##Reading in the Data##

student <- read\_csv("ames\_student.csv")

## Warning: Missing column names filled in: 'X1' [1]

##   
## ── Column specification ────────────────────────────────────────────────────────  
## cols(  
## .default = col\_character(),  
## X1 = col\_double(),  
## Lot\_Frontage = col\_double(),  
## Lot\_Area = col\_double(),  
## Year\_Built = col\_double(),  
## Year\_Remod\_Add = col\_double(),  
## Mas\_Vnr\_Area = col\_double(),  
## BsmtFin\_SF\_1 = col\_double(),  
## BsmtFin\_SF\_2 = col\_double(),  
## Bsmt\_Unf\_SF = col\_double(),  
## Total\_Bsmt\_SF = col\_double(),  
## First\_Flr\_SF = col\_double(),  
## Second\_Flr\_SF = col\_double(),  
## Low\_Qual\_Fin\_SF = col\_double(),  
## Gr\_Liv\_Area = col\_double(),  
## Bsmt\_Full\_Bath = col\_double(),  
## Bsmt\_Half\_Bath = col\_double(),  
## Full\_Bath = col\_double(),  
## Half\_Bath = col\_double(),  
## Bedroom\_AbvGr = col\_double(),  
## Kitchen\_AbvGr = col\_double()  
## # ... with 15 more columns  
## )  
## ℹ Use `spec()` for the full column specifications.

#summary(student)  
#str(student)  
#glimpse(student)

competition <- read\_csv("ames\_competition.csv")

##   
## ── Column specification ────────────────────────────────────────────────────────  
## cols(  
## .default = col\_character(),  
## ID = col\_double(),  
## Lot\_Frontage = col\_double(),  
## Lot\_Area = col\_double(),  
## Year\_Built = col\_double(),  
## Year\_Remod\_Add = col\_double(),  
## Mas\_Vnr\_Area = col\_double(),  
## BsmtFin\_SF\_1 = col\_double(),  
## BsmtFin\_SF\_2 = col\_double(),  
## Bsmt\_Unf\_SF = col\_double(),  
## Total\_Bsmt\_SF = col\_double(),  
## First\_Flr\_SF = col\_double(),  
## Second\_Flr\_SF = col\_double(),  
## Low\_Qual\_Fin\_SF = col\_double(),  
## Gr\_Liv\_Area = col\_double(),  
## Bsmt\_Full\_Bath = col\_double(),  
## Bsmt\_Half\_Bath = col\_double(),  
## Full\_Bath = col\_double(),  
## Half\_Bath = col\_double(),  
## Bedroom\_AbvGr = col\_double(),  
## Kitchen\_AbvGr = col\_double()  
## # ... with 15 more columns  
## )  
## ℹ Use `spec()` for the full column specifications.

##Consolidating Variables##

student2 <- student %>% dplyr::select("Above\_Median", "Lot\_Area", "Neighborhood", "House\_Style", "Overall\_Cond", "Year\_Built", "Roof\_Style", "Exter\_Cond", "Central\_Air", "Full\_Bath", "Pool\_QC", "Kitchen\_Qual")

##Recoding variables to factors##

student2 <- student2 %>%   
 mutate(Above\_Median = as\_factor(Above\_Median)) %>%  
 mutate(Above\_Median = fct\_recode(Above\_Median, "No" = "0", "Yes" = "1" )) %>%   
  
 mutate(Neighborhood = as\_factor(Neighborhood)) %>%  
   
 mutate(House\_Style = as\_factor(House\_Style)) %>%  
   
 mutate(Overall\_Cond = as\_factor(Overall\_Cond)) %>%   
 mutate(Overall\_Cond = fct\_recode(Overall\_Cond, "Very Poor" = "1", "Poor" = "2","Fair" = "3", "Below\_Average" = "4","Average" = "5", "Above\_Average" = "6", "Good" = "7", "Very\_Good" = "8", "Excellent" = "9", "Very\_Excellent" = "10")) %>%  
   
 mutate(Roof\_Style = as\_factor(Roof\_Style)) %>%  
   
 mutate(Exter\_Cond = as\_factor(Exter\_Cond)) %>%   
 mutate(Exter\_Cond = fct\_recode(Exter\_Cond, "Poor" = "0", "Fair" = "1", "Typical" = "2", "Good" = "3", "Excellent" = "4")) %>%  
   
 mutate(Central\_Air = as\_factor(Central\_Air)) %>%  
 mutate(Central\_Air = fct\_recode(Central\_Air, "N" = "0", "Y" = "1" )) %>%  
   
 mutate(Full\_Bath = as\_factor(Full\_Bath)) %>%  
   
 mutate(Pool\_QC = as\_factor(Pool\_QC)) %>%  
 mutate(Pool\_QC = fct\_recode(Pool\_QC, "No\_Pool" = "0", "Fair" = "1", "Typical" = "2", "Good" = "3", "Excellent" = "4")) %>%  
   
 mutate(Kitchen\_Qual = as\_factor(Kitchen\_Qual)) %>%   
 mutate(Kitchen\_Qual = fct\_recode(Kitchen\_Qual, "Poor" = "0", "Fair" = "1", "Typical" = "2", "Good" = "3", "Excellent" = "4"))

## Warning: Unknown levels in `f`: 0, 1

## Warning: Unknown levels in `f`: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

## Warning: Unknown levels in `f`: 0, 1, 2, 3, 4

## Warning: Unknown levels in `f`: 0, 1

## Warning: Unknown levels in `f`: 0, 1, 2, 3, 4  
  
## Warning: Unknown levels in `f`: 0, 1, 2, 3, 4

##Examining individual neighborhoods##

table(student2$Neighborhood)

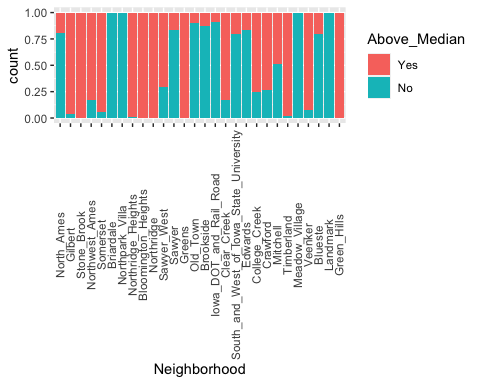
##   
## North\_Ames Gilbert   
## 327 109   
## Stone\_Brook Northwest\_Ames   
## 39 95   
## Somerset Briardale   
## 119 22   
## Northpark\_Villa Northridge\_Heights   
## 15 105   
## Bloomington\_Heights Northridge   
## 21 50   
## Sawyer\_West Sawyer   
## 82 109   
## Greens Old\_Town   
## 7 181   
## Brookside Iowa\_DOT\_and\_Rail\_Road   
## 74 57   
## Clear\_Creek South\_and\_West\_of\_Iowa\_State\_University   
## 31 35   
## Edwards College\_Creek   
## 129 183   
## Crawford Mitchell   
## 77 79   
## Timberland Meadow\_Village   
## 60 24   
## Veenker Blueste   
## 16 5   
## Landmark Green\_Hills   
## 1 1

##Splitting the data##

set.seed(123)   
student\_split = initial\_split(student2, prop = 0.7, strata = Above\_Median) #70% in training  
train = training(student\_split)  
test = testing(student\_split)

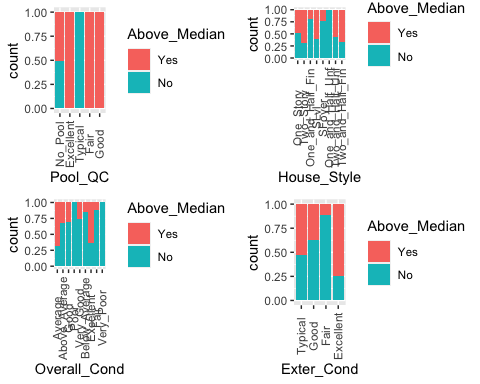
##Plotting neighborhoods first##

p1 = ggplot(train, aes(x = Neighborhood, fill = Above\_Median)) + geom\_bar(position = "fill") + theme(axis.text.x = element\_text(angle = 90, vjust=0.5))  
grid.arrange(p1)



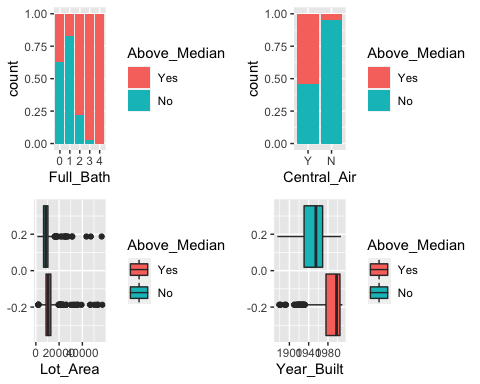
##Plotting more bar charts##

p1 = ggplot(train, aes(x = Pool\_QC, fill = Above\_Median)) + geom\_bar(position = "fill") + theme(axis.text.x = element\_text(angle = 90, vjust=0.5))  
p2 = ggplot(train, aes(x = House\_Style, fill = Above\_Median)) + geom\_bar(position = "fill") + theme(axis.text.x = element\_text(angle = 90, vjust=0.5))  
p3 = ggplot(train, aes(x = Overall\_Cond, fill = Above\_Median)) + geom\_bar(position = "fill") + theme(axis.text.x = element\_text(angle = 90, vjust=0.5))  
p4 = ggplot(train, aes(x = Exter\_Cond, fill = Above\_Median)) + geom\_bar(position = "fill") + theme(axis.text.x = element\_text(angle = 90, vjust=0.5))  
grid.arrange(p1,p2,p3,p4)

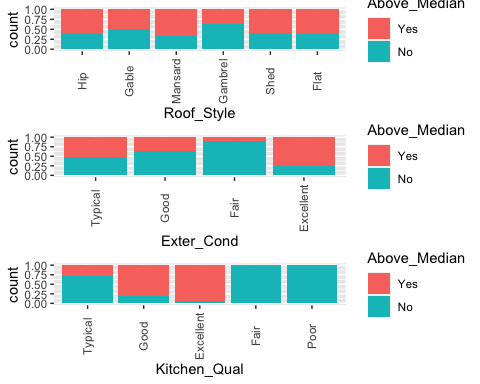


##Plotting nominal variables via box-plots##

p1 = ggplot(train, aes(x = Full\_Bath, fill = Above\_Median)) + geom\_bar(position = "fill")  
p2 = ggplot(train, aes(x = Central\_Air, fill = Above\_Median)) + geom\_bar(position = "fill")  
p3 = ggplot(train, aes(x = Lot\_Area, fill = Above\_Median)) + geom\_boxplot()  
p4 = ggplot(train, aes(x = Year\_Built, fill = Above\_Median)) + geom\_boxplot()  
grid.arrange(p1,p2,p3,p4)



p1 = ggplot(train, aes(x = Roof\_Style, fill = Above\_Median)) + geom\_bar(position = "fill") + theme(axis.text.x = element\_text(angle = 90, vjust=0.5))  
p2 = ggplot(train, aes(x = Exter\_Cond, fill = Above\_Median)) + geom\_bar(position = "fill") + theme(axis.text.x = element\_text(angle = 90, vjust=0.5))  
p3 = ggplot(train, aes(x = Kitchen\_Qual, fill = Above\_Median)) + geom\_bar(position = "fill") + theme(axis.text.x = element\_text(angle = 90, vjust=0.5))  
grid.arrange(p1,p2,p3)



##Further examining variables that stuck out in our visuals##

t1 = table(train$Above\_Median, train$Overall\_Cond)  
prop.table(t1, margin = 2)

##   
## Average Above\_Average Good Poor Very\_Good Below\_Average  
## Yes 0.6826196 0.3283019 0.3010204 0.0000000 0.2567568 0.1481481  
## No 0.3173804 0.6716981 0.6989796 1.0000000 0.7432432 0.8518519  
##   
## Excellent Fair Very\_Poor  
## Yes 0.6315789 0.1153846 0.0000000  
## No 0.3684211 0.8846154 1.0000000

t1 = table(train$Above\_Median, train$Central\_Air)  
prop.table(t1, margin = 2)

##   
## Y N  
## Yes 0.54347826 0.04854369  
## No 0.45652174 0.95145631

t1 = table(train$Above\_Median, train$Full\_Bath)  
prop.table(t1, margin = 2)

##   
## 0 1 2 3 4  
## Yes 0.37500000 0.17027864 0.77882038 0.97142857 1.00000000  
## No 0.62500000 0.82972136 0.22117962 0.02857143 0.00000000

t1 = table(train$Above\_Median, train$Pool\_QC)  
prop.table(t1, margin = 2)

##   
## No\_Pool Excellent Typical Fair Good  
## Yes 0.5073273 1.0000000 0.0000000 1.0000000 1.0000000  
## No 0.4926727 0.0000000 1.0000000 0.0000000 0.0000000

folds = vfold\_cv(train, v = 10)