Capstone

Read data and assign to *appointments*

appointments <- read.csv("Final\_Data.csv")  
zipcodes <- read.csv("zipcodes.csv")

### Data Summary and Structure

## keptstatus date time length   
## Kept :288246 4/11/17: 1300 13:00:00: 40627 Min. : 10   
## Missed: 54616 4/13/17: 1300 8:00:00 : 36476 1st Qu.: 60   
## 4/18/17: 1282 10:00:00: 35541 Median : 60   
## 4/19/17: 1282 11:00:00: 34899 Mean : 57   
## 4/12/17: 1280 14:00:00: 34440 3rd Qu.: 60   
## 5/2/17 : 1272 9:00:00 : 33510 Max. :600   
## (Other):335146 (Other) :127369   
## datesched age gender   
## 7/11/17: 1588 Min. : 0.00 Female :190399   
## 5/1/17 : 1327 1st Qu.: 17.00 Male :152061   
## 5/2/17 : 1288 Median : 34.00 Other : 76   
## 5/3/17 : 1236 Mean : 35.56 Unknown: 326   
## 5/8/17 : 1202 3rd Qu.: 54.00   
## 5/30/17: 1198 Max. :264.00   
## (Other):335023   
## billtype priormiss priorkept   
## Commercial : 78296 Min. : 0.000 Min. : 0.00   
## DMAP :264565 1st Qu.: 1.000 1st Qu.: 2.00   
## To Be Assigned: 1 Median : 2.000 Median : 6.00   
## Mean : 2.451 Mean : 8.02   
## 3rd Qu.: 3.000 3rd Qu.: 11.00   
## Max. :117.000 Max. :676.00   
##   
## distance zip specialty   
## Min. : 0.0 AP : 29129 A:246965   
## 1st Qu.: 0.0 BL : 22388 B: 84147   
## Median : 3.0 BT : 17440 C: 5623   
## Mean : 10.8 BH : 13337 D: 5512   
## 3rd Qu.: 9.0 AF : 10887 E: 42   
## Max. :2688.0 AJ : 10816 F: 525   
## NA's :974 (Other):238865 G: 48   
## remindresult   
## Answered - No Response:180874   
## Not Called : 63501   
## Answered - Confirmed : 49108   
## Failed : 27944   
## Left Message : 18433   
## Answered - Reschedule : 1369   
## (Other) : 1633

## 'data.frame': 342862 obs. of 14 variables:  
## $ keptstatus : Factor w/ 2 levels "Kept","Missed": 1 1 1 1 2 1 1 1 1 1 ...  
## $ date : Factor w/ 361 levels "1/10/17","1/10/18",..: 320 320 320 320 320 320 320 320 320 320 ...  
## $ time : Factor w/ 89 levels "0:00:00","10:00:00",..: 65 78 78 78 78 78 78 78 78 78 ...  
## $ length : int 90 60 120 60 60 60 60 60 60 90 ...  
## $ datesched : Factor w/ 1475 levels "1/10/12","1/10/13",..: 1242 39 618 1120 1089 1141 1148 1363 1332 387 ...  
## $ age : int 7 75 31 45 49 71 49 38 36 13 ...  
## $ gender : Factor w/ 4 levels "Female","Male",..: 2 1 2 2 2 2 2 1 2 2 ...  
## $ billtype : Factor w/ 3 levels "Commercial","DMAP",..: 2 1 2 2 1 2 1 1 2 2 ...  
## $ priormiss : int 1 2 1 6 5 6 8 0 2 3 ...  
## $ priorkept : int 3 5 5 15 6 6 20 0 5 12 ...  
## $ distance : int 41 29 5 5 0 5 0 539 0 4 ...  
## $ zip : Factor w/ 50 levels "AA","AB","AC",..: 16 38 38 38 38 38 38 38 45 34 ...  
## $ specialty : Factor w/ 7 levels "A","B","C","D",..: 1 1 1 2 2 1 1 1 2 1 ...  
## $ remindresult: Factor w/ 9 levels "Answered - Canceled",..: 7 2 7 3 3 2 7 9 9 3 ...

Convert time and date variables to POSIXct Convert zip to character

appointments$time <- as.POSIXct(appointments$time, format = "%H:%M:%S")  
appointments$date <- as.POSIXct(appointments$date, format = "%m/%d/%y")  
appointments$datesched <- as.POSIXct(appointments$datesched,   
 format = "%m/%d/%y")  
###appointments$zip <- as.character(appointments$zip)

Calculating percent of missed appointments overall

missed\_rate <- length(which(appointments$keptstatus == "Missed")) /  
 length(appointments$keptstatus)  
missed\_rate

## [1] 0.1592944

### Build Data Dictionary using dataMeta

var\_desc <- 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",  
 "Reminder Call result")  
var\_type <- c(1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1)  
linker <- dataMeta::build\_linker(appointments, variable\_description = var\_desc,  
 variable\_type = var\_type)  
dictionary <- dataMeta::build\_dict(my.data = appointments, linker = linker,  
 option\_description = NULL,  
 prompt\_varopts = FALSE)  
knitr::kable(dictionary, format = "html",  
 caption = "Data Dictionary for Original Dataset")

Data Dictionary for Original Dataset

variable\_name

variable\_description

variable\_options

age

Patient Age

0 to 264

billtype

Billing Type

DMAP

Commercial

To Be Assigned

date

Appointment Date

2016-09-01 to 2018-01-31

datesched

Date Appointment was scheduled

2011-10-18 to 2018-01-31

distance

Patient Distance From Office in Miles

NA to NA

gender

Patient Gender

Male

Female

Other

Unknown

keptstatus

Dependent Variable Kept or Missed

Kept

Missed

length

Appointment Length in Minutes

10 to 600

priorkept

Number of prior kept appointments

0 to 676

priormiss

Number of prior missed appointments

0 to 117

remindresult

Reminder Call result

Left Message

Answered - Confirmed

Answered - No Response

Not Called

Failed

Answered - Reschedule

No Answer

Busy

Answered - Canceled

specialty

Provider Primary Specialty

A

B

G

C

D

E

F

time

Appointment Time

2018-03-15 00:00:00 to 2018-03-15 21:00:00

zip

Office Zip Code - Anonymized

AP

BL

BS

BH

BC

BT

BR

AJ

AO

BU

AL

BB

BQ

AW

AF

BI

AZ

BO

AG

AV

BN

BK

BM

BE

BF

AS

AT

BV

AI

AU

AK

AN

AM

AD

BA

AB

AR

BG

AY

AX

BX

AQ

AH

BJ

BP

AC

AA

BD

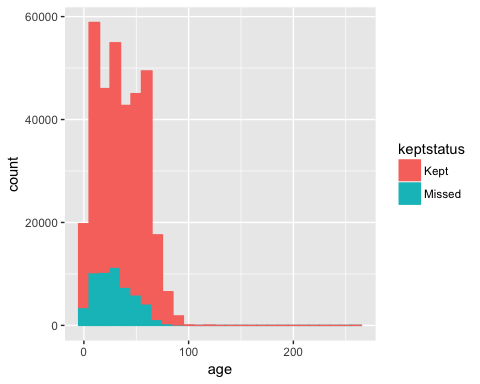
BW

AE

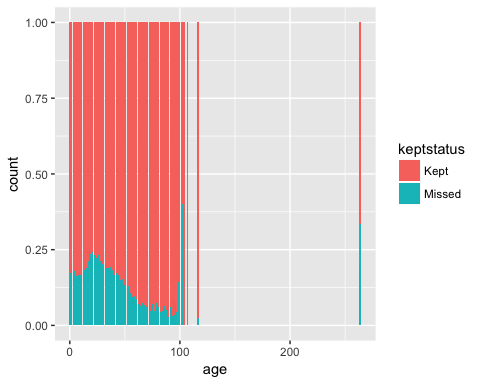
### Data Exploration

#### age

ggplot(appointments, aes(x = age, group = keptstatus, col = keptstatus,  
 fill = keptstatus)) +  
 geom\_histogram(binwidth = 10)



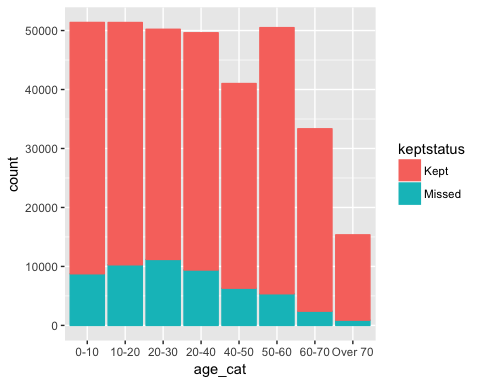
ggplot(data = appointments) +  
 geom\_bar(mapping = aes(x = age, fill = keptstatus), position = "fill")



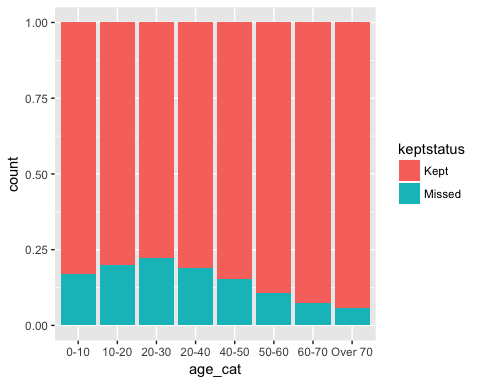
Ranges from 0-264, so there are obviously some impossible values. Ratio of missed appointments decreases with age in general.

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

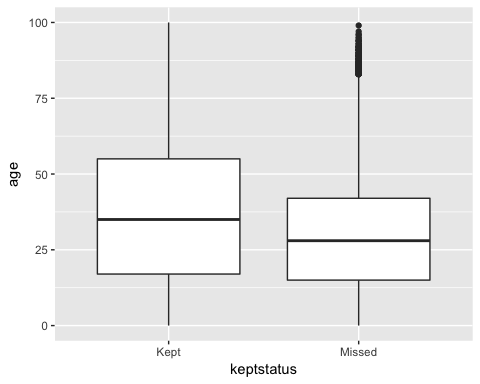
appointments\_2 <- appointments %>%  
 filter(age <= 100)  
  
appointments\_2 <- appointments\_2 %>%  
 mutate(age\_cat = cut(age, breaks = c(-1, 10, 20, 30, 40, 50, 60, 70, 101),  
 labels = c("0-10", "10-20", "20-30", "20-40", "40-50",  
 "50-60", "60-70", "Over 70")))  
  
  
ggplot(appointments\_2, aes(x = age\_cat, group = keptstatus, col = keptstatus,  
 fill = keptstatus)) +  
 stat\_count()



ggplot(data = appointments\_2) +  
 geom\_bar(mapping = aes(x = age\_cat, fill = keptstatus), position = "fill")



ggplot(data = appointments\_2, aes(x = keptstatus, y = age)) +  
 geom\_boxplot()

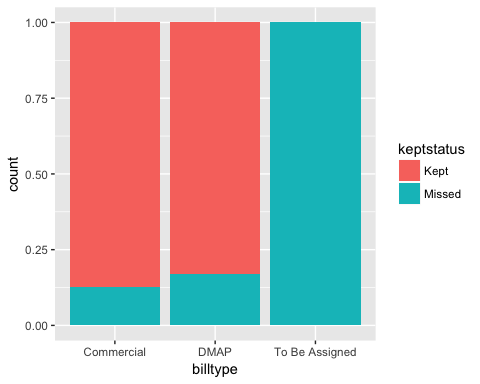


#### billtype

table(appointments\_2$billtype)

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

ggplot(data = appointments\_2) +  
 geom\_bar(mapping = aes(x = billtype, fill = keptstatus), position = "fill")

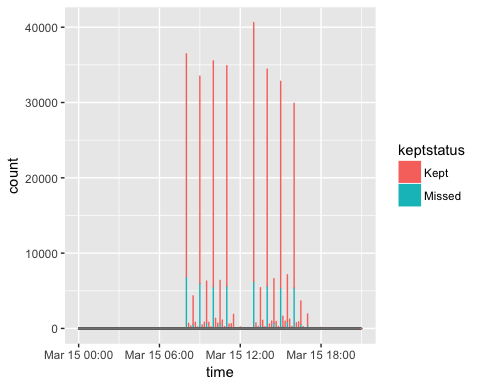


Only one row has *To Be Assigned* value and will just be removed There is a minor difference between billing types

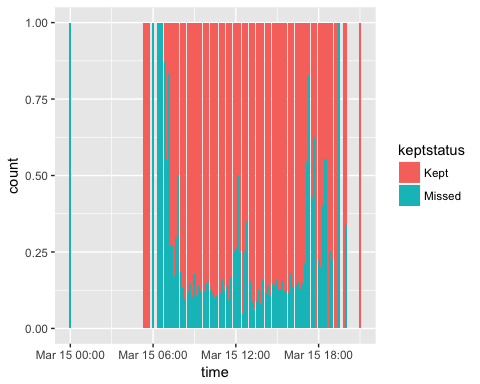
appointments\_2 <- subset(appointments\_2,  
 appointments\_2$billtype != "To Be Assigned")

#### time

ggplot(data = appointments\_2,  
 aes(x = time, group = keptstatus, col = keptstatus, fill = keptstatus)) +  
 geom\_histogram(binwidth = 1)

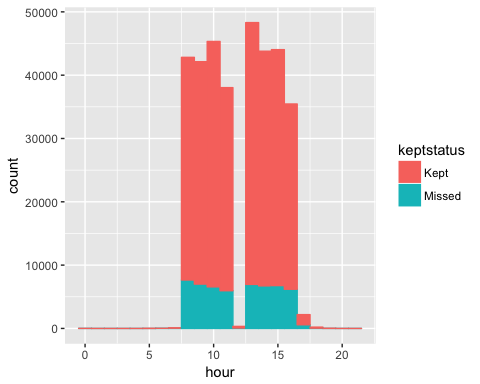


ggplot(data = appointments\_2) +  
 geom\_bar(mapping = aes(x = time, fill = keptstatus), position = "fill")

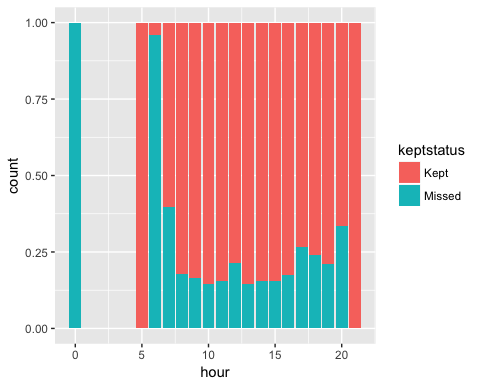


Creating new *hour* variable and replotting

appointments\_2 <- appointments\_2 %>%  
 mutate(hour = lubridate::hour(appointments\_2$time))  
   
ggplot(data = appointments\_2,   
 aes(x = hour, group = keptstatus, col = keptstatus, fill = keptstatus)) +  
 geom\_histogram(binwidth = 1)



ggplot(data = appointments\_2) +  
 geom\_bar(mapping = aes(x = hour, fill = keptstatus), 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 minor.

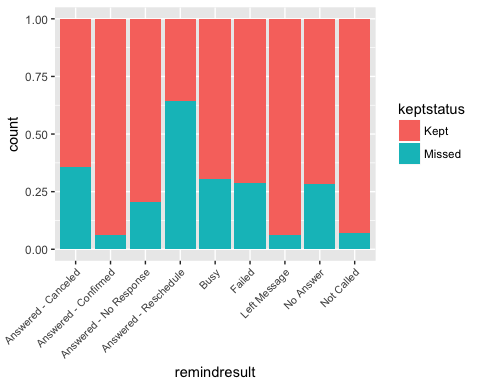
#### remindresult

table(appointments\_2$remindresult)

##   
## 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”

ggplot(data = appointments\_2) +  
 geom\_bar(aes(x = remindresult, fill = keptstatus), position = "fill") +  
 theme(axis.text.x = element\_text(size = 8, angle = 45,  
 hjust = 1, vjust = 1))



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

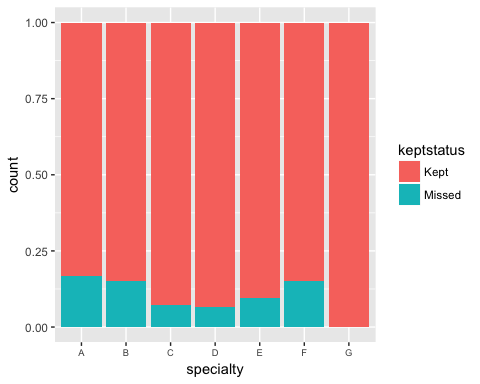
#### specialty

table(appointments\_2$specialty)

##   
## A B C D E F G   
## 246904 84114 5619 5512 42 525 48

48 missing values. E and F have few observations.

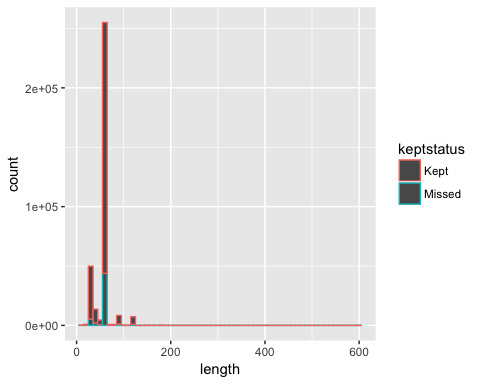
ggplot(data = appointments\_2) +  
 geom\_bar(aes(x = specialty, fill = keptstatus), position = "fill") +  
 theme(axis.text.x = element\_text(size = 7))



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

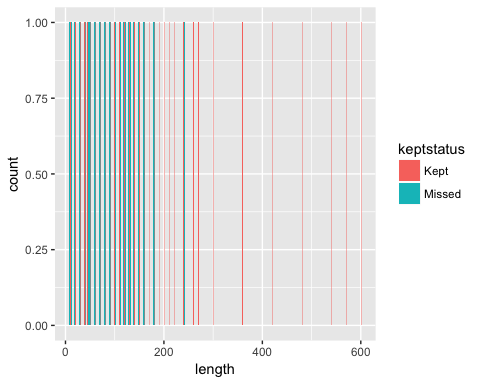
#### length

ggplot(appointments\_2, aes(x = length, group = keptstatus, col = keptstatus)) +  
 geom\_histogram(binwidth = 10)

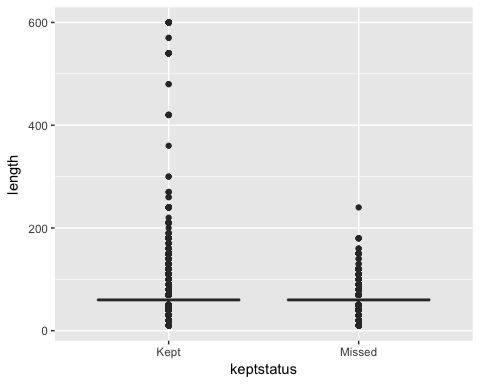


ggplot(data = appointments\_2) +  
 geom\_bar(mapping = aes(x = length, fill = keptstatus), position = "fill")

## Warning: position\_stack requires non-overlapping x intervals



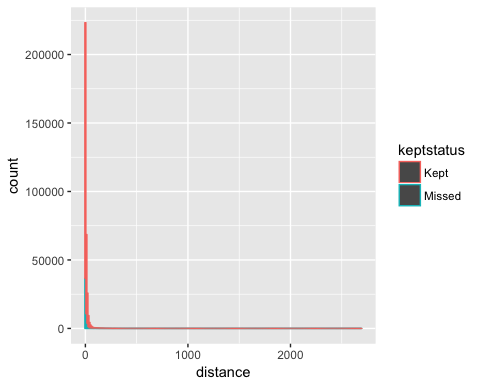
ggplot(data = appointments\_2, aes(x = keptstatus, y = length)) +  
 geom\_boxplot()



#### distance

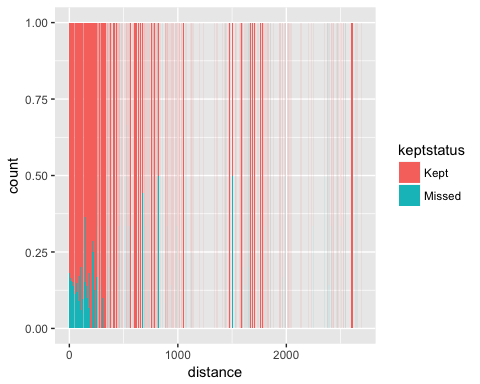
ggplot(appointments\_2, aes(x = distance, group = keptstatus, col = keptstatus)) +  
 geom\_histogram(binwidth = 10)

## Warning: Removed 972 rows containing non-finite values (stat\_bin).



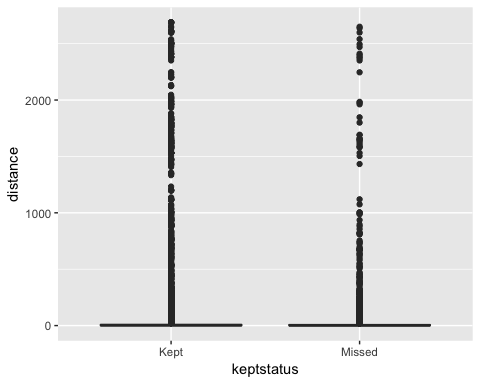
ggplot(data = appointments\_2) +  
 geom\_bar(mapping = aes(x = distance, fill = keptstatus), position = "fill")

## Warning: Removed 972 rows containing non-finite values (stat\_count).



ggplot(data = appointments\_2, aes(x = keptstatus, y = distance)) +  
 geom\_boxplot()

## Warning: Removed 972 rows containing non-finite values (stat\_boxplot).



Create new variables

percent\_missed = percent of prior appointments missed.

appointments\_3 <- appointments\_2 %>%  
 mutate(missed = ifelse(appointments\_2$keptstatus == "Missed", 1,0)) %>%  
 mutate(percent\_missed = priormiss / (priormiss + priorkept)) %>%  
 mutate(new = ifelse(appointments\_2$priormiss == 0 & appointments\_2$priorkept == 0, 1, 0)) %>%  
 mutate(leadtime = as.integer((date - datesched)/86400)) %>% #results are in seconds. Should convert to days for simplicity  
 mutate(weekday = as.factor(strftime(date, "%A")))  
#remove zipcodes$zip <- as.character(zipcodes$zip)  
appointments\_3 <- dplyr::left\_join(appointments\_3, zipcodes, by = "zip")  
str(appointments\_3)

## 'data.frame': 342764 obs. of 23 variables:  
## $ keptstatus : Factor w/ 2 levels "Kept","Missed": 1 1 1 1 2 1 1 1 1 1 ...  
## $ date : POSIXct, format: "2016-09-01" "2016-09-01" ...  
## $ time : POSIXct, format: "2018-03-15 05:30:00" "2018-03-15 08:00:00" ...  
## $ length : int 90 60 120 60 60 60 60 60 60 90 ...  
## $ datesched : POSIXct, format: "2016-08-01" "2016-01-18" ...  
## $ age : int 7 75 31 45 49 71 49 38 36 13 ...  
## $ gender : Factor w/ 4 levels "Female","Male",..: 2 1 2 2 2 2 2 1 2 2 ...  
## $ billtype : Factor w/ 3 levels "Commercial","DMAP",..: 2 1 2 2 1 2 1 1 2 2 ...  
## $ priormiss : int 1 2 1 6 5 6 8 0 2 3 ...  
## $ priorkept : int 3 5 5 15 6 6 20 0 5 12 ...  
## $ distance : int 41 29 5 5 0 5 0 539 0 4 ...  
## $ zip : Factor w/ 50 levels "AA","AB","AC",..: 16 38 38 38 38 38 38 38 45 34 ...  
## $ specialty : Factor w/ 7 levels "A","B","C","D",..: 1 1 1 2 2 1 1 1 2 1 ...  
## $ remindresult : Factor w/ 9 levels "Answered - Canceled",..: 7 2 7 3 3 2 7 9 9 3 ...  
## $ age\_cat : Factor w/ 8 levels "0-10","10-20",..: 1 8 4 5 5 8 5 4 4 2 ...  
## $ hour : int 5 8 8 8 8 8 8 8 8 8 ...  
## $ missed : num 0 0 0 0 1 0 0 0 0 0 ...  
## $ percent\_missed: num 0.25 0.286 0.167 0.286 0.455 ...  
## $ new : num 0 0 0 0 0 0 0 1 0 0 ...  
## $ leadtime : int 31 226 210 85 65 51 49 0 2 663 ...  
## $ weekday : Factor w/ 5 levels "Friday","Monday",..: 3 3 3 3 3 3 3 3 3 3 ...  
## $ county : Factor w/ 24 levels "A","B","C","D",..: 16 9 9 9 9 9 9 9 20 12 ...  
## $ size : int 2 4 4 4 4 4 4 4 4 4 ...

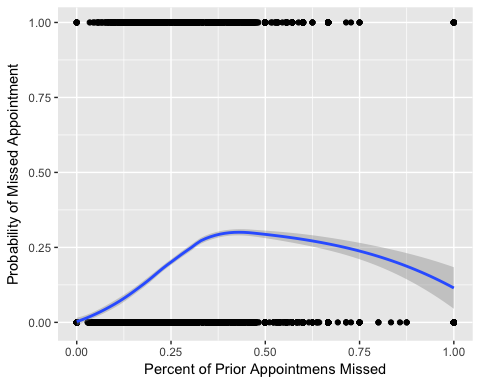
#### percent\_missed

Create random subset and plot

appointments\_sample\_05 <- appointments\_3 %>%  
 sample\_frac(size = 0.05, replace = FALSE)  
  
ggplot(data = appointments\_sample\_05, aes(x = percent\_missed, y = missed)) +  
 geom\_point() +  
 stat\_smooth(method = "loess") +  
 xlab("Percent of Prior Appointmens Missed") +  
 ylab("Probability of Missed Appointment")

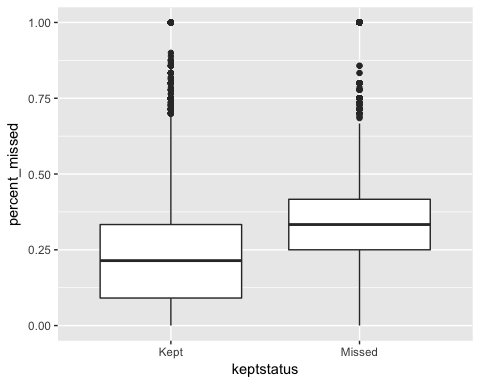
## Warning: Removed 1032 rows containing non-finite values (stat\_smooth).

## Warning: Removed 1032 rows containing missing values (geom\_point).



ggplot(data = appointments\_3, aes(x = keptstatus, y = percent\_missed)) +  
 geom\_boxplot()

## Warning: Removed 22338 rows containing non-finite values (stat\_boxplot).

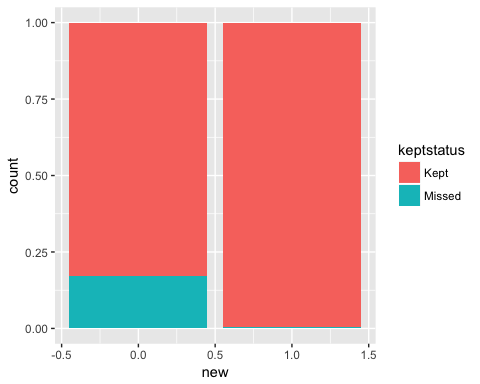


#### New

table(appointments\_3$new)

##   
## 0 1   
## 320426 22338

ggplot(data = appointments\_3) +  
 geom\_bar(mapping = aes(x = new, fill = keptstatus), position = "fill")



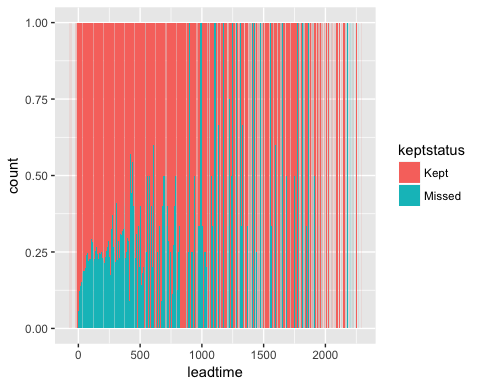
New patients have a very high percentage of kept appointments. 22k of 342k appointments are first-time, or about 6.4%

#### Time Differential

table(appointments\_3$new)

##   
## 0 1   
## 320426 22338

ggplot(data = appointments\_3) +  
 geom\_bar(mapping = aes(x = leadtime, fill = keptstatus), position = "fill")

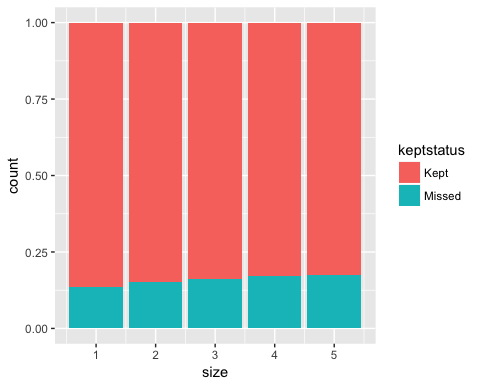


#### Size

table(appointments\_3$new)

##   
## 0 1   
## 320426 22338

ggplot(data = appointments\_3) +  
 geom\_bar(mapping = aes(x = size, fill = keptstatus), position = "fill")

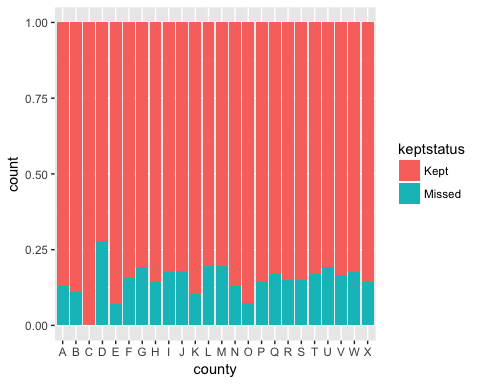


#### County

table(appointments\_3$new)

##   
## 0 1   
## 320426 22338

ggplot(data = appointments\_3) +  
 geom\_bar(mapping = aes(x = county, fill = keptstatus), position = "fill")



### Modeling

Create Modeling Data

mod\_data <- appointments\_3 #%>%  
mod\_data$new <- as.factor(mod\_data$new)  
mod\_data$percent\_missed <- as.integer(mod\_data$percent\_missed \* 100)  
#Replace NAs with mean  
mod\_data$percent\_missed <- mod\_data$percent\_missed %>%  
 tidyr::replace\_na(mean(mod\_data$percent\_missed, na.rm = TRUE))  
  
#Check for NAs  
sapply(mod\_data, function(x) sum(is.na(x)))

## keptstatus date time length datesched   
## 0 0 0 0 0   
## age gender billtype priormiss priorkept   
## 0 0 0 0 0   
## distance zip specialty remindresult age\_cat   
## 972 0 0 0 0   
## hour missed percent\_missed new leadtime   
## 0 0 0 0 0   
## weekday county size   
## 0 0 0

## Distance hase 972 NA values, will replace with mean  
mod\_data$distance <- mod\_data$distance %>%  
 tidyr::replace\_na(median(mod\_data$distance, na.rm = TRUE))  
  
str(mod\_data)

## 'data.frame': 342764 obs. of 23 variables:  
## $ keptstatus : Factor w/ 2 levels "Kept","Missed": 1 1 1 1 2 1 1 1 1 1 ...  
## $ date : POSIXct, format: "2016-09-01" "2016-09-01" ...  
## $ time : POSIXct, format: "2018-03-15 05:30:00" "2018-03-15 08:00:00" ...  
## $ length : int 90 60 120 60 60 60 60 60 60 90 ...  
## $ datesched : POSIXct, format: "2016-08-01" "2016-01-18" ...  
## $ age : int 7 75 31 45 49 71 49 38 36 13 ...  
## $ gender : Factor w/ 4 levels "Female","Male",..: 2 1 2 2 2 2 2 1 2 2 ...  
## $ billtype : Factor w/ 3 levels "Commercial","DMAP",..: 2 1 2 2 1 2 1 1 2 2 ...  
## $ priormiss : int 1 2 1 6 5 6 8 0 2 3 ...  
## $ priorkept : int 3 5 5 15 6 6 20 0 5 12 ...  
## $ distance : num 41 29 5 5 0 5 0 539 0 4 ...  
## $ zip : Factor w/ 50 levels "AA","AB","AC",..: 16 38 38 38 38 38 38 38 45 34 ...  
## $ specialty : Factor w/ 7 levels "A","B","C","D",..: 1 1 1 2 2 1 1 1 2 1 ...  
## $ remindresult : Factor w/ 9 levels "Answered - Canceled",..: 7 2 7 3 3 2 7 9 9 3 ...  
## $ age\_cat : Factor w/ 8 levels "0-10","10-20",..: 1 8 4 5 5 8 5 4 4 2 ...  
## $ hour : int 5 8 8 8 8 8 8 8 8 8 ...  
## $ missed : num 0 0 0 0 1 0 0 0 0 0 ...  
## $ 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 ...  
## $ leadtime : int 31 226 210 85 65 51 49 0 2 663 ...  
## $ weekday : Factor w/ 5 levels "Friday","Monday",..: 3 3 3 3 3 3 3 3 3 3 ...  
## $ county : Factor w/ 24 levels "A","B","C","D",..: 16 9 9 9 9 9 9 9 20 12 ...  
## $ size : int 2 4 4 4 4 4 4 4 4 4 ...

Divide rf\_sample into train, validate, and test sets

###split <- sample(3, nrow(rf\_sample), replace = TRUE, prob = c(0.60, 0.20, 0.20))  
train <- mod\_data[1:205660,]  
validate <- mod\_data[205660:274200,]  
test <- mod\_data[274201:nrow(mod\_data),]  
  
table(train$keptstatus)

##   
## Kept Missed   
## 174601 31059

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

##   
## Kept Missed   
## 31059 5864

train\_kept <- train2[train2$keptstatus == "Kept",]  
train\_missed <- train[train$keptstatus == "Missed",]  
  
train\_balanced <- rbind(train\_kept, train\_missed)  
table(train\_balanced$keptstatus)

##   
## Kept Missed   
## 31059 31059

### Logistic Regression Model

model1 <- caret::train(keptstatus ~ age + remindresult + specialty + billtype + hour + percent\_missed + length + gender + distance + new + leadtime + weekday + county, data = train\_balanced, method = "glm")  
model1$finalModel  
confusionMatrix(model1)  
##p\_glm <- predict(glm, train)  
#caret::confusionMatrix(p\_glm, train$kept\_status)

### Random Forest Model

Using randomForest Package

rf <- randomForest(keptstatus ~ age + remindresult + specialty + billtype + hour + percent\_missed + length + gender + distance + new + leadtime + weekday + county, data = train\_balanced, ntree = 250)  
#Takes about 30 seconds to run

Using caret Package

# Look at number of cvs and repeats for faster run-time  
control <- caret::trainControl(method = "repeatedcv", number = 10, repeats = 3)  
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

rftrain <- caret::train(keptstatus ~ age + remindresult + specialty + billtype + hour + percent\_missed + length + gender + distance + new + leadtime + weekday + county, data = train\_balanced, method = "rf", metric = metric, tuneGrid = tunegrid, trControl = control)

caret::confusionMatrix(rftrain)

control <- caret::trainControl(method = "repeatedcv", number = 10, repeats = 3, search = "random")  
seed <- 7  
metric <- "Accuracy"  
set.seed(seed)  
mtry <- 3  
tunegrid <- expand.grid(.mtry = mtry)

### 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)

print(rftrain2)  
plot(rf)  
varImpPlot(rf)  
varUsed(rf)  
p\_rf <- predict(rf, test)  
caret::confusionMatrix(p\_rf, test$keptstatus)