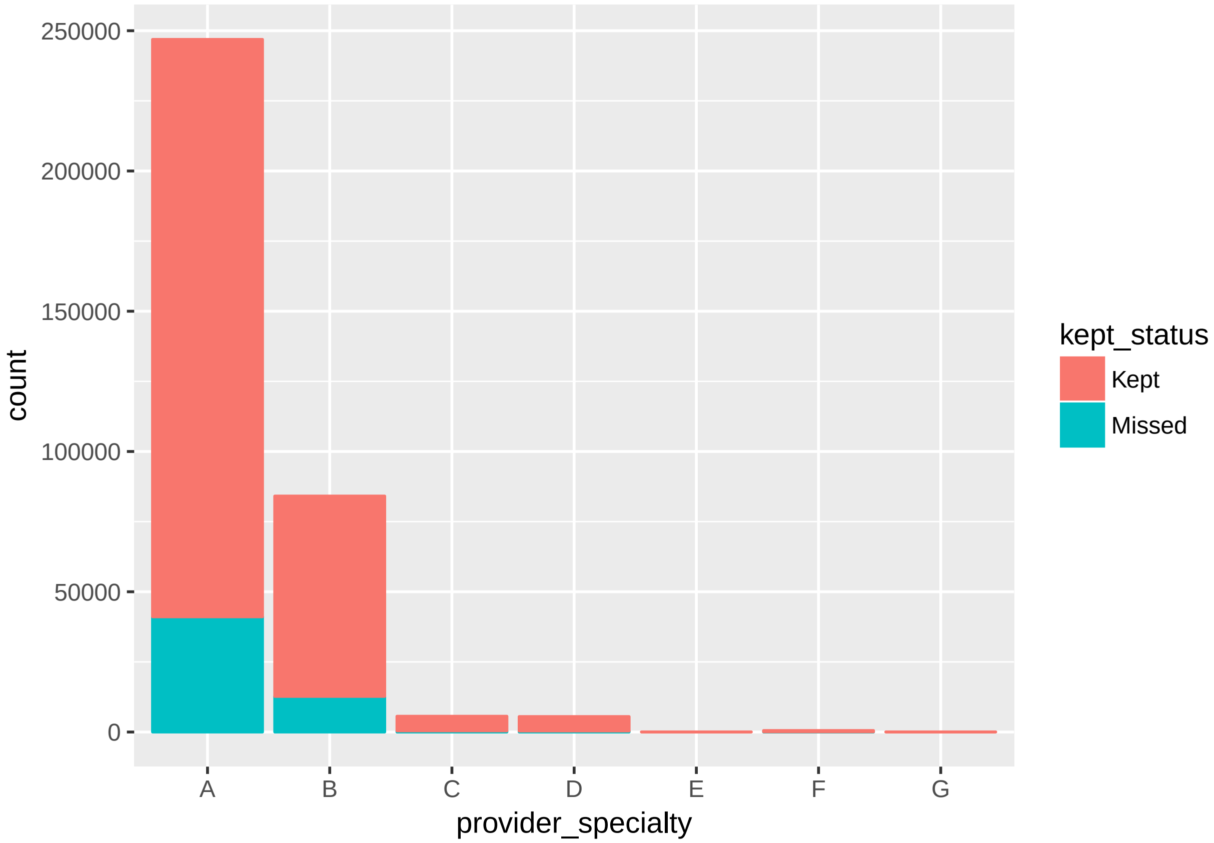
```r
ggplot(data = appointments_2) +
    geom_bar(
        mapping = aes(x = remind_call_result, fill = kept_status),
        position = "fill"
) +
    theme(
        axis.text.x = element_text(size = 8, angle = 45,
        hjust = 1, vjust = 1))
```



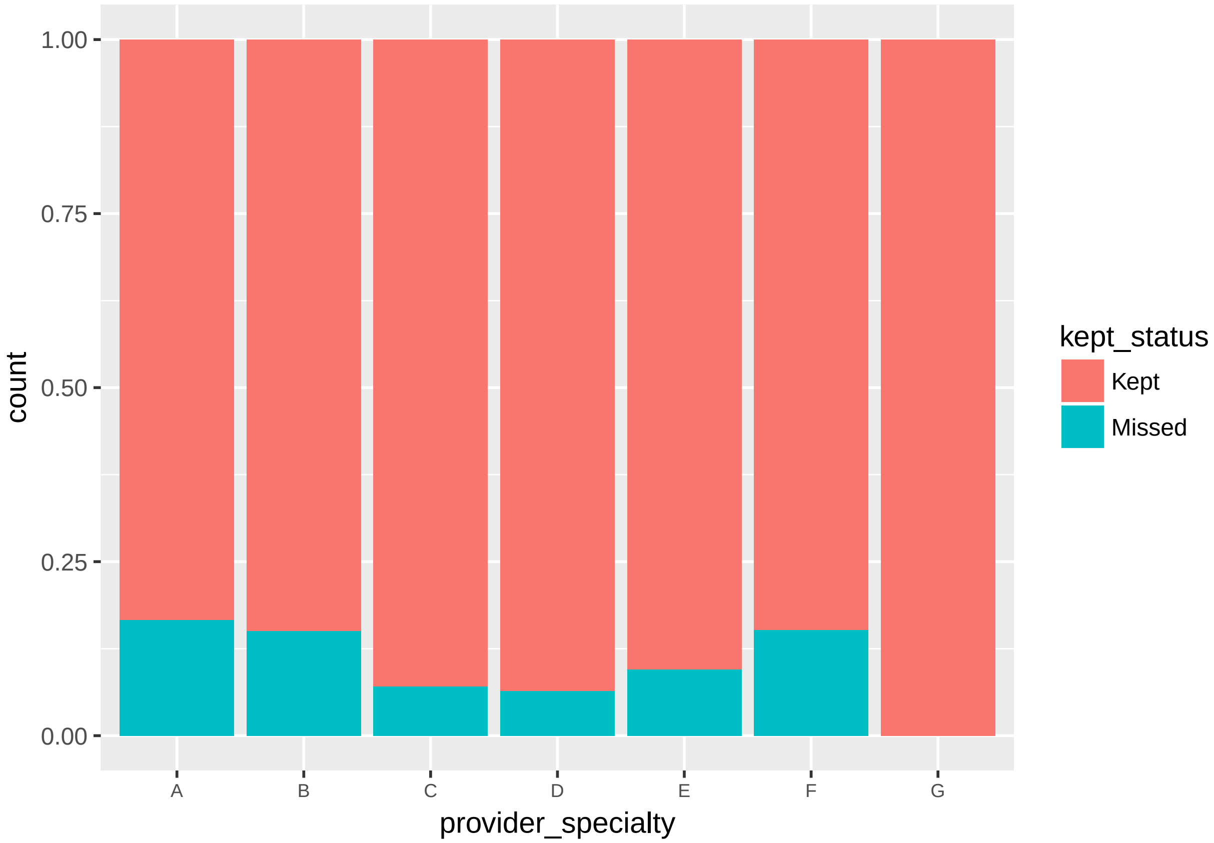
~65% of appointments with “Answered - Cancelled” and ~35% with “Answered-Reschedule” still kept their appointments, however, very few observations in these categories.

provider_specialty

```
ggplot(
    data = appointments_2,
    mapping = aes(x = provider_specialty, col = kept_status, fill = kept_status)
) +
    stat_count()
```



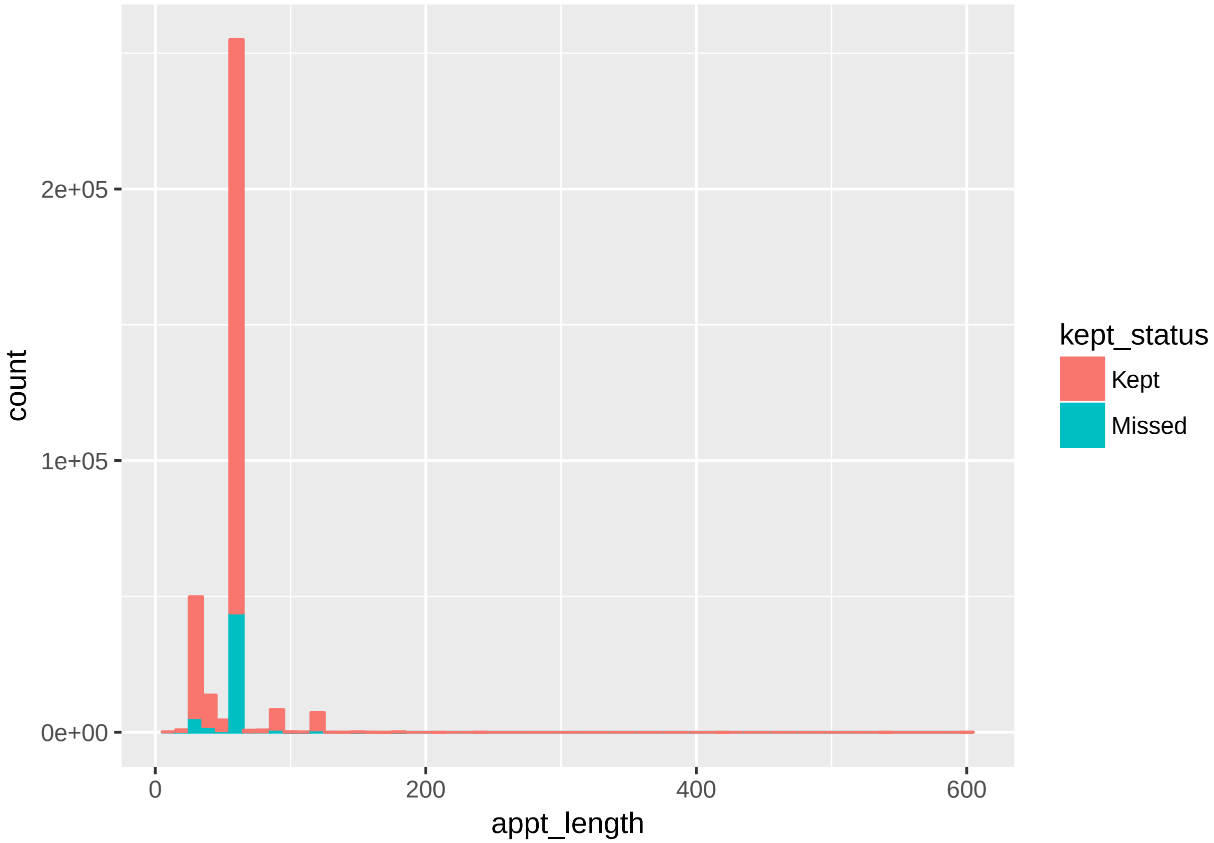
```
ggplot(data = appointments_2) +
    geom_bar(
        mapping = aes(x = provider_specialty, fill = kept_status),
        position = "fill"
) +
    theme(axis.text.x = element_text(size = 7))
```



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

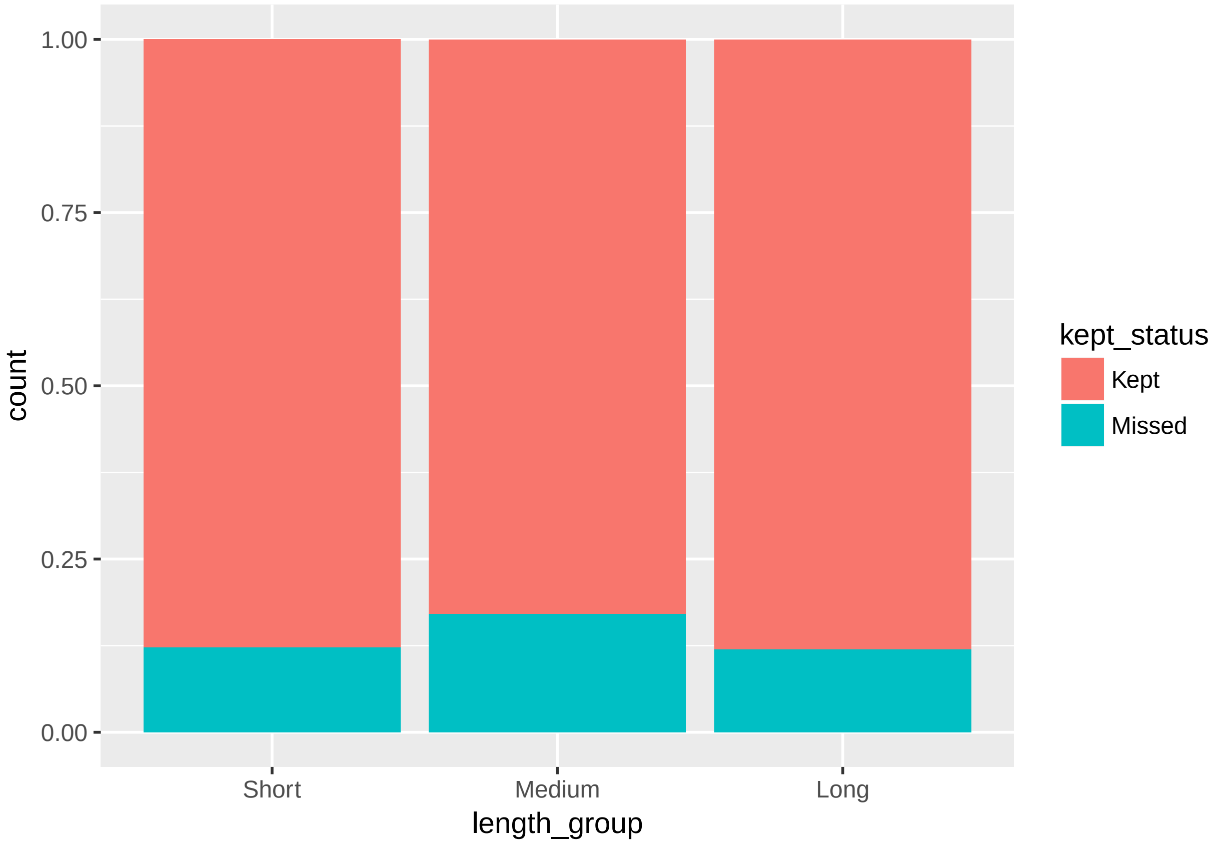
appt_length

```
ggplot(
    data = appointments_2,
    mapping = aes(x = appt_length, col = kept_status, fill = kept_status)
) +
    geom_histogram(binwidth = 10)
```



Most Appointments appear to be 60 minutes long. 30-minute appointments are next most popular.

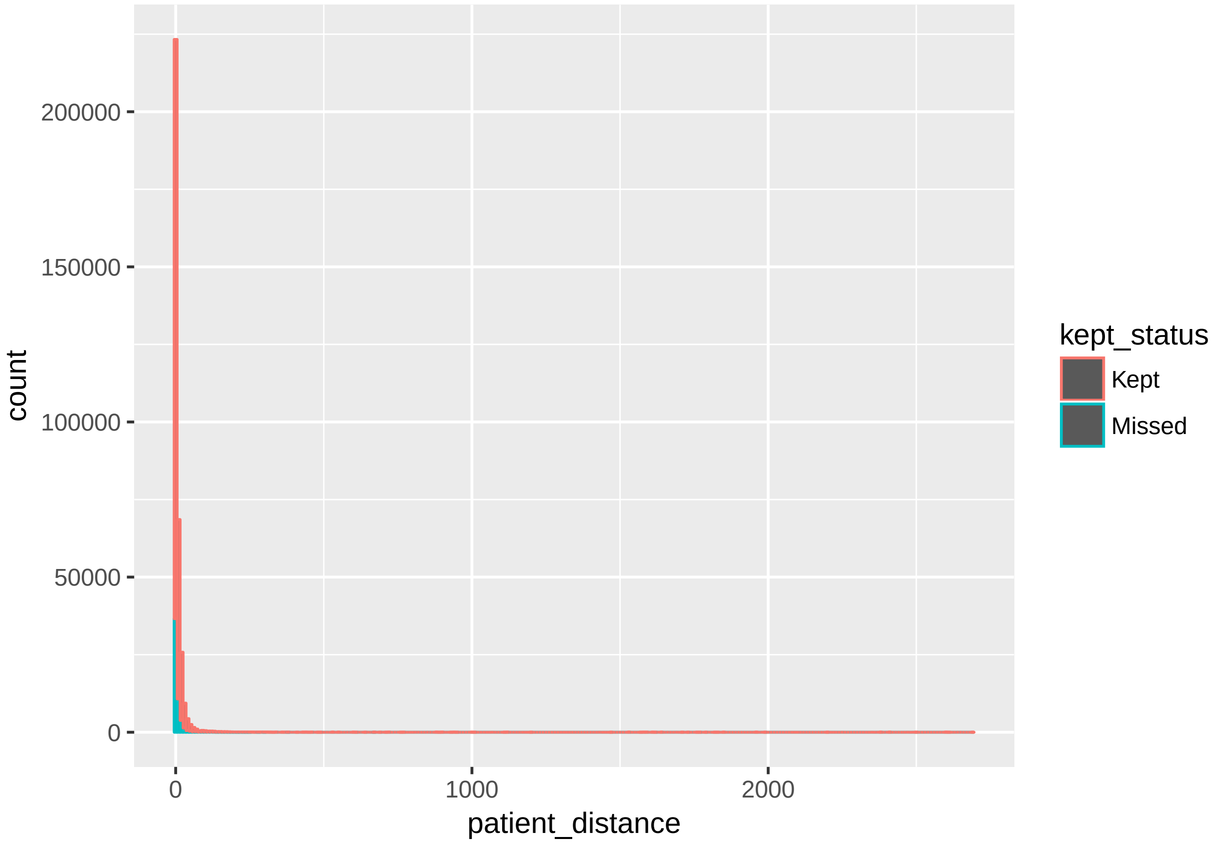
```r
appointments_2 <- appointments_2 %>%
    mutate(length_group = cut(appt_length, breaks = c(-1, 45, 75, 1000), labels = c("Short", "Medium",

ggplot(data = appointments_2) +
    geom_bar(mapping = aes(x = length_group, fill = kept_status), position = "fill")
```



patient_distance

```
ggplot(appointments_2, aes(x = patient_distance, group = kept_status, col = kept_status)) +
    geom_histogram(binwidth = 10)
```

```
## Warning: Removed 972 rows containing non-finite values (stat_bin).
```



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

```
appointments_2$patient_distance <- appointments_2$patient_distance %>%
    tidyr::replace_na(median(appointments_2$patient_distance, na.rm = TRUE))
```

Create new variables

percent_missed = percent of prior appointments missed. New represents represents first time appointments appt_lead_time is the difference between the day the appointment was scheduled and the day of the appointment.

```
appointments_3 <- appointments_2 %>%
    mutate(percent_missed = prior_missed / (prior_missed + prior_kept)) %>%
    mutate(new = ifelse(appointments_2$prior_missed == 0 & appointments_2$prior_kept == 0, 1, 0)) %>%
    mutate(appt_lead_time = date(appt_datetime) - date(date_scheduled)) %>%
    mutate(weekday = strftime(appt_datetime, "%A"))
```

Add city_size and county_code from zipcode data.

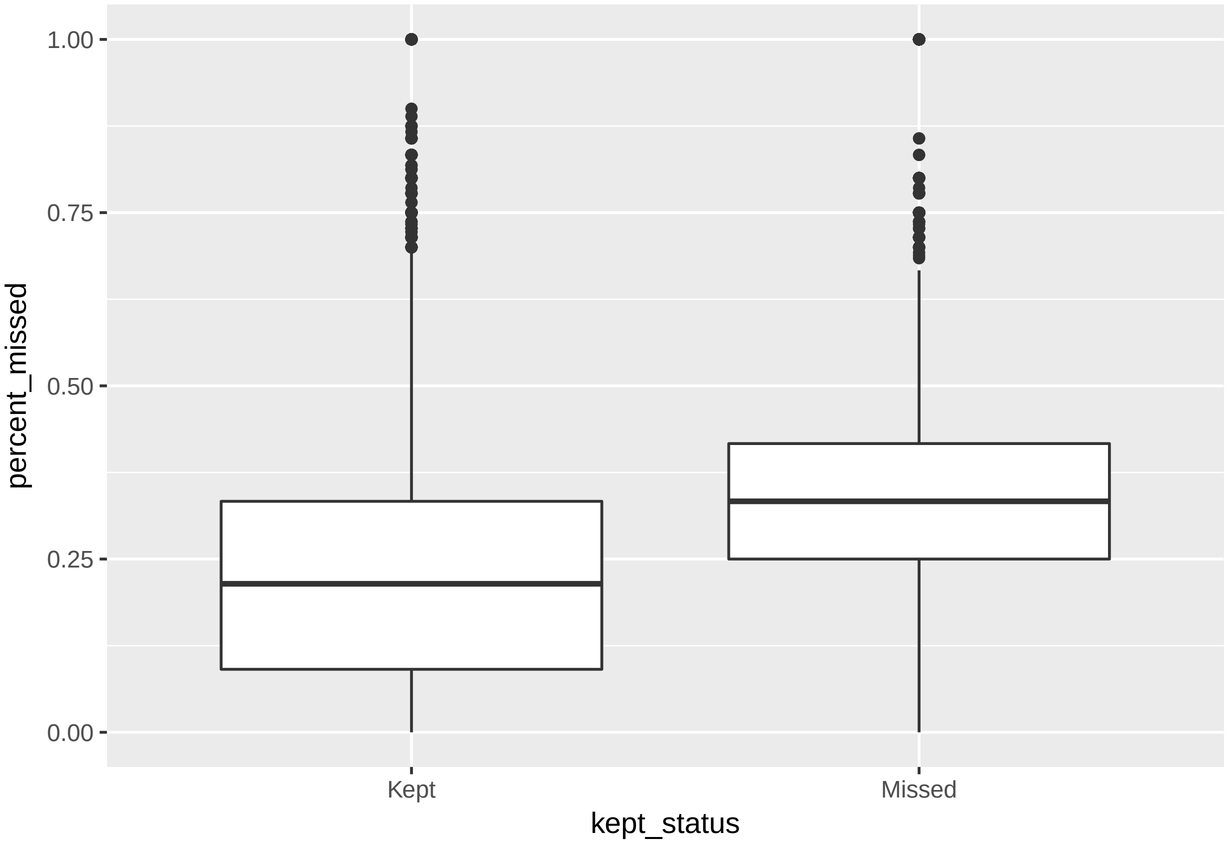
```
appointments_3 <- dplyr::left_join(appointments_3, zipcodes, by = "office_zip")
```

percent_missed

Create random subset and plot

```
ggplot(data = appointments_3, aes(x = kept_status, y = percent_missed)) +
    geom_boxplot()
```

```
## Warning: Removed 22338 rows containing non-finite values (stat_boxplot).
```

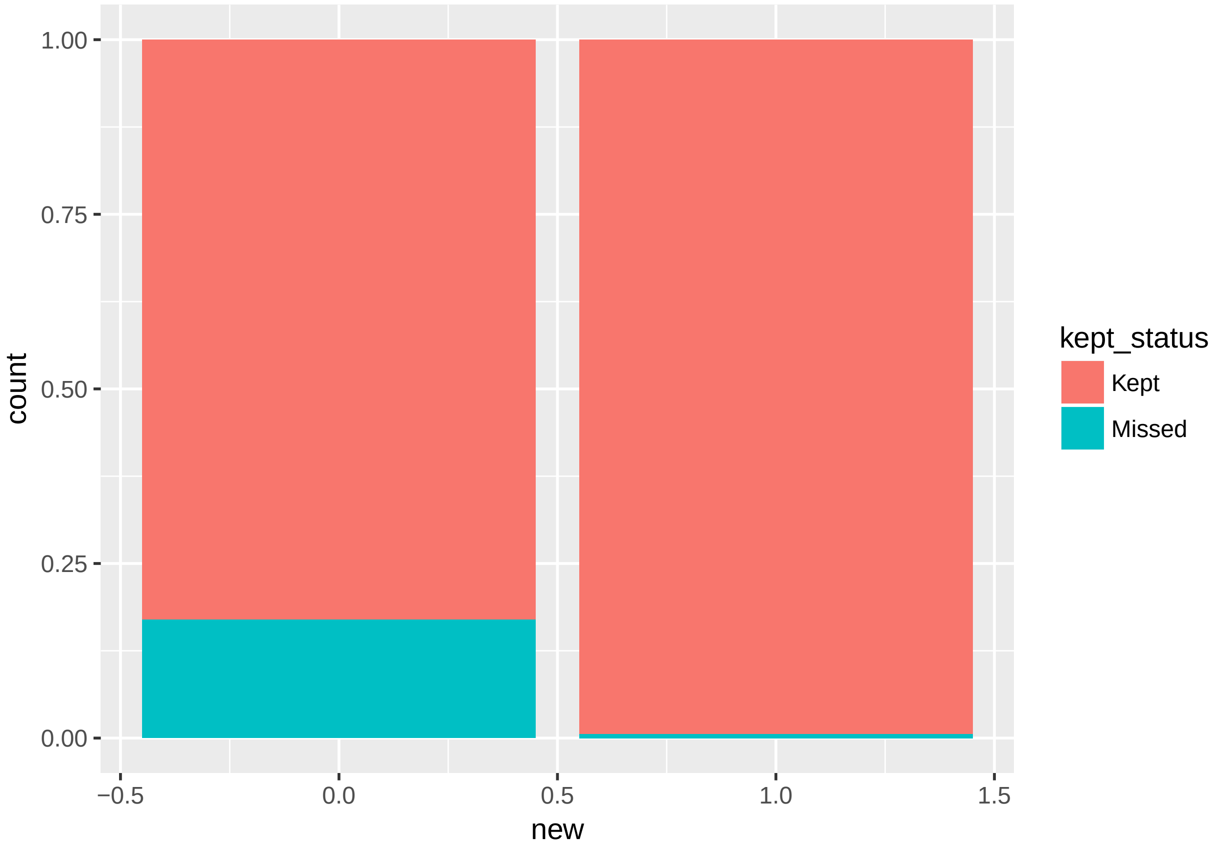


new

```
table(appointments_3$new)
```

```
##
##      0      1
## 320426  22338
```

```
ggplot(data = appointments_3) +
    geom_bar(mapping = aes(x = new, fill = kept_status), position = "fill")
```

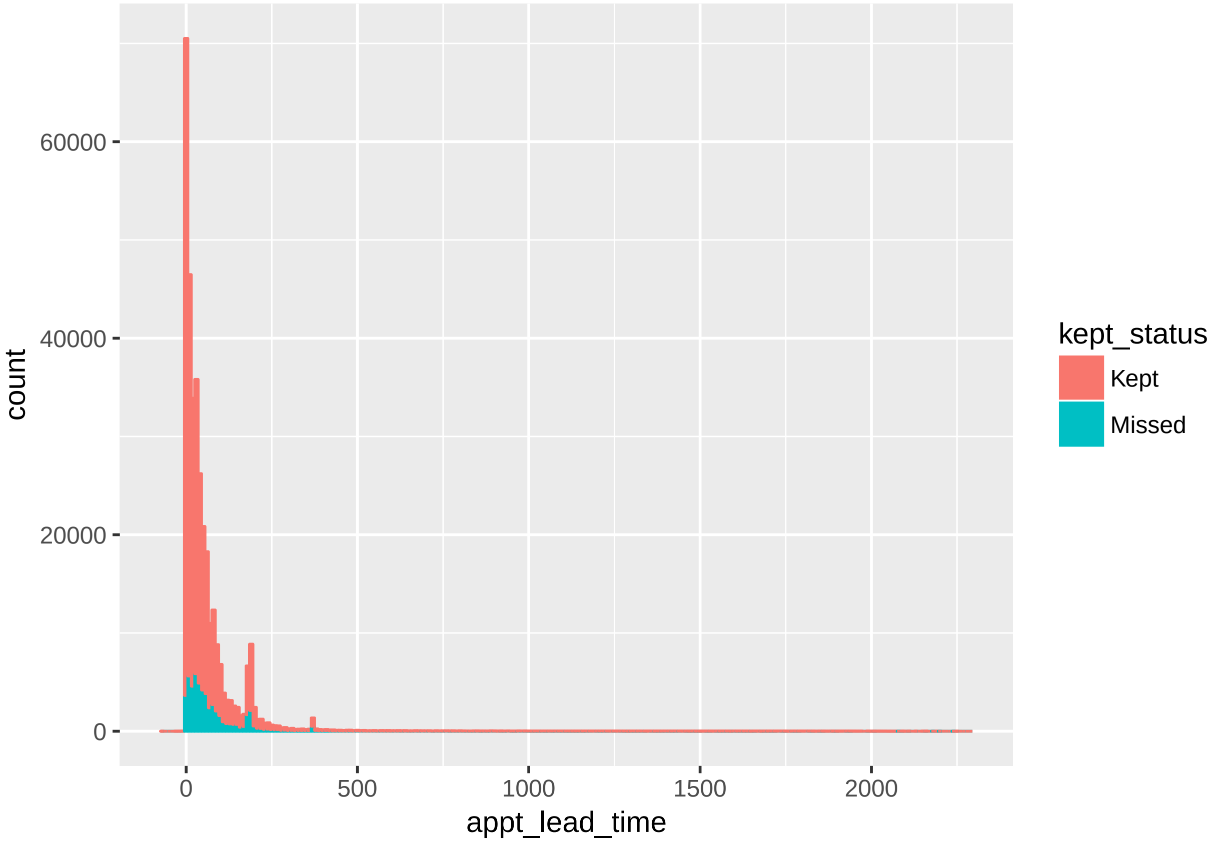


New patients have a very high percentage of kept appointments. 22k of 342k appointments are first-time, or about 6.4%

appt_lead_time

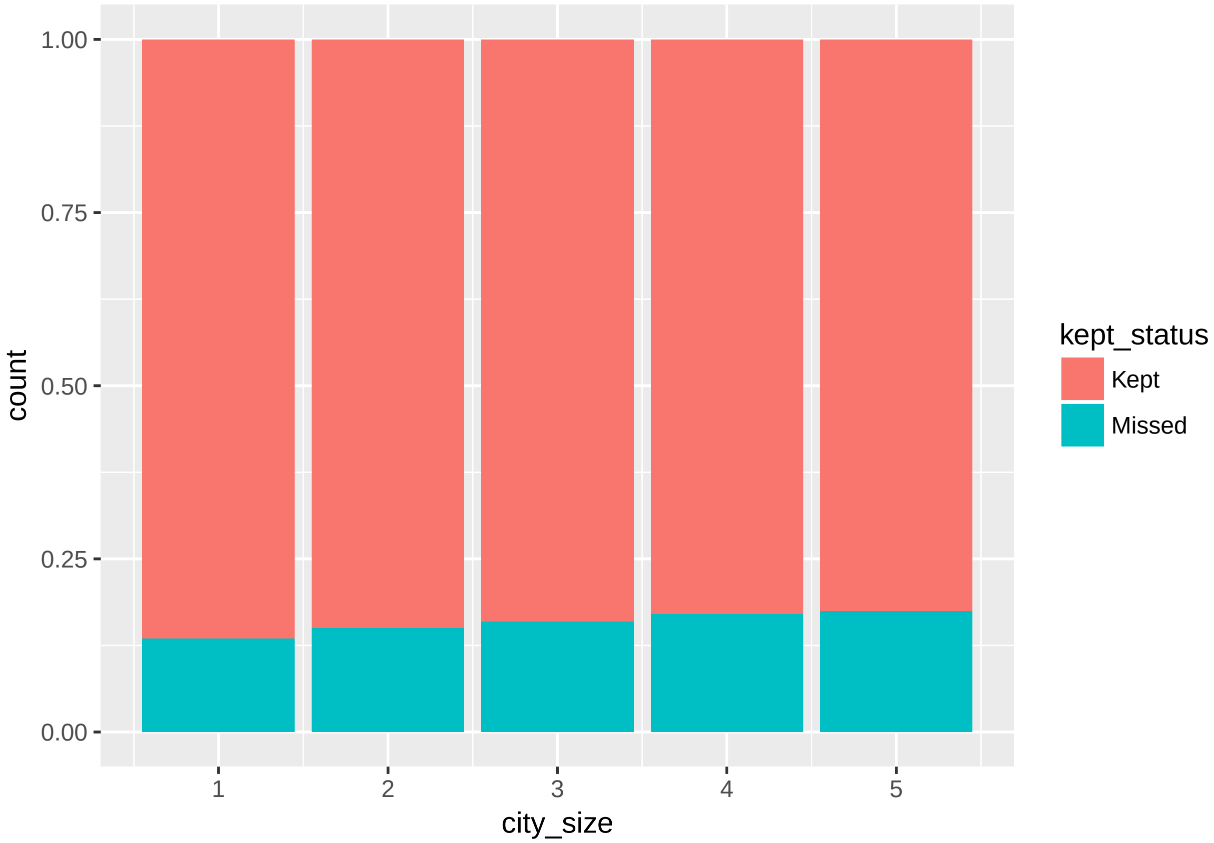
```
ggplot(appointments_3, aes(x = appt_lead_time, col = kept_status, fill = kept_status)) +
    geom_histogram(binwidth = 10)
```

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



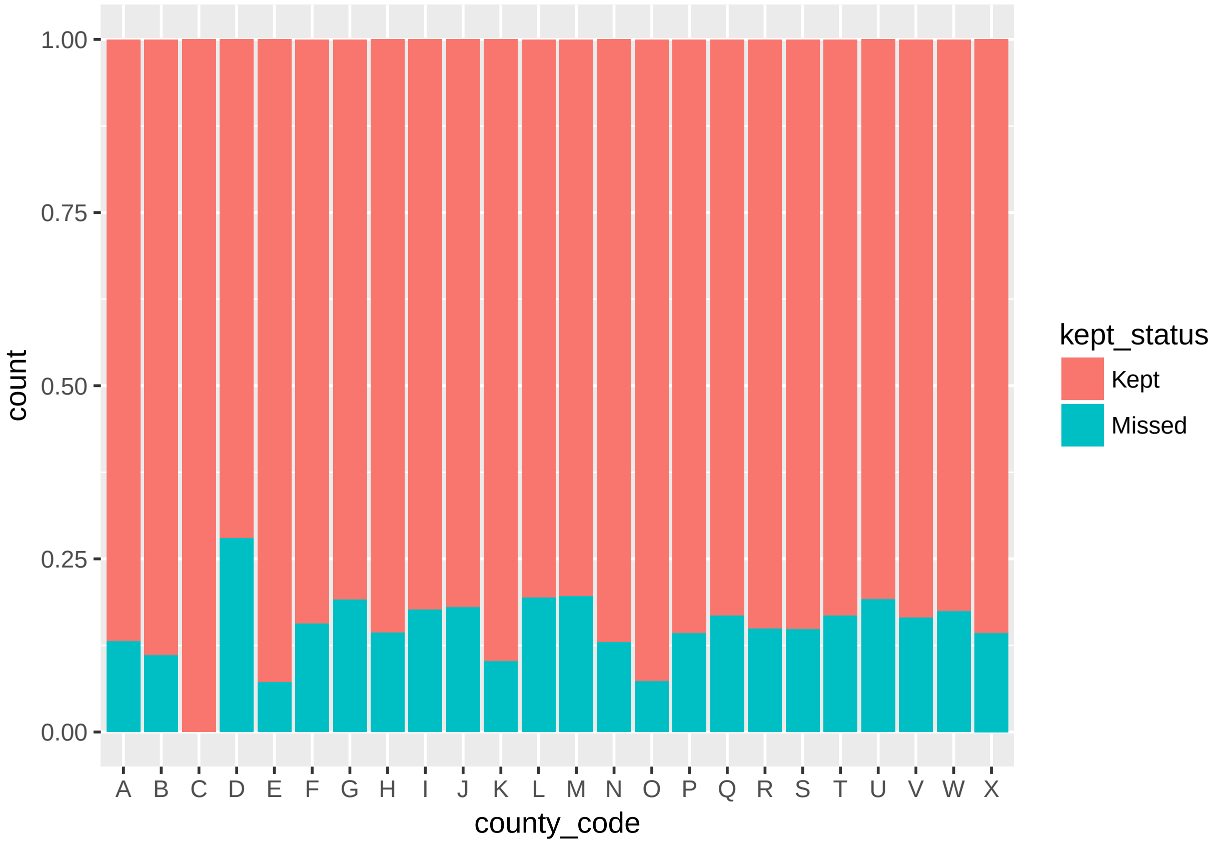
city__size

```
ggplot(data = appointments_3) +
    geom_bar(mapping = aes(x = city_size, fill = kept_status), position = "fill")
```



county_code

```
ggplot(data = appointments_3) +
    geom_bar(mapping = aes(x = county_code, fill = kept_status), position = "fill")
```



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

Modeling

Create Modeling Data

```
model_data <- appointments_3 #%>%
model_data$new <- as.factor(model_data$new)
model_data$percent_missed <- as.integer(model_data$percent_missed * 100)
#Replace NAs with mean
model_data$percent_missed <- model_data$percent_missed %>%
    tidyr::replace_na(mean(model_data$percent_missed, na.rm = TRUE))
factor_columns <- c("kept_status", "patient_gender", "billing_type",
"office_zip", "provider_specialty", "remind_call_result", "hour", "weekday",
"county_code", "length_group")
model_data[factor_columns] <- lapply(model_data[factor_columns], factor)
#Check for NAs
sapply(model_data, function(x) sum(is.na(x)))
```

```
##        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                  0                  0
## provider_specialty remind_call_result      appt_datetime
##                  0                  0                  0
##             missed            age_cat               hour
##                  0                  0                  0
##       length_group     percent_missed                new
##                  0                  0                  0
##     appt_lead_time            weekday        county_code
##                  0                  0                  0
##          city_size
##                  0
```

```
str(model_data)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    342764 obs. of  25 variables:
##  $ kept_status       : Factor w/ 2 levels "Kept","Missed": 1 1 1 1 2 1 1 1 1 1 ...
##  $ appt_date         : chr  "9/1/16" "9/1/16" "9/1/16" "9/1/16" ...
##  $ appt_time         :Classes 'hms', 'difftime'  atomic [1:342764] 19800 28800 28800 28800 28800 2880
##   .. ..- attr(*, "units")= chr "secs"
##  $ appt_length       : int  90 60 120 60 60 60 60 60 60 90 ...
##  $ date_scheduled    : POSIXct, format: "2016-08-01" "2016-01-18" ...
##  $ patient_age       : int  7 75 31 45 49 71 49 38 36 13 ...
##  $ patient_gender    : Factor w/ 4 levels "Female","Male",..: 2 1 2 2 2 2 2 1 2 2 ...
##  $ billing_type      : Factor w/ 2 levels "Commercial","DMAP": 2 1 2 2 1 2 1 1 2 2 ...
##  $ 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  : num  41 29 5 5 0 5 0 539 0 4 ...
##  $ office_zip        : Factor w/ 50 levels "AA","AB","AC",..: 16 38 38 38 38 38 38 38 45 34 ...
##  $ provider_specialty: Factor w/ 7 levels "A","B","C","D",..: 1 1 1 2 2 1 1 1 2 1 ...
##  $ remind_call_result: Factor w/ 9 levels "Answered - Canceled",..: 7 2 7 3 3 2 7 9 9 3 ...
##  $ appt_datetime     : POSIXct, format: "2016-09-01 05:30:00" "2016-09-01 08:00:00" ...
##  $ missed            : num  0 0 0 0 1 0 0 0 0 0 ...
##  $ age_cat           : Factor w/ 8 levels "0-10","10-20",..: 1 8 4 5 5 8 5 4 4 2 ...
##  $ hour              : Factor w/ 18 levels "0","5","6","7",..: 2 5 5 5 5 5 5 5 5 5 ...
##  $ length_group      : Factor w/ 3 levels "Short","Medium",..: 3 2 3 2 2 2 2 2 2 3 ...
##  $ percent_missed    : num  25 28 16 28 45 ...
##  $ new               : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 2 1 1 ...
##  $ appt_lead_time    :Class 'difftime'  atomic [1:342764] 31 227 211 85 65 51 49 0 2 664 ...
##   .. ..- attr(*, "units")= chr "days"
##  $ weekday           : Factor w/ 6 levels "Friday","Monday",..: 6 4 4 4 4 4 4 4 4 4 ...
##  $ county_code       : Factor w/ 24 levels "A","B","C","D",..: 16 9 9 9 9 9 9 9 20 12 ...
##  $ city_size         : int  2 4 4 4 4 4 4 4 4 4 ...
```

Divide model_data into train, validate, and test sets

```
train <- model_data[1:205660,]
validate <- model_data[205660:274200,]
test <- model_data[274201:nrow(model_data),]

table(train$kept_status)
```

```
##
##   Kept Missed
## 174601  31059
```

```
train2 <- train[168738:205660,]
table(train2$keptstatus)
```

```
## Warning: Unknown or uninitialised column: 'keptstatus'.
```

```
## < table of extent 0 >
```

```
train_kept <- train2[train2$kept_status == "Kept",]
train_missed <- train[train$kept_status == "Missed",]

train_balanced <- rbind(train_kept, train_missed)
table(train_balanced$kept_status)
```

```
##
##   Kept Missed
##  31059  31059
```

Logistic Regression Model

```

model1 <- caret::train(kept_status ~ age + remindresult + specialty + billtype + hour + percent_missed +
model1$finalModel
confusionMatrix(model1)
##p_glm <- predict(glm, train)
#caret::confusionMatrix(p_glm, train$kept_status)
```

Random Forest Model

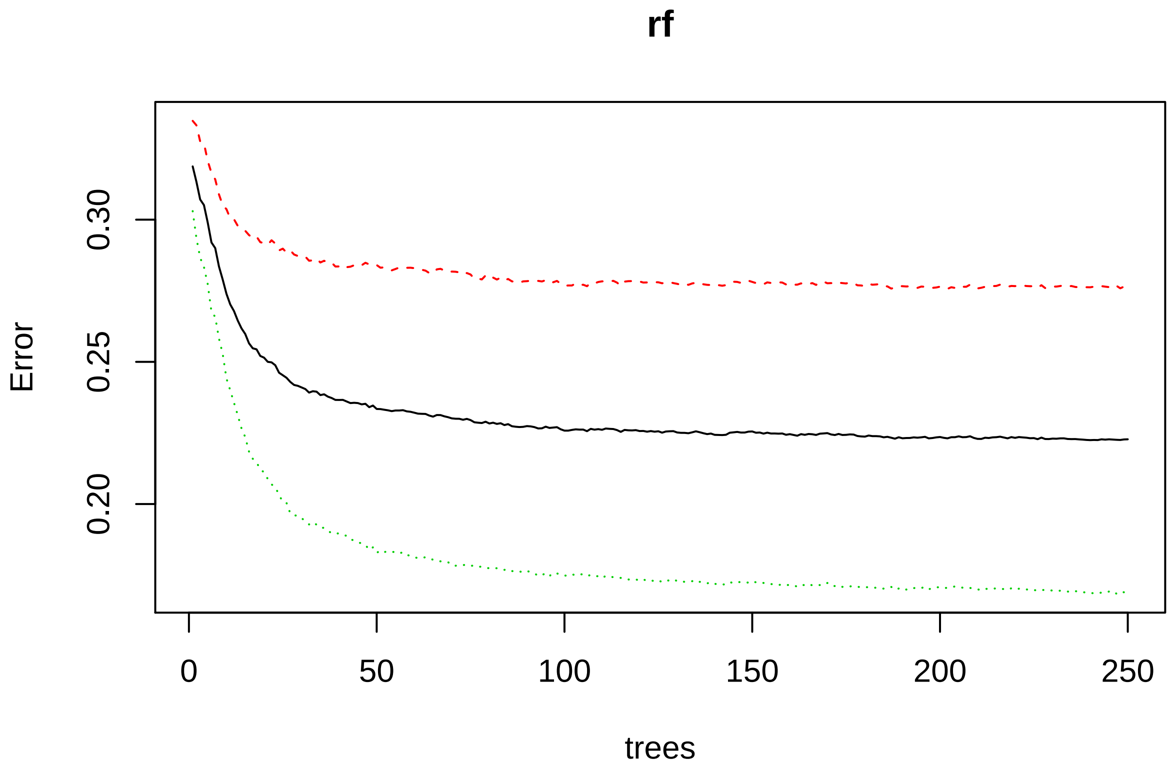
Using randomForest Package

```
rf <- randomForest(kept_status ~ patient_age + remind_call_result + provider_specialty + billing_type +
#Takes about 30 seconds to run
```

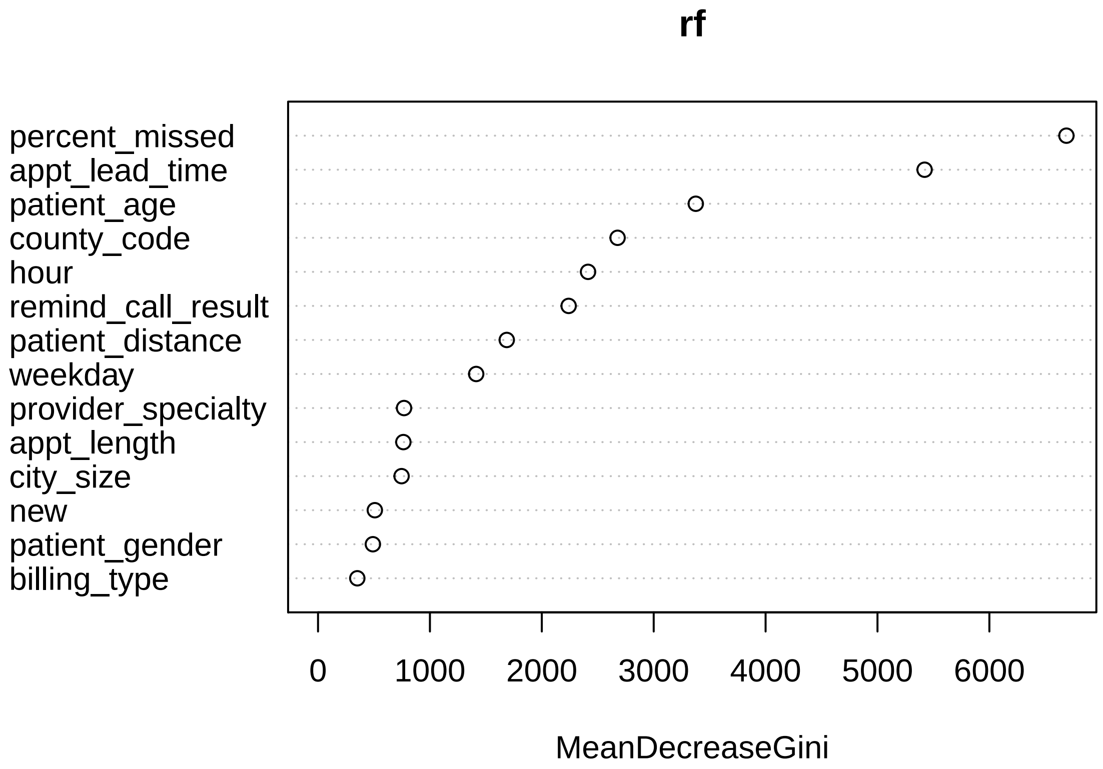
```
print(rf)
```

```
##
## Call:
##  randomForest(formula = kept_status ~ patient_age + remind_call_result +      provider_specialty + bi
##                Type of random forest: classification
##                      Number of trees: 250
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 22.28%
## Confusion matrix:
##         Kept Missed class.error
## Kept   22465   8594   0.2766992
## Missed  5244  25815   0.1688399
```

```
plot(rf)
```



```
varImpPlot(rf)
```



Using caret Package

```
# Look at number of cvs and repeats for faster run-time
control <- caret::trainControl(method = "cv", number = 3, allowParallel = TRUE)
```

```r
seed <- 7
metric <- "Accuracy"
set.seed(seed)
mtry <- 3
tunegrid <- expand.grid(.mtry = mtry)

#Train on subset to see how long it will take.  Takes ~ 1.5 hours
```

```r
rftrain <- caret::train(kept_status ~ patient_age + remind_call_result + provider_specialty + billing_ty
```

```r
caret::confusionMatrix(rftrain)
```

```r
control <- caret::trainControl(method = "oob", number = 10, repeats = 3, search = "random")
```

```
## Warning: `repeats` has no meaning for this resampling method.
```

```r
seed <- 7
metric <- "Accuracy"
set.seed(seed)
mtry <- 3
tunegrid <- expand.grid(.mtry = mtry)
```

```r
### Below code takes a long time to run, need to consider ways to shorten it
rftrain3 <- caret::train(keptstatus ~ age + remindresult + specialty +
                         billtype + hour + percent_missed + length + gender +
                         distance + new + leadtime + weekday + county,
                         data = train_balanced, method = "rf", metric = metric,
                         tuneLength = 15, trControl = control)
```

```r
print(rftrain2)
plot(rf)
varImpPlot(rf)
varUsed(rf)
p_rf <- predict(rf, test)
caret::confusionMatrix(p_rf, test$keptstatus)
```