R Notebook

Code for housing data.

```
#read in data and inspect it
housingdat = read.csv("train.csv")
names(housingdat)
```

```
##  [1] "Id"            "MSSubClass"    "MSZoning"      "LotFrontage"
##  [5] "LotArea"       "Street"        "Alley"         "LotShape"
##  [9] "LandContour"   "Utilities"     "LotConfig"     "LandSlope"
## [13] "Neighborhood"  "Condition1"    "Condition2"    "BldgType"
## [17] "HouseStyle"    "OverallQual"   "OverallCond"   "YearBuilt"
## [21] "YearRemodAdd"  "RoofStyle"     "RoofMatl"      "Exterior1st"
## [25] "Exterior2nd"   "MasVnrType"    "MasVnrArea"    "ExterQual"
## [29] "ExterCond"     "Foundation"    "BsmtQual"      "BsmtCond"
## [33] "BsmtExposure"  "BsmtFinType1"  "BsmtFinSF1"    "BsmtFinType2"
## [37] "BsmtFinSF2"    "BsmtUnfSF"     "TotalBsmtSF"   "Heating"
## [41] "HeatingQC"     "CentralAir"    "Electrical"    "X1stFlrSF"
## [45] "X2ndFlrSF"     "LowQualFinSF"  "GrLivArea"     "BsmtFullBath"
## [49] "BsmtHalfBath"  "FullBath"      "HalfBath"      "BedroomAbvGr"
## [53] "KitchenAbvGr"  "KitchenQual"   "TotRmsAbvGrd"  "Functional"
## [57] "Fireplaces"    "FireplaceQu"   "GarageType"    "GarageYrBlt"
## [61] "GarageFinish"  "GarageCars"    "GarageArea"    "GarageQual"
## [65] "GarageCond"    "PavedDrive"    "WoodDeckSF"    "OpenPorchSF"
## [69] "EnclosedPorch" "X3SsnPorch"    "ScreenPorch"   "PoolArea"
## [73] "PoolQC"        "Fence"         "MiscFeature"   "MiscVal"
## [77] "MoSold"        "YrSold"        "SaleType"      "SaleCondition"
## [81] "SalePrice"
```

```
#get your data (you need to adjust for whichever variables you were assigned. I had 40 thru 60)
housing = housingdat[, 40:60]
names(housing)
```

```
##  [1] "Heating"      "HeatingQC"    "CentralAir"   "Electrical"
##  [5] "X1stFlrSF"    "X2ndFlrSF"    "LowQualFinSF" "GrLivArea"
##  [9] "BsmtFullBath" "BsmtHalfBath" "FullBath"     "HalfBath"
## [13] "BedroomAbvGr" "KitchenAbvGr" "KitchenQual"  "TotRmsAbvGrd"
## [17] "Functional"   "Fireplaces"   "FireplaceQu"  "GarageType"
## [21] "GarageYrBlt"
```

```
str(housing)
```

```
## 'data.frame':    1460 obs. of  21 variables:
##  $ Heating     : Factor w/ 6 levels "Floor","GasA",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ HeatingQC   : Factor w/ 5 levels "Ex","Fa","Gd",..: 1 1 1 3 1 1 1 1 3 1 ...
##  $ CentralAir  : Factor w/ 2 levels "N","Y": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Electrical  : Factor w/ 5 levels "FuseA","FuseF",..: 5 5 5 5 5 5 5 5 2 5 ...
##  $ X1stFlrSF   : int  856 1262 920 961 1145 796 1694 1107 1022 1077 ...
##  $ X2ndFlrSF   : int  854 0 866 756 1053 566 0 983 752 0 ...
##  $ LowQualFinSF: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ GrLivArea   : int  1710 1262 1786 1717 2198 1362 1694 2090 1774 1077 ...
##  $ BsmtFullBath: int  1 0 1 1 1 1 1 1 0 1 ...
##  $ BsmtHalfBath: int  0 1 0 0 0 0 0 0 0 0 ...
```

```
##  $ FullBath    : int  2 2 2 1 2 1 2 2 2 1 ...
##  $ HalfBath    : int  1 0 1 0 1 1 0 1 0 0 ...
##  $ BedroomAbvGr: int  3 3 3 3 4 1 3 3 2 2 ...
##  $ KitchenAbvGr: int  1 1 1 1 1 1 1 1 2 2 ...
##  $ KitchenQual : Factor w/ 4 levels "Ex","Fa","Gd",..: 3 4 3 3 3 4 3 4 4 4 ...
##  $ TotRmsAbvGrd: int  8 6 6 7 9 5 7 7 8 5 ...
##  $ Functional  : Factor w/ 7 levels "Maj1","Maj2",..: 7 7 7 7 7 7 7 7 3 7 ...
##  $ Fireplaces  : int  0 1 1 1 1 0 1 2 2 2 ...
##  $ FireplaceQu : Factor w/ 5 levels "Ex","Fa","Gd",..: NA 5 5 3 5 NA 3 5 5 5 ...
##  $ GarageType  : Factor w/ 6 levels "2Types","Attchd",..: 2 2 2 6 2 2 2 2 6 2 ...
##  $ GarageYrBlt : int  2003 1976 2001 1998 2000 1993 2004 1973 1931 1939 ...
```

```
#how many NA values do we have?
NaData = apply(housing, 2, function(x) sum(is.na(x)))
NaData
```

```
##      Heating    HeatingQC   CentralAir   Electrical    X1stFlrSF
##            0            0            0            1            0
##    X2ndFlrSF LowQualFinSF    GrLivArea BsmtFullBath BsmtHalfBath
##            0            0            0            0            0
##     FullBath     HalfBath BedroomAbvGr KitchenAbvGr  KitchenQual
##            0            0            0            0            0
## TotRmsAbvGrd   Functional   Fireplaces  FireplaceQu   GarageType
##            0            0            0          690           81
##  GarageYrBlt
##           81
```

FireplaceQu = NA means no fireplace. We may consider changing this to "none". GarageYrBlt = NA likely means there is no information available. We should probably leave this as NA. GarageType = NA means no garage. We may consider changing this to "none".

```
#get logical list (and then convert to vector) to get names of integer and factor valued variables
library(purrr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:purrr':
##
##     contains, order_by
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
# housing %>%
str(mtcars)
```

```
## 'data.frame':    32 obs. of  11 variables:
##  $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
##  $ cyl : num  6 6 4 6 8 6 8 4 4 6 ...
##  $ disp: num  160 160 108 258 360 ...
##  $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
```

```
##  $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
##  $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
##  $ qsec: num  16.5 17 18.6 19.4 17 ...
##  $ vs  : num  0 0 1 1 0 1 0 1 1 1 ...
##  $ am  : num  1 1 1 0 0 0 0 0 0 0 ...
##  $ gear: num  4 4 4 3 3 3 3 4 4 4 ...
##  $ carb: num  4 4 1 1 2 1 4 2 2 4 ...
```

```
m = housing %>%
  map(is.factor)
um = unlist(m)

yesfactor = um[um == TRUE]
yesinteger = um[um == FALSE]
```

Split data into 2 datasets-1 with factor-valued variables and another with numeric variables.

```
############################################################
#function to get a list of integer and factor variables
############################################################
my_integers = list(NULL)
my_factors = list(NULL)
for (i in seq_along(names(housing))){
  if (class(housing[,i]) == "integer"){
    my_integers[[i]] = housing[,i]
  }
  else if (class(housing[,i]) == "factor"){
    my_factors[[i]] = housing[, i]
  }
}
```

Do more manipulation to extract the 2 datasets

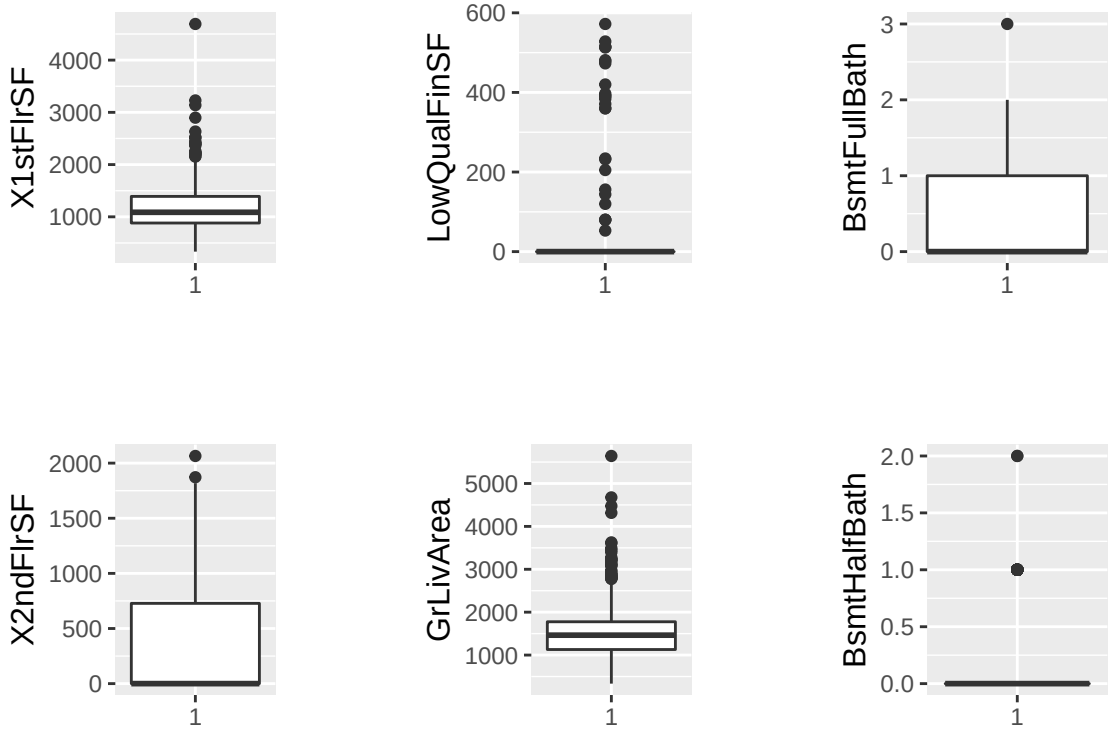
```
# Now we have to get rid of null entries
m = lapply(my_integers, function(x) is.null(x))
n = lapply(my_factors, function(x) is.null(x))
onlyints = my_integers[m == FALSE]
onlyfactors = my_factors[n == FALSE]

#make the lists into a dataframe
intsdf = as.data.frame(onlyints)
factsdf = as.data.frame(onlyfactors)
intnames = names(yesinteger)
factnames = names(yesfactor)

#assign proper names to variables!
names(intsdf) = intnames
names(factsdf) = factnames
names(intsdf)
```

```
##  [1] "X1stFlrSF"    "X2ndFlrSF"    "LowQualFinSF" "GrLivArea"
##  [5] "BsmtFullBath" "BsmtHalfBath" "FullBath"     "HalfBath"
##  [9] "BedroomAbvGr" "KitchenAbvGr" "TotRmsAbvGrd" "Fireplaces"
## [13] "GarageYrBlt"
```

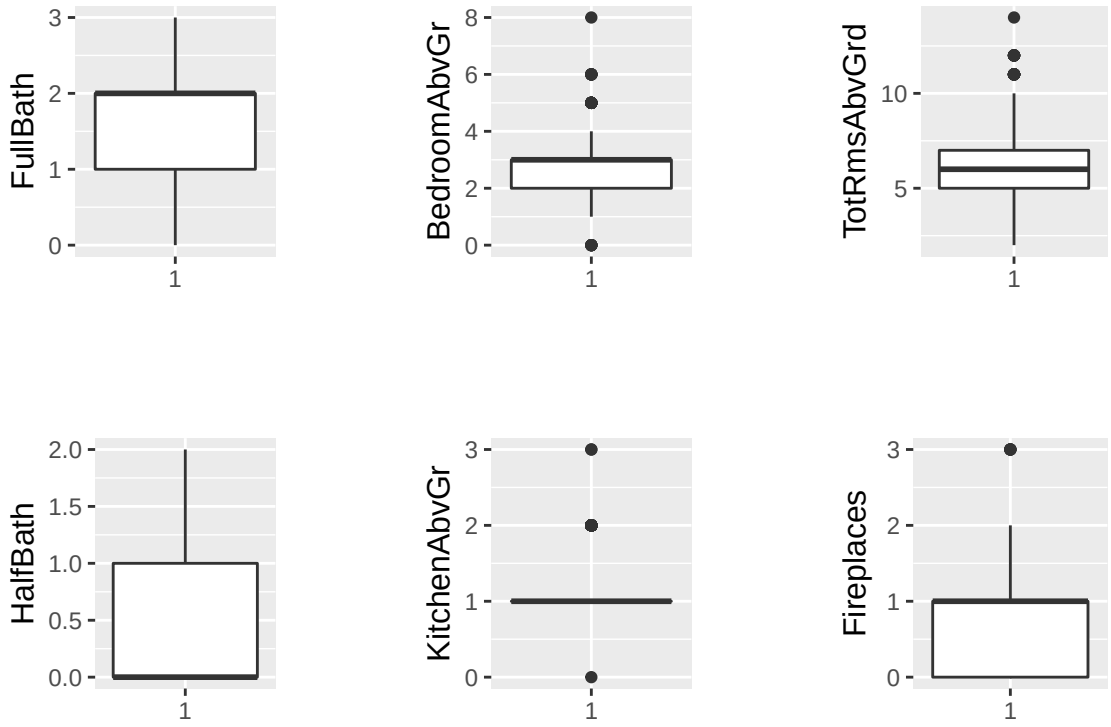
```
names(factsdf)
```

```
## [1] "Heating"     "HeatingQC"   "CentralAir"  "Electrical"  "KitchenQual"
## [6] "Functional"  "FireplaceQu" "GarageType"
```

```
#############################################################################################################
# now we can create a function to output boxplots and histograms of integer and factor variables.
#############################################################################################################
library(ggplot2)
#integer valued variable plots (boxplots)
myplots_ints = function(data){
  allvars=names(data)
  varcols = ncol(data)
  varnames = allvars
  listofplots=list(NULL)
  for (i in seq_along(varnames)){
    listofplots[[i]]=
      ggplot(data, aes_string(x = factor(1), y = varnames[i])) +
      geom_boxplot(width = .8) +
      theme(axis.title.x = element_blank(),
            plot.margin = unit(c(1,1,1,1), "cm"),
            axis.title.y = element_text(size=12))
  }
  return(listofplots)
}

#run funciton on intsdf and get lots of boxplots
try1 = myplots_ints(intsdf)
multiplot(plotlist = try1[1:6], cols = 3)
```

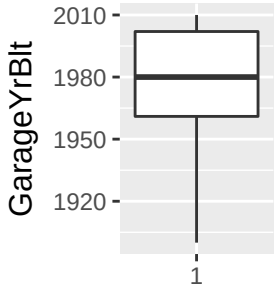


```
multiplot(plotlist = try1[7:12], cols = 3)
```




```
multiplot(plotlist = try1[13:18], cols = 3)
```

```
## Warning: Removed 81 rows containing non-finite values (stat_boxplot).
```




```
## NULL
## NULL
## NULL
## NULL
## NULL
```

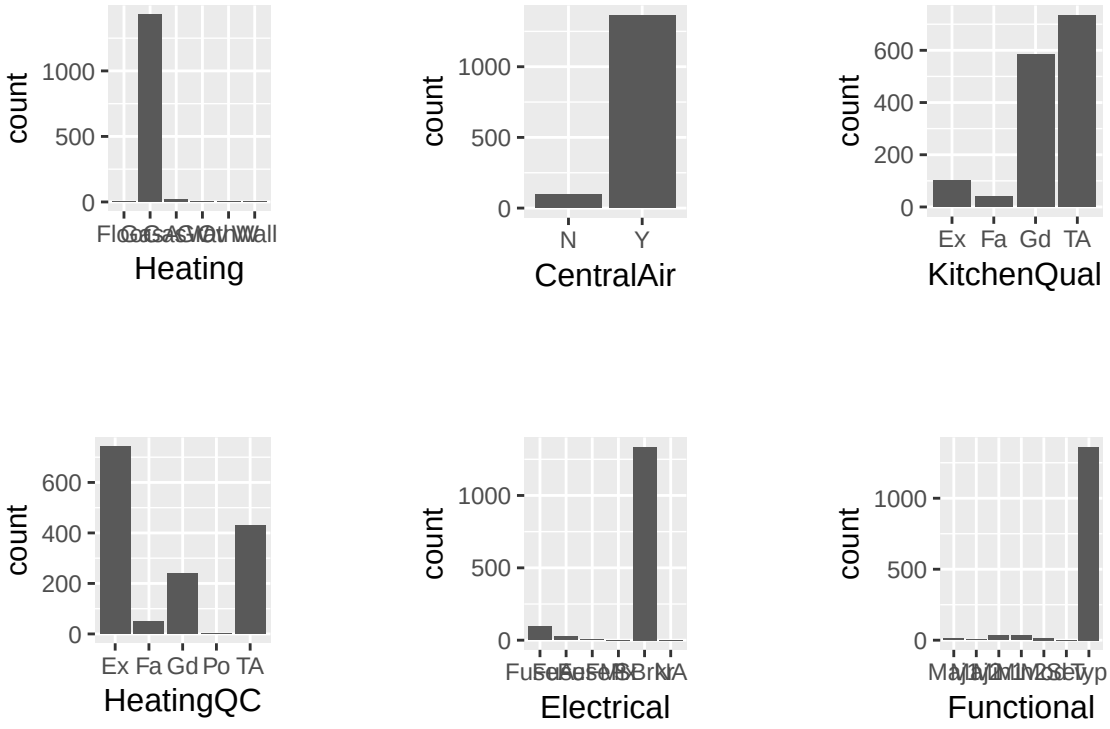
Do the same for factor variables. Generally, data is fairly unbalanced.

```
#histogram function for factor valued variables
myplots_facts = function(data){
  allvars=names(data)
  varcols = ncol(data)
  varnames = allvars
  listofplots=list(NULL)
  for (i in seq_along(varnames)){
    listofplots[[i]]=
      ggplot(data, aes_string(varnames[i])) +
      geom_histogram(stat = "count") +
      theme(
            plot.margin = unit(c(1,1,1,1), "cm"),
            axis.title.x = element_text(size=12))
  }
  return(listofplots)
}


# run on the factor dataframe
```

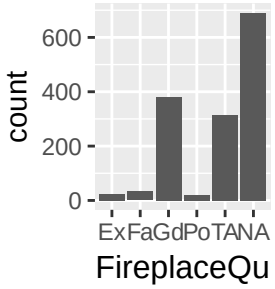
```
try2 = myplots_facts(factsdf)
```

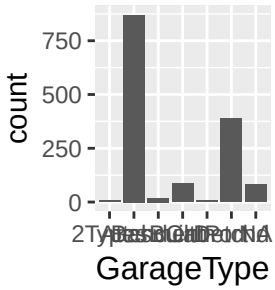
```
## Warning: Ignoring unknown parameters: binwidth, bins, pad

## Warning: Ignoring unknown parameters: binwidth, bins, pad

## Warning: Ignoring unknown parameters: binwidth, bins, pad

## Warning: Ignoring unknown parameters: binwidth, bins, pad

## Warning: Ignoring unknown parameters: binwidth, bins, pad

## Warning: Ignoring unknown parameters: binwidth, bins, pad

## Warning: Ignoring unknown parameters: binwidth, bins, pad

## Warning: Ignoring unknown parameters: binwidth, bins, pad
```

```
multiplot(plotlist = try2[1:6], cols = 3)
```




```
multiplot(plotlist = try2[7:12], cols = 3)
```






```
## NULL
## NULL
## NULL
## NULL
```

```
#multiplot(plotlist = try2[13:18], cols = 3)
```

Do some variable selection with random forests.

```
library(randomForest)
```

```
## randomForest 4.6-12
```

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

```
##
## Attaching package: 'randomForest'
```

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

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

```
housenames = names(housingdat)
paste(housenames,collapse = "+")
```

```
## [1] "Id+MSSubClass+MSZoning+LotFrontage+LotArea+Street+Alley+LotShape+LandContour+Utilities+LotConfi
```

X2ndFlrSF and GRLivArea and ScreenPorch appear to be among the most important variables.

```
myRF = randomForest(SalePrice ~ MSSubClass+MSZoning+LotFrontage+LotArea+Street+Alley+LotShape+LandContou
                    Neighborhood+Condition1+Condition2+BldgType+HouseStyle+OverallQual+OverallCond+Yea
                    RoofMatl+Exterior1st+Exterior2nd+MasVnrType+MasVnrArea +ExterQual+ExterCond+Founda
                    BsmtFinType1+BsmtFinSF1+BsmtFinType2+BsmtFinSF2+BsmtUnfSF  +TotalBsmtSF+Heating+He
                    KitchenQual+ TotRmsAbvGrd+Functional+Fireplaces+FireplaceQu+GarageType +
                    X2ndFlrSF+LowQualFinSF+GrLivArea+BsmtFullBath+BsmtHalfBath+FullBath+HalfBath+Bedro
                    +GarageYrBlt+GarageFinish+GarageCars+GarageArea+GarageQual + GarageCond+PavedDrive
                    X3SsnPorch + ScreenPorch + PoolArea + MoSold,
                    data = housingdat, na.action = na.omit)
varImpPlot(myRF)
```