options(tidyverse.quiet = TRUE)  
library(tidyverse)  
library(tidymodels)

## ── Attaching packages ────────────────────────────────────── tidymodels 1.0.0 ──

## ✔ broom 1.0.0 ✔ rsample 1.1.0  
## ✔ dials 1.0.0 ✔ tune 1.0.1  
## ✔ infer 1.0.3 ✔ workflows 1.1.0  
## ✔ modeldata 1.0.1 ✔ workflowsets 1.0.0  
## ✔ parsnip 1.0.2 ✔ yardstick 1.1.0  
## ✔ recipes 1.0.1

## ── Conflicts ───────────────────────────────────────── tidymodels\_conflicts() ──  
## ✖ scales::discard() masks purrr::discard()  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ recipes::fixed() masks stringr::fixed()  
## ✖ dplyr::lag() masks stats::lag()  
## ✖ yardstick::spec() masks readr::spec()  
## ✖ recipes::step() masks stats::step()  
## • Search for functions across packages at https://www.tidymodels.org/find/

library(esquisse)  
library(GGally)

## Registered S3 method overwritten by 'GGally':  
## method from   
## +.gg ggplot2

library(ggcorrplot)  
library(gridExtra) #I may need to put plots in a grid for easier viewing

##   
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':  
##   
## combine

library(leaps)  
library(skimr)  
library(glmnet) #for lasso and ridge

## Loading required package: Matrix

##   
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':  
##   
## expand, pack, unpack

## Loaded glmnet 4.1-4

library(rpart) #for classification trees

##   
## Attaching package: 'rpart'

## The following object is masked from 'package:dials':  
##   
## prune

library(rpart.plot) #for plotting trees  
library(RColorBrewer) #better visualization of classification trees  
library(rattle) #better visualization of classification trees

## Loading required package: bitops

##   
## Attaching package: 'bitops'

## The following object is masked from 'package:Matrix':  
##   
## %&%

## Rattle: A free graphical interface for data science with R.  
## Version 5.5.1 Copyright (c) 2006-2021 Togaware Pty Ltd.  
## Type 'rattle()' to shake, rattle, and roll your data.

library(caret) #for easy confusion matrix creation

## Loading required package: lattice

##   
## Attaching package: 'caret'

## The following objects are masked from 'package:yardstick':  
##   
## precision, recall, sensitivity, specificity

## The following object is masked from 'package:purrr':  
##   
## lift

library(ranger)

##   
## Attaching package: 'ranger'

## The following object is masked from 'package:rattle':  
##   
## importance

library(randomForest)

## randomForest 4.7-1.1

## Type rfNews() to see new features/changes/bug fixes.

##   
## Attaching package: 'randomForest'

## The following object is masked from 'package:ranger':  
##   
## importance

## The following object is masked from 'package:rattle':  
##   
## importance

## The following object is masked from 'package:gridExtra':  
##   
## combine

## The following object is masked from 'package:dplyr':  
##   
## combine

## The following object is masked from 'package:ggplot2':  
##   
## margin

library(vip)

##   
## Attaching package: 'vip'

## The following object is masked from 'package:utils':  
##   
## vi

library(e1071) #often needed for various statistical tasks

##   
## Attaching package: 'e1071'

## The following object is masked from 'package:tune':  
##   
## tune

## The following object is masked from 'package:rsample':  
##   
## permutations

## The following object is masked from 'package:parsnip':  
##   
## tune

library(ROCR)

#Read in the dataset

ames\_homesales <- read\_csv("ames\_homesales.csv")

## Rows: 2053 Columns: 81  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (47): MS\_SubClass, MS\_Zoning, Street, Alley, Lot\_Shape, Land\_Contour, Ut...  
## dbl (34): Lot\_Frontage, Lot\_Area, Year\_Built, Year\_Remod\_Add, Mas\_Vnr\_Area, ...  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

#skim(ames\_homesales) #Good news, no missing data  
#summary(ames\_homesales)

ames\_homesales[sapply(ames\_homesales, is.character)] <- lapply(ames\_homesales[sapply(ames\_homesales, is.character)],   
 as.factor)   
  
#Response Variable, Above\_Median, is a character vector. Also, let's convert all the variables that are characters to factors

#str(ames\_homesales)

ames <- ames\_homesales %>%  
select(c(1:78,81))  
#Removing columns 79 & 80 (Longitude and Latitude) to reduce the complexity since these are variables that don't add to our predictions.

table(ames$Above\_Median) #1010 properties not above median, 1043 above median

##   
## No Yes   
## 1010 1043

#this is the response variable and the dataset appears balanced with only slightly more than half of the properties selling above median

t1 = table(ames$Above\_Median, ames$Year\_Sold) #create a table object  
prop.table(t1, margin = 2 ) #crosstab with proportions

##   
## 2006 2007 2008 2009 2010  
## No 0.4909502 0.4669339 0.4966292 0.5043860 0.5165877  
## Yes 0.5090498 0.5330661 0.5033708 0.4956140 0.4834123

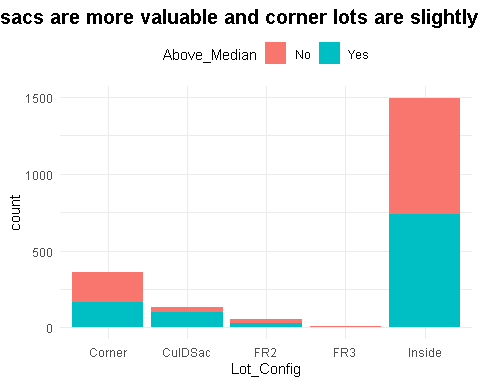
#Proportion of homes selling above (or below) median (response variable) by year sold

t2 = table(ames$Above\_Median, ames$Lot\_Config)  
prop.table(t2, margin = 2)

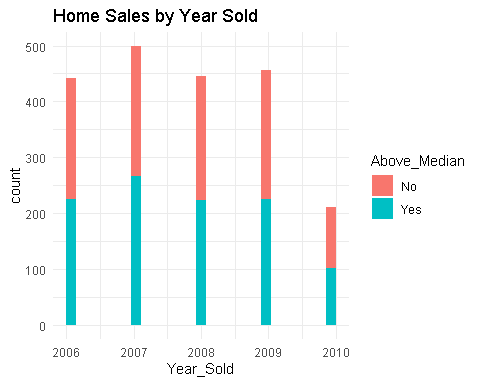
##   
## Corner CulDSac FR2 FR3 Inside  
## No 0.5320334 0.2518519 0.4821429 0.3750000 0.5050167  
## Yes 0.4679666 0.7481481 0.5178571 0.6250000 0.4949833

#Proportion of properties selling above median by lot config. CulDeSacs are selling above median, as are FR3.

ggplot(ames) +  
 aes(x = Lot\_Config, fill = Above\_Median) +  
 geom\_bar() +  
 scale\_fill\_hue(direction = 1) +  
 labs(title = " Cul-de-sacs are more valuable and corner lots are slightly less valuable") +  
 theme\_minimal() +  
 theme(legend.position = "top", plot.title = element\_text(size = 15L, face = "bold",   
 hjust = 0.5), plot.subtitle = element\_text(size = 12L, hjust = 0.5))

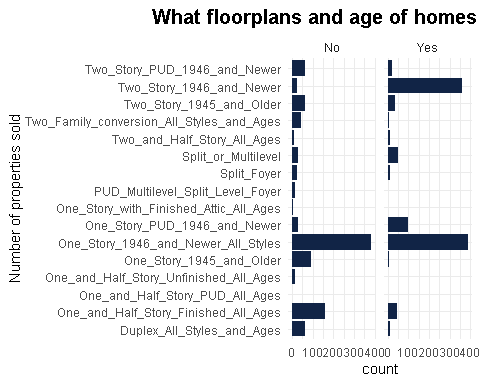


#How many properties sold each year?  
ggplot(ames) +  
 aes(x = Year\_Sold, fill = Above\_Median) +  
 geom\_histogram(bins = 30L) +  
 scale\_fill\_hue(direction = 1) +  
 labs(title = "Home Sales by Year Sold") +  
 theme\_minimal()

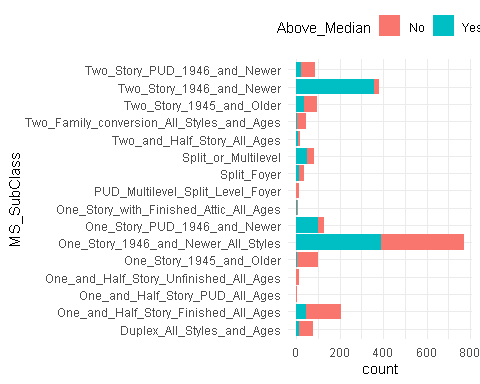


#We learn that sales dropped in 2010 but as we already knew from the table, proportion of above and below median was uniform across the years.

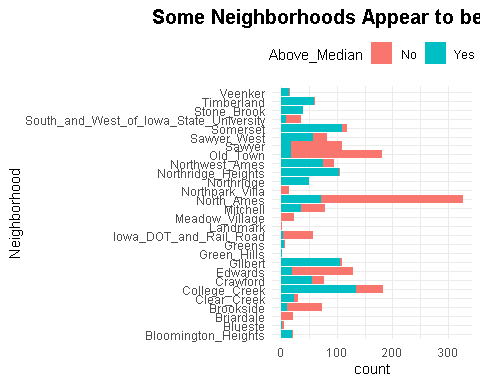
#esquisser  
  
  
ggplot(ames) +  
 aes(x = MS\_SubClass) +  
 geom\_bar(fill = "#112446") +  
 labs(x = "Number of properties sold",   
 title = "What floorplans and age of homes were selling?") +  
 coord\_flip() +  
 theme\_minimal() +  
 theme(plot.title = element\_text(size = 15L,   
 face = "bold", hjust = 0.5), plot.subtitle = element\_text(size = 12L, hjust = 0.5)) +  
 facet\_wrap(vars(Above\_Median))



ggplot(ames) +  
 aes(x = MS\_SubClass, fill = Above\_Median) +  
 geom\_bar() +  
 scale\_fill\_hue(direction = 1) +  
 coord\_flip() +  
 theme\_minimal() +  
 theme(legend.position = "top", plot.title = element\_text(size = 15L,   
 face = "bold", hjust = 0.5), plot.subtitle = element\_text(size = 12L, hjust = 0.5))



ggplot(ames) +  
 aes(x = Neighborhood, fill = Above\_Median) +  
 geom\_bar() +  
 scale\_fill\_hue(direction = 1) +  
 labs(title = "Some Neighborhoods Appear to be Hot Sellers") +  
 coord\_flip() +  
 theme\_minimal() +  
 theme(legend.position = "top",   
 plot.title = element\_text(size = 15L, face = "bold", hjust = 0.5), plot.subtitle = element\_text(size = 12L,   
 hjust = 0.5))



table(ames$Street) #Only 7 properties in this category are in "gravel" category, all others are "street"

##   
## Grvl Pave   
## 7 2046

table(ames$Utilities) #Only 1 property is "no sewr" all other are "public"

##   
## AllPub NoSewr   
## 2052 1

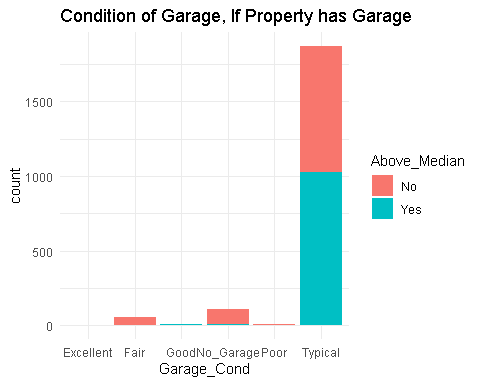
table(ames$Lot\_Shape)

##   
## Irregular Moderately\_Irregular Regular   
## 11 53 1275   
## Slightly\_Irregular   
## 714

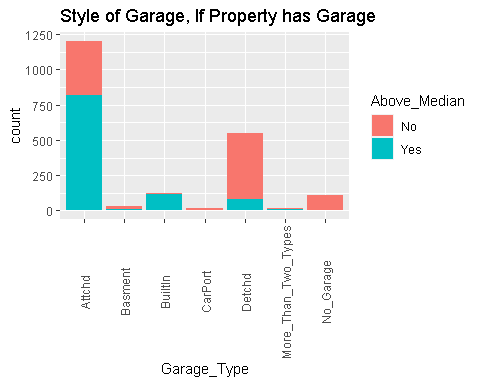
table(ames$Half\_Bath) #1300 have no half bath, 736 have 1 half bath and 17 have 2 half baths

##   
## 0 1 2   
## 1300 736 17

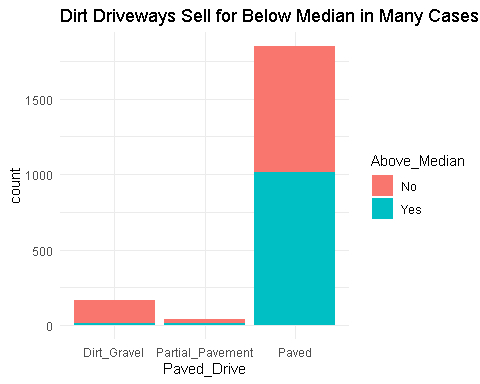
ggplot(ames) +  
 aes(x = Garage\_Cond, fill = Above\_Median) +  
 geom\_bar() +  
 scale\_fill\_hue(direction = 1) +  
 labs(title = "Condition of Garage, If Property has Garage") +  
 theme\_minimal()



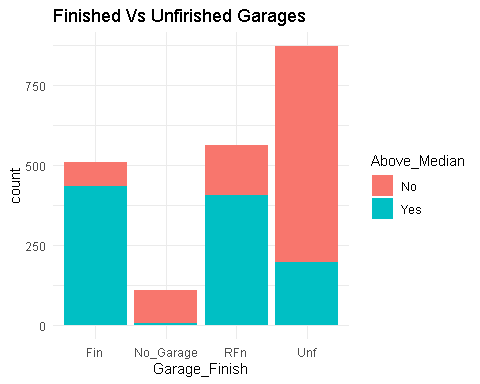
ggplot(ames) +  
 aes(x = Garage\_Type, fill = Above\_Median) +  
 geom\_bar() +  
 scale\_fill\_hue(direction = 1) +  
 labs(title = "Style of Garage, If Property has Garage") +  
 theme(axis.text.x = element\_text(angle = 90,vjust = .5))



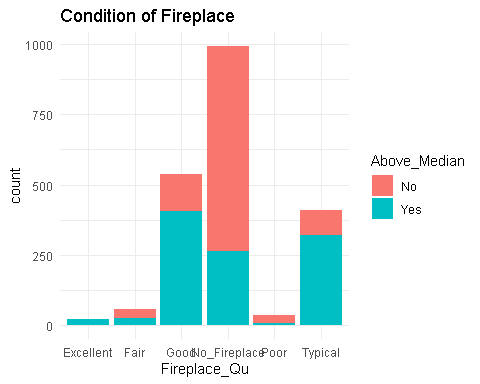
ggplot(ames) +  
 aes(x = Paved\_Drive, fill = Above\_Median) +  
 geom\_bar() +  
 scale\_fill\_hue(direction = 1) +  
 labs(title = "Dirt Driveways Sell for Below Median in Many Cases") +  
 theme\_minimal()



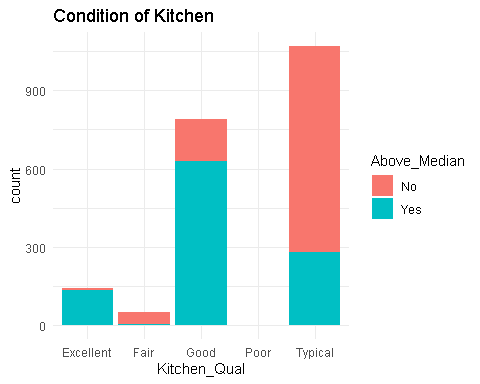
ggplot(ames) +  
 aes(x = Garage\_Finish, fill = Above\_Median) +  
 geom\_bar() +  
 scale\_fill\_hue(direction = 1) +  
 labs(title = "Finished Vs Unfirished Garages") +  
 theme\_minimal()



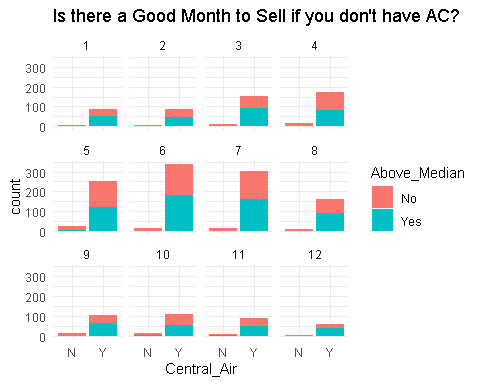
ggplot(ames) +  
 aes(x = Fireplace\_Qu, fill = Above\_Median) +  
 geom\_bar() +  
 scale\_fill\_hue(direction = 1) +  
 labs(title = "Condition of Fireplace") +  
 theme\_minimal()



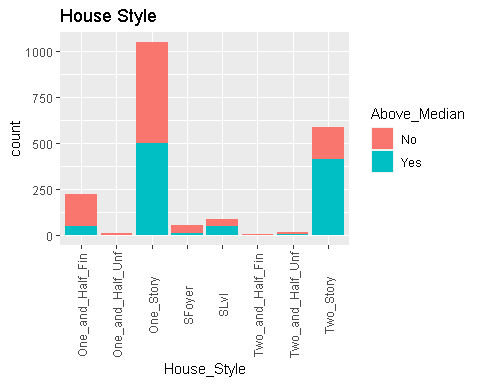
ggplot(ames) +  
 aes(x = Kitchen\_Qual, fill = Above\_Median) +  
 geom\_bar() +  
 scale\_fill\_hue(direction = 1) +  
 labs(title = "Condition of Kitchen") +  
 theme\_minimal()



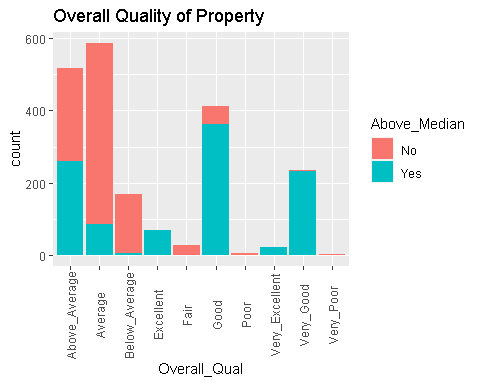
ggplot(ames) +  
 aes(x = Central\_Air, fill = Above\_Median) +  
 geom\_bar() +  
 scale\_fill\_hue(direction = 1) +  
 labs(title = "Is there a Good Month to Sell if you don't have AC?") +  
 theme\_minimal() +  
 facet\_wrap(vars(Mo\_Sold))



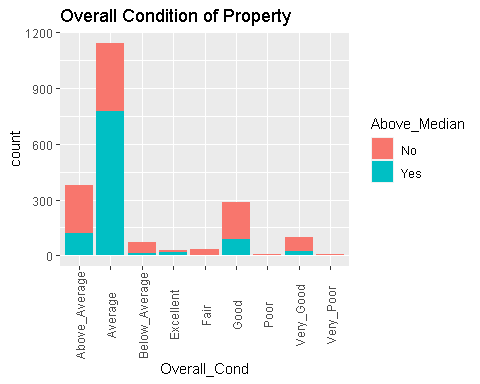
ggplot(ames) +  
 aes(x = House\_Style, fill = Above\_Median) +  
 geom\_bar() +  
 scale\_fill\_hue(direction = 1) +  
 labs(title = "House Style") +  
 theme(axis.text.x = element\_text(angle = 90,vjust = .5))



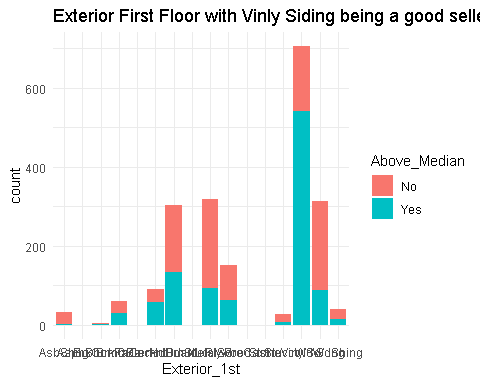
ggplot(ames) +  
 aes(x = Overall\_Qual, fill = Above\_Median) +  
 geom\_bar() +  
 scale\_fill\_hue(direction = 1) +  
 labs(title = "Overall Quality of Property") +  
 theme(axis.text.x = element\_text(angle = 90,vjust = .5))



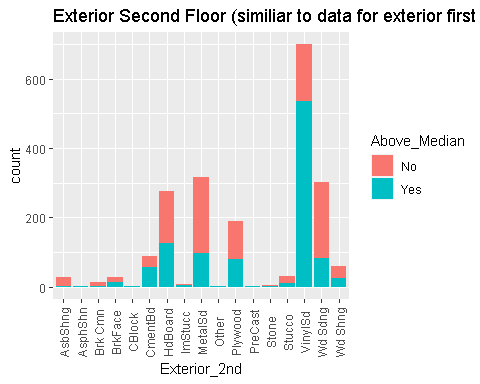
ggplot(ames) +  
 aes(x = Overall\_Cond, fill = Above\_Median) +  
 geom\_bar() +  
 scale\_fill\_hue(direction = 1) +  
 labs(title = "Overall Condition of Property") +  
 theme(axis.text.x = element\_text(angle = 90,vjust = .5))



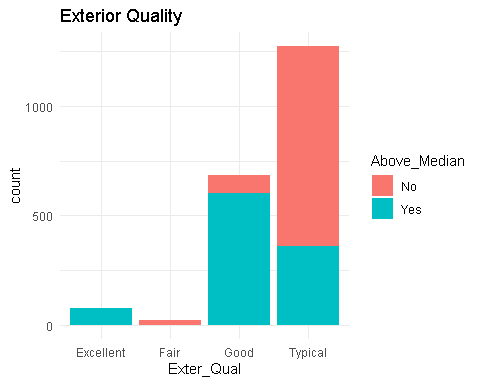
ggplot(ames) +  
 aes(x = Exterior\_1st, fill = Above\_Median) +  
 geom\_bar() +  
 scale\_fill\_hue(direction = 1) +  
 labs(title = "Exterior First Floor with Vinly Siding being a good seller") +  
 theme\_minimal()



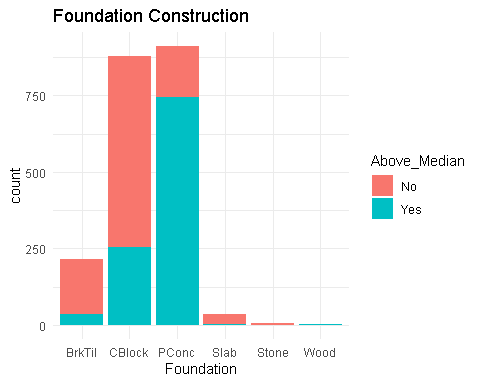
ggplot(ames) +  
 aes(x = Exterior\_2nd, fill = Above\_Median) +  
 geom\_bar() +  
 scale\_fill\_hue(direction = 1) +  
 labs(title = "Exterior Second Floor (similiar to data for exterior first floor)") +  
 theme(axis.text.x = element\_text(angle = 90,vjust = .5))



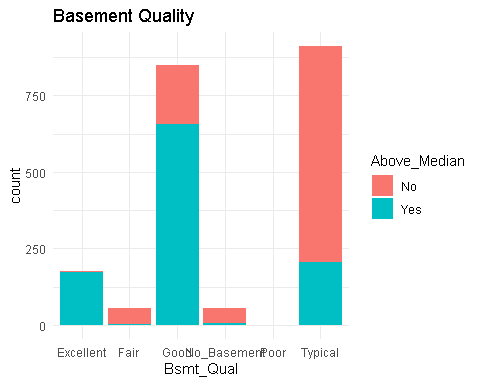
ggplot(ames) +  
 aes(x = Exter\_Qual, fill = Above\_Median) +  
 geom\_bar() +  
 scale\_fill\_hue(direction = 1) +  
 labs(title = "Exterior Quality") +  
 theme\_minimal()



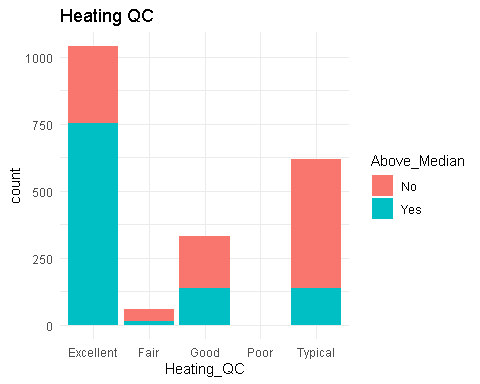
ggplot(ames) +  
 aes(x = Foundation, fill = Above\_Median) +  
 geom\_bar() +  
 scale\_fill\_hue(direction = 1) +  
 labs(title = "Foundation Construction") +  
 theme\_minimal()



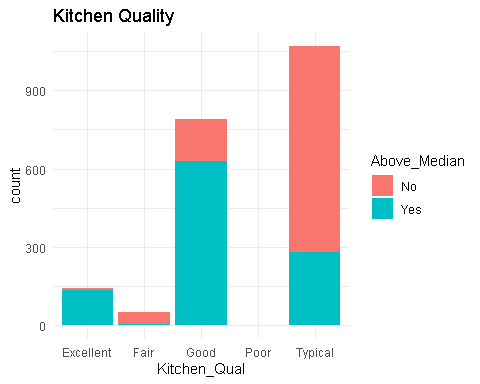
ggplot(ames) +  
 aes(x = Bsmt\_Qual, fill = Above\_Median) +  
 geom\_bar() +  
 scale\_fill\_hue(direction = 1) +  
 labs(title = "Basement Quality") +  
 theme\_minimal()



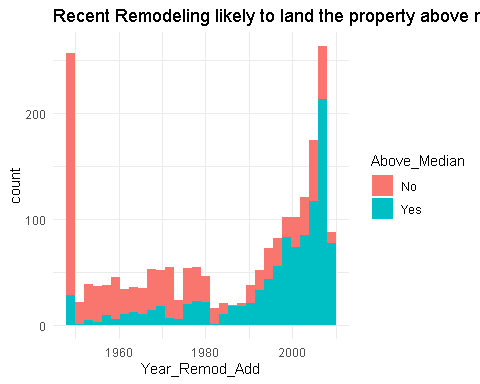
ggplot(ames) +  
 aes(x = Heating\_QC, fill = Above\_Median) +  
 geom\_bar() +  
 scale\_fill\_hue(direction = 1) +  
 labs(title = "Heating QC") +  
 theme\_minimal()



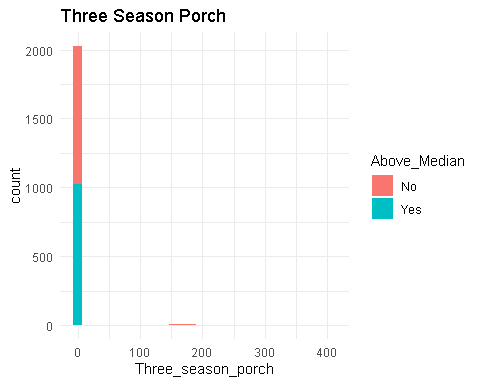
ggplot(ames) +  
 aes(x = Kitchen\_Qual, fill = Above\_Median) +  
 geom\_bar() +  
 scale\_fill\_hue(direction = 1) +  
 labs(title = "Kitchen Quality") +  
 theme\_minimal()



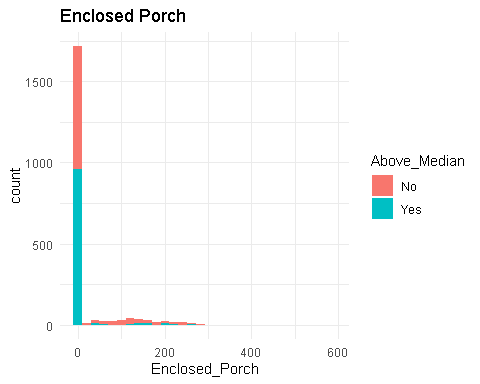
ggplot(ames) +  
 aes(x = Year\_Remod\_Add, fill = Above\_Median) +  
 geom\_histogram(bins = 30L) +  
 scale\_fill\_hue(direction = 1) +  
 labs(title = "Recent Remodeling likely to land the property above median") +  
 theme\_minimal()



ggplot(ames) +  
 aes(x = Three\_season\_porch, fill = Above\_Median) +  
 geom\_histogram(bins = 30L) +  
 scale\_fill\_hue(direction = 1) +  
 labs(title = "Three Season Porch") +  
 theme\_minimal()



ggplot(ames) +  
 aes(x = Enclosed\_Porch, fill = Above\_Median) +  
 geom\_histogram(bins = 30L) +  
 scale\_fill\_hue(direction = 1) +  
 labs(title = "Enclosed Porch") +  
 theme\_minimal()

 #Filter out some outliers #```{r}

ames %>% filter(Lot\_Area >= 1300L & Lot\_Area <= 28280L) %>% filter(Wood\_Deck\_SF >= 0L & Wood\_Deck\_SF <= 302L) %>% ggplot() + aes(x = ““, y = Lot\_Area, fill = Above\_Median) + geom\_boxplot() + scale\_fill\_hue(direction = 1) + labs(title =”Lot Area”) + theme\_minimal()

ames %>% filter(Lot\_Area >= 1300L & Lot\_Area <= 28280L) %>% filter(Wood\_Deck\_SF >= 0L & Wood\_Deck\_SF <= 302L) %>% ggplot() + aes(x = ““, y = Year\_Built, fill = Above\_Median) + geom\_boxplot() + scale\_fill\_hue(direction = 1) + labs(title =”Newer Houses Have higher Selling Prices”) + theme\_minimal()

ames %>% filter(Lot\_Area >= 1300L & Lot\_Area <= 28280L) %>% filter(Wood\_Deck\_SF >= 0L & Wood\_Deck\_SF <= 302L) %>% ggplot() + aes(x = Wood\_Deck\_SF, fill = Above\_Median) + geom\_histogram(bins = 30L) + scale\_fill\_hue(direction = 1) + labs(title = “Wood Decks from 100 - 300 square feet are good selling points”) + theme\_minimal()

ames %>% filter(Lot\_Area >= 1300L & Lot\_Area <= 28280L) %>% filter(Wood\_Deck\_SF >= 0L & Wood\_Deck\_SF <= 302L) %>% ggplot() + aes(x = ““, y = First\_Flr\_SF, fill = Above\_Median) + geom\_boxplot() + scale\_fill\_hue(direction = 1) + labs(title =”First Floor Sf”) + theme\_minimal()

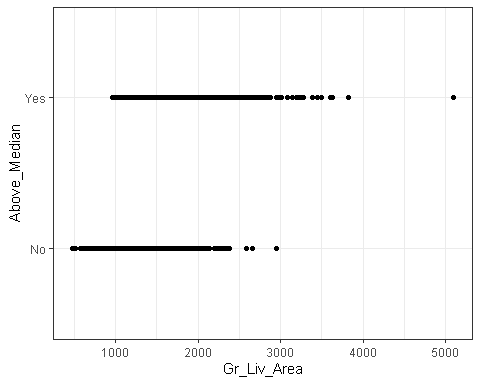
ames %>% filter(Lot\_Area >= 1300L & Lot\_Area <= 28280L) %>% filter(Mas\_Vnr\_Area >= 0L & Mas\_Vnr\_Area <= 502L) %>% filter(Wood\_Deck\_SF >= 0L & Wood\_Deck\_SF <= 302L) %>% ggplot() + aes(x = Full\_Bath, fill = Above\_Median) + geom\_histogram(bins = 30L) + scale\_fill\_hue(direction = 1) + labs(title = “Two Full Bathrooms is a predictor for above median value”) + theme\_minimal()

ames %>% filter(Lot\_Area >= 1300L & Lot\_Area <= 28280L) %>% filter(Mas\_Vnr\_Area >= 0L & Mas\_Vnr\_Area <= 502L) %>% filter(Wood\_Deck\_SF >= 0L & Wood\_Deck\_SF <= 302L) %>% ggplot() + aes(x = Full\_Bath, fill = Above\_Median) + geom\_histogram(bins = 30L) + scale\_fill\_hue(direction = 1) + labs(title = “Two Full Bathrooms is a predictor for above median value”) + theme\_minimal()```

t9 = table(ames$Above\_Median, ames$Full\_Bath)  
prop.table(t9, margin = 2)

##   
## 0 1 2 3 4  
## No 0.70000000 0.82500000 0.22500000 0.02439024 0.00000000  
## Yes 0.30000000 0.17500000 0.77500000 0.97560976 1.00000000

ggplot(ames, aes(x=Gr\_Liv\_Area, y=Above\_Median)) + geom\_point() + theme\_bw()



#Some outliers are homes with large square footage, above 3000 sq feet

Ames\_MS\_SubClass <- ames\_homesales %>%  
 group\_by(MS\_SubClass) %>%  
 summarize(freq = n()) %>%  
 arrange(desc(freq))  
  
Ames\_MS\_SubClass

## # A tibble: 16 × 2  
## MS\_SubClass freq  
## <fct> <int>  
## 1 One\_Story\_1946\_and\_Newer\_All\_Styles 772  
## 2 Two\_Story\_1946\_and\_Newer 383  
## 3 One\_and\_Half\_Story\_Finished\_All\_Ages 204  
## 4 One\_Story\_PUD\_1946\_and\_Newer 129  
## 5 One\_Story\_1945\_and\_Older 98  
## 6 Two\_Story\_1945\_and\_Older 95  
## 7 Two\_Story\_PUD\_1946\_and\_Newer 85  
## 8 Split\_or\_Multilevel 82  
## 9 Duplex\_All\_Styles\_and\_Ages 76  
## 10 Two\_Family\_conversion\_All\_Styles\_and\_Ages 46  
## 11 Split\_Foyer 34  
## 12 Two\_and\_Half\_Story\_All\_Ages 17  
## 13 One\_and\_Half\_Story\_Unfinished\_All\_Ages 13  
## 14 PUD\_Multilevel\_Split\_Level\_Foyer 12  
## 15 One\_Story\_with\_Finished\_Attic\_All\_Ages 6  
## 16 One\_and\_Half\_Story\_PUD\_All\_Ages 1

#t3 = table(ames$Above\_Median, ames$MS\_SubClass) #create a table object  
#t3

#28 Neighborhoods are represented. Some neighborhoods have very few (only 1) home sale while others have many (327)  
Ames\_gp\_neighborhoods <- ames\_homesales %>%  
 group\_by(Neighborhood) %>%  
 summarize(freq = n()) %>%  
 arrange(desc(freq))  
  
Ames\_gp\_neighborhoods

## # A tibble: 28 × 2  
## Neighborhood freq  
## <fct> <int>  
## 1 North\_Ames 327  
## 2 College\_Creek 183  
## 3 Old\_Town 181  
## 4 Edwards 129  
## 5 Somerset 119  
## 6 Gilbert 109  
## 7 Sawyer 109  
## 8 Northridge\_Heights 105  
## 9 Northwest\_Ames 95  
## 10 Sawyer\_West 82  
## # … with 18 more rows

t8 = table(ames$Above\_Median, ames$Neighborhood) #create a table object  
prop.table(t8, margin = 2)

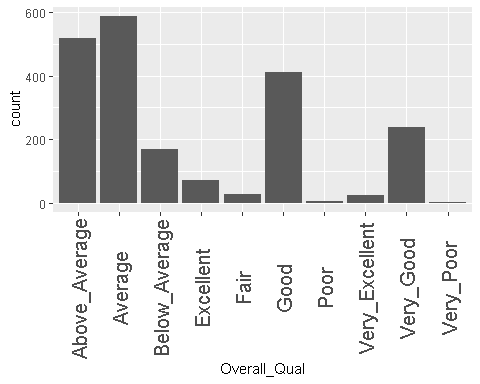
##   
## Bloomington\_Heights Blueste Briardale Brookside Clear\_Creek  
## No 0.09523810 0.80000000 1.00000000 0.86486486 0.22580645  
## Yes 0.90476190 0.20000000 0.00000000 0.13513514 0.77419355  
##   
## College\_Creek Crawford Edwards Gilbert Green\_Hills Greens  
## No 0.26775956 0.27272727 0.84496124 0.02752294 0.00000000 0.14285714  
## Yes 0.73224044 0.72727273 0.15503876 0.97247706 1.00000000 0.85714286  
##   
## Iowa\_DOT\_and\_Rail\_Road Landmark Meadow\_Village Mitchell North\_Ames  
## No 0.92982456 1.00000000 1.00000000 0.54430380 0.77981651  
## Yes 0.07017544 0.00000000 0.00000000 0.45569620 0.22018349  
##   
## Northpark\_Villa Northridge Northridge\_Heights Northwest\_Ames Old\_Town  
## No 1.00000000 0.00000000 0.00952381 0.20000000 0.90055249  
## Yes 0.00000000 1.00000000 0.99047619 0.80000000 0.09944751  
##   
## Sawyer Sawyer\_West Somerset South\_and\_West\_of\_Iowa\_State\_University  
## No 0.84403670 0.30487805 0.07563025 0.74285714  
## Yes 0.15596330 0.69512195 0.92436975 0.25714286  
##   
## Stone\_Brook Timberland Veenker  
## No 0.00000000 0.01666667 0.06250000  
## Yes 1.00000000 0.98333333 0.93750000

#crosstab with proportions  
  
#Proportion of homes selling above (or below) median by neighborhood

Ames\_Overall\_Qual <- ames\_homesales %>%  
 group\_by(Overall\_Qual) %>%  
 summarize(freq = n()) %>%  
 arrange(desc(freq))  
  
Ames\_Overall\_Qual

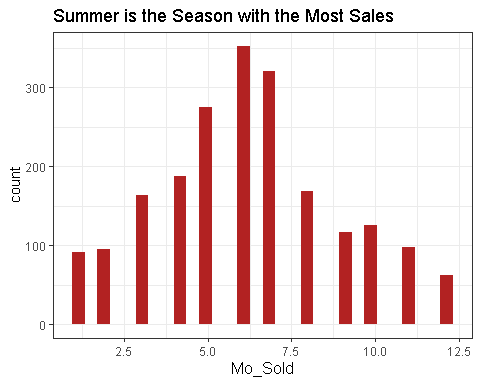
## # A tibble: 10 × 2  
## Overall\_Qual freq  
## <fct> <int>  
## 1 Average 587  
## 2 Above\_Average 518  
## 3 Good 411  
## 4 Very\_Good 237  
## 5 Below\_Average 169  
## 6 Excellent 70  
## 7 Fair 28  
## 8 Very\_Excellent 24  
## 9 Poor 6  
## 10 Very\_Poor 3

ggplot(ames\_homesales, aes(Overall\_Qual))+  
 geom\_bar() +  
 theme(axis.text.x = element\_text(angle = 90, vjust = .5, size = 15))



#Looks like some of these conditions are outliers

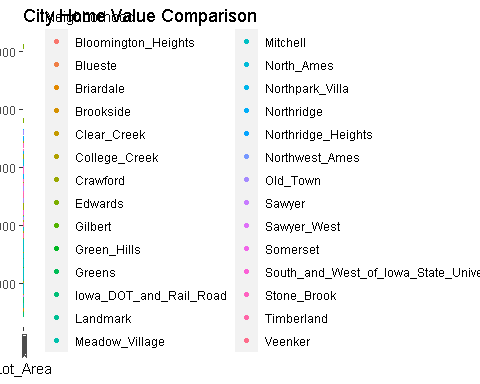
ggplot(ames) +  
 aes(x = Mo\_Sold) +  
 geom\_histogram(bins = 30L, fill = "#B22222") +  
 labs(title = "Summer is the Season with the Most Sales") +   
 theme\_bw() +  
 theme(axis.title.x = element\_text(size = 12L))

 ##Part TWo

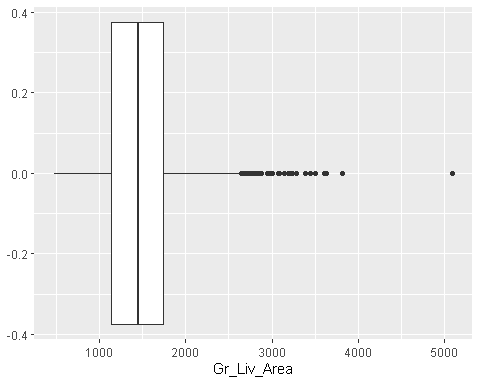
#Let's group the neighborhoods with very few sales into a category called "other". This gives us 23 neighborhoods (one is "other") instead of 28  
#We may neeed to step dummy the response variable because it's not numeric.

ames\_recipe = recipe(Gr\_Liv\_Area~Neighborhood,ames) %>%  
 step\_other(Neighborhood, threshold =.01) %>%  
 step\_dummy(all\_nominal())

ggplot(ames, aes(x = Lot\_Area, y = Gr\_Liv\_Area, color = Neighborhood))+ geom\_point()+   
 labs(title = "City Home Value Comparison", ) +   
 theme(axis.text.x = element\_text(angle = 90, vjust = .5, size = 5))+ scale\_y\_continuous(name = "Living Area Square Footage", labels = scales::comma)



ggplot(ames, aes(Gr\_Liv\_Area))+   
 geom\_boxplot()



table(ames$Gr\_Liv\_Area > 3000) #Sixteen properties are outliers, they are large properties with more than 3000 square feet

##   
## FALSE TRUE   
## 2037 16

#Outliers in living area with 16 of the homes being over 3000 sq feet. Let's remove these 16 observations, which is .77% of the total dataset.  
  
ames <-ames %>%  
 filter(Gr\_Liv\_Area<3000)

#Some neighborhoods have very few (only 1) home sale while others have many (327)

Ames\_gp\_neighborhoods <- ames %>%  
 group\_by(Neighborhood) %>%  
 summarize(freq = n()) %>%  
 arrange(desc(freq))

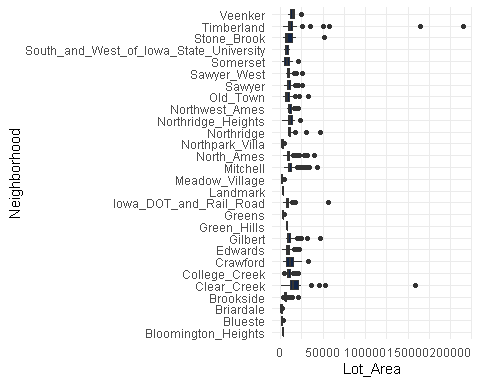
Ames\_gp\_neighborhoods

## # A tibble: 28 × 2  
## Neighborhood freq  
## <fct> <int>  
## 1 North\_Ames 326  
## 2 College\_Creek 183  
## 3 Old\_Town 180  
## 4 Edwards 127  
## 5 Somerset 118  
## 6 Gilbert 109  
## 7 Sawyer 109  
## 8 Northridge\_Heights 104  
## 9 Northwest\_Ames 95  
## 10 Sawyer\_West 82  
## # … with 18 more rows

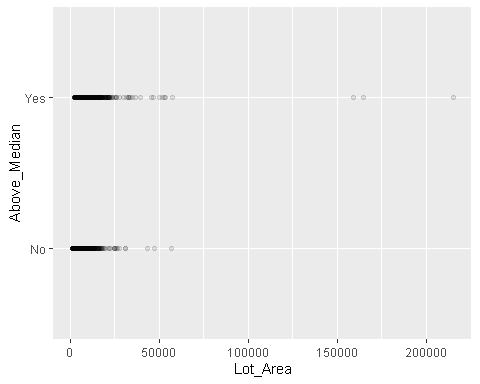
#Year Sold is an integer variable, but we want to examine each year as it’s own category. Therefore, we convert Year\_Sold into a factor.

ames\_yr\_as\_factor <- ames %>%  
 mutate(Year\_Sold = as\_factor(Year\_Sold))

ggplot(ames) +  
 aes(x = Lot\_Area, y = Neighborhood) +  
 geom\_boxplot(fill = "#112446") +  
 theme\_minimal()



ggplot(ames, aes(x=Lot\_Area, y=Above\_Median))+  
 geom\_point(alpha=.1)



ames1 <- ames %>%  
select(c(1:4,6:8,10:79))  
#Removing columns 5 & 9 (Street and Utilities) as these variables that don't add to our predictions.

#Remove Alley, Land\_Slope, Condition\_1, Condition\_2, Sale\_Condition, Sale\_type, BsmtFin\_Type\_2\_Misc\_features as all have no predictive value  
  
ames2 <-ames1 %>%  
 select(-Alley,-Land\_Slope,-Sale\_Condition,-Condition\_1,-Condition\_2,-Sale\_Type,-Misc\_Feature,-BsmtFin\_Type\_2,-Pool\_Area,-Pool\_QC,-Fence)  
  
  
#str(ames2)

#Here is another method of paring down ames2 dataset to just the variables that seemed to have predictive value based on my ggplots and proportion tables

ames2 <- ames2 %>%  
 select(MS\_SubClass, MS\_Zoning, Overall\_Qual, Overall\_Cond,Exterior\_1st,Exterior\_2nd,Mas\_Vnr\_Type,Exter\_Qual,Foundation,Bsmt\_Qual,Heating\_QC,Kitchen\_Qual,Fireplace\_Qu,Year\_Remod\_Add,Year\_Built,First\_Flr\_SF,Full\_Bath,Neighborhood,Above\_Median, Garage\_Type)

#Splitting data

ames2\_split = initial\_split(ames2, prop = 0.7, strata = Above\_Median) #70% in training  
train = training(ames2\_split)   
test = testing(ames2\_split)

Now that we have the split data, let’s build a classification tree. Here we use caret to manage the model building.  
#```{r} #ames2\_recipe = recipe(Above\_Median ~ ., train) %>% step\_dummy(all\_nominal())

tree\_model = decision\_tree() %>% set\_engine(“rpart”, model = TRUE) %>% #don’t forget the model = TRUE flag set\_mode(“classification”)

ames2\_wflow = workflow() %>% add\_model(tree\_model) %>% add\_recipe(ames2\_recipe)

ames2\_fit = fit(ames2\_wflow, train)

Let's take a look at our tree (a few ways)   
#```{r}  
#look at the tree's fit  
ames2\_fit %>%  
 pull\_workflow\_fit() %>%  
 pluck("fit")

#```{r} #extract the tree’s fit from the fit object tree = ames2\_fit %>% pull\_workflow\_fit() %>% pluck(“fit”)

#plot the tree rpart.plot(tree)

#```{r}  
#alternative  
fancyRpartPlot(tree, tweak=1.5)

Look at the “rpart” complexity parameter “cp”. Auto tuning to grow the right size tree #{r} ames2\_fit$fit$fit$fit$cptable Predictions on training set  
#{r} treepred = predict(ames2\_fit, train, type = "class") head(treepred)

Caret confusion matrix and accuracy, etc. calcs  
#{r} confusionMatrix(treepred$.pred\_class,train$Above\_Median,positive="Yes") #predictions first then actual

Predictions on testing set  
#{r} treepred\_test = predict(ames2\_fit, test, type = "class") head(treepred\_test)

Caret confusion matrix and accuracy, etc. calcs  
#{r} confusionMatrix(treepred\_test$.pred\_class,test$Above\_Median,positive="Yes") #predictions first then actual #Shall we attempt to hand tune this instrument? Create our folds  
#{r} folds = vfold\_cv(train, v = 5)

#```{r} ames2\_recipe = recipe(Above\_Median ~., train) %>% step\_dummy(all\_nominal(),-all\_outcomes())

tree\_model = decision\_tree(cost\_complexity = tune()) %>% set\_engine(“rpart”, model = TRUE) %>% #don’t forget the model = TRUE flag set\_mode(“classification”)

tree\_grid = grid\_regular(cost\_complexity(), levels = 25) #try 25 sensible values for cp

ames2\_wflow = workflow() %>% add\_model(tree\_model) %>% add\_recipe(ames2\_recipe)

tree\_res = ames2\_wflow %>% tune\_grid( resamples = folds, grid = tree\_grid )

tree\_res

Borrowed code from: https://www.tidymodels.org/start/tuning/  
#```{r}  
tree\_res %>%  
 collect\_metrics() %>%  
 ggplot(aes(cost\_complexity, mean)) +  
 geom\_line(size = 1.5, alpha = 0.6) +  
 geom\_point(size = 2) +  
 facet\_wrap(~ .metric, scales = "free", nrow = 2)

#```{r} best\_tree = tree\_res %>% select\_best(“accuracy”)

best\_tree

#```{r}  
final\_wf =   
 ames2\_wflow %>%   
 finalize\_workflow(best\_tree)

#```{r} final\_fit = fit(final\_wf, train)

tree = final\_fit %>% pull\_workflow\_fit() %>% pluck(“fit”)

fancyRpartPlot(tree, tweak = 1.9)

#Predictions on training set   
#```{r}  
treepred = predict(final\_fit, train, type = "class")  
head(treepred)

Caret confusion matrix and accuracy, etc. calcs  
#{r} confusionMatrix(treepred$.pred\_class,train$Above\_Median,positive="Yes") #predictions first then actual

Predictions on testing set  
#{r} treepred\_test = predict(final\_fit, test, type = "class") head(treepred\_test)

Caret confusion matrix and accuracy, etc. calcs  
#{r} confusionMatrix(treepred\_test$.pred\_class,test$Above\_Median,positive="Yes") #predictions first then actual

#Random FOrest

ames3\_recipe = recipe(Above\_Median ~ ., train) %>%  
 step\_other(Neighborhood, threshold = 0.01) %>% #collapses small Neighborhoods into an "Other" group  
 step\_dummy(all\_nominal(),-all\_outcomes()) #dummify categorical variables except for response variable  
   
   
rf\_model = rand\_forest() %>%   
 set\_engine("ranger", importance = "permutation") %>% #added importance metric  
 set\_mode("classification")  
  
ames3\_wflow =   
 workflow() %>%   
 add\_model(rf\_model) %>%   
 add\_recipe(ames3\_recipe)  
  
ames3\_fit = fit(ames3\_wflow, train)

Predictions

trainpredrf = predict(ames3\_fit, train)  
head(trainpredrf)

## # A tibble: 6 × 1  
## .pred\_class  
## <fct>   
## 1 No   
## 2 No   
## 3 No   
## 4 No   
## 5 No   
## 6 No

Confusion matrix

confusionMatrix(trainpredrf$.pred\_class, train$Above\_Median,   
 positive = "Yes")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 694 19  
## Yes 13 699  
##   
## Accuracy : 0.9775   
## 95% CI : (0.9684, 0.9846)  
## No Information Rate : 0.5039   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.9551   
##   
## Mcnemar's Test P-Value : 0.3768   
##   
## Sensitivity : 0.9735   
## Specificity : 0.9816   
## Pos Pred Value : 0.9817   
## Neg Pred Value : 0.9734   
## Prevalence : 0.5039   
## Detection Rate : 0.4905   
## Detection Prevalence : 0.4996   
## Balanced Accuracy : 0.9776   
##   
## 'Positive' Class : Yes   
##

Predictions on test

testpredrf = predict(ames3\_fit, test)  
head(testpredrf)

## # A tibble: 6 × 1  
## .pred\_class  
## <fct>   
## 1 No   
## 2 No   
## 3 Yes   
## 4 Yes   
## 5 Yes   
## 6 Yes

confusionMatrix(testpredrf$.pred\_class, test$Above\_Median,   
 positive = "Yes")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 288 48  
## Yes 15 261  
##   
## Accuracy : 0.8971   
## 95% CI : (0.8702, 0.92)  
## No Information Rate : 0.5049   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.7943   
##   
## Mcnemar's Test P-Value : 5.539e-05   
##   
## Sensitivity : 0.8447   
## Specificity : 0.9505   
## Pos Pred Value : 0.9457   
## Neg Pred Value : 0.8571   
## Prevalence : 0.5049   
## Detection Rate : 0.4265   
## Detection Prevalence : 0.4510   
## Balanced Accuracy : 0.8976   
##   
## 'Positive' Class : Yes   
##

Save the model to a file to load later (if needed)

saveRDS(ames3\_fit, "ames3\_fit.rds")

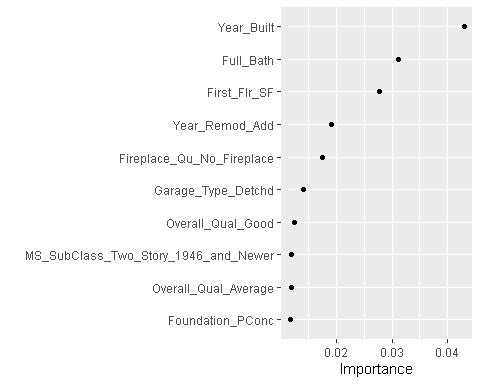
Load the model

ames3\_fit = readRDS("ames3\_fit.rds")

Check out variable importance

ames3\_fit %>% pull\_workflow\_fit() %>% vip(geom = "point")

## Warning: `pull\_workflow\_fit()` was deprecated in workflows 0.2.3.  
## Please use `extract\_fit\_parsnip()` instead.



#Logistic Regression

Let’s build a model with all variables.

ames2\_model =   
 logistic\_reg() %>% #note the use of logistic\_reg and mode = "classification"  
 set\_engine("glm") #standard logistic regression engine is glm  
  
ames2\_recipe = recipe(Above\_Median ~ ., train) %>%  
 step\_dummy(all\_nominal(), -all\_outcomes())  
  
logreg\_wf = workflow() %>%  
 add\_recipe(ames2\_recipe) %>%   
 add\_model(ames2\_model)  
  
ames2\_fit3 = fit(logreg\_wf, train)

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

summary(ames2\_fit3$fit$fit$fit)

##   
## Call:  
## stats::glm(formula = ..y ~ ., family = stats::binomial, data = data)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -8.4904 -0.1064 0.0000 0.0488 3.3593   
##   
## Coefficients: (8 not defined because of singularities)  
## Estimate Std. Error  
## (Intercept) -4.336e+01 8.487e+04  
## Year\_Remod\_Add 1.366e-02 1.106e-02  
## Year\_Built 1.410e-02 1.882e-02  
## First\_Flr\_SF 5.944e-03 9.390e-04  
## Full\_Bath 1.089e+00 3.638e-01  
## MS\_SubClass\_One\_and\_Half\_Story\_Finished\_All\_Ages 3.762e+00 1.134e+00  
## MS\_SubClass\_One\_and\_Half\_Story\_PUD\_All\_Ages 1.754e+03 2.179e+05  
## MS\_SubClass\_One\_and\_Half\_Story\_Unfinished\_All\_Ages -2.679e+01 1.105e+03  
## MS\_SubClass\_One\_Story\_1945\_and\_Older 4.917e+00 1.683e+00  
## MS\_SubClass\_One\_Story\_1946\_and\_Newer\_All\_Styles 7.507e-01 8.907e-01  
## MS\_SubClass\_One\_Story\_PUD\_1946\_and\_Newer 8.242e-01 1.481e+00  
## MS\_SubClass\_One\_Story\_with\_Finished\_Attic\_All\_Ages -1.097e+00 3.941e+00  
## MS\_SubClass\_PUD\_Multilevel\_Split\_Level\_Foyer -1.265e+01 1.409e+02  
## MS\_SubClass\_Split\_Foyer 2.454e+00 1.177e+00  
## MS\_SubClass\_Split\_or\_Multilevel 2.013e+00 1.024e+00  
## MS\_SubClass\_Two\_and\_Half\_Story\_All\_Ages 6.297e+00 1.875e+00  
## MS\_SubClass\_Two\_Family\_conversion\_All\_Styles\_and\_Ages 6.609e-01 1.644e+00  
## MS\_SubClass\_Two\_Story\_1945\_and\_Older 3.538e+00 1.360e+00  
## MS\_SubClass\_Two\_Story\_1946\_and\_Newer 4.605e+00 1.125e+00  
## MS\_SubClass\_Two\_Story\_PUD\_1946\_and\_Newer -2.000e+01 1.582e+04  
## MS\_Zoning\_C\_all 1.171e+01 2.633e+03  
## MS\_Zoning\_Floating\_Village\_Residential 2.462e+01 5.415e+04  
## MS\_Zoning\_I\_all NA NA  
## MS\_Zoning\_Residential\_High\_Density 2.198e+01 2.630e+03  
## MS\_Zoning\_Residential\_Low\_Density 2.089e+01 2.630e+03  
## MS\_Zoning\_Residential\_Medium\_Density 1.853e+01 2.630e+03  
## Overall\_Qual\_Average -1.094e+00 3.654e-01  
## Overall\_Qual\_Below\_Average -2.267e+00 1.010e+00  
## Overall\_Qual\_Excellent -7.258e+00 2.710e+00  
## Overall\_Qual\_Fair -7.102e+00 5.074e+03  
## Overall\_Qual\_Good 1.099e+00 5.091e-01  
## Overall\_Qual\_Poor 1.701e+01 1.656e+02  
## Overall\_Qual\_Very\_Excellent 4.081e+01 7.497e+04  
## Overall\_Qual\_Very\_Good 3.509e+00 2.256e+00  
## Overall\_Qual\_Very\_Poor 4.504e+15 1.246e+07  
## Overall\_Cond\_Average -7.785e-01 4.206e-01  
## Overall\_Cond\_Below\_Average 1.728e-01 7.475e-01  
## Overall\_Cond\_Excellent 1.874e+00 1.849e+00  
## Overall\_Cond\_Fair -3.736e+00 1.602e+00  
## Overall\_Cond\_Good 8.403e-01 4.949e-01  
## Overall\_Cond\_Poor -4.265e+01 5.952e+05  
## Overall\_Cond\_Very\_Good -5.576e-01 7.316e-01  
## Overall\_Cond\_Very\_Poor -7.514e+00 2.611e+03  
## Exterior\_1st\_AsphShn NA NA  
## Exterior\_1st\_BrkComm 1.741e+01 5.490e+04  
## Exterior\_1st\_BrkFace 1.085e+00 3.233e+00  
## Exterior\_1st\_CBlock -4.504e+15 1.246e+07  
## Exterior\_1st\_CemntBd 2.186e+00 3.184e+00  
## Exterior\_1st\_HdBoard -9.980e-01 3.087e+00  
## Exterior\_1st\_ImStucc 1.206e+03 7.194e+05  
## Exterior\_1st\_MetalSd -9.807e-01 3.467e+00  
## Exterior\_1st\_Plywood -2.387e+00 3.093e+00  
## Exterior\_1st\_PreCast 4.504e+15 6.826e+07  
## Exterior\_1st\_Stone NA NA  
## Exterior\_1st\_Stucco -5.759e+00 3.415e+00  
## Exterior\_1st\_VinylSd -3.466e+00 3.452e+00  
## Exterior\_1st\_Wd.Sdng -1.612e+00 3.090e+00  
## Exterior\_1st\_WdShing -1.779e+00 3.281e+00  
## Exterior\_2nd\_AsphShn 2.888e+01 3.371e+05  
## Exterior\_2nd\_Brk.Cmn 9.699e+00 6.295e+04  
## Exterior\_2nd\_BrkFace 3.091e+00 3.869e+00  
## Exterior\_2nd\_CBlock NA NA  
## Exterior\_2nd\_CmentBd 3.541e+00 3.366e+00  
## Exterior\_2nd\_HdBoard 5.461e+00 3.670e+00  
## Exterior\_2nd\_ImStucc -1.202e+03 6.348e+05  
## Exterior\_2nd\_MetalSd 5.469e+00 3.991e+00  
## Exterior\_2nd\_Other -1.494e+01 3.242e+05  
## Exterior\_2nd\_Plywood 5.574e+00 3.604e+00  
## Exterior\_2nd\_PreCast NA NA  
## Exterior\_2nd\_Stone -1.295e+01 3.019e+05  
## Exterior\_2nd\_Stucco 8.810e+00 4.112e+00  
## Exterior\_2nd\_VinylSd 7.336e+00 3.951e+00  
## Exterior\_2nd\_Wd.Sdng 5.058e+00 3.661e+00  
## Exterior\_2nd\_Wd.Shng 6.502e+00 3.785e+00  
## Mas\_Vnr\_Type\_BrkFace 2.985e+00 1.323e+00  
## Mas\_Vnr\_Type\_CBlock NA NA  
## Mas\_Vnr\_Type\_None 2.190e+00 1.315e+00  
## Mas\_Vnr\_Type\_Stone 2.606e+00 1.520e+00  
## Exter\_Qual\_Fair -1.996e+01 3.446e+04  
## Exter\_Qual\_Good -2.195e+01 3.446e+04  
## Exter\_Qual\_Typical -2.239e+01 3.446e+04  
## Foundation\_CBlock -3.748e-01 7.337e-01  
## Foundation\_PConc 1.159e+00 7.859e-01  
## Foundation\_Slab 2.803e+01 3.998e+04  
## Foundation\_Stone -1.767e+01 2.101e+04  
## Foundation\_Wood NA NA  
## Bsmt\_Qual\_Fair -1.949e+00 2.103e+00  
## Bsmt\_Qual\_Good -1.614e+00 1.307e+00  
## Bsmt\_Qual\_No\_Basement -3.269e+01 3.998e+04  
## Bsmt\_Qual\_Poor 1.905e+00 1.417e+02  
## Bsmt\_Qual\_Typical -2.452e+00 1.343e+00  
## Heating\_QC\_Fair 9.156e-01 8.592e-01  
## Heating\_QC\_Good 5.499e-01 4.482e-01  
## Heating\_QC\_Poor NA NA  
## Heating\_QC\_Typical 1.379e-01 4.031e-01  
## Kitchen\_Qual\_Fair -4.504e+15 1.246e+07  
## Kitchen\_Qual\_Good -2.283e+00 1.228e+00  
## Kitchen\_Qual\_Poor -2.225e+01 3.423e+05  
## Kitchen\_Qual\_Typical -2.455e+00 1.196e+00  
## Fireplace\_Qu\_Fair -2.334e+01 7.752e+04  
## Fireplace\_Qu\_Good -2.347e+01 7.752e+04  
## Fireplace\_Qu\_No\_Fireplace -2.488e+01 7.752e+04  
## Fireplace\_Qu\_Poor -2.499e+01 7.752e+04  
## Fireplace\_Qu\_Typical -2.237e+01 7.752e+04  
## Neighborhood\_Blueste 1.691e+00 1.635e+05  
## Neighborhood\_Briardale 2.228e+01 1.582e+04  
## Neighborhood\_Brookside 3.146e+00 2.241e+00  
## Neighborhood\_Clear\_Creek 4.276e+00 2.234e+00  
## Neighborhood\_College\_Creek 2.466e+00 1.798e+00  
## Neighborhood\_Crawford 3.673e+00 2.081e+00  
## Neighborhood\_Edwards 1.002e+00 2.004e+00  
## Neighborhood\_Gilbert 2.298e+00 1.952e+00  
## Neighborhood\_Green\_Hills 5.831e+01 3.274e+05  
## Neighborhood\_Greens 1.826e+00 2.963e+00  
## Neighborhood\_Iowa\_DOT\_and\_Rail\_Road 4.070e+00 2.626e+00  
## Neighborhood\_Landmark -4.824e-01 4.481e+05  
## Neighborhood\_Meadow\_Village -2.130e+01 5.396e+04  
## Neighborhood\_Mitchell 2.068e+00 1.883e+00  
## Neighborhood\_North\_Ames 1.423e+00 1.905e+00  
## Neighborhood\_Northpark\_Villa -2.146e+01 9.011e+04  
## Neighborhood\_Northridge 2.035e+01 3.718e+04  
## Neighborhood\_Northridge\_Heights 2.189e+01 1.582e+04  
## Neighborhood\_Northwest\_Ames 1.716e+00 1.898e+00  
## Neighborhood\_Old\_Town 1.296e+00 2.404e+00  
## Neighborhood\_Sawyer 1.579e+00 1.925e+00  
## Neighborhood\_Sawyer\_West 1.449e+00 1.757e+00  
## Neighborhood\_Somerset 2.146e+01 5.171e+04  
## Neighborhood\_South\_and\_West\_of\_Iowa\_State\_University 7.278e-01 2.311e+00  
## Neighborhood\_Stone\_Brook 3.900e+01 4.511e+04  
## Neighborhood\_Timberland 4.594e+00 2.264e+00  
## Neighborhood\_Veenker 3.847e+00 2.923e+00  
## Garage\_Type\_Basment -1.066e+00 1.080e+00  
## Garage\_Type\_BuiltIn 1.731e+00 1.286e+00  
## Garage\_Type\_CarPort 1.182e+00 2.000e+00  
## Garage\_Type\_Detchd -1.445e+00 4.950e-01  
## Garage\_Type\_More\_Than\_Two\_Types 7.096e-01 1.108e+00  
## Garage\_Type\_No\_Garage -1.660e+00 1.203e+00  
## z value Pr(>|z|)   
## (Intercept) -1.000e-03 0.999592   
## Year\_Remod\_Add 1.235e+00 0.216840   
## Year\_Built 7.490e-01 0.453723   
## First\_Flr\_SF 6.330e+00 2.45e-10 \*\*\*  
## Full\_Bath 2.993e+00 0.002766 \*\*   
## MS\_SubClass\_One\_and\_Half\_Story\_Finished\_All\_Ages 3.316e+00 0.000912 \*\*\*  
## MS\_SubClass\_One\_and\_Half\_Story\_PUD\_All\_Ages 8.000e-03 0.993579   
## MS\_SubClass\_One\_and\_Half\_Story\_Unfinished\_All\_Ages -2.400e-02 0.980654   
## MS\_SubClass\_One\_Story\_1945\_and\_Older 2.922e+00 0.003480 \*\*   
## MS\_SubClass\_One\_Story\_1946\_and\_Newer\_All\_Styles 8.430e-01 0.399334   
## MS\_SubClass\_One\_Story\_PUD\_1946\_and\_Newer 5.570e-01 0.577856   
## MS\_SubClass\_One\_Story\_with\_Finished\_Attic\_All\_Ages -2.780e-01 0.780714   
## MS\_SubClass\_PUD\_Multilevel\_Split\_Level\_Foyer -9.000e-02 0.928499   
## MS\_SubClass\_Split\_Foyer 2.086e+00 0.037013 \*   
## MS\_SubClass\_Split\_or\_Multilevel 1.965e+00 0.049368 \*   
## MS\_SubClass\_Two\_and\_Half\_Story\_All\_Ages 3.358e+00 0.000786 \*\*\*  
## MS\_SubClass\_Two\_Family\_conversion\_All\_Styles\_and\_Ages 4.020e-01 0.687753   
## MS\_SubClass\_Two\_Story\_1945\_and\_Older 2.602e+00 0.009269 \*\*   
## MS\_SubClass\_Two\_Story\_1946\_and\_Newer 4.094e+00 4.23e-05 \*\*\*  
## MS\_SubClass\_Two\_Story\_PUD\_1946\_and\_Newer -1.000e-03 0.998991   
## MS\_Zoning\_C\_all 4.000e-03 0.996450   
## MS\_Zoning\_Floating\_Village\_Residential 0.000e+00 0.999637   
## MS\_Zoning\_I\_all NA NA   
## MS\_Zoning\_Residential\_High\_Density 8.000e-03 0.993332   
## MS\_Zoning\_Residential\_Low\_Density 8.000e-03 0.993661   
## MS\_Zoning\_Residential\_Medium\_Density 7.000e-03 0.994379   
## Overall\_Qual\_Average -2.994e+00 0.002756 \*\*   
## Overall\_Qual\_Below\_Average -2.244e+00 0.024808 \*   
## Overall\_Qual\_Excellent -2.679e+00 0.007393 \*\*   
## Overall\_Qual\_Fair -1.000e-03 0.998883   
## Overall\_Qual\_Good 2.159e+00 0.030837 \*   
## Overall\_Qual\_Poor 1.030e-01 0.918191   
## Overall\_Qual\_Very\_Excellent 1.000e-03 0.999566   
## Overall\_Qual\_Very\_Good 1.555e+00 0.119905   
## Overall\_Qual\_Very\_Poor 3.614e+08 < 2e-16 \*\*\*  
## Overall\_Cond\_Average -1.851e+00 0.064209 .   
## Overall\_Cond\_Below\_Average 2.310e-01 0.817222   
## Overall\_Cond\_Excellent 1.013e+00 0.310966   
## Overall\_Cond\_Fair -2.332e+00 0.019676 \*   
## Overall\_Cond\_Good 1.698e+00 0.089506 .   
## Overall\_Cond\_Poor 0.000e+00 0.999943   
## Overall\_Cond\_Very\_Good -7.620e-01 0.445964   
## Overall\_Cond\_Very\_Poor -3.000e-03 0.997704   
## Exterior\_1st\_AsphShn NA NA   
## Exterior\_1st\_BrkComm 0.000e+00 0.999747   
## Exterior\_1st\_BrkFace 3.360e-01 0.737134   
## Exterior\_1st\_CBlock -3.614e+08 < 2e-16 \*\*\*  
## Exterior\_1st\_CemntBd 6.860e-01 0.492438   
## Exterior\_1st\_HdBoard -3.230e-01 0.746490   
## Exterior\_1st\_ImStucc 2.000e-03 0.998663   
## Exterior\_1st\_MetalSd -2.830e-01 0.777297   
## Exterior\_1st\_Plywood -7.720e-01 0.440310   
## Exterior\_1st\_PreCast 6.598e+07 < 2e-16 \*\*\*  
## Exterior\_1st\_Stone NA NA   
## Exterior\_1st\_Stucco -1.687e+00 0.091696 .   
## Exterior\_1st\_VinylSd -1.004e+00 0.315435   
## Exterior\_1st\_Wd.Sdng -5.220e-01 0.601875   
## Exterior\_1st\_WdShing -5.420e-01 0.587764   
## Exterior\_2nd\_AsphShn 0.000e+00 0.999932   
## Exterior\_2nd\_Brk.Cmn 0.000e+00 0.999877   
## Exterior\_2nd\_BrkFace 7.990e-01 0.424351   
## Exterior\_2nd\_CBlock NA NA   
## Exterior\_2nd\_CmentBd 1.052e+00 0.292752   
## Exterior\_2nd\_HdBoard 1.488e+00 0.136779   
## Exterior\_2nd\_ImStucc -2.000e-03 0.998489   
## Exterior\_2nd\_MetalSd 1.370e+00 0.170532   
## Exterior\_2nd\_Other 0.000e+00 0.999963   
## Exterior\_2nd\_Plywood 1.547e+00 0.121919   
## Exterior\_2nd\_PreCast NA NA   
## Exterior\_2nd\_Stone 0.000e+00 0.999966   
## Exterior\_2nd\_Stucco 2.142e+00 0.032166 \*   
## Exterior\_2nd\_VinylSd 1.857e+00 0.063358 .   
## Exterior\_2nd\_Wd.Sdng 1.382e+00 0.167092   
## Exterior\_2nd\_Wd.Shng 1.718e+00 0.085841 .   
## Mas\_Vnr\_Type\_BrkFace 2.257e+00 0.024032 \*   
## Mas\_Vnr\_Type\_CBlock NA NA   
## Mas\_Vnr\_Type\_None 1.665e+00 0.095826 .   
## Mas\_Vnr\_Type\_Stone 1.715e+00 0.086431 .   
## Exter\_Qual\_Fair -1.000e-03 0.999538   
## Exter\_Qual\_Good -1.000e-03 0.999492   
## Exter\_Qual\_Typical -1.000e-03 0.999481   
## Foundation\_CBlock -5.110e-01 0.609457   
## Foundation\_PConc 1.475e+00 0.140270   
## Foundation\_Slab 1.000e-03 0.999441   
## Foundation\_Stone -1.000e-03 0.999329   
## Foundation\_Wood NA NA   
## Bsmt\_Qual\_Fair -9.270e-01 0.354105   
## Bsmt\_Qual\_Good -1.235e+00 0.216868   
## Bsmt\_Qual\_No\_Basement -1.000e-03 0.999348   
## Bsmt\_Qual\_Poor 1.300e-02 0.989278   
## Bsmt\_Qual\_Typical -1.825e+00 0.068003 .   
## Heating\_QC\_Fair 1.066e+00 0.286602   
## Heating\_QC\_Good 1.227e+00 0.219855   
## Heating\_QC\_Poor NA NA   
## Heating\_QC\_Typical 3.420e-01 0.732307   
## Kitchen\_Qual\_Fair -3.614e+08 < 2e-16 \*\*\*  
## Kitchen\_Qual\_Good -1.859e+00 0.063046 .   
## Kitchen\_Qual\_Poor 0.000e+00 0.999948   
## Kitchen\_Qual\_Typical -2.053e+00 0.040098 \*   
## Fireplace\_Qu\_Fair 0.000e+00 0.999760   
## Fireplace\_Qu\_Good 0.000e+00 0.999758   
## Fireplace\_Qu\_No\_Fireplace 0.000e+00 0.999744   
## Fireplace\_Qu\_Poor 0.000e+00 0.999743   
## Fireplace\_Qu\_Typical 0.000e+00 0.999770   
## Neighborhood\_Blueste 0.000e+00 0.999992   
## Neighborhood\_Briardale 1.000e-03 0.998876   
## Neighborhood\_Brookside 1.404e+00 0.160315   
## Neighborhood\_Clear\_Creek 1.914e+00 0.055663 .   
## Neighborhood\_College\_Creek 1.372e+00 0.170146   
## Neighborhood\_Crawford 1.765e+00 0.077531 .   
## Neighborhood\_Edwards 5.000e-01 0.617208   
## Neighborhood\_Gilbert 1.177e+00 0.239074   
## Neighborhood\_Green\_Hills 0.000e+00 0.999858   
## Neighborhood\_Greens 6.160e-01 0.537684   
## Neighborhood\_Iowa\_DOT\_and\_Rail\_Road 1.550e+00 0.121246   
## Neighborhood\_Landmark 0.000e+00 0.999999   
## Neighborhood\_Meadow\_Village 0.000e+00 0.999685   
## Neighborhood\_Mitchell 1.098e+00 0.272143   
## Neighborhood\_North\_Ames 7.470e-01 0.455328   
## Neighborhood\_Northpark\_Villa 0.000e+00 0.999810   
## Neighborhood\_Northridge 1.000e-03 0.999563   
## Neighborhood\_Northridge\_Heights 1.000e-03 0.998896   
## Neighborhood\_Northwest\_Ames 9.040e-01 0.365921   
## Neighborhood\_Old\_Town 5.390e-01 0.589804   
## Neighborhood\_Sawyer 8.200e-01 0.412203   
## Neighborhood\_Sawyer\_West 8.250e-01 0.409326   
## Neighborhood\_Somerset 0.000e+00 0.999669   
## Neighborhood\_South\_and\_West\_of\_Iowa\_State\_University 3.150e-01 0.752803   
## Neighborhood\_Stone\_Brook 1.000e-03 0.999310   
## Neighborhood\_Timberland 2.029e+00 0.042422 \*   
## Neighborhood\_Veenker 1.316e+00 0.188236   
## Garage\_Type\_Basment -9.870e-01 0.323849   
## Garage\_Type\_BuiltIn 1.346e+00 0.178218   
## Garage\_Type\_CarPort 5.910e-01 0.554380   
## Garage\_Type\_Detchd -2.918e+00 0.003518 \*\*   
## Garage\_Type\_More\_Than\_Two\_Types 6.400e-01 0.521905   
## Garage\_Type\_No\_Garage -1.379e+00 0.167774   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1975.38 on 1424 degrees of freedom  
## Residual deviance: 468.11 on 1296 degrees of freedom  
## AIC: 726.11  
##   
## Number of Fisher Scoring iterations: 25

predictions = predict(ames2\_fit3, train, type="prob") #develop predicted probabilities

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

head(predictions)

## # A tibble: 6 × 2  
## .pred\_No .pred\_Yes  
## <dbl> <dbl>  
## 1 0.997 2.96e- 3  
## 2 0.792 2.08e- 1  
## 3 0.974 2.57e- 2  
## 4 1.00 6.61e- 6  
## 5 1 2.22e-16  
## 6 0.892 1.08e- 1

Let’s extract just the “Yes” prediction.

predictions = predict(ames2\_fit3, train, type="prob")[2]

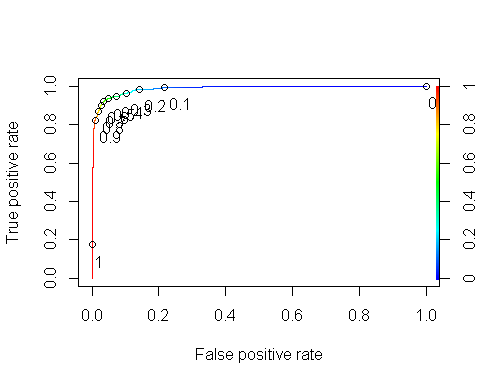
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :  
## prediction from a rank-deficient fit may be misleading

head(predictions)

## # A tibble: 6 × 1  
## .pred\_Yes  
## <dbl>  
## 1 2.96e- 3  
## 2 2.08e- 1  
## 3 2.57e- 2  
## 4 6.61e- 6  
## 5 2.22e-16  
## 6 1.08e- 1

Threshold selection

#Change this next line to the names of your predictions and the response variable in the training data frame  
ROCRpred = prediction(predictions, train$Above\_Median)   
  
###You shouldn't need to ever change the next two lines:  
ROCRperf = performance(ROCRpred, "tpr", "fpr")  
plot(ROCRperf, colorize=TRUE, print.cutoffs.at=seq(0,1,by=0.1), text.adj=c(-0.2,1.7))

 Area under the curve (AUC). AUC is a measure of the strength of the model. Values closer to 1 are better. Can be used to compare models.

as.numeric(performance(ROCRpred, "auc")@y.values)

## [1] 0.9868496

# Let’s just choose a few key predictor variables

Let’s build a model with kitchen qual and house age and type variables.

ames2\_model =   
 logistic\_reg() %>% #note the use of logistic\_reg and mode = "classification"  
 set\_engine("glm") #standard logistic regression engine is glm  
  
ames2\_recipe = recipe(Above\_Median ~ Kitchen\_Qual + MS\_SubClass, train) %>%  
 step\_dummy(all\_nominal(), -all\_outcomes())  
  
logreg\_wf = workflow() %>%  
 add\_recipe(ames2\_recipe) %>%   
 add\_model(ames2\_model)  
  
ames2\_fit3 = fit(logreg\_wf, train)

summary(ames2\_fit3$fit$fit$fit)

##   
## Call:  
## stats::glm(formula = ..y ~ ., family = stats::binomial, data = data)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.7306 -0.7654 0.0979 0.6352 3.3956   
##   
## Coefficients:  
## Estimate Std. Error  
## (Intercept) 1.52740 0.69108  
## Kitchen\_Qual\_Fair -5.81031 1.20041  
## Kitchen\_Qual\_Good -1.63439 0.49227  
## Kitchen\_Qual\_Poor -18.37412 2399.54478  
## Kitchen\_Qual\_Typical -3.84010 0.49122  
## MS\_SubClass\_One\_and\_Half\_Story\_Finished\_All\_Ages 0.28065 0.54576  
## MS\_SubClass\_One\_and\_Half\_Story\_PUD\_All\_Ages -16.45909 2399.54477  
## MS\_SubClass\_One\_and\_Half\_Story\_Unfinished\_All\_Ages -15.43422 665.79581  
## MS\_SubClass\_One\_Story\_1945\_and\_Older -1.47896 0.80190  
## MS\_SubClass\_One\_Story\_1946\_and\_Newer\_All\_Styles 1.23477 0.50221  
## MS\_SubClass\_One\_Story\_PUD\_1946\_and\_Newer 1.74163 0.58674  
## MS\_SubClass\_One\_Story\_with\_Finished\_Attic\_All\_Ages -0.40102 1.29245  
## MS\_SubClass\_PUD\_Multilevel\_Split\_Level\_Foyer -14.25337 979.61015  
## MS\_SubClass\_Split\_Foyer 1.50410 0.65033  
## MS\_SubClass\_Split\_or\_Multilevel 2.21986 0.56869  
## MS\_SubClass\_Two\_and\_Half\_Story\_All\_Ages 1.99420 0.87186  
## MS\_SubClass\_Two\_Family\_conversion\_All\_Styles\_and\_Ages 0.04652 0.81640  
## MS\_SubClass\_Two\_Story\_1945\_and\_Older 0.74623 0.58216  
## MS\_SubClass\_Two\_Story\_1946\_and\_Newer 3.81086 0.55307  
## MS\_SubClass\_Two\_Story\_PUD\_1946\_and\_Newer -0.30879 0.60800  
## z value Pr(>|z|)   
## (Intercept) 2.210 0.02709 \*   
## Kitchen\_Qual\_Fair -4.840 1.30e-06 \*\*\*  
## Kitchen\_Qual\_Good -3.320 0.00090 \*\*\*  
## Kitchen\_Qual\_Poor -0.008 0.99389   
## Kitchen\_Qual\_Typical -7.817 5.39e-15 \*\*\*  
## MS\_SubClass\_One\_and\_Half\_Story\_Finished\_All\_Ages 0.514 0.60708   
## MS\_SubClass\_One\_and\_Half\_Story\_PUD\_All\_Ages -0.007 0.99453   
## MS\_SubClass\_One\_and\_Half\_Story\_Unfinished\_All\_Ages -0.023 0.98151   
## MS\_SubClass\_One\_Story\_1945\_and\_Older -1.844 0.06514 .   
## MS\_SubClass\_One\_Story\_1946\_and\_Newer\_All\_Styles 2.459 0.01394 \*   
## MS\_SubClass\_One\_Story\_PUD\_1946\_and\_Newer 2.968 0.00299 \*\*   
## MS\_SubClass\_One\_Story\_with\_Finished\_Attic\_All\_Ages -0.310 0.75635   
## MS\_SubClass\_PUD\_Multilevel\_Split\_Level\_Foyer -0.015 0.98839   
## MS\_SubClass\_Split\_Foyer 2.313 0.02073 \*   
## MS\_SubClass\_Split\_or\_Multilevel 3.903 9.48e-05 \*\*\*  
## MS\_SubClass\_Two\_and\_Half\_Story\_All\_Ages 2.287 0.02218 \*   
## MS\_SubClass\_Two\_Family\_conversion\_All\_Styles\_and\_Ages 0.057 0.95456   
## MS\_SubClass\_Two\_Story\_1945\_and\_Older 1.282 0.19990   
## MS\_SubClass\_Two\_Story\_1946\_and\_Newer 6.890 5.56e-12 \*\*\*  
## MS\_SubClass\_Two\_Story\_PUD\_1946\_and\_Newer -0.508 0.61154   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1975.4 on 1424 degrees of freedom  
## Residual deviance: 1153.7 on 1405 degrees of freedom  
## AIC: 1193.7  
##   
## Number of Fisher Scoring iterations: 15

Let’s build a model with Overall\_Qual and remodel year variables.

ames2\_model =   
 logistic\_reg() %>% #note the use of logistic\_reg and mode = "classification"  
 set\_engine("glm") #standard logistic regression engine is glm  
  
ames2\_recipe = recipe(Above\_Median ~ Year\_Remod\_Add + Overall\_Qual, train) %>%  
 step\_dummy(all\_nominal(), -all\_outcomes())  
  
logreg\_wf = workflow() %>%  
 add\_recipe(ames2\_recipe) %>%   
 add\_model(ames2\_model)  
  
ames2\_fit3 = fit(logreg\_wf, train)

summary(ames2\_fit3$fit$fit$fit)

##   
## Call:  
## stats::glm(formula = ..y ~ ., family = stats::binomial, data = data)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.98596 -0.52176 0.00035 0.41503 2.63585   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -6.316e+01 8.278e+00 -7.630 2.35e-14 \*\*\*  
## Year\_Remod\_Add 3.186e-02 4.176e-03 7.628 2.38e-14 \*\*\*  
## Overall\_Qual\_Average -1.684e+00 1.833e-01 -9.186 < 2e-16 \*\*\*  
## Overall\_Qual\_Below\_Average -2.850e+00 4.748e-01 -6.003 1.94e-09 \*\*\*  
## Overall\_Qual\_Excellent 3.159e+00 1.021e+00 3.094 0.00197 \*\*   
## Overall\_Qual\_Fair -1.620e+01 5.384e+02 -0.030 0.97599   
## Overall\_Qual\_Good 1.854e+00 2.327e-01 7.967 1.63e-15 \*\*\*  
## Overall\_Qual\_Poor -1.571e+01 1.190e+03 -0.013 0.98947   
## Overall\_Qual\_Very\_Excellent 1.585e+01 6.190e+02 0.026 0.97957   
## Overall\_Qual\_Very\_Good 3.796e+00 7.240e-01 5.243 1.58e-07 \*\*\*  
## Overall\_Qual\_Very\_Poor -1.556e+01 1.385e+03 -0.011 0.99104   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1975.4 on 1424 degrees of freedom  
## Residual deviance: 1033.8 on 1414 degrees of freedom  
## AIC: 1055.8  
##   
## Number of Fisher Scoring iterations: 15

Develop predictions

predictions = predict(ames2\_fit3, train, type="prob") #develop predicted probabilities  
head(predictions)

## # A tibble: 6 × 2  
## .pred\_No .pred\_Yes  
## <dbl> <dbl>  
## 1 0.915 0.0854  
## 2 0.599 0.401   
## 3 0.706 0.294   
## 4 0.886 0.114   
## 5 0.591 0.409   
## 6 0.529 0.471

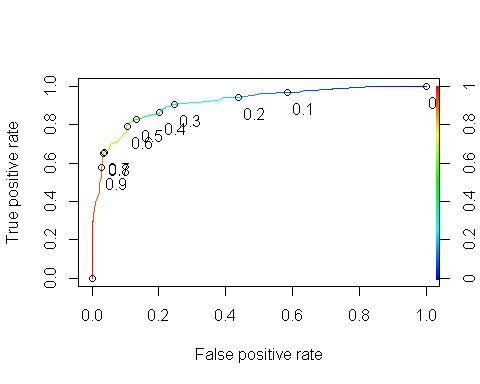
Let’s extract just the “Yes” prediction.

predictions = predict(ames2\_fit3, train, type="prob")[2]  
head(predictions)

## # A tibble: 6 × 1  
## .pred\_Yes  
## <dbl>  
## 1 0.0854  
## 2 0.401   
## 3 0.294   
## 4 0.114   
## 5 0.409   
## 6 0.471

Threshold selection

#Change this next line to the names of your predictions and the response variable in the training data frame  
ROCRpred = prediction(predictions, train$Above\_Median)   
  
###You shouldn't need to ever change the next two lines:  
ROCRperf = performance(ROCRpred, "tpr", "fpr")  
plot(ROCRperf, colorize=TRUE, print.cutoffs.at=seq(0,1,by=0.1), text.adj=c(-0.2,1.7))

 Area under the curve (AUC). AUC is a measure of the strength of the model. Values closer to 1 are better. Can be used to compare models.

as.numeric(performance(ROCRpred, "auc")@y.values)

## [1] 0.9146941

Let’s build a model with full bath, year of remodel, overall quality and first floor sq ft variables.

ames2\_model =   
 logistic\_reg() %>% #note the use of logistic\_reg and mode = "classification"  
 set\_engine("glm") #standard logistic regression engine is glm  
  
ames2\_recipe = recipe(Above\_Median ~ Full\_Bath + Year\_Remod\_Add + Overall\_Qual + First\_Flr\_SF, train) %>%  
 step\_dummy(all\_nominal(), -all\_outcomes())  
  
logreg\_wf = workflow() %>%  
 add\_recipe(ames2\_recipe) %>%   
 add\_model(ames2\_model)  
  
ames2\_fit3 = fit(logreg\_wf, train)

summary(ames2\_fit3$fit$fit$fit)

##   
## Call:  
## stats::glm(formula = ..y ~ ., family = stats::binomial, data = data)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.91286 -0.40469 0.00019 0.36378 2.88298   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -7.084e+01 9.758e+00 -7.260 3.87e-13 \*\*\*  
## Full\_Bath 1.505e+00 1.733e-01 8.682 < 2e-16 \*\*\*  
## Year\_Remod\_Add 3.315e-02 4.885e-03 6.787 1.14e-11 \*\*\*  
## First\_Flr\_SF 2.620e-03 3.134e-04 8.359 < 2e-16 \*\*\*  
## Overall\_Qual\_Average -1.627e+00 2.099e-01 -7.753 8.99e-15 \*\*\*  
## Overall\_Qual\_Below\_Average -2.808e+00 5.165e-01 -5.436 5.44e-08 \*\*\*  
## Overall\_Qual\_Excellent 1.142e+00 1.064e+00 1.073 0.283   
## Overall\_Qual\_Fair -1.746e+01 7.898e+02 -0.022 0.982   
## Overall\_Qual\_Good 1.509e+00 2.566e-01 5.880 4.11e-09 \*\*\*  
## Overall\_Qual\_Poor -1.468e+01 1.829e+03 -0.008 0.994   
## Overall\_Qual\_Very\_Excellent 1.367e+01 9.691e+02 0.014 0.989   
## Overall\_Qual\_Very\_Good 3.223e+00 7.688e-01 4.192 2.77e-05 \*\*\*  
## Overall\_Qual\_Very\_Poor -1.587e+01 2.006e+03 -0.008 0.994   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1975.38 on 1424 degrees of freedom  
## Residual deviance: 840.91 on 1412 degrees of freedom  
## AIC: 866.91  
##   
## Number of Fisher Scoring iterations: 16

Develop predictions

predictions = predict(ames2\_fit3, train, type="prob") #develop predicted probabilities  
head(predictions)

## # A tibble: 6 × 2  
## .pred\_No .pred\_Yes  
## <dbl> <dbl>  
## 1 0.974 0.0264  
## 2 0.802 0.198   
## 3 0.828 0.172   
## 4 0.986 0.0141  
## 5 0.932 0.0678  
## 6 0.456 0.544

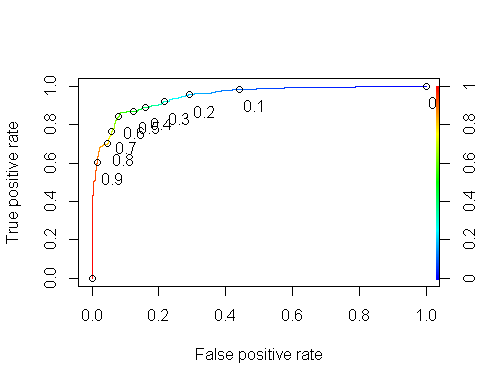
Let’s extract just the “Yes” prediction.

predictions = predict(ames2\_fit3, train, type="prob")[2]  
head(predictions)

## # A tibble: 6 × 1  
## .pred\_Yes  
## <dbl>  
## 1 0.0264  
## 2 0.198   
## 3 0.172   
## 4 0.0141  
## 5 0.0678  
## 6 0.544

Threshold selection

#Change this next line to the names of your predictions and the response variable in the training data frame  
ROCRpred = prediction(predictions, train$Above\_Median)   
  
###You shouldn't need to ever change the next two lines:  
ROCRperf = performance(ROCRpred, "tpr", "fpr")  
plot(ROCRperf, colorize=TRUE, print.cutoffs.at=seq(0,1,by=0.1), text.adj=c(-0.2,1.7))

 Area under the curve (AUC). AUC is a measure of the strength of the model. Values closer to 1 are better. Can be used to compare models.

as.numeric(performance(ROCRpred, "auc")@y.values)

## [1] 0.9469422

#Build on the test with full bath, year of remodel, overall quality and first floor sq ft variables.

ames2\_model =   
 logistic\_reg() %>% #note the use of logistic\_reg and mode = "classification"  
 set\_engine("glm") #standard logistic regression engine is glm  
  
ames2\_recipe = recipe(Above\_Median ~ Full\_Bath + Year\_Remod\_Add + Overall\_Qual + First\_Flr\_SF, test) %>%  
 step\_dummy(all\_nominal(), -all\_outcomes())  
  
logreg\_wf = workflow() %>%  
 add\_recipe(ames2\_recipe) %>%   
 add\_model(ames2\_model)  
  
ames2\_fit3 = fit(logreg\_wf, test)

summary(ames2\_fit3$fit$fit$fit)

##   
## Call:  
## stats::glm(formula = ..y ~ ., family = stats::binomial, data = data)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -3.5413 -0.3643 0.0001 0.4153 2.8117   
##   
## Coefficients: (1 not defined because of singularities)  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -4.825e+01 1.402e+01 -3.441 0.00058 \*\*\*  
## Full\_Bath 1.617e+00 2.570e-01 6.294 3.10e-10 \*\*\*  
## Year\_Remod\_Add 2.180e-02 7.054e-03 3.091 0.00200 \*\*   
## First\_Flr\_SF 2.513e-03 4.667e-04 5.385 7.24e-08 \*\*\*  
## Overall\_Qual\_Average -1.922e+00 3.274e-01 -5.869 4.38e-09 \*\*\*  
## Overall\_Qual\_Below\_Average -3.122e+00 7.806e-01 -4.000 6.34e-05 \*\*\*  
## Overall\_Qual\_Excellent 1.478e+01 8.772e+02 0.017 0.98655   
## Overall\_Qual\_Fair -1.629e+01 1.166e+03 -0.014 0.98886   
## Overall\_Qual\_Good 8.530e-01 3.485e-01 2.448 0.01437 \*   
## Overall\_Qual\_Poor -1.505e+01 2.787e+03 -0.005 0.99569   
## Overall\_Qual\_Very\_Excellent 1.395e+01 1.696e+03 0.008 0.99344   
## Overall\_Qual\_Very\_Good 3.020e+00 1.051e+00 2.874 0.00406 \*\*   
## Overall\_Qual\_Very\_Poor NA NA NA NA   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 848.35 on 611 degrees of freedom  
## Residual deviance: 374.55 on 600 degrees of freedom  
## AIC: 398.55  
##   
## Number of Fisher Scoring iterations: 16

Develop predictions

predictions = predict(ames2\_fit3, test, type="prob") #develop predicted probabilities

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

head(predictions)

## # A tibble: 6 × 2  
## .pred\_No .pred\_Yes  
## <dbl> <dbl>  
## 1 0.436 0.564  
## 2 0.648 0.352  
## 3 0.296 0.704  
## 4 0.00950 0.991  
## 5 0.152 0.848  
## 6 0.0312 0.969

Let’s extract just the “Yes” prediction.

predictions = predict(ames2\_fit3, test, type="prob")[2]

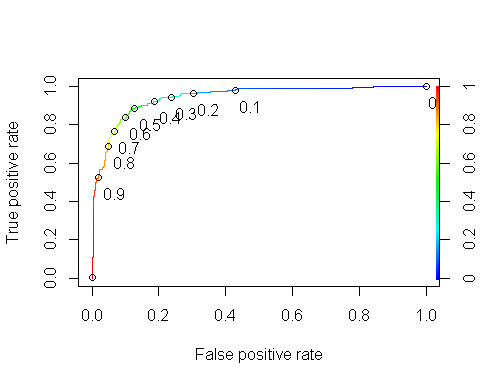
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :  
## prediction from a rank-deficient fit may be misleading

head(predictions)

## # A tibble: 6 × 1  
## .pred\_Yes  
## <dbl>  
## 1 0.564  
## 2 0.352  
## 3 0.704  
## 4 0.991  
## 5 0.848  
## 6 0.969

Threshold selection

#Change this next line to the names of your predictions and the response variable in the training data frame  
ROCRpred = prediction(predictions, test$Above\_Median)   
  
###You shouldn't need to ever change the next two lines:  
ROCRperf = performance(ROCRpred, "tpr", "fpr")  
plot(ROCRperf, colorize=TRUE, print.cutoffs.at=seq(0,1,by=0.1), text.adj=c(-0.2,1.7))



as.numeric(performance(ROCRpred, "auc")@y.values)

## [1] 0.9422709