573 Group Project

2025-04-13

The Model

Preparing the Data

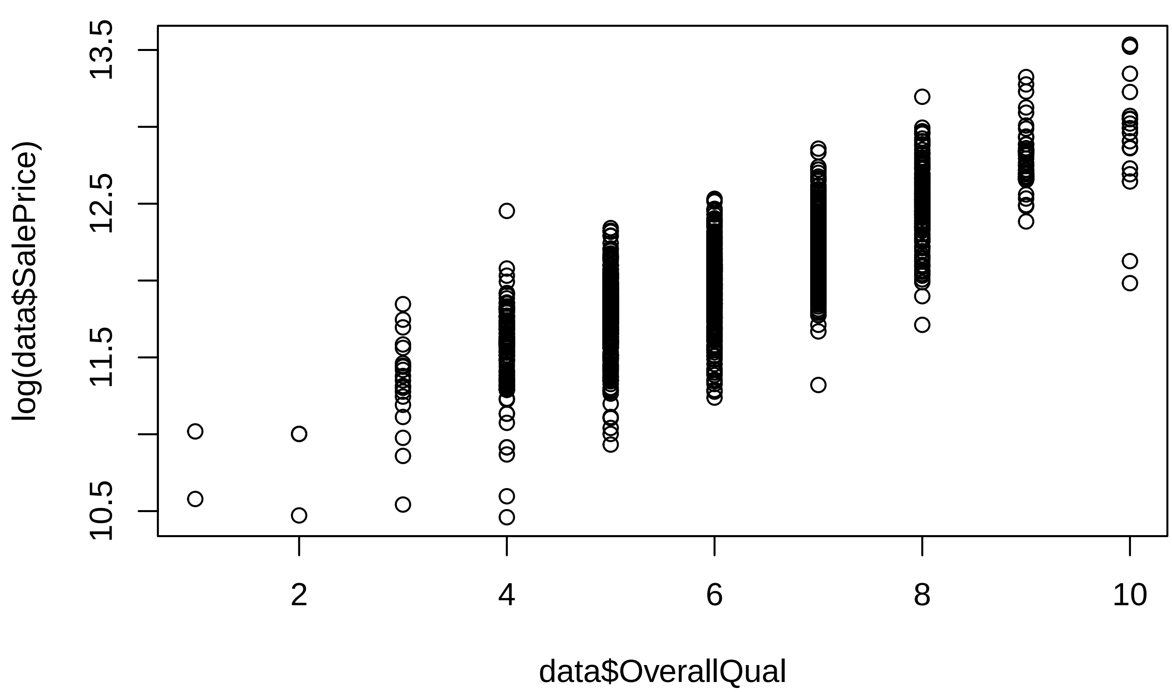
```r
set.seed(1)
data <- read.csv("train.csv")
data[is.na(data)]<-0

# factor the appropriate columns
data$MSSubClass <- factor(data$MSSubClass)
data$MSZoning <- factor(data$MSZoning)
data$Street <- factor(data$Street)
data$Alley <- factor(data$Alley)
data$LotShape <- factor(data$LotShape)
data$LandContour <- factor(data$LandContour)
data$Utilities <- factor(data$Utilities)
data$LotConfig <- factor(data$LotConfig)
data$LandSlope <- factor(data$LandSlope)
data$Neighborhood <- factor(data$Neighborhood)
data$Condition1 <- factor(data$Condition1)
data$Condition2 <- factor(data$Condition2)
data$BldgType <- factor(data$BldgType)
data$HouseStyle <- factor(data$HouseStyle)
data$RoofStyle <- factor(data$RoofStyle)
data$RoofMatl <- factor(data$RoofMatl)
data$Exterior1st <- factor(data$Exterior1st)
data$Exterior2nd <- factor(data$Exterior2nd)
data$MasVnrType <- factor(data$MasVnrType)
data$ExterQual <- factor(data$ExterQual)
data$ExterCond <- factor(data$ExterCond)
data$Foundation <- factor(data$Foundation)
data$BsmtQual <- factor(data$BsmtQual)
data$BsmtCond <- factor(data$BsmtCond)
data$BsmtExposure <- factor(data$BsmtExposure)
data$BsmtFinType1 <- factor(data$BsmtFinType1)
data$BsmtFinType2 <- factor(data$BsmtFinType2)
data$Heating <- factor(data$Heating)
data$HeatingQC <- factor(data$HeatingQC)
data$CentralAir <- factor(data$CentralAir)
data$Electrical <- factor(data$Electrical)
data$KitchenQual <- factor(data$KitchenQual)
data$Functional <- factor(data$Functional)
data$FireplaceQu <- factor(data$FireplaceQu)
data$GarageType <- factor(data$GarageType)
```

```r
data$GarageFinish <- factor(data$GarageFinish)
data$GarageQual <- factor(data$GarageQual)
data$GarageCond <- factor(data$GarageCond)
data$PavedDrive <- factor(data$PavedDrive)
data$PoolQC <- factor(data$PoolQC)
data$Fence <- factor(data$Fence)
data$MiscFeature <- factor(data$MiscFeature)
data$SaleType <- factor(data$SaleType)
data$SaleCondition <- factor(data$SaleCondition)
```

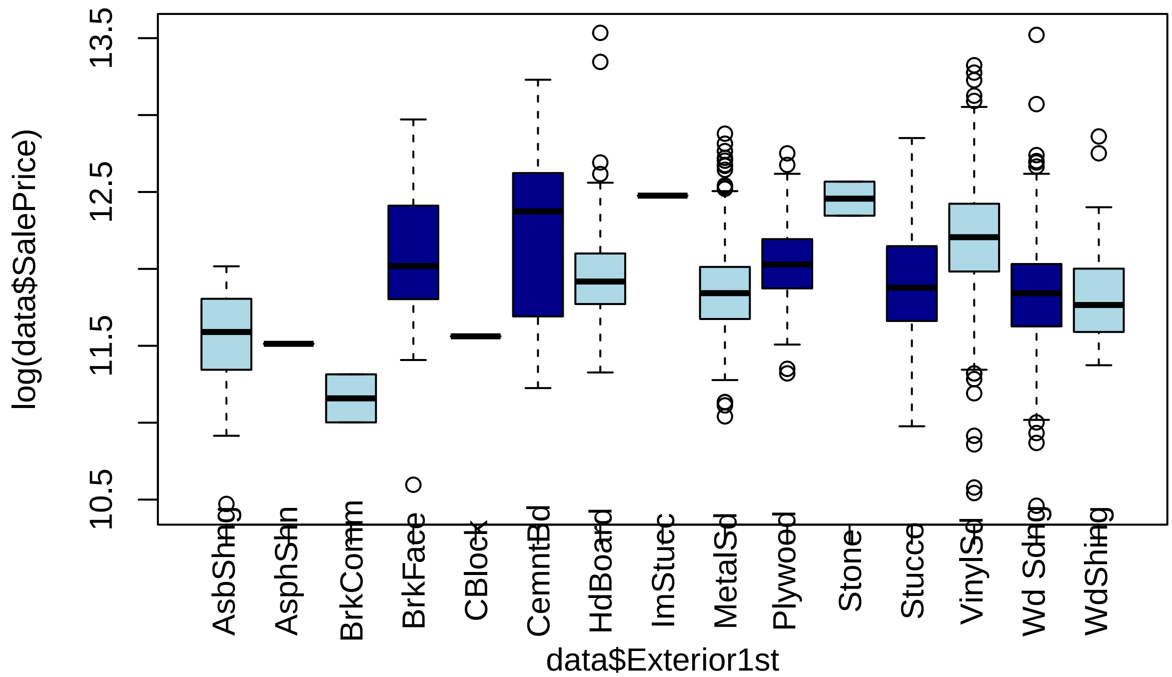
Exploratory visuals

This confirms that the data broadly looks how we expect it to.

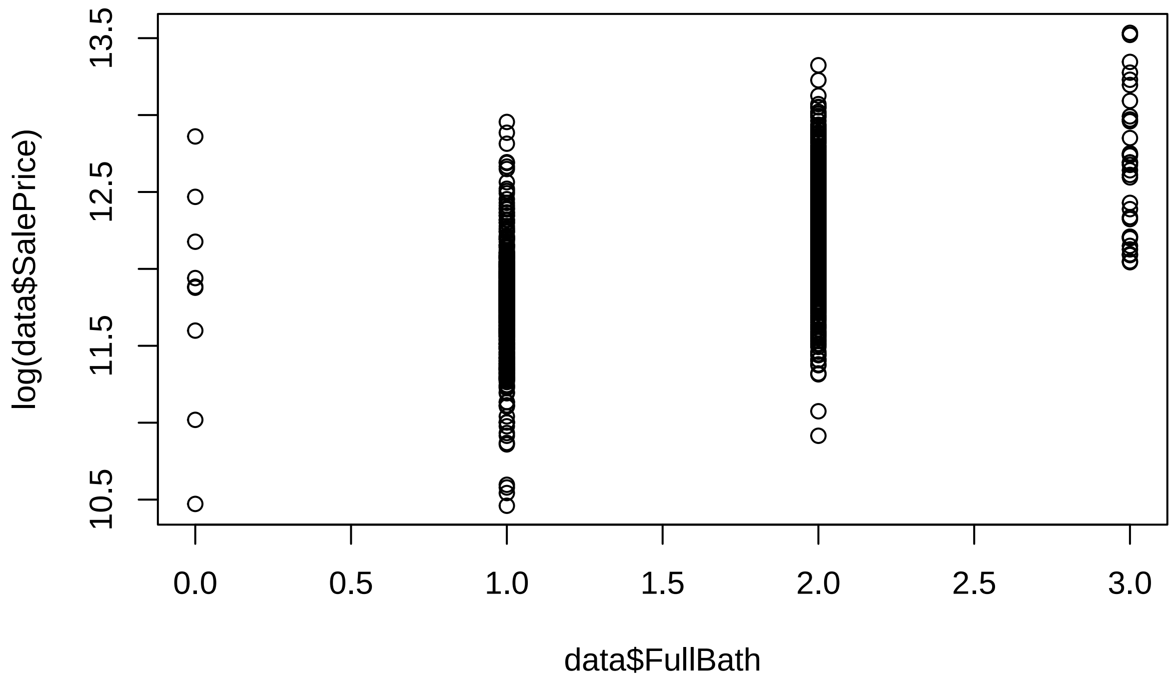
```r
plot(data$OverallQual,log(data$SalePrice))
```




```r
bp <- boxplot(log(data$SalePrice) ~ data$Exterior1st, data = data, col = c("lightblue", "darkblue"), xa
tick <- seq_along(bp$names)
axis(1, at = tick, labels = FALSE)
text(tick, par("usr")[3] - 0.3, bp$names, srt = 90, xpd = TRUE)
```




```
plot(data$FullBath,log(data$SalePrice))
```



Here, I examine which factor variables might be worth dropping.

```
library(ggplot2)

plotbar <- function(i){
     if (is.factor(get(i))){
         ggplot(data, aes(x=get(i))) + geom_bar() + ggtitle(i)
     }
}
```

Breaking this into two chunks to more easily isolate the function

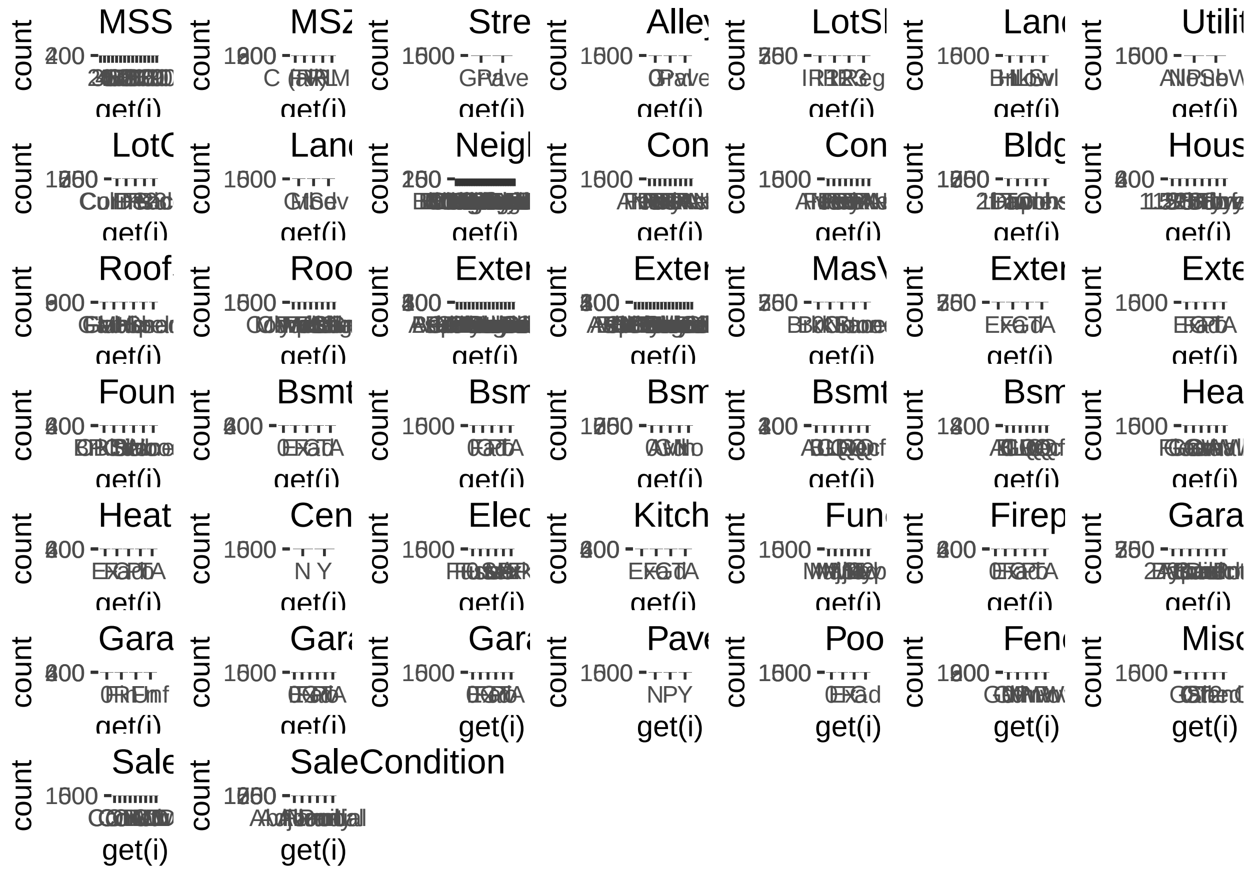
```
attach(data)
library(gridExtra)
```

```
## Warning: package 'gridExtra' was built under R version 4.4.3
```

```
plots <- lapply(names(data),plotbar)
plots[sapply(plots, is.null)] <- NULL # drop the null values
length(plots)
```

```
## [1] 44
```

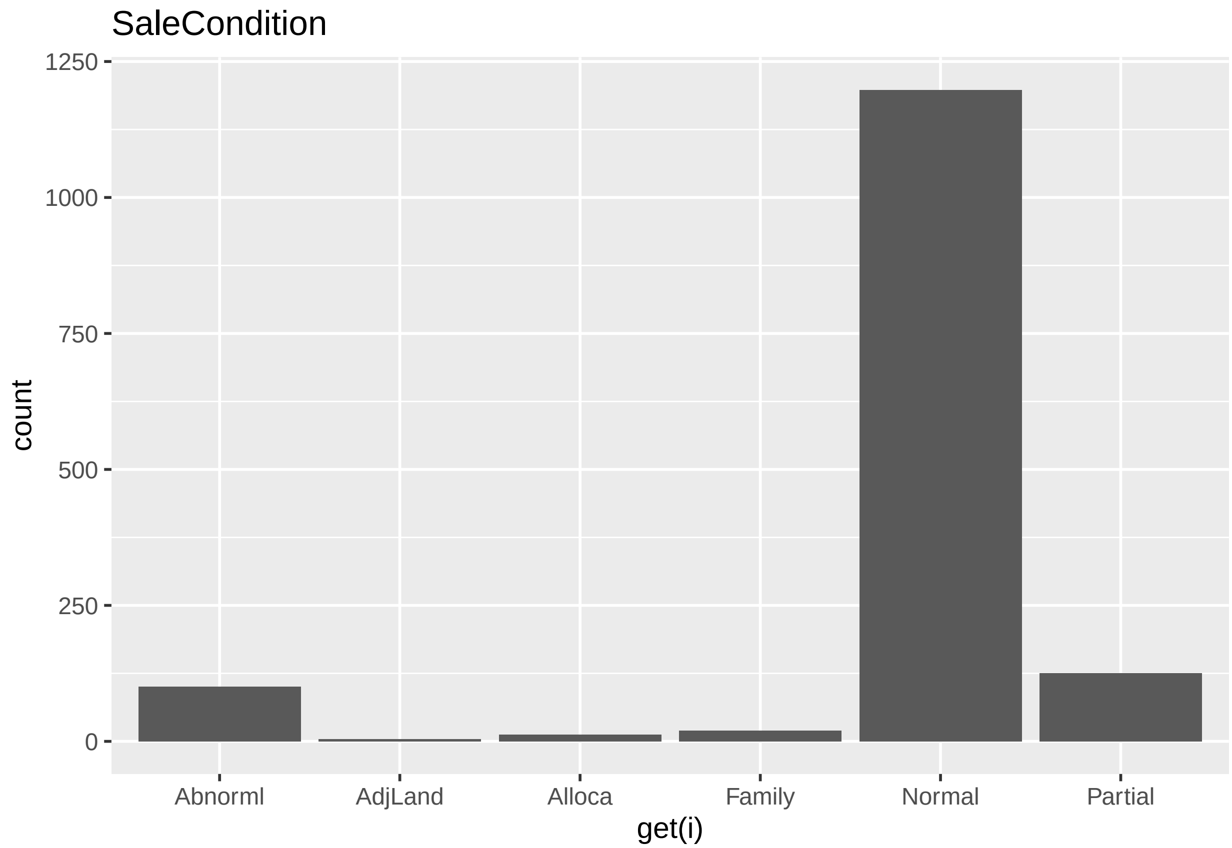
```
gridExtra::grid.arrange( grobs = plots, nrow = 7) # you might want to run this row separately and use t
```



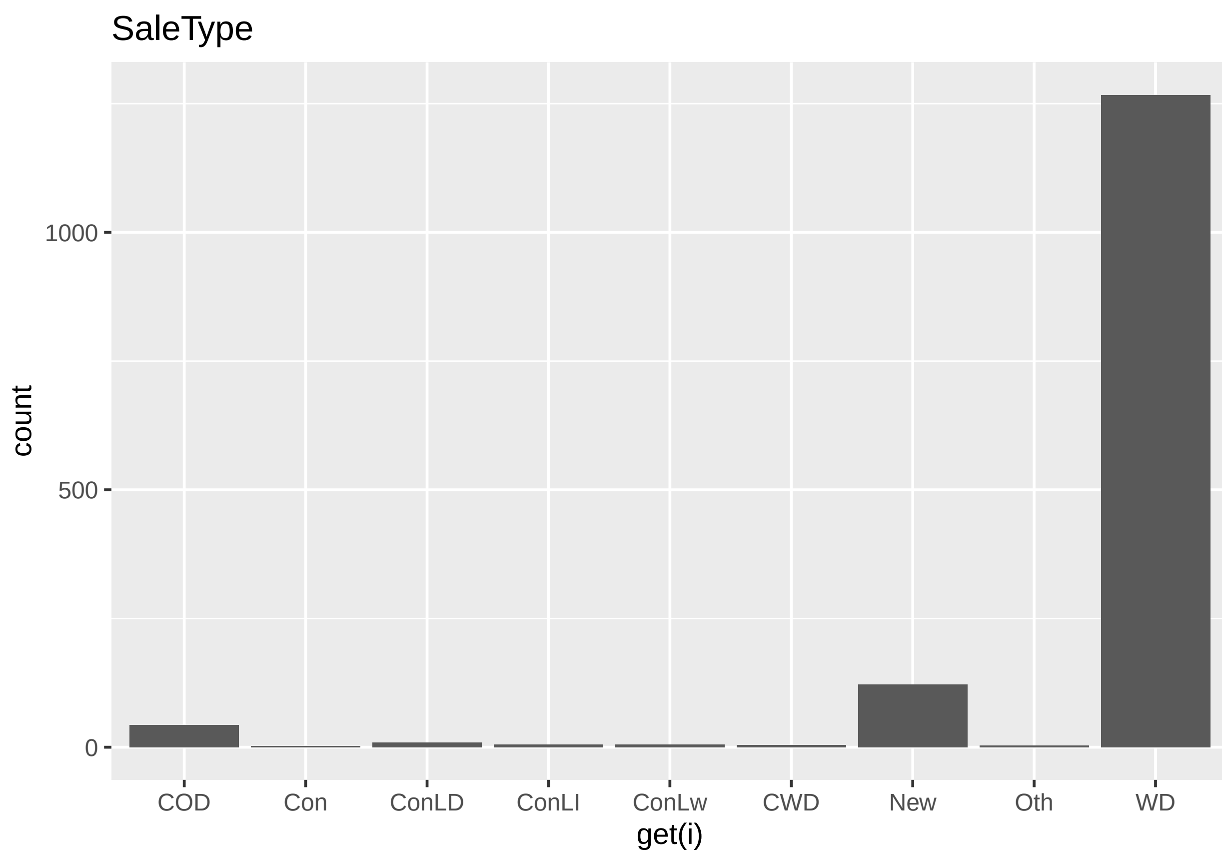
```r
# Clearing data - Lily
library(forcats)
```

```
## Warning: package 'forcats' was built under R version 4.4.3
```

```r
data.cl <- data[!sapply(data,is.factor)][-1]

#SaleCondition
plotbar("SaleCondition") #to find how to aggregate
```



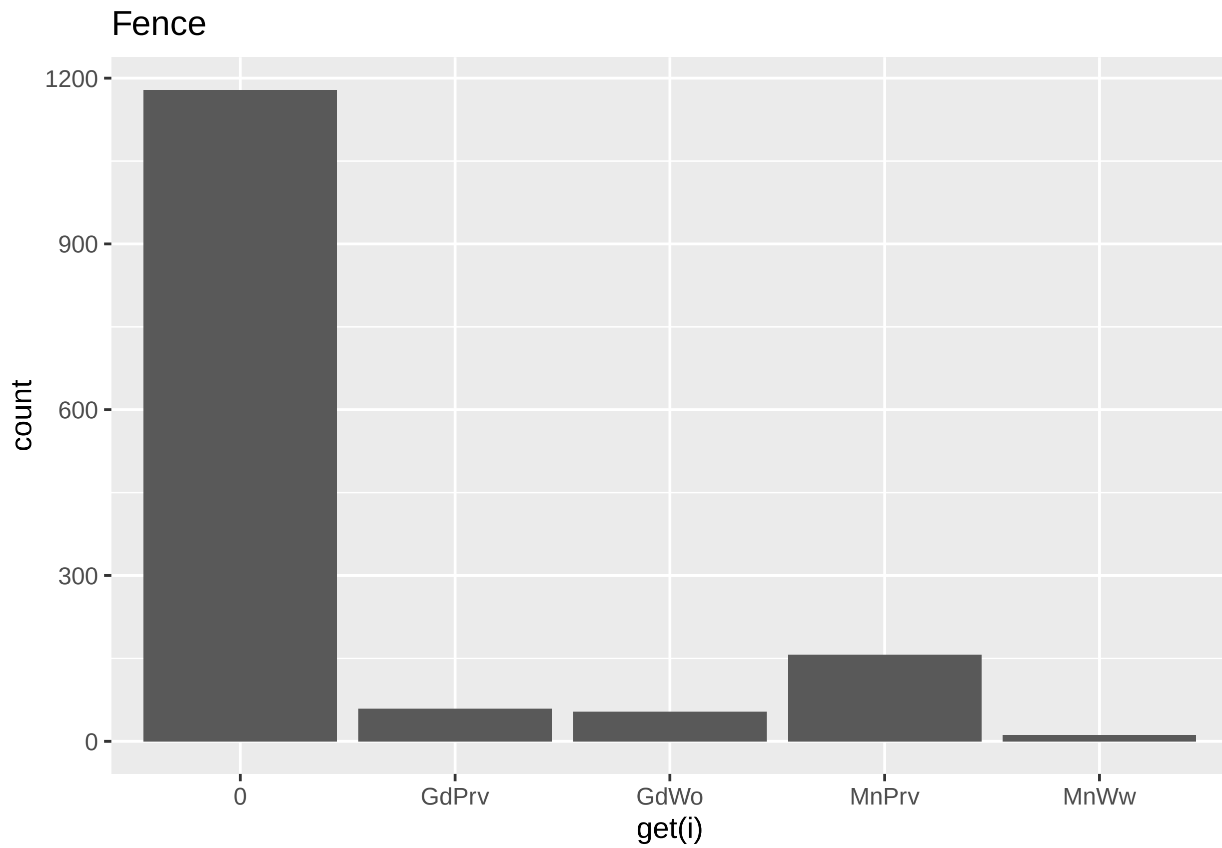

```
data.cl$SaleCondition <- fct_collapse(SaleCondition,Abnormal= c("Abnorml","AdjLand","Alloca","Family"),I

#SaleType
plotbar("SaleType") #to find how to aggregate
```



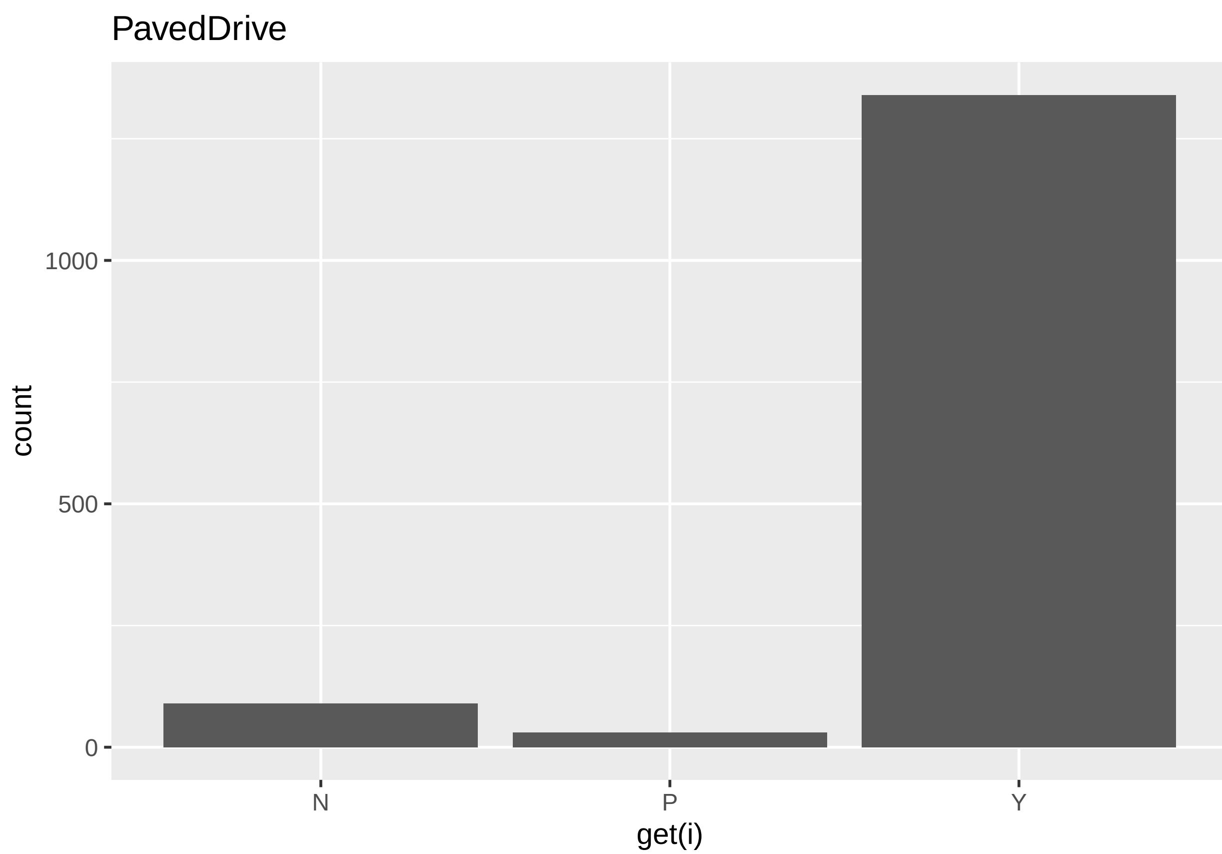

```
data.cl$SaleType <- fct_collapse(SaleType,New = "New",Warranty = c("WD","CWD", "VWD"), Others = c ("COD
```

```
## Warning: Unknown levels in `f`: VWD, ConLl
```

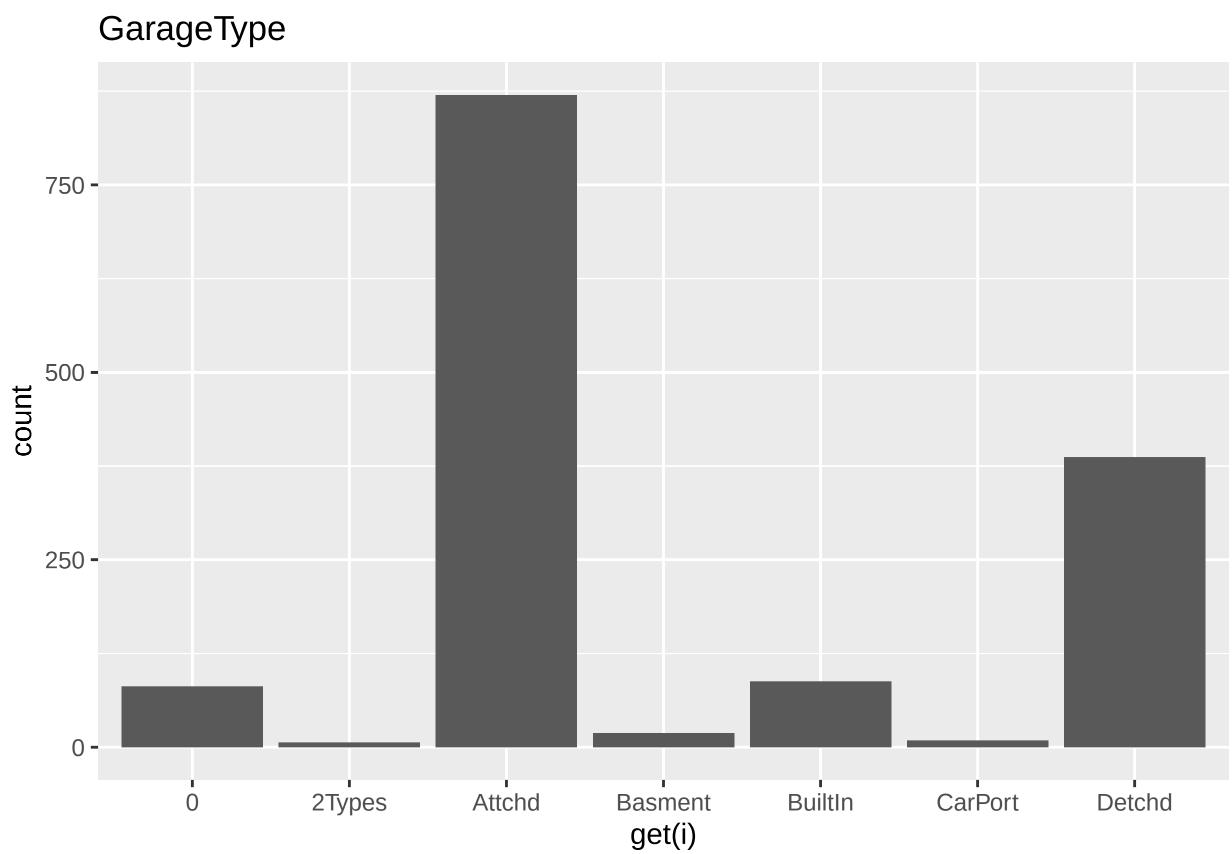
```
#Fence
plotbar("Fence") #to find how to aggregate
```



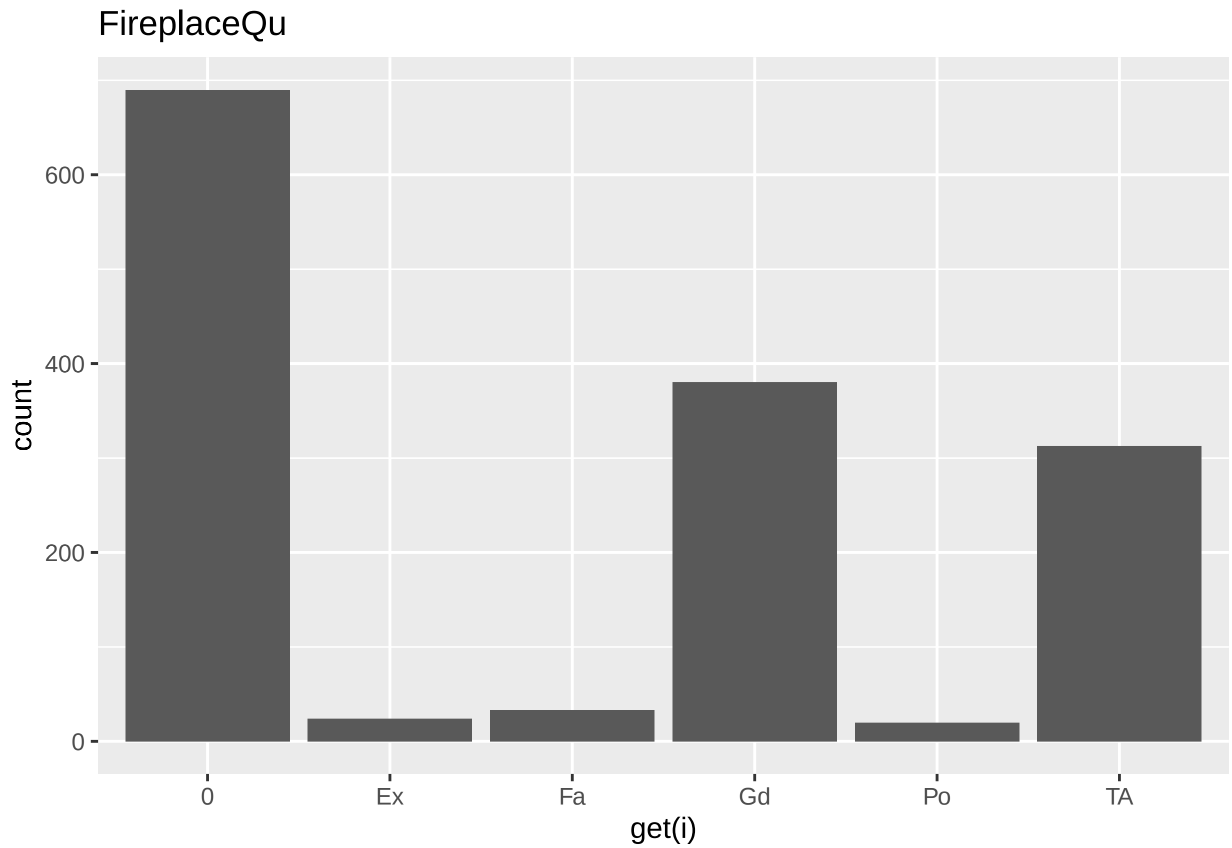

```
data.cl$Fence <- fct_collapse(Fence,No ="0",Good = c("GdPrv","GdWo"),Mini = c("MnPrv","MnWw"))

#PavedDrive
plotbar("PavedDrive") #to find how to aggregate
```



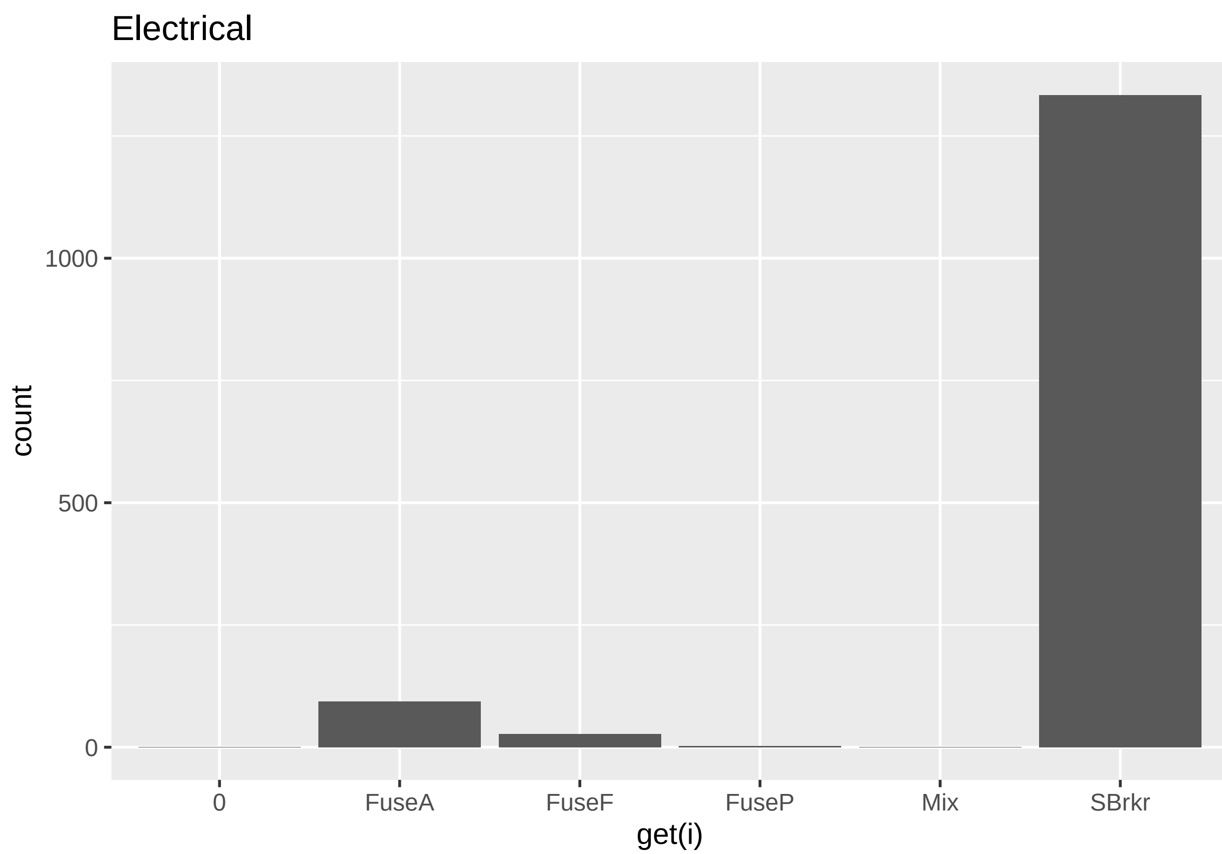

```r
data.cl$PavedDrive <- fct_collapse(PavedDrive,NP= c("N","P"))

#GarageType
plotbar("GarageType")
```



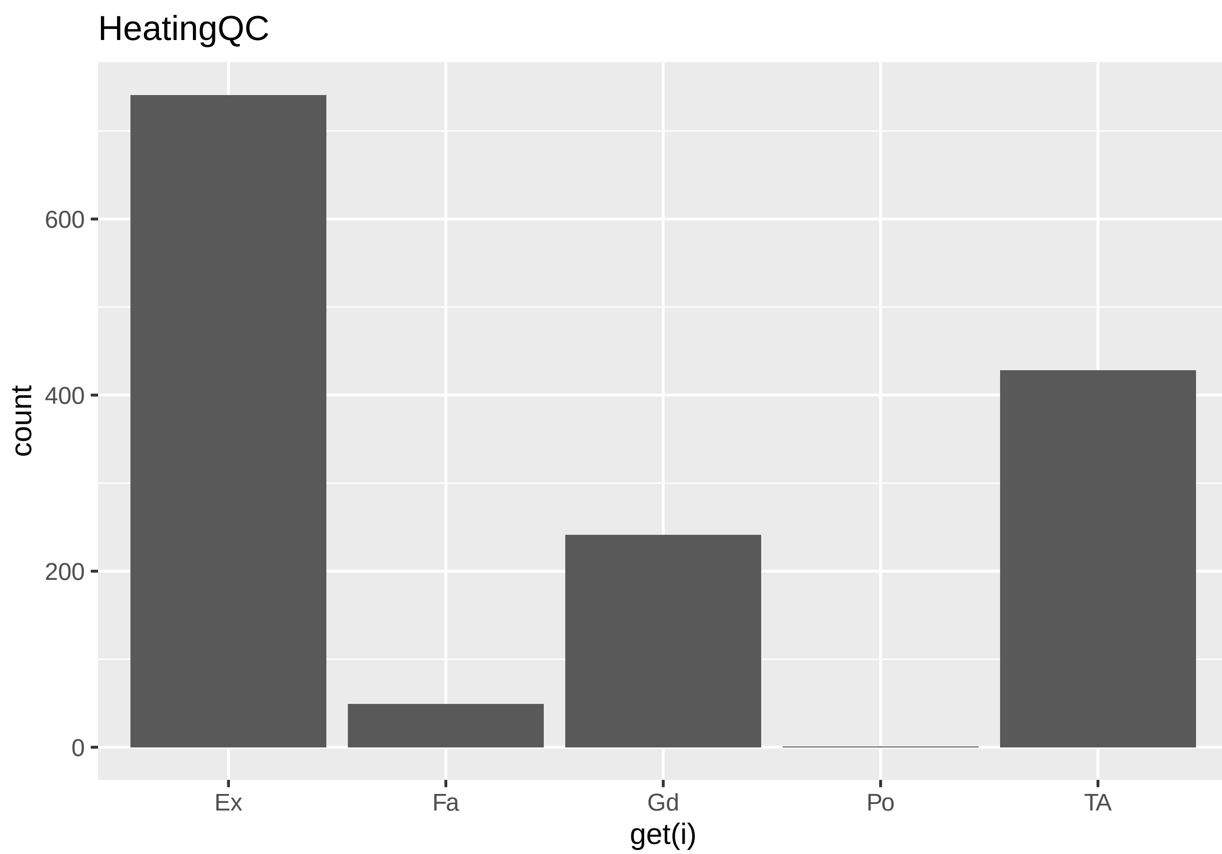

```
data.cl$GarageType <- fct_collapse(GarageType,Attached = c("Attchd", "BuiltIn"),Detached = "Detchd",Othe


#FireplaceQu
plotbar("FireplaceQu")
```



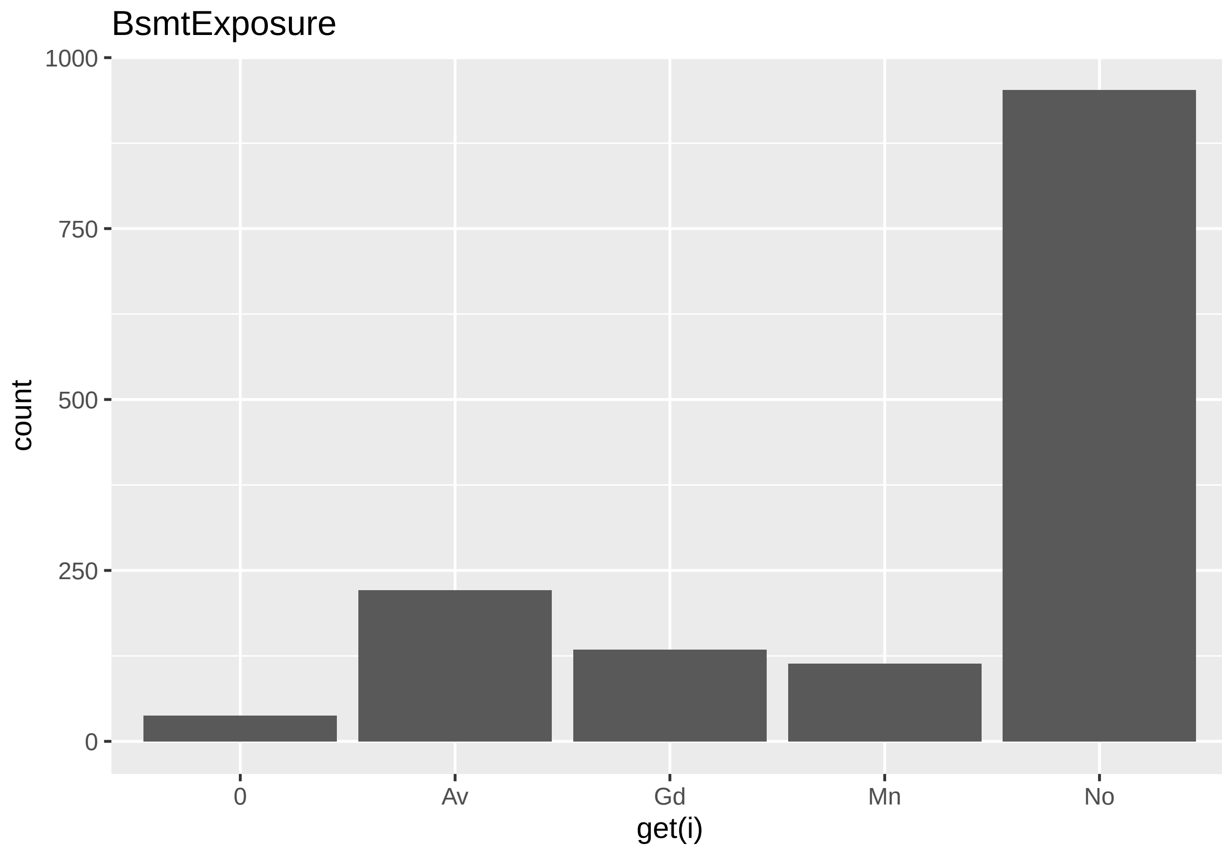

```
data.cl$FireplaceQu <- fct_collapse(FireplaceQu,Good=c("Ex","Gd"), Aver = c("TA","Fa"), Bad=c("Po","0"))

#Electrical
plotbar("Electrical")
```



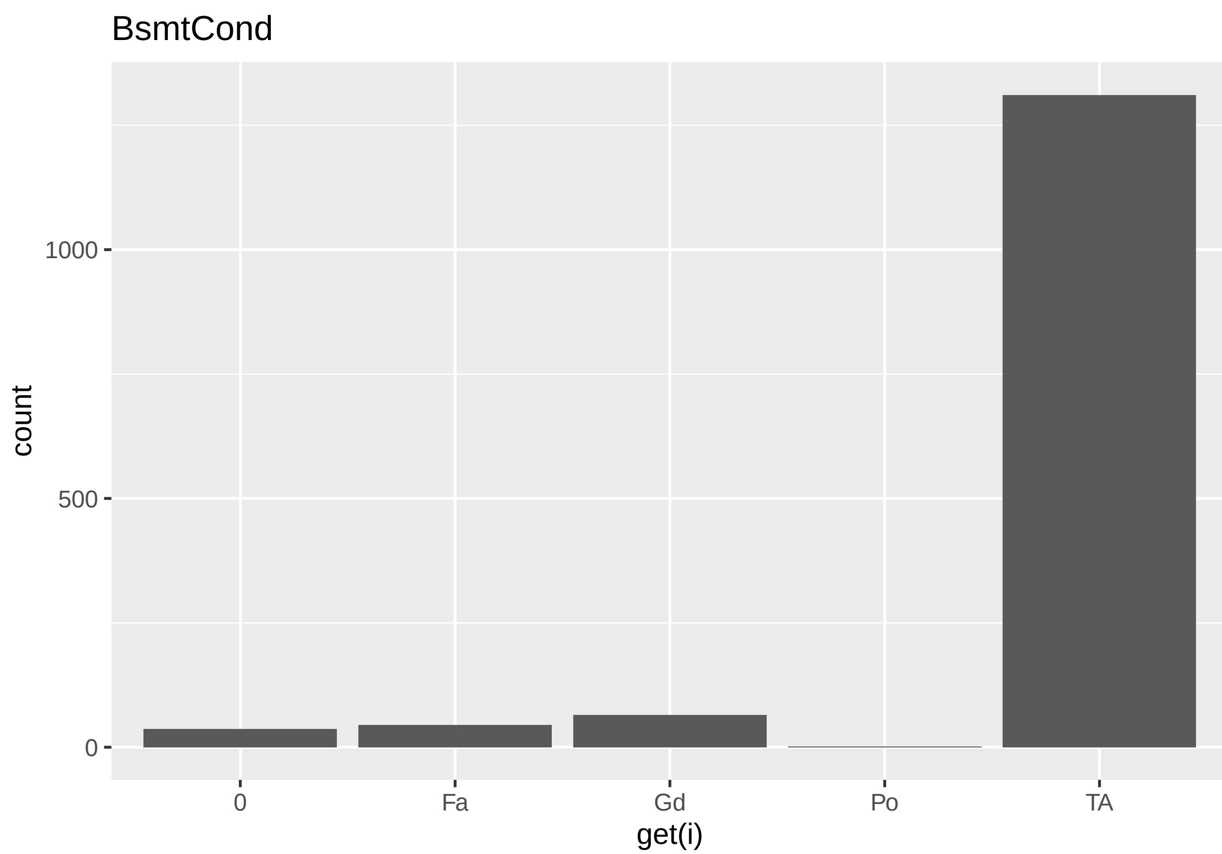

```r
data.cl$Electrical <- fct_collapse(Electrical,Stand = "SBrkr", Other = c("FuseA","FuseF","FuseP", "Mix"

#HeatingQC
plotbar("HeatingQC")
```




```
data.cl$HeatingQC <- fct_collapse(HeatingQC,Good =  c("Ex","Gd"),   Ave = "TA", Bad = c("Fa","Po"))

#"BsmtExposure"
plotbar("BsmtExposure")
```



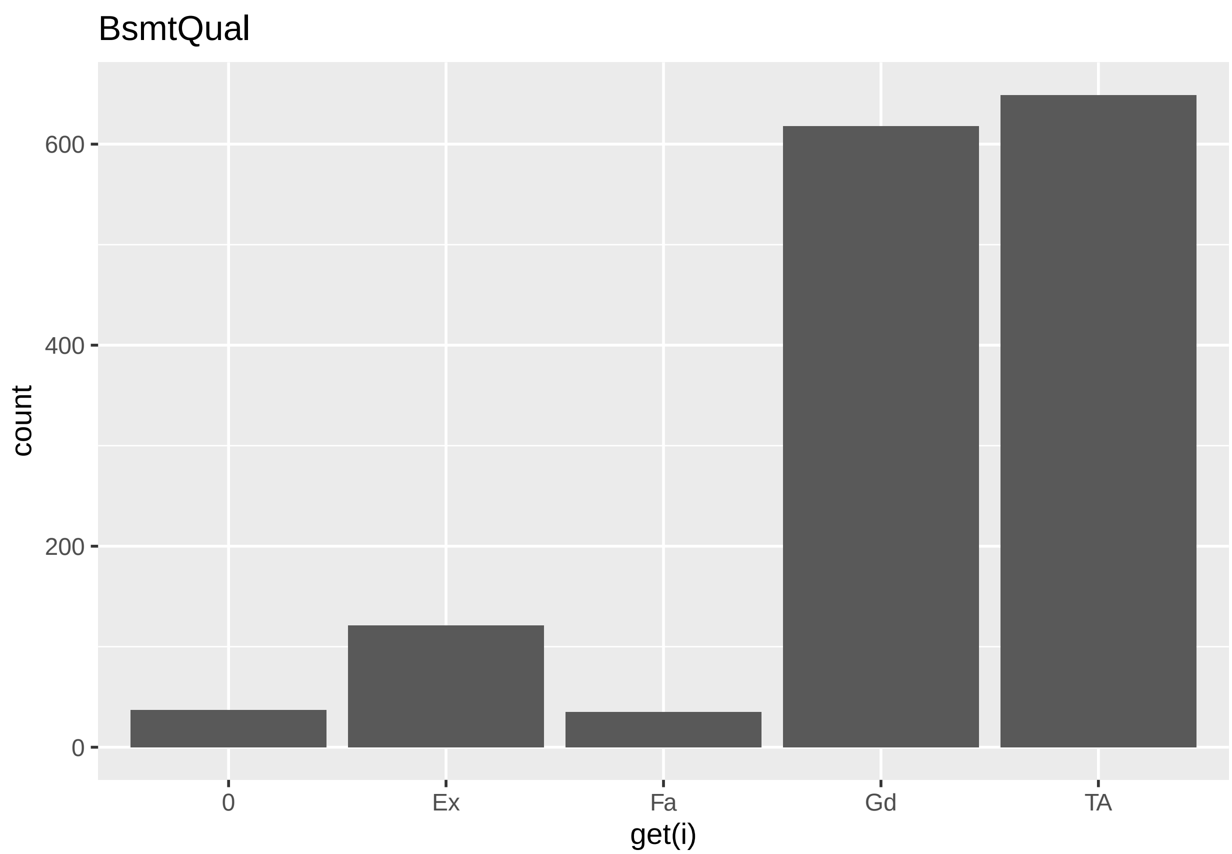

```
data.cl$BsmtExposure <- fct_collapse(BsmtExposure, Good =  "Gd",AbovMin = c("Av","Mn"), NoE = "No", NoB

#"BsmtCond"
plotbar("BsmtCond")
```




```
data.cl$BsmtCond <- fct_collapse(BsmtCond, Typical = "TA",NonTy = c("Ex","Gd","Fa","Po","0"))
```

```
## Warning: Unknown levels in `f`: Ex
```

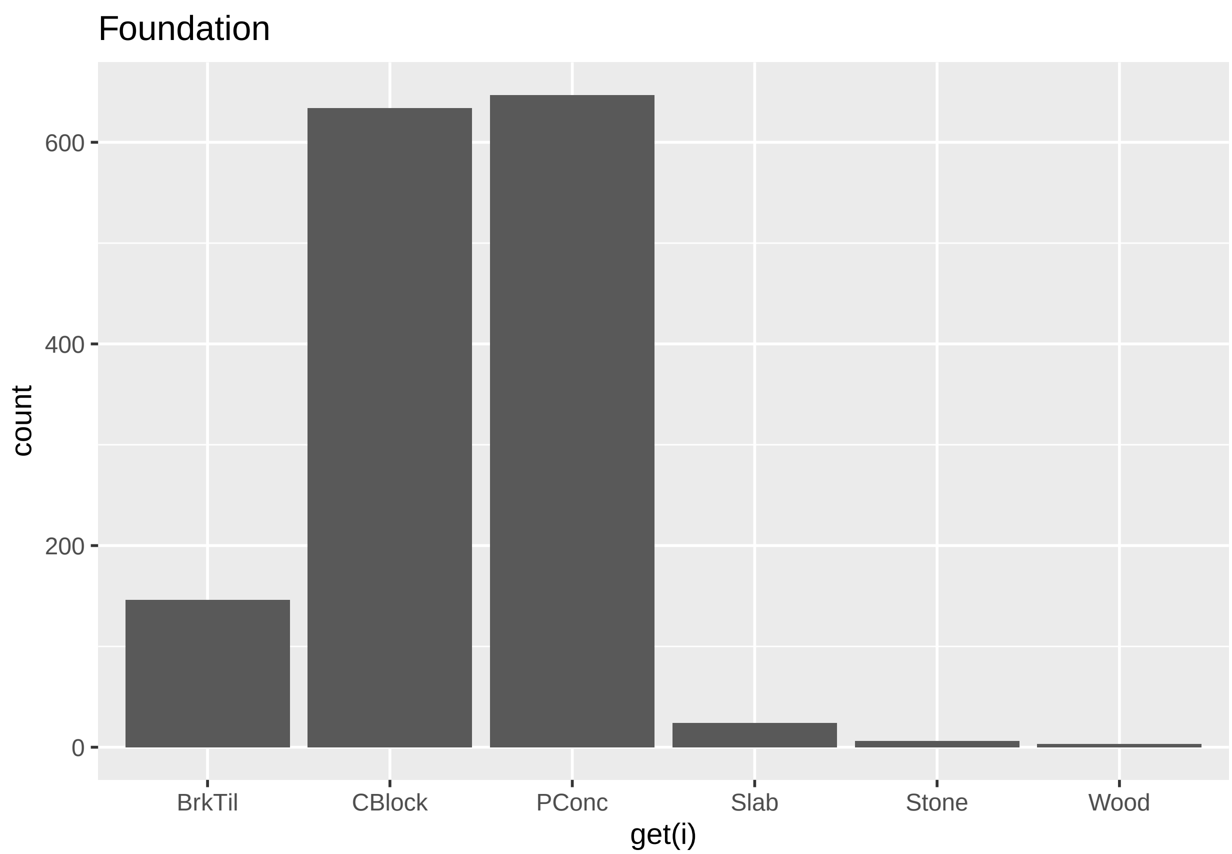
```
#"BsmtQual"
plotbar("BsmtQual")
```



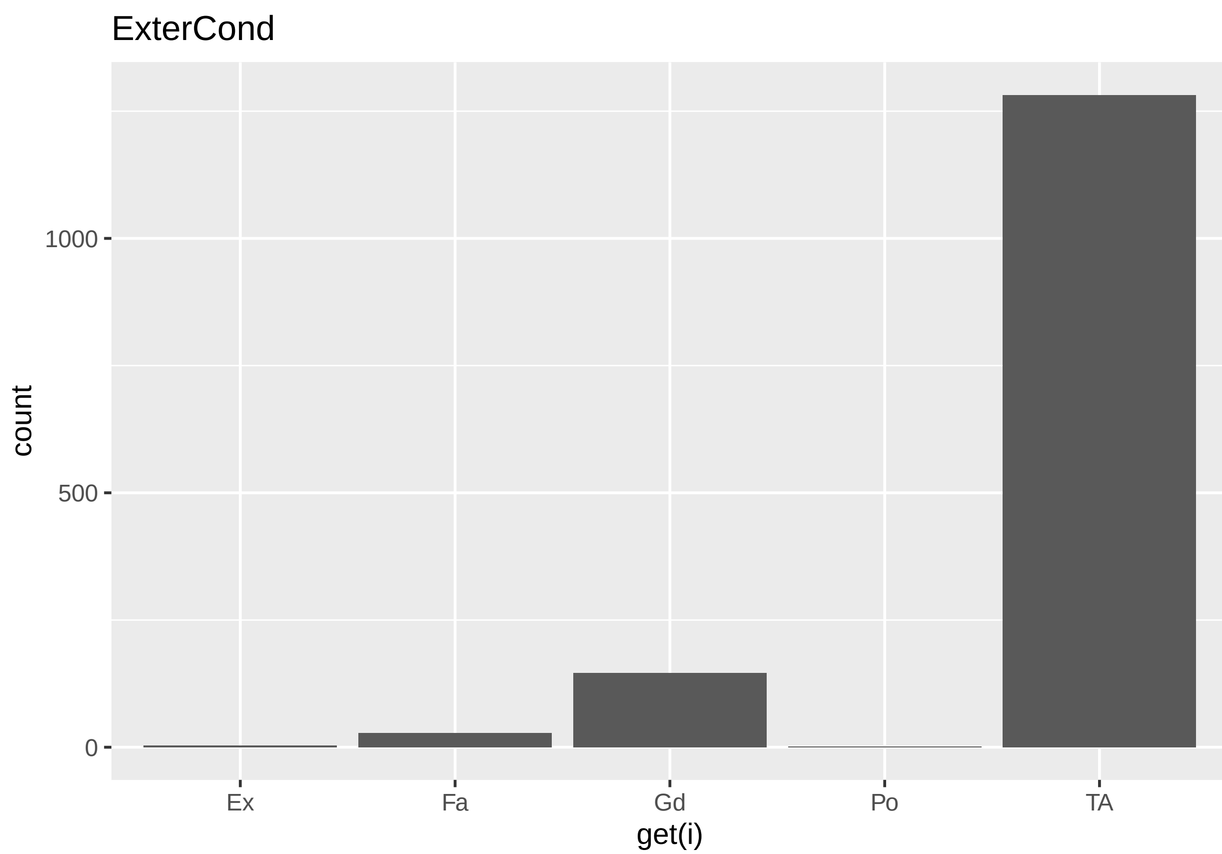

```
data.cl$BsmtQual <- fct_collapse(BsmtQual,Good=c("Ex","Gd"),    Aver = c("TA","Fa"),    Bad=c("Po","0"))
```

```
## Warning: Unknown levels in `f`: Po
```

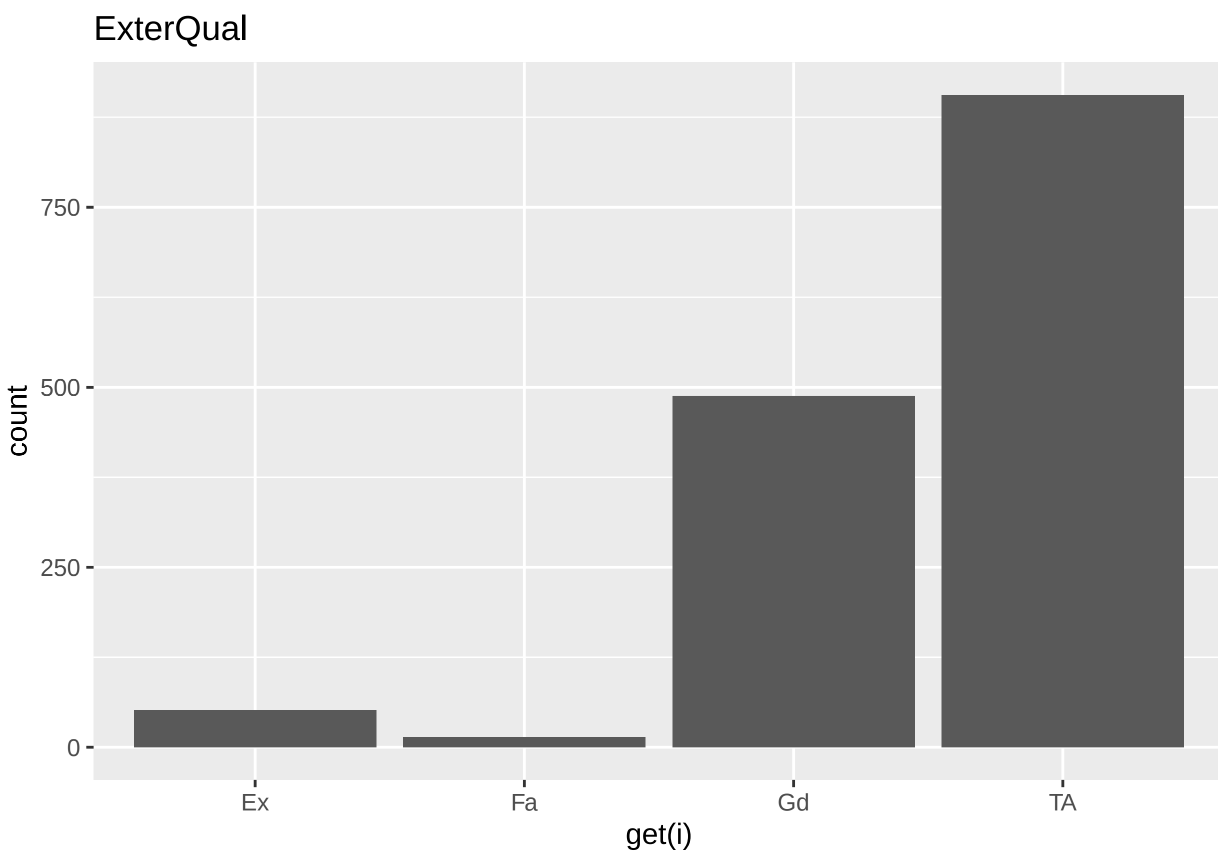
```
#"Foundation"
plotbar("Foundation")
```




```
data.cl$Foundation <- fct_collapse(Foundation,BrkTil= "BrkTil",CBlock = "CBlock",PConc = "PConc",Oth = c

#"ExterCond"
plotbar("ExterCond")
```



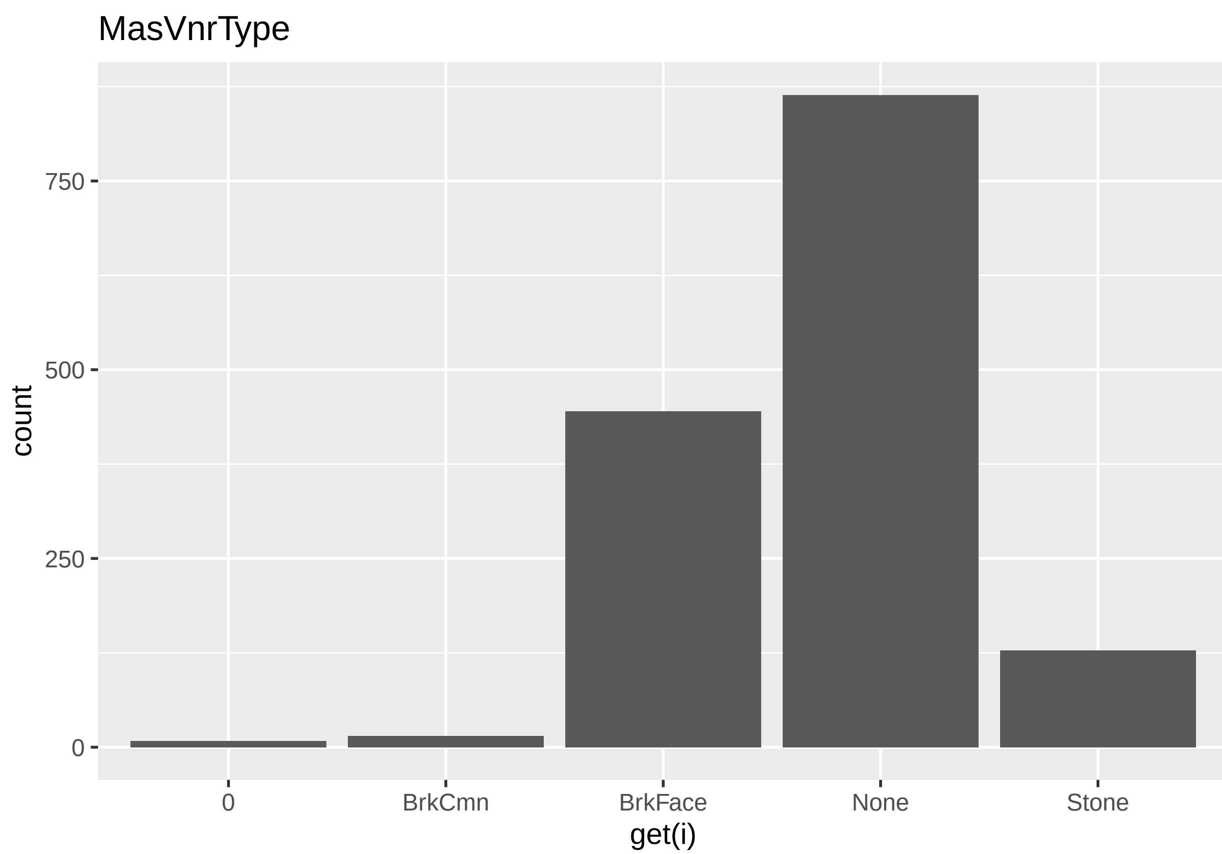

```
data.cl$ExterCond <- fct_collapse(ExterCond,Good =  c("Ex","Gd"),   Ave = "TA", Bad = c("Fa","Po"))

#ExterQual
plotbar("ExterQual")
```




```
data.cl$ExterQual <- fct_collapse(ExterQual,Good =  c("Ex","Gd"), Ave = "TA",   Bad = c("Fa","Po"))
```

```
## Warning: Unknown levels in `f`: Po
```

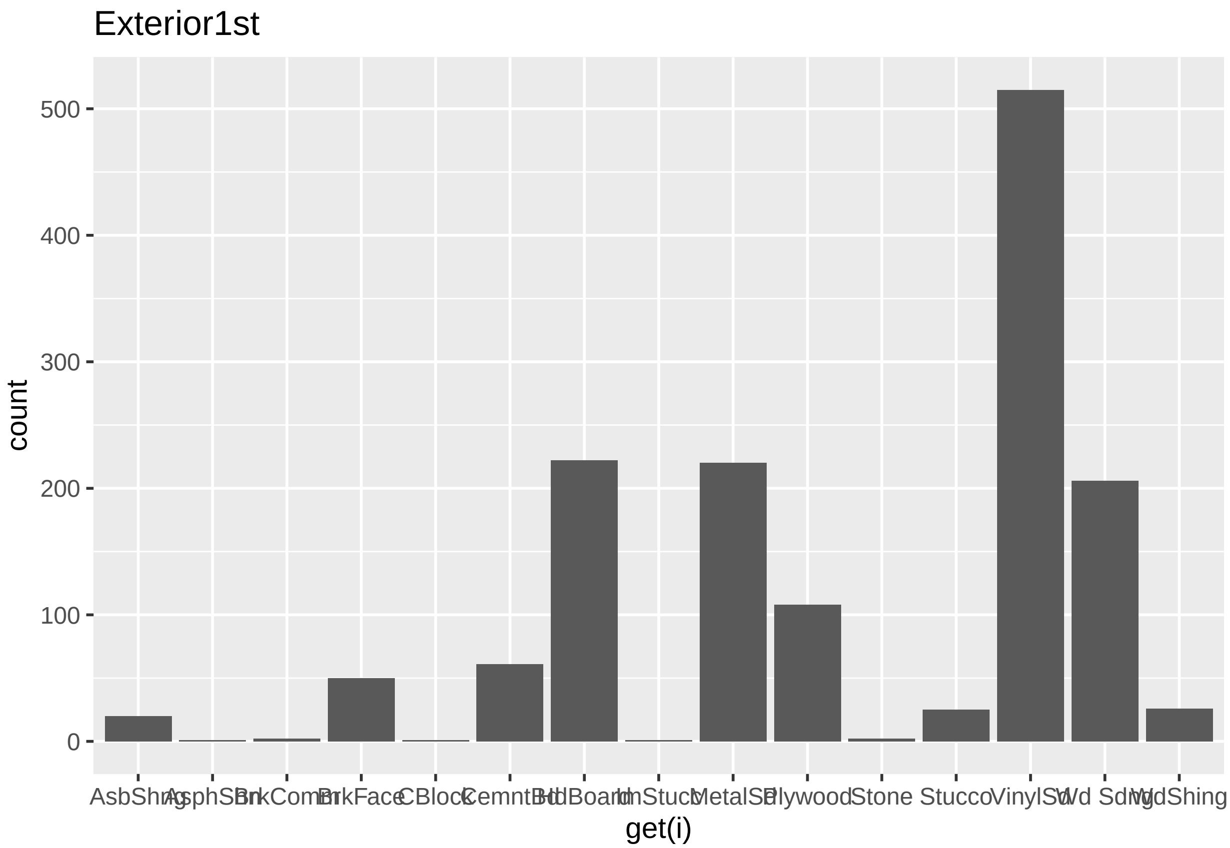
```
#"MasVnrType"
plotbar("MasVnrType")
```




```
data.cl$MasVnrType <- fct_collapse(MasVnrType, Brk= c("BrkCmn","BrkFace")    ,Stone = "Stone",    Other =
```

```
## Warning: Unknown levels in `f`: CBlock
```

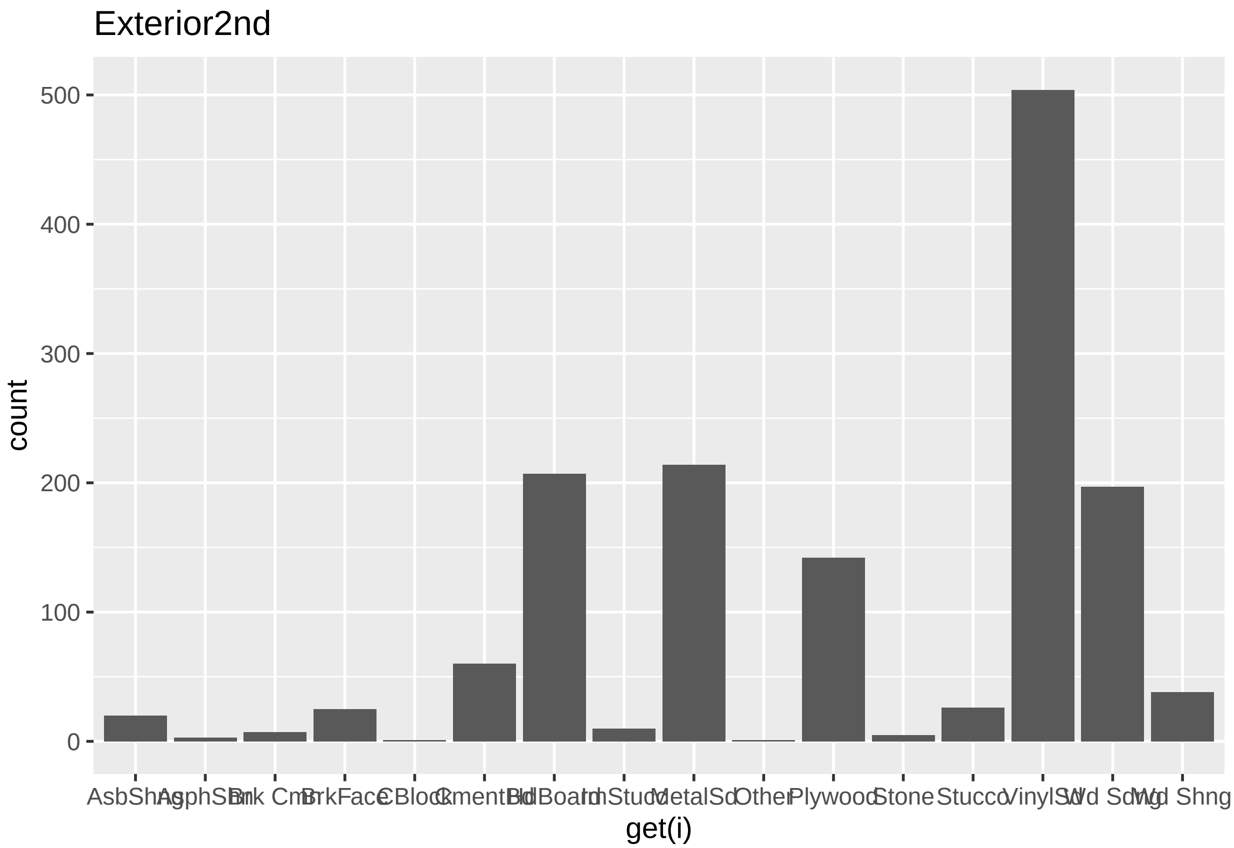
```
#"Exterior1st"
plotbar("Exterior1st")
```




```
data.cl$Exterior1st <- fct_collapse(Exterior1st, Brk = c("BrkComm","BrkFace"), Wood = c("Wd Sdng", "WdSh
```

```
## Warning: Unknown levels in `f`: PreCast, Other
```

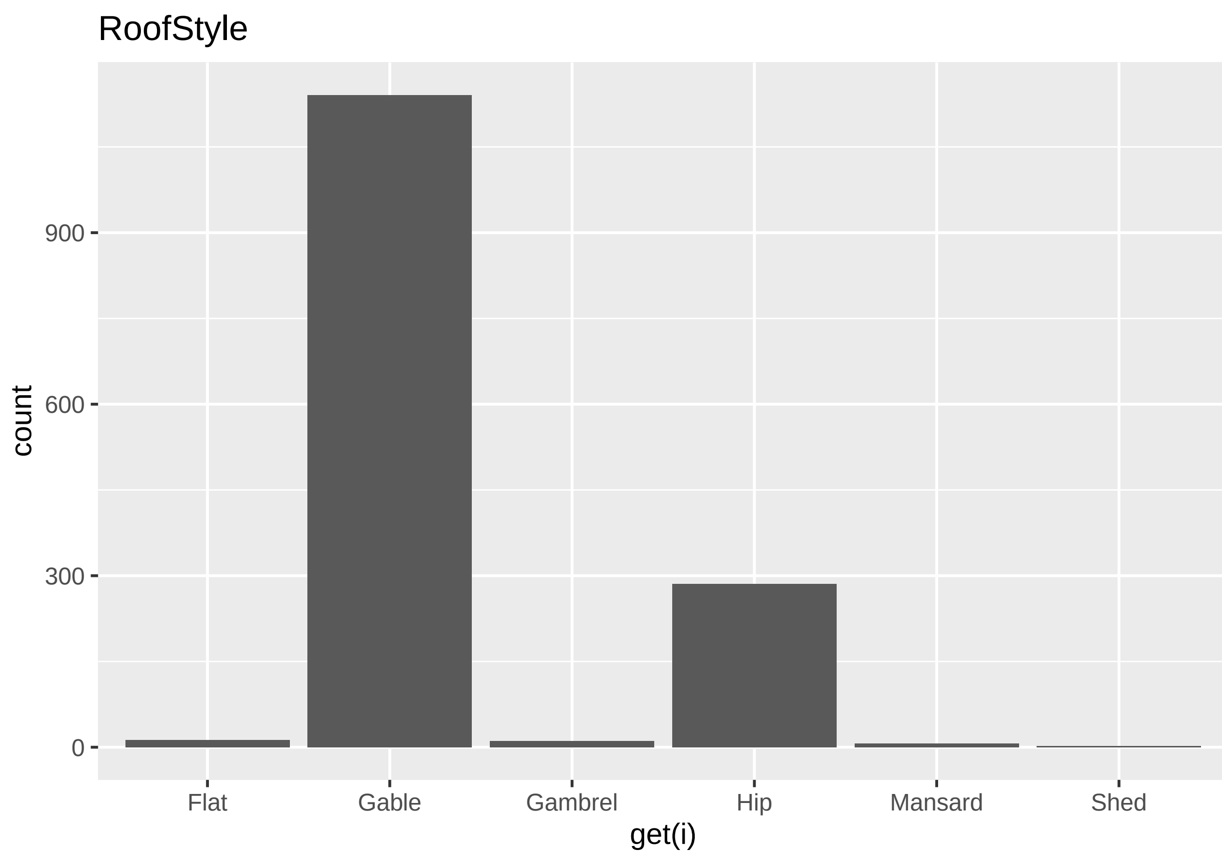
```
#"Exterior2nd"
plotbar("Exterior2nd")
```



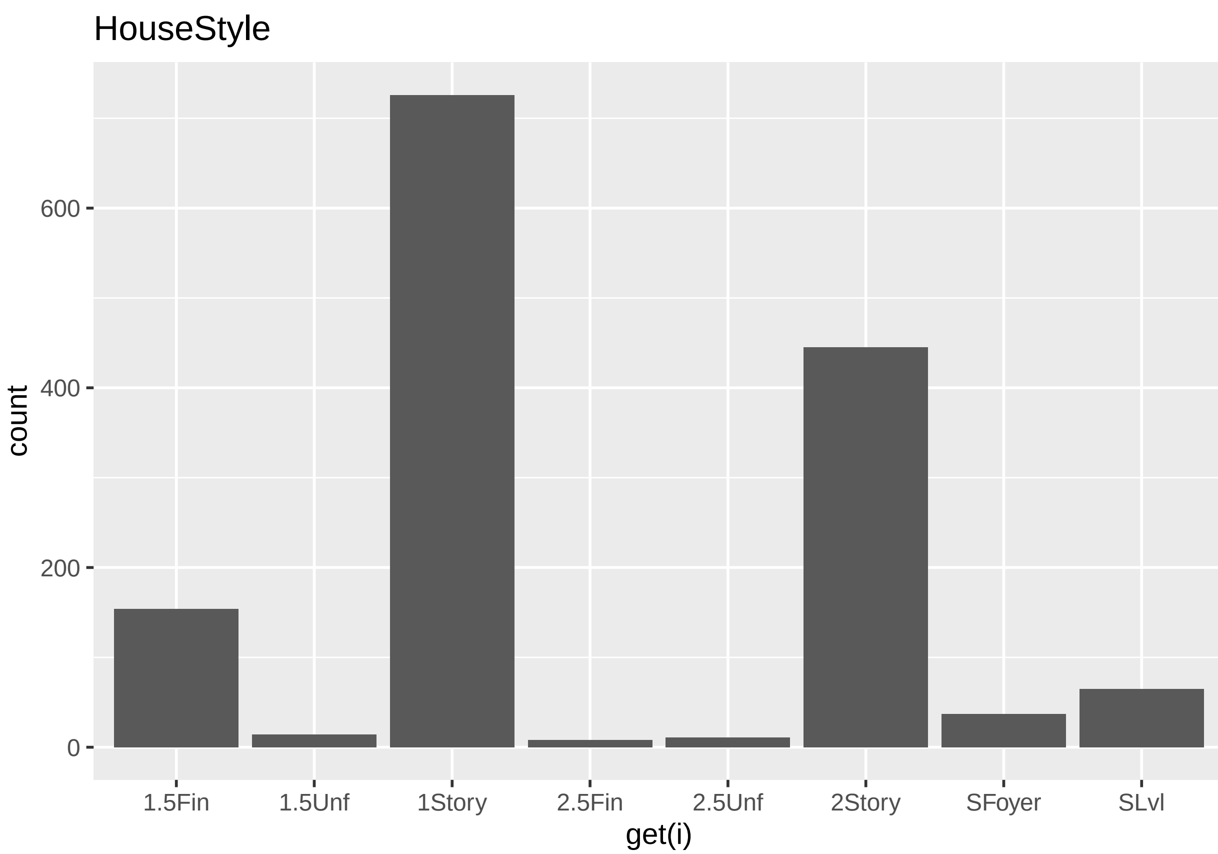

```
data.cl$Exterior2nd <- fct_collapse(Exterior2nd, Brk = c("BrkComm","BrkFace"), Wood = c("Wd Sdng", "WdSh
```

```
## Warning: Unknown levels in `f`: BrkComm, WdShing, CemntBd, PreCast
```

```
#"RoofStyle"
plotbar("RoofStyle")
```



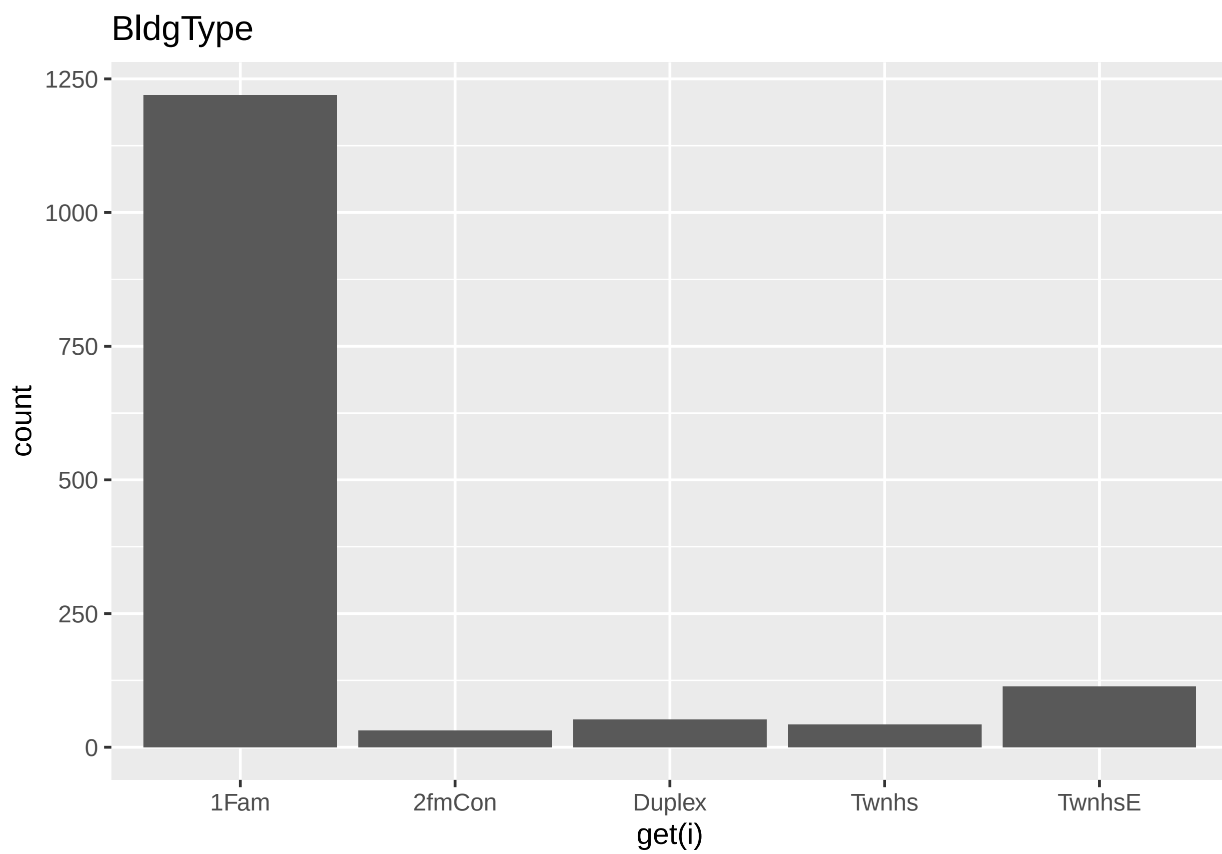

```
data.cl$RoofStyle <- fct_collapse(RoofStyle,Gable = "Gable",    Hip = "Hip",    Others = c ("Flat","Gaml

#"HouseStyle"
plotbar("HouseStyle")
```




```
data.cl$HouseStyle <- fct_collapse(HouseStyle,OneStory = "1Story", OnenHalfStory = c ("1.5Fin", "1.5Unf
```

```
## Warning: Unknown levels in `f`: 2.5 Fin
```

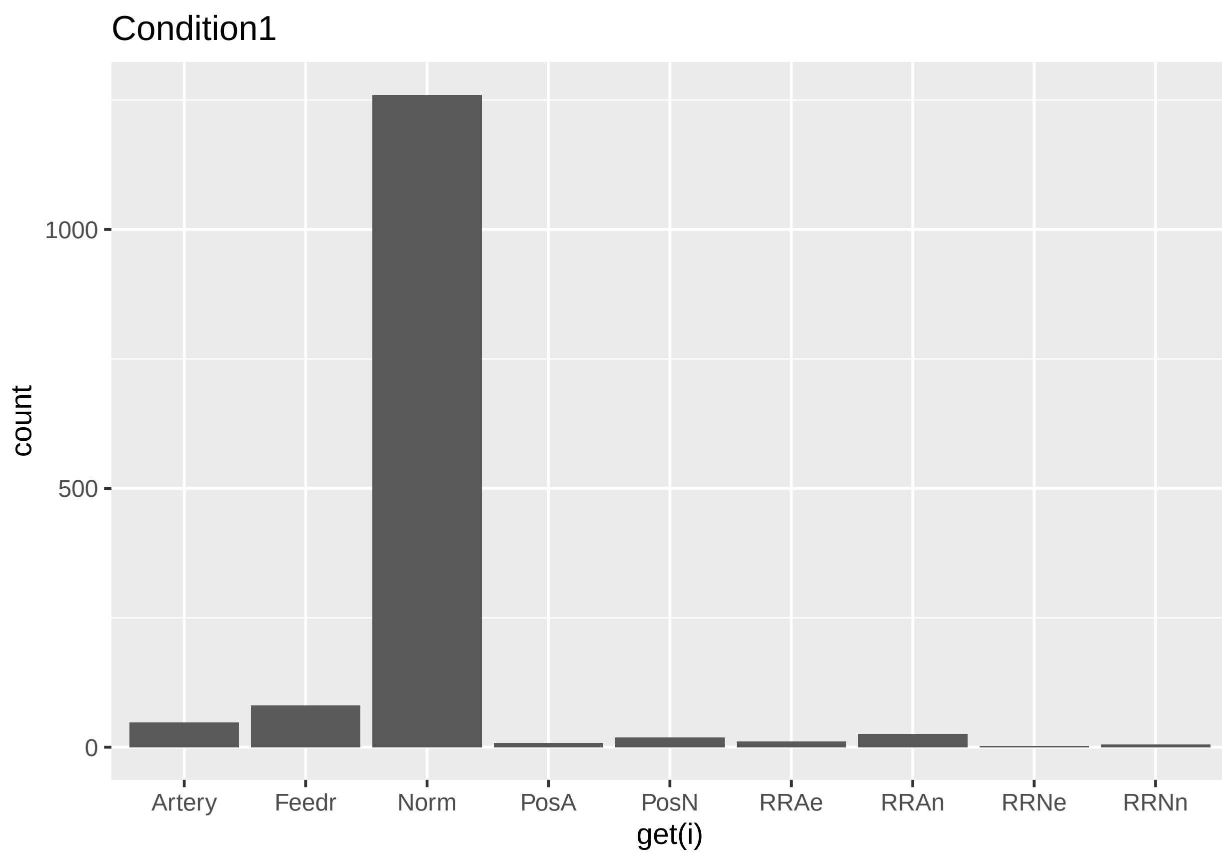
```
#"BldgType"
plotbar("BldgType")
```



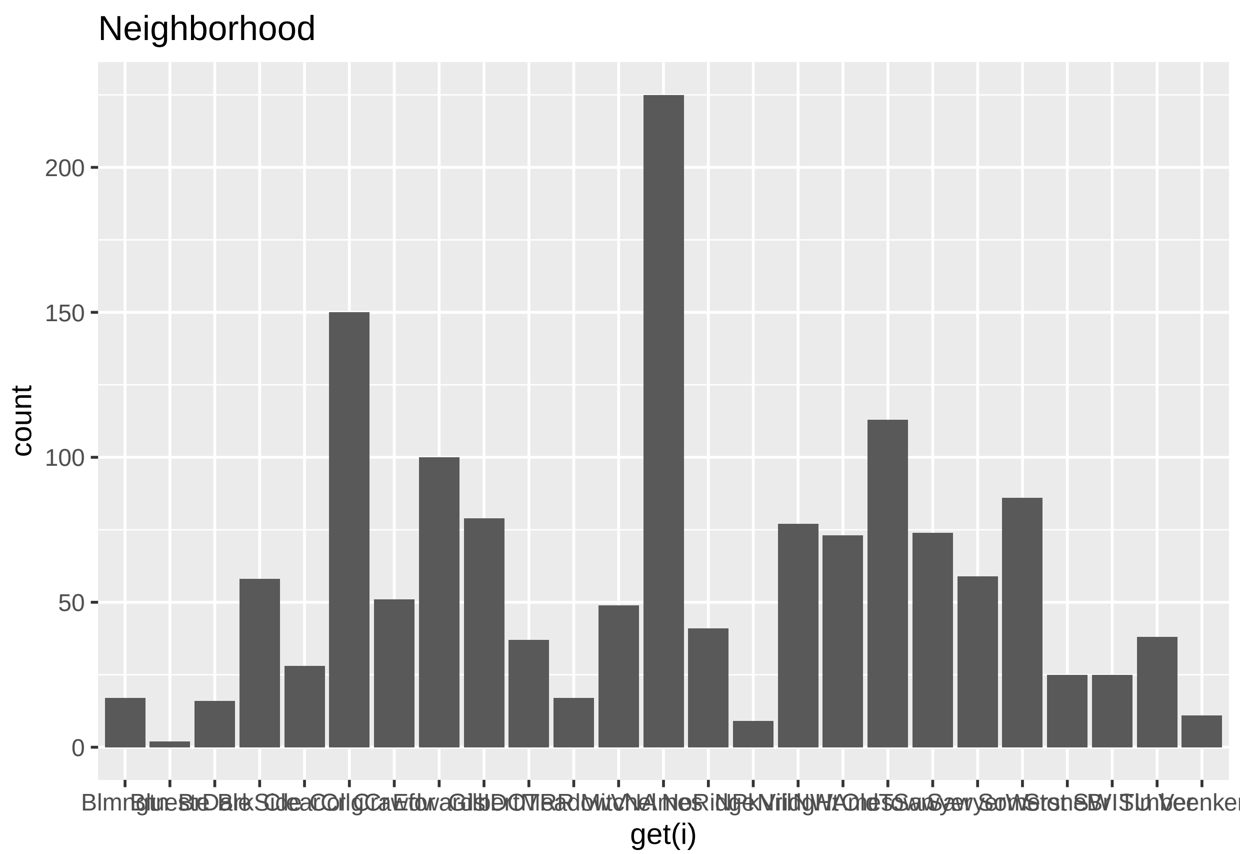

```
data.cl$BldgType <- fct_collapse(BldgType,OneFam = "1Fam",  TwoFam = c ( "2FmCon", "Duplex")    , Twn =
```

```
## Warning: Unknown levels in `f`: 2FmCon, Twnhsl
```

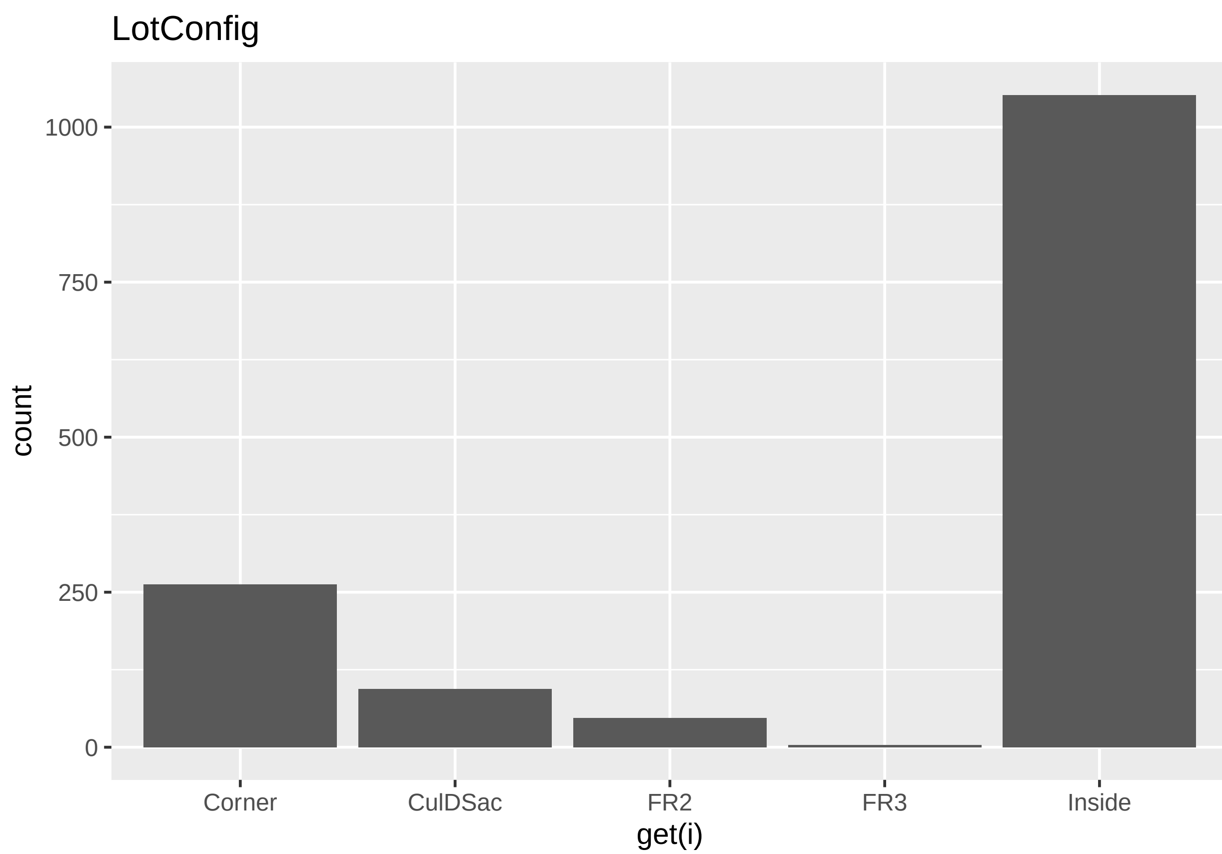
```
#"Condition1"
plotbar("Condition1")
```



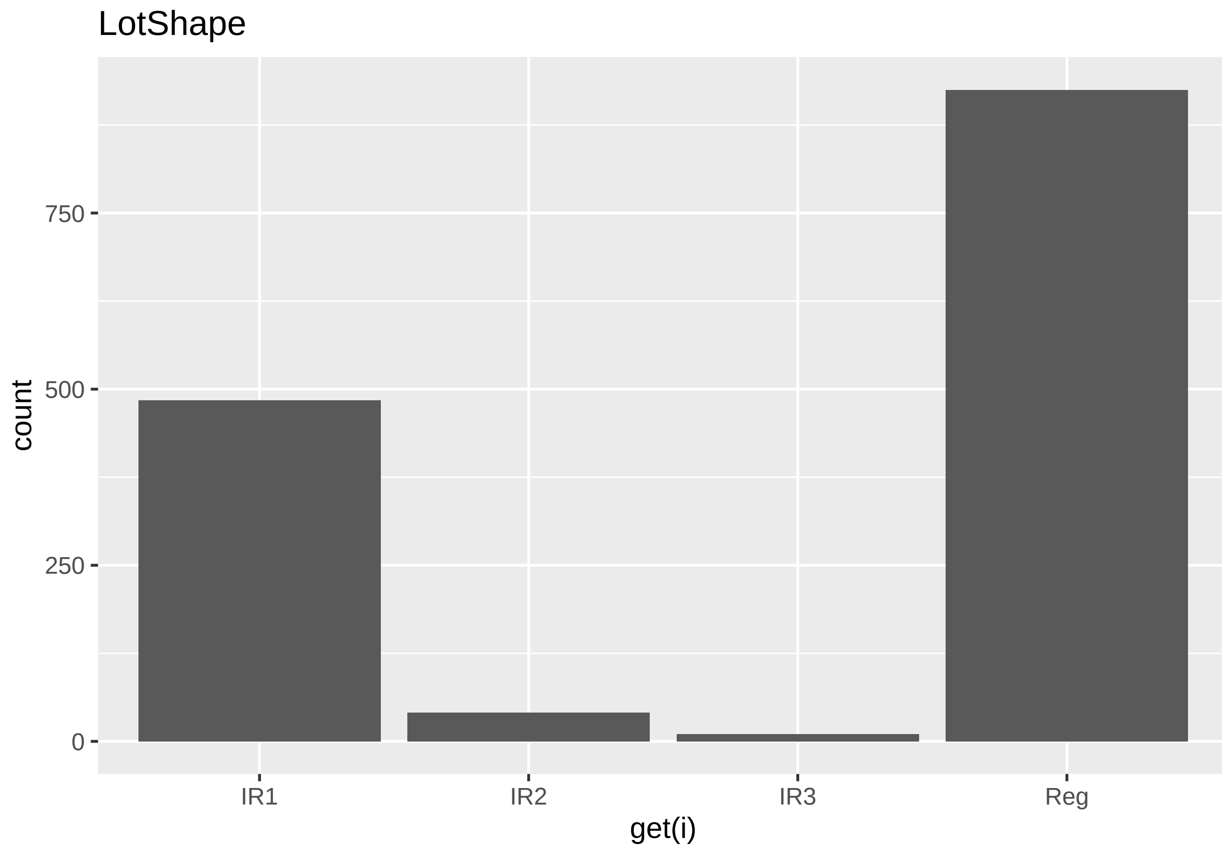

```r
data.cl$Condition1 <- fct_collapse(Condition1, Norm = "Norm", Other = c ("Artery", "Feedr", "RRNn", "RRA

#"Neighborhood"
plotbar("Neighborhood")
```



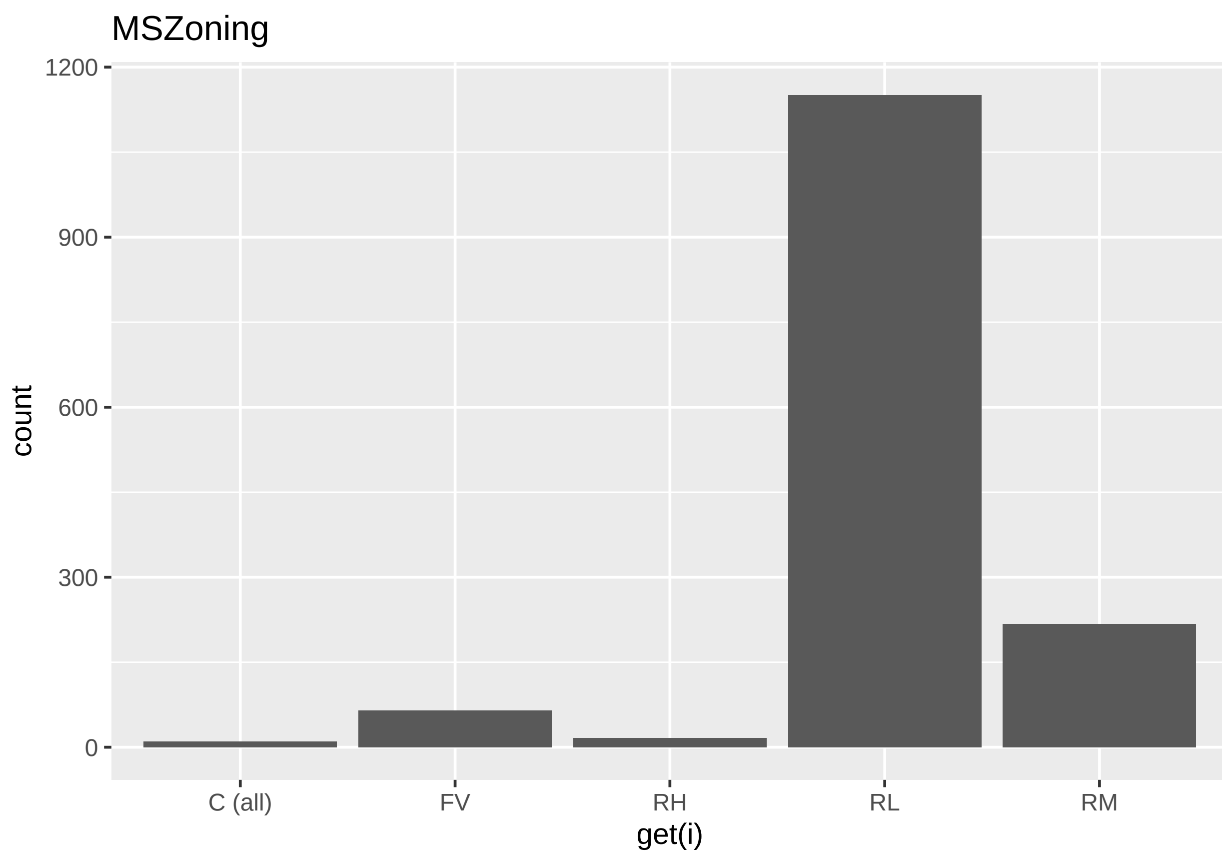

```
data.cl$Neighborhood <- fct_collapse(Neighborhood, North = c ("NWAmes", "NAmes", "NoRidge", "NPkVill",

#"LotConfig"
plotbar("LotConfig")
```




```
data.cl$LotConfig <- fct_collapse(LotConfig,Standard = c ("Inside", "Corner"),  Premium = c ("CulDSac",

#"LotShape"
plotbar("LotShape")
```



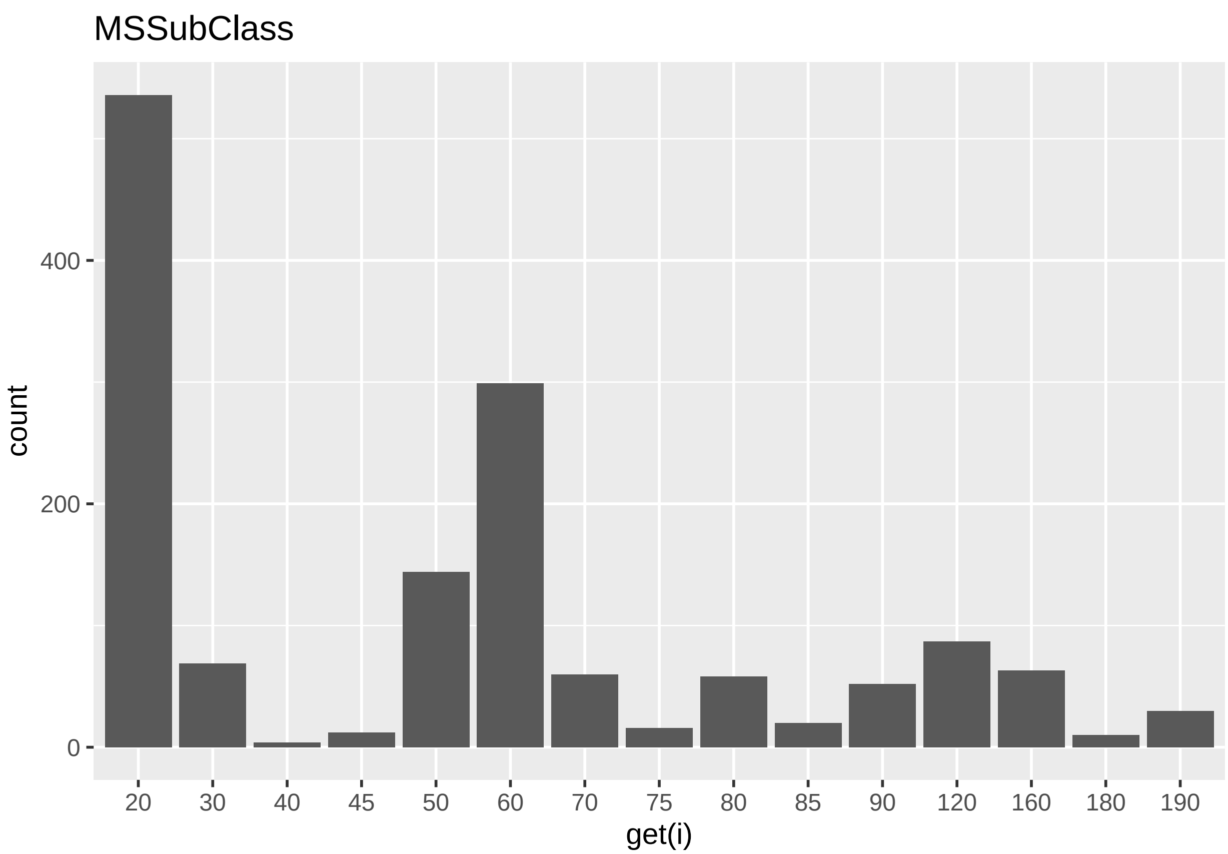

```
data.cl$LotShape <- fct_collapse(LotShape,Regular = "Reg", Irregular = c("IR1","IR2","IR3"))

#"MSZoning"
plotbar("MSZoning")
```




```
data.cl$MSZoning <- fct_collapse(MSZoning,Residentiallow = c ("RL", "RP")   ,ResidentialMedHi = c ("RM"
```

```
## Warning: Unknown levels in `f`: RP, A, I, C
```

```
#"MSSubClass"
plotbar("MSSubClass")
```




```
data.cl$MSSubClass <- fct_collapse(MSSubClass, story1=c(20,30,40,120), story1.5=c(45,50,150),story2=c(6
```

```
## Warning: Unknown levels in `f`: 150
```

Preparing the test data

Here, I redo all of the cleaning steps, but with the test data instead

```
test <- read.csv("test.csv")
test[is.na(test)]<-0

# factor the appropriate columns
test$MSSubClass <- factor(test$MSSubClass)
test$MSZoning <- factor(test$MSZoning)
test$Street <- factor(test$Street)
test$Alley <- factor(test$Alley)
test$LotShape <- factor(test$LotShape)
test$LandContour <- factor(test$LandContour)
test$Utilities <- factor(test$Utilities)
test$LotConfig <- factor(test$LotConfig)
test$LandSlope <- factor(test$LandSlope)
test$Neighborhood <- factor(test$Neighborhood)
test$Condition1 <- factor(test$Condition1)
test$Condition2 <- factor(test$Condition2)
test$BldgType <- factor(test$BldgType)
test$HouseStyle <- factor(test$HouseStyle)
test$RoofStyle <- factor(test$RoofStyle)
```

```r
test$RoofMatl <- factor(test$RoofMatl)
test$Exterior1st <- factor(test$Exterior1st)
test$Exterior2nd <- factor(test$Exterior2nd)
test$MasVnrType <- factor(test$MasVnrType)
test$ExterQual <- factor(test$ExterQual)
test$ExterCond <- factor(test$ExterCond)
test$Foundation <- factor(test$Foundation)
test$BsmtQual <- factor(test$BsmtQual)
test$BsmtCond <- factor(test$BsmtCond)
test$BsmtExposure <- factor(test$BsmtExposure)
test$BsmtFinType1 <- factor(test$BsmtFinType1)
test$BsmtFinType2 <- factor(test$BsmtFinType2)
test$Heating <- factor(test$Heating)
test$HeatingQC <- factor(test$HeatingQC)
test$CentralAir <- factor(test$CentralAir)
test$Electrical <- factor(test$Electrical)
test$KitchenQual <- factor(test$KitchenQual)
test$Functional <- factor(test$Functional)
test$FireplaceQu <- factor(test$FireplaceQu)
test$GarageType <- factor(test$GarageType)
test$GarageFinish <- factor(test$GarageFinish)
test$GarageQual <- factor(test$GarageQual)
test$GarageCond <- factor(test$GarageCond)
test$PavedDrive <- factor(test$PavedDrive)
test$PoolQC <- factor(test$PoolQC)
test$Fence <- factor(test$Fence)
test$MiscFeature <- factor(test$MiscFeature)
test$SaleType <- factor(test$SaleType)
test$SaleCondition <- factor(test$SaleCondition)

# Clearing test - Lily
library(forcats)

test.cl <- test[!sapply(test,is.factor)][-1]

#SaleCondition
#plotbar("SaleCondition") #to find how to aggregate
test.cl$SaleCondition <- fct_collapse(test$SaleCondition,Abnormal= c("Abnorml","AdjLand","Alloca","Famil

#SaleType
#plotbar("SaleType") #to find how to aggregate
test.cl$SaleType <- fct_collapse(test$SaleType,New = "New",Warranty = c("WD","CWD", "VWD"), Others = c
```

```
## Warning: Unknown levels in `f`: VWD, ConLl
```

```r
#Fence
#plotbar("Fence") #to find how to aggregate
test.cl$Fence <- fct_collapse(test$Fence,No ="0",Good = c("GdPrv","GdWo"),Mini = c("MnPrv","MnWw"))

#PavedDrive
#plotbar("PavedDrive") #to find how to aggregate
test.cl$PavedDrive <- fct_collapse(test$PavedDrive,NP= c("N","P"))

#GarageType
```

```r
#plotbar("GarageType")
test.cl$GarageType <- fct_collapse(test$GarageType,Attached = c("Attchd", "BuiltIn"),Detached = "Detchd


#FireplaceQu
#plotbar("FireplaceQu")
test.cl$FireplaceQu <- fct_collapse(test$FireplaceQu,Good=c("Ex","Gd"), Aver = c("TA","Fa"), Bad=c("Po"

#Electrical
#plotbar("Electrical")
test.cl$Electrical <- fct_collapse(test$Electrical,Stand = "SBrkr", Other = c("FuseA","FuseF","FuseP",
```

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

```r
#HeatingQC
#plotbar("HeatingQC")
test.cl$HeatingQC <- fct_collapse(test$HeatingQC,Good =  c("Ex","Gd"),  Ave = "TA", Bad = c("Fa","Po"))

#"BsmtExposure"
#plotbar("BsmtExposure")
test.cl$BsmtExposure <- fct_collapse(test$BsmtExposure, Good =  "Gd",AbovMin = c("Av","Mn"), NoE = "No"

#"BsmtCond"
#plotbar("BsmtCond")
test.cl$BsmtCond <- fct_collapse(test$BsmtCond, Typical = "TA",NonTy = c("Ex","Gd","Fa","Po","0"))
```

```
## Warning: Unknown levels in `f`: Ex
```

```r
#"BsmtQual"
#plotbar("BsmtQual")
test.cl$BsmtQual <- fct_collapse(test$BsmtQual,Good=c("Ex","Gd"),   Aver = c("TA","Fa"),    Bad=c("Po",
```

```
## Warning: Unknown levels in `f`: Po
```

```r
#"Foundation"
#plotbar("Foundation")
test.cl$Foundation <- fct_collapse(test$Foundation,BrkTil= "BrkTil",CBlock = "CBlock",PConc = "PConc",O

#"ExterCond"
#plotbar("ExterCond")
test.cl$ExterCond <- fct_collapse(test$ExterCond,Good =  c("Ex","Gd"),  Ave = "TA", Bad = c("Fa","Po"))

#ExterQual
#plotbar("ExterQual")
test.cl$ExterQual <- fct_collapse(test$ExterQual,Good =  c("Ex","Gd"), Ave = "TA",  Bad = c("Fa","Po"))
```

```
## Warning: Unknown levels in `f`: Po
```

```r
#"MasVnrType"
#plotbar("MasVnrType")
test.cl$MasVnrType <- fct_collapse(test$MasVnrType, Brk= c("BrkCmn","BrkFace")  ,Stone = "Stone",   Oth
```

```
## Warning: Unknown levels in `f`: CBlock
```

```r
#"Exterior1st"
#plotbar("Exterior1st")
test.cl$Exterior1st <- fct_collapse(test$Exterior1st, Brk = c("BrkComm","BrkFace"), Wood = c("Wd Sdng",
```

```
## Warning: Unknown levels in `f`: Stone, PreCast, ImStucc, Other
```

```r
#"Exterior2nd"
#plotbar("Exterior2nd")
test.cl$Exterior2nd <- fct_collapse(test$Exterior2nd, Brk = c("BrkComm","BrkFace"), Wood = c("Wd Sdng",
```

```
## Warning: Unknown levels in `f`: BrkComm, WdShing, CemntBd, PreCast, Other
```

```r
#"RoofStyle"
#plotbar("RoofStyle")
test.cl$RoofStyle <- fct_collapse(test$RoofStyle,Gable = "Gable",   Hip = "Hip",    Others = c ("Flat",

#"HouseStyle"
#plotbar("HouseStyle")
test.cl$HouseStyle <- fct_collapse(test$HouseStyle,OneStory = "1Story", OnenHalfStory = c ("1.5Fin", "1
```

```
## Warning: Unknown levels in `f`: 2.5 Fin
```

```r
#"BldgType"
#plotbar("BldgType")
test.cl$BldgType <- fct_collapse(test$BldgType,OneFam = "1Fam", TwoFam = c ( "2FmCon", "Duplex")    , T
```

```
## Warning: Unknown levels in `f`: 2FmCon, Twnhsl
```

```r
#"Condition1"
#plotbar("Condition1")
test.cl$Condition1 <- fct_collapse(test$Condition1, Norm = "Norm", Other = c ("Artery", "Feedr", "RRNn"

#"Neighborhood"
#plotbar("Neighborhood")
test.cl$Neighborhood <- fct_collapse(test$Neighborhood, North = c ("NWAmes", "NAmes", "NoRidge", "NPkVi

#"LotConfig"
#plotbar("LotConfig")
test.cl$LotConfig <- fct_collapse(test$LotConfig,Standard = c ("Inside", "Corner"), Premium = c ("CulDS

#"LotShape"
#plotbar("LotShape")
test.cl$LotShape <- fct_collapse(test$LotShape,Regular = "Reg", Irregular = c("IR1","IR2","IR3"))

#"MSZoning"
#plotbar("MSZoning")
test.cl$MSZoning <- fct_collapse(test$MSZoning,Residentiallow = c ("RL", "RP")  ,ResidentialMedHi = c (
```

```
## Warning: Unknown levels in `f`: RP, A, I, C
```

```r
#"MSSubClass"
#plotbar("MSSubClass")
test.cl$MSSubClass <- fct_collapse(test$MSSubClass, story1=c(20,30,40,120), story1.5=c(45,50,150),story
```

linear regression - Ari

Since the outcome variable is housing prices, we will examine whether log transformation of the outcome variable will better fit our data:

```r
linbase <- glm(SalePrice~.,data = data.cl)
linlog <- glm(log(SalePrice)~., data = data.cl)
summary(linbase)
```

```
##
## Call:
## glm(formula = SalePrice ~ ., data = data.cl)
##
## Coefficients: (2 not defined because of singularities)
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)            -4.052e+05  1.364e+06  -0.297 0.766440
## LotFrontage            -2.438e+01  2.897e+01  -0.842 0.400121
## LotArea                 1.982e-01  1.015e-01   1.953 0.051028 .
## OverallQual             1.499e+04  1.225e+03  12.240  < 2e-16 ***
## OverallCond             5.623e+03  1.070e+03   5.254 1.72e-07 ***
## YearBuilt               2.177e+02  8.692e+01   2.505 0.012367 *
## YearRemodAdd            4.652e+01  6.961e+01   0.668 0.504088
## MasVnrArea              4.022e+01  7.420e+00   5.421 7.02e-08 ***
## BsmtFinSF1              1.005e+01  5.677e+00   1.770 0.076978 .
## BsmtFinSF2              9.796e-01  7.485e+00   0.131 0.895900
## BsmtUnfSF               8.786e-01  5.567e+00   0.158 0.874607
## TotalBsmtSF                    NA         NA      NA       NA
## X1stFlrSF               4.063e+01  6.527e+00   6.225 6.40e-10 ***
## X2ndFlrSF               6.279e+01  6.297e+00   9.971  < 2e-16 ***
## LowQualFinSF            2.827e+01  2.318e+01   1.219 0.222957
## GrLivArea                      NA         NA      NA       NA
## BsmtFullBath            8.426e+03  2.493e+03   3.379 0.000748 ***
## BsmtHalfBath            2.014e+03  3.875e+03   0.520 0.603361
## FullBath                5.442e+03  2.755e+03   1.975 0.048436 *
## HalfBath                2.698e+03  2.657e+03   1.015 0.310187
## BedroomAbvGr           -7.919e+03  1.713e+03  -4.622 4.17e-06 ***
## KitchenAbvGr           -1.761e+04  7.236e+03  -2.433 0.015085 *
## TotRmsAbvGrd            4.795e+03  1.212e+03   3.956 8.01e-05 ***
## Fireplaces              6.690e+03  3.060e+03   2.186 0.028954 *
## GarageYrBlt            -4.466e+01  7.465e+01  -0.598 0.549783
## GarageCars              1.535e+04  2.884e+03   5.320 1.21e-07 ***
## GarageArea             -3.749e+00  9.969e+00  -0.376 0.706948
## WoodDeckSF              2.233e+01  7.657e+00   2.916 0.003608 **
## OpenPorchSF            -4.158e+00  1.474e+01  -0.282 0.777889
## EnclosedPorch           1.843e+01  1.612e+01   1.143 0.253110
## X3SsnPorch              3.741e+01  2.916e+01   1.283 0.199754
## ScreenPorch             4.424e+01  1.605e+01   2.757 0.005918 **
## PoolArea               -5.838e+00  2.297e+01  -0.254 0.799420
## MiscVal                -1.259e+00  1.753e+00  -0.718 0.472996
## MoSold                 -4.333e+02  3.241e+02  -1.337 0.181516
## YrSold                 -1.112e+02  6.746e+02  -0.165 0.869085
## SaleConditionNormal     4.635e+03  3.163e+03   1.465 0.143042
## SaleConditionPartial   -1.259e+04  1.896e+04  -0.664 0.507016
## SaleTypeConLI           1.227e+04  1.521e+04   0.807 0.419857
## SaleTypeWarranty        5.333e+03  4.510e+03   1.183 0.237179
## SaleTypeNew             4.144e+04  1.924e+04   2.154 0.031401 *
## FenceGood              -2.839e+03  3.389e+03  -0.838 0.402339
## FenceMini               4.110e+03  2.888e+03   1.423 0.154970
## PavedDriveY             8.681e+02  3.751e+03   0.231 0.817044
## GarageTypeOther         6.409e+04  1.449e+05   0.442 0.658410
## GarageTypeAttached      6.637e+04  1.454e+05   0.457 0.648058
## GarageTypeDetached      7.028e+04  1.454e+05   0.483 0.628995
## FireplaceQuGood        -2.422e+02  4.149e+03  -0.058 0.953460
```

```
## FireplaceQuAver          -6.003e+03  4.240e+03  -1.416 0.157036
## ElectricalStand          -3.971e+03  3.487e+03  -1.139 0.255012
## HeatingQCBad              8.050e+02  5.185e+03   0.155 0.876638
## HeatingQCAve             -1.182e+03  2.345e+03  -0.504 0.614281
## BsmtExposureAbovMin       1.807e+04  3.211e+04   0.563 0.573619
## BsmtExposureGood          3.408e+04  3.224e+04   1.057 0.290617
## BsmtExposureNoE           1.098e+04  3.208e+04   0.342 0.732192
## BsmtCondTypical           2.960e+03  3.368e+03   0.879 0.379508
## BsmtQualGood             -1.559e+04  3.378e+04  -0.461 0.644546
## BsmtQualAver             -1.411e+04  3.364e+04  -0.419 0.674951
## FoundationCBlock         -9.483e+02  3.950e+03  -0.240 0.810320
## FoundationPConc           6.383e+03  4.435e+03   1.439 0.150370
## FoundationOth            -5.890e+03  8.604e+03  -0.685 0.493722
## ExterCondBad              7.529e+03  7.558e+03   0.996 0.319405
## ExterCondAve              1.227e+03  3.126e+03   0.392 0.694878
## ExterQualBad              2.505e+03  1.076e+04   0.233 0.815878
## ExterQualAve             -8.907e+03  2.957e+03  -3.012 0.002646 **
## MasVnrTypeBrk            -6.865e+03  1.199e+04  -0.573 0.567017
## MasVnrTypeOther           2.955e+03  1.181e+04   0.250 0.802379
## MasVnrTypeStone          -2.426e+03  1.215e+04  -0.200 0.841762
## Exterior1stBrk            3.057e+04  9.486e+03   3.222 0.001303 **
## Exterior1stStone_Cement   2.561e+04  1.554e+04   1.648 0.099660 .
## Exterior1stWood           1.668e+04  7.738e+03   2.156 0.031292 *
## Exterior1stMetal          1.766e+04  1.235e+04   1.430 0.152994
## Exterior1stVinyl          1.683e+04  1.088e+04   1.547 0.122087
## Exterior2ndBrk Cmn       -9.221e+03  1.458e+04  -0.632 0.527228
## Exterior2ndBrk           -9.873e+03  1.099e+04  -0.898 0.369234
## Exterior2ndStone_Cement  -1.323e+04  1.571e+04  -0.842 0.399722
## Exterior2ndCmentBd       -1.513e+03  1.590e+04  -0.095 0.924201
## Exterior2ndWood          -8.206e+03  6.940e+03  -1.182 0.237233
## Exterior2ndMetal         -4.110e+03  1.197e+04  -0.343 0.731387
## Exterior2ndVinyl         -4.404e+03  1.009e+04  -0.436 0.662575
## Exterior2ndWd Shng       -1.131e+04  8.148e+03  -1.388 0.165250
## RoofStyleGable            5.745e+03  6.139e+03   0.936 0.349469
## RoofStyleHip              1.086e+04  6.449e+03   1.683 0.092580 .
## HouseStyleOneStory        1.836e+04  1.035e+04   1.773 0.076375 .
## HouseStyle2.5Fin         -1.610e+04  1.795e+04  -0.897 0.370115
## HouseStyleTwonHalfStory  -2.378e+04  1.490e+04  -1.596 0.110763
## HouseStyleTwoStory       -1.007e+04  1.001e+04  -1.006 0.314707
## HouseStyleSplit          -2.304e+03  1.198e+04  -0.192 0.847478
## BldgType2fmCon           -7.770e+03  1.022e+04  -0.760 0.447161
## BldgTypeTwoFam           -5.004e+03  9.801e+03  -0.511 0.609769
## BldgTypeTwnhs            -2.063e+04  6.234e+03  -3.309 0.000961 ***
## BldgTypeTwn              -1.780e+04  4.344e+03  -4.098 4.41e-05 ***
## Condition1Norm            1.581e+04  2.597e+03   6.085 1.51e-09 ***
## NeighborhoodEast          1.335e+04  9.629e+03   1.386 0.165960
## NeighborhoodHighEnd       3.094e+04  9.926e+03   3.118 0.001862 **
## NeighborhoodCentral       1.287e+04  9.370e+03   1.373 0.169952
## NeighborhoodWest          1.307e+04  9.378e+03   1.394 0.163586
## NeighborhoodNorth         2.379e+04  9.295e+03   2.559 0.010600 *
## LotConfigPremium          5.526e+03  3.104e+03   1.780 0.075232 .
## LotShapeRegular          -2.657e+02  2.061e+03  -0.129 0.897421
## MSZoningOther             5.509e+03  1.241e+04   0.444 0.657255
## MSZoningResidentialMedHi  9.111e+03  1.131e+04   0.806 0.420521
```

```
## MSZoningResidentiallow    1.379e+04  1.143e+04   1.206 0.227928
## MSSubClassstory1.5        1.664e+03  1.007e+04   0.165 0.868731
## MSSubClassstory2          2.344e+02  9.053e+03   0.026 0.979351
## MSSubClassother           4.772e+03  9.705e+03   0.492 0.623004
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 1008907426)
##
##     Null deviance: 9.2079e+12  on 1459  degrees of freedom
## Residual deviance: 1.3681e+12  on 1356  degrees of freedom
## AIC: 34514
##
## Number of Fisher Scoring iterations: 2
```

```r
summary(linlog)
```

```
##
## Call:
## glm(formula = log(SalePrice) ~ ., data = data.cl)
##
## Coefficients: (2 not defined because of singularities)
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               1.367e+01  5.670e+00   2.411 0.016035 *
## LotFrontage              -2.159e-04  1.204e-04  -1.793 0.073243 .
## LotArea                   1.093e-06  4.219e-07   2.591 0.009676 **
## OverallQual               6.981e-02  5.093e-03  13.707  < 2e-16 ***
## OverallCond               4.379e-02  4.449e-03   9.841  < 2e-16 ***
## YearBuilt                 1.385e-03  3.614e-04   3.832 0.000133 ***
## YearRemodAdd              7.370e-04  2.894e-04   2.546 0.010993 *
## MasVnrArea                3.042e-05  3.085e-05   0.986 0.324152
## BsmtFinSF1                1.481e-05  2.360e-05   0.628 0.530382
## BsmtFinSF2               -4.651e-06  3.112e-05  -0.149 0.881206
## BsmtUnfSF                -1.589e-05  2.314e-05  -0.687 0.492396
## TotalBsmtSF                      NA         NA      NA       NA
## X1stFlrSF                 2.139e-04  2.713e-05   7.881 6.61e-15 ***
## X2ndFlrSF                 1.865e-04  2.618e-05   7.122 1.71e-12 ***
## LowQualFinSF              2.436e-04  9.639e-05   2.528 0.011598 *
## GrLivArea                        NA         NA      NA       NA
## BsmtFullBath              5.645e-02  1.037e-02   5.445 6.13e-08 ***
## BsmtHalfBath              1.457e-02  1.611e-02   0.905 0.365840
## FullBath                  3.452e-02  1.145e-02   3.014 0.002626 **
## HalfBath                  2.614e-02  1.105e-02   2.366 0.018139 *
## BedroomAbvGr              3.261e-04  7.123e-03   0.046 0.963496
## KitchenAbvGr             -6.595e-02  3.008e-02  -2.192 0.028542 *
## TotRmsAbvGrd              1.697e-02  5.039e-03   3.368 0.000778 ***
## Fireplaces                1.623e-02  1.272e-02   1.276 0.202209
## GarageYrBlt              -1.504e-04  3.104e-04  -0.484 0.628122
## GarageCars                6.217e-02  1.199e-02   5.184 2.50e-07 ***
## GarageArea                2.265e-05  4.145e-05   0.547 0.584810
## WoodDeckSF                1.062e-04  3.184e-05   3.336 0.000873 ***
## OpenPorchSF               2.059e-05  6.128e-05   0.336 0.736958
## EnclosedPorch             1.744e-04  6.703e-05   2.601 0.009394 **
## X3SsnPorch                2.105e-04  1.212e-04   1.736 0.082743 .
## ScreenPorch               2.969e-04  6.673e-05   4.450 9.30e-06 ***
```

```
## PoolArea                -1.497e-04  9.550e-05  -1.567 0.117291
## MiscVal                 -4.166e-06  7.289e-06  -0.571 0.567767
## MoSold                  -2.118e-04  1.348e-03  -0.157 0.875151
## YrSold                  -3.924e-03  2.805e-03  -1.399 0.162041
## SaleConditionNormal      5.531e-02  1.315e-02   4.206 2.77e-05 ***
## SaleConditionPartial    -9.144e-03  7.885e-02  -0.116 0.907694
## SaleTypeConLI           -1.982e-02  6.322e-02  -0.314 0.753940
## SaleTypeWarranty        -1.429e-02  1.875e-02  -0.762 0.446082
## SaleTypeNew              9.533e-02  7.997e-02   1.192 0.233481
## FenceGood               -1.935e-02  1.409e-02  -1.373 0.169998
## FenceMini                9.867e-04  1.201e-02   0.082 0.934521
## PavedDriveY              2.073e-02  1.560e-02   1.329 0.184047
## GarageTypeOther          2.797e-01  6.026e-01   0.464 0.642600
## GarageTypeAttached       3.158e-01  6.044e-01   0.522 0.601460
## GarageTypeDetached       3.079e-01  6.046e-01   0.509 0.610699
## FireplaceQuGood          4.842e-02  1.725e-02   2.807 0.005072 **
## FireplaceQuAver          2.343e-02  1.763e-02   1.329 0.184059
## ElectricalStand         -6.631e-03  1.450e-02  -0.457 0.647464
## HeatingQCBad            -4.454e-02  2.156e-02  -2.066 0.038998 *
## HeatingQCAve            -1.731e-02  9.749e-03  -1.775 0.076050 .
## BsmtExposureAbovMin      9.456e-02  1.335e-01   0.708 0.478830
## BsmtExposureGood         1.342e-01  1.340e-01   1.001 0.316848
## BsmtExposureNoE          7.320e-02  1.334e-01   0.549 0.583152
## BsmtCondTypical          1.242e-02  1.400e-02   0.887 0.375296
## BsmtQualGood             4.947e-02  1.404e-01   0.352 0.724665
## BsmtQualAver             3.486e-02  1.399e-01   0.249 0.803207
## FoundationCBlock         1.818e-02  1.642e-02   1.107 0.268482
## FoundationPConc          5.699e-02  1.844e-02   3.091 0.002037 **
## FoundationOth            2.867e-02  3.577e-02   0.801 0.423064
## ExterCondBad            -2.024e-02  3.142e-02  -0.644 0.519545
## ExterCondAve             1.654e-02  1.300e-02   1.273 0.203268
## ExterQualBad             2.309e-02  4.472e-02   0.516 0.605612
## ExterQualAve            -1.713e-02  1.230e-02  -1.394 0.163697
## MasVnrTypeBrk            7.363e-03  4.984e-02   0.148 0.882578
## MasVnrTypeOther          7.889e-03  4.908e-02   0.161 0.872332
## MasVnrTypeStone          1.587e-02  5.051e-02   0.314 0.753438
## Exterior1stBrk           1.349e-01  3.944e-02   3.421 0.000643 ***
## Exterior1stStone_Cement -9.777e-04  6.462e-02  -0.015 0.987931
## Exterior1stWood          2.830e-02  3.217e-02   0.880 0.379087
## Exterior1stMetal         3.730e-02  5.136e-02   0.726 0.467758
## Exterior1stVinyl         6.035e-02  4.522e-02   1.335 0.182245
## Exterior2ndBrk Cmn      -8.299e-02  6.062e-02  -1.369 0.171255
## Exterior2ndBrk          -5.385e-02  4.570e-02  -1.178 0.238862
## Exterior2ndStone_Cement -4.251e-04  6.530e-02  -0.007 0.994807
## Exterior2ndCmentBd       5.180e-02  6.611e-02   0.784 0.433449
## Exterior2ndWood         -1.112e-04  2.885e-02  -0.004 0.996925
## Exterior2ndMetal         2.151e-02  4.977e-02   0.432 0.665605
## Exterior2ndVinyl        -1.041e-02  4.195e-02  -0.248 0.804017
## Exterior2ndWd Shng      -2.690e-02  3.387e-02  -0.794 0.427229
## RoofStyleGable          -1.196e-02  2.552e-02  -0.469 0.639391
## RoofStyleHip            -1.789e-03  2.681e-02  -0.067 0.946803
## HouseStyleOneStory       3.327e-02  4.305e-02   0.773 0.439726
## HouseStyle2.5Fin        -4.275e-02  7.464e-02  -0.573 0.566947
## HouseStyleTwonHalfStory  1.632e-02  6.196e-02   0.263 0.792242
```

```
## HouseStyleTwoStory        2.958e-03  4.161e-02   0.071 0.943331
## HouseStyleSplit          -8.196e-03  4.980e-02  -0.165 0.869292
## BldgType2fmCon           -3.165e-03  4.248e-02  -0.075 0.940617
## BldgTypeTwoFam            2.387e-02  4.075e-02   0.586 0.558055
## BldgTypeTwnhs            -9.785e-02  2.592e-02  -3.775 0.000167 ***
## BldgTypeTwn              -4.998e-02  1.806e-02  -2.768 0.005724 **
## Condition1Norm            7.126e-02  1.080e-02   6.599 5.92e-11 ***
## NeighborhoodEast          3.392e-02  4.003e-02   0.847 0.396935
## NeighborhoodHighEnd       1.422e-01  4.127e-02   3.445 0.000589 ***
## NeighborhoodCentral       6.763e-02  3.895e-02   1.736 0.082755 .
## NeighborhoodWest          5.415e-02  3.899e-02   1.389 0.165108
## NeighborhoodNorth         8.597e-02  3.865e-02   2.225 0.026278 *
## LotConfigPremium          1.740e-02  1.290e-02   1.348 0.177843
## LotShapeRegular           4.096e-04  8.568e-03   0.048 0.961882
## MSZoningOther             3.768e-01  5.161e-02   7.302 4.83e-13 ***
## MSZoningResidentialMedHi  2.870e-01  4.701e-02   6.106 1.34e-09 ***
## MSZoningResidentiallow    3.568e-01  4.754e-02   7.506 1.10e-13 ***
## MSSubClassstory1.5        2.481e-02  4.186e-02   0.593 0.553500
## MSSubClassstory2         -2.645e-02  3.764e-02  -0.703 0.482307
## MSSubClassother          -8.523e-04  4.035e-02  -0.021 0.983150
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.01743899)
##
##     Null deviance: 232.801  on 1459  degrees of freedom
## Residual deviance:  23.647  on 1356  degrees of freedom
## AIC: -1666.2
##
## Number of Fisher Scoring iterations: 2
```

```r
library(boot)
cv.glm(data.cl, linbase , K = 5)$delta[1]
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful cases
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful cases

## [1] 1336497296
```

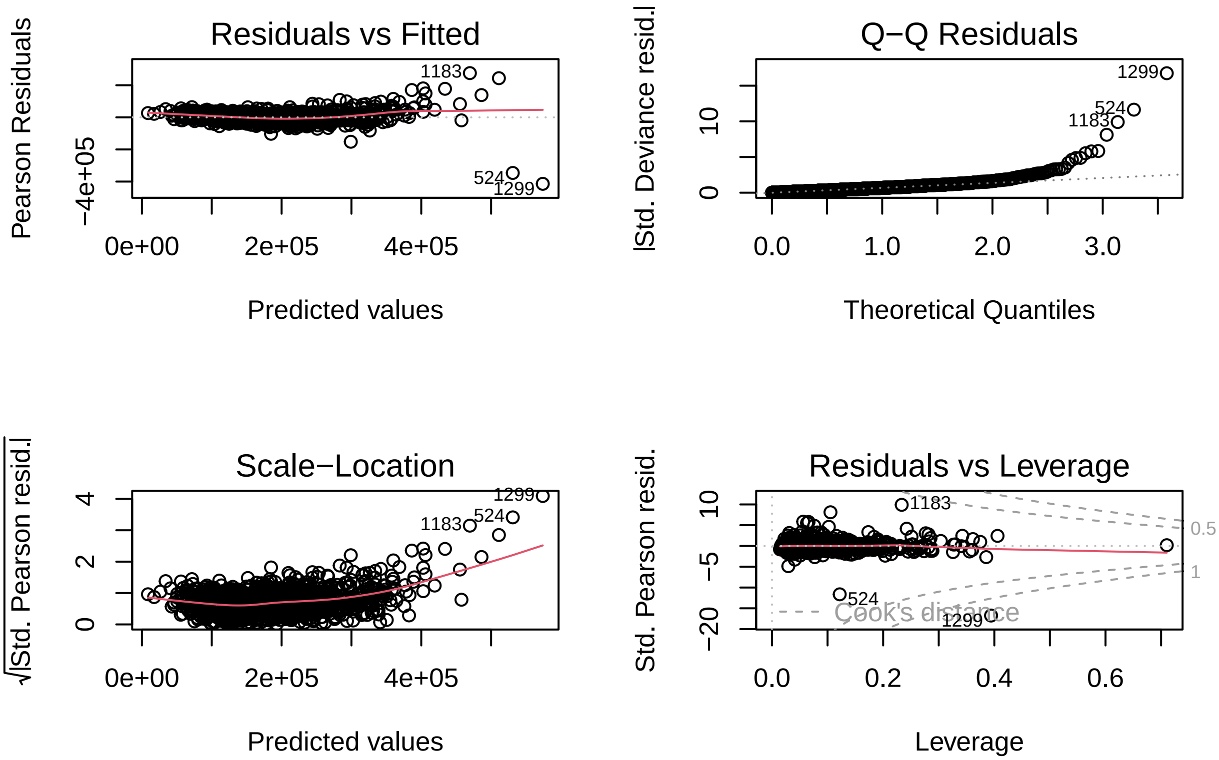
```r
cv.glm(data.cl, linlog , K = 5)$delta[1]
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful cases
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful cases

## [1] 0.02390744
```

Keep in mind that, since y-values are on different axes, it is entirely inappropriate to compare cv output. This was a proof of concept. See the residual plots below for model evaluations:

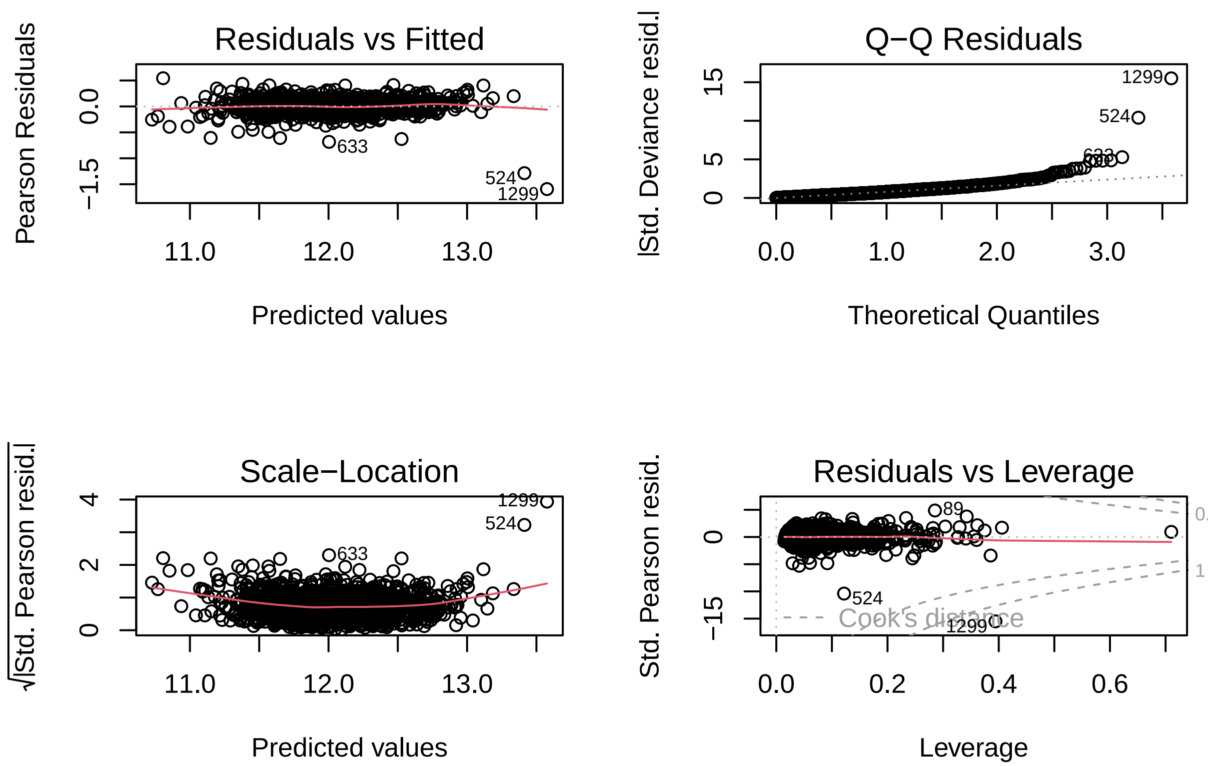
```r
par(mfrow=c(2,2))
plot(linbase)
```

```
## Warning: not plotting observations with leverage one:
##   949
```




```r
plot(linlog)
```

```
## Warning: not plotting observations with leverage one:
##   949
```



By visual inspection, the models seem similar, though the log outcome variable may be slightly more appropriate, which aligns with our intuition that prices should be measured on a log scale.

Kaggle Submission - Linear Regression

```
outlin <- test[1]
#exponentiate predictions
outlin$SalePrice <- exp(predict(linlog, test.cl))
write.csv(outlin, "outlinlog", row.names=FALSE)
```

This submission had an RMSE of 0.14107 (so an MSE of 0.01990074).

LASSO/ridge/elastic net - Ari

```
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-8
```

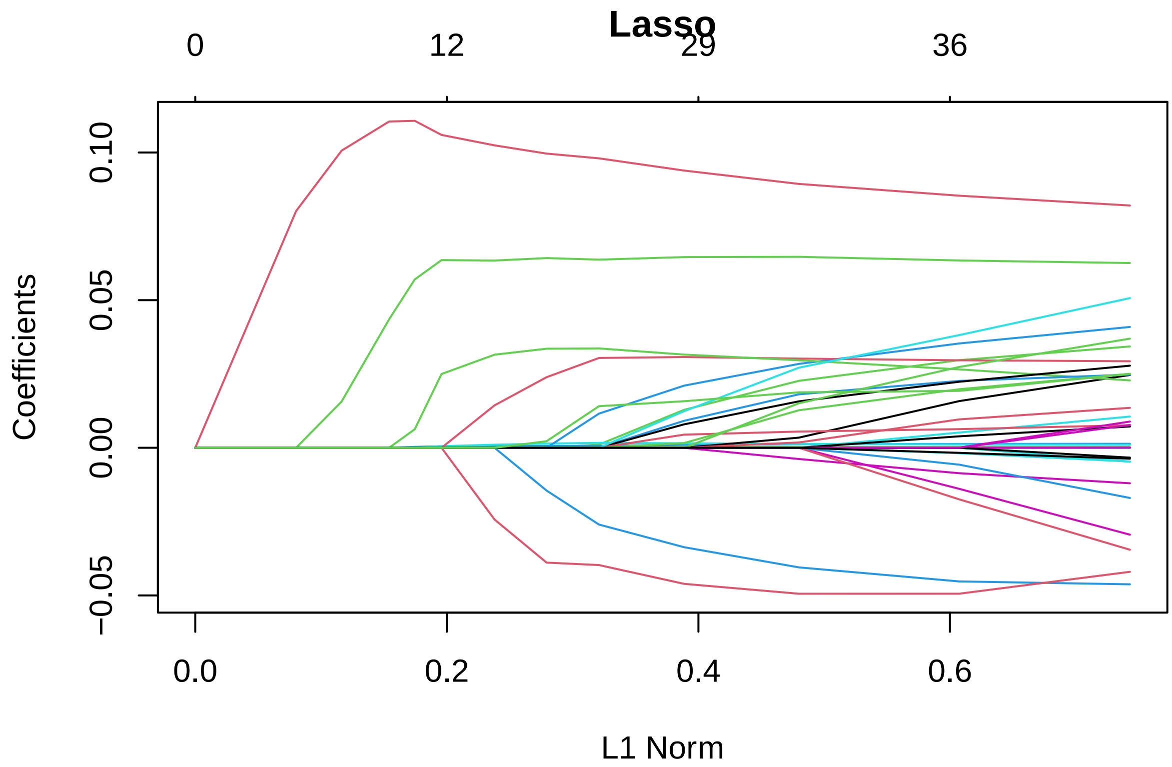
```
x <- model.matrix(log(SalePrice)~.,data.cl)[, -1]
y <- log(data.cl$SalePrice)
```

Lasso

```
grid <- 10^ seq(10, -2, length = 100)
lasso.mod <- glmnet(x, y, alpha = 1, lambda = grid)
```

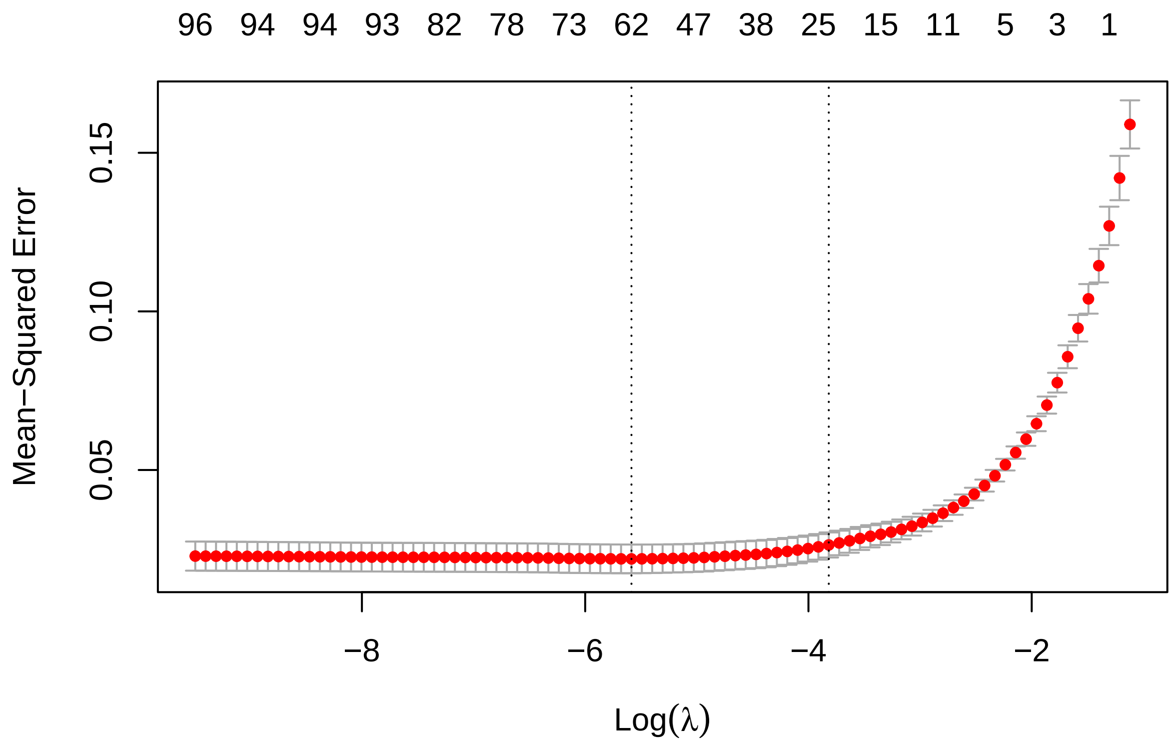
```r
plot(lasso.mod, main="Lasso")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```



Cross Validation:

```r
cv.out <- cv.glmnet(x,y, alpha = 1)
plot(cv.out)
```




```
cv.out$lambda.min
```

```
## [1] 0.003751818
```

```
min(cv.out$cvm)
```

```
## [1] 0.02197524
```

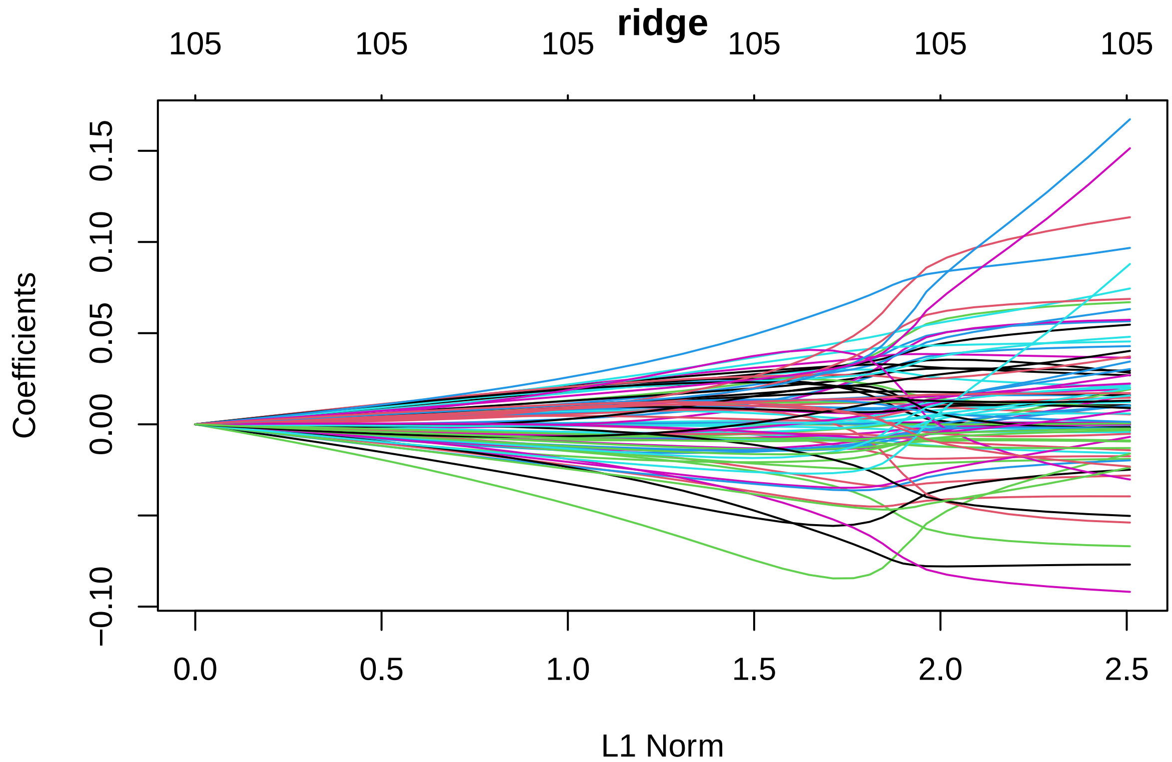
Using a lasso regression, the lowest mean Cross-Validated error is 0.02197524, with lambda of 0.003751818.

```
outlasso <- test[1]
#exponentiate predictions
outlasso$SalePrice <- exp(predict(lasso.mod, s = cv.out$lambda.min, newx = model.matrix(test$Id~., test
write.csv(outlasso, "outlasso.csv", row.names=FALSE)
```

Kaggle Submission - Lasso

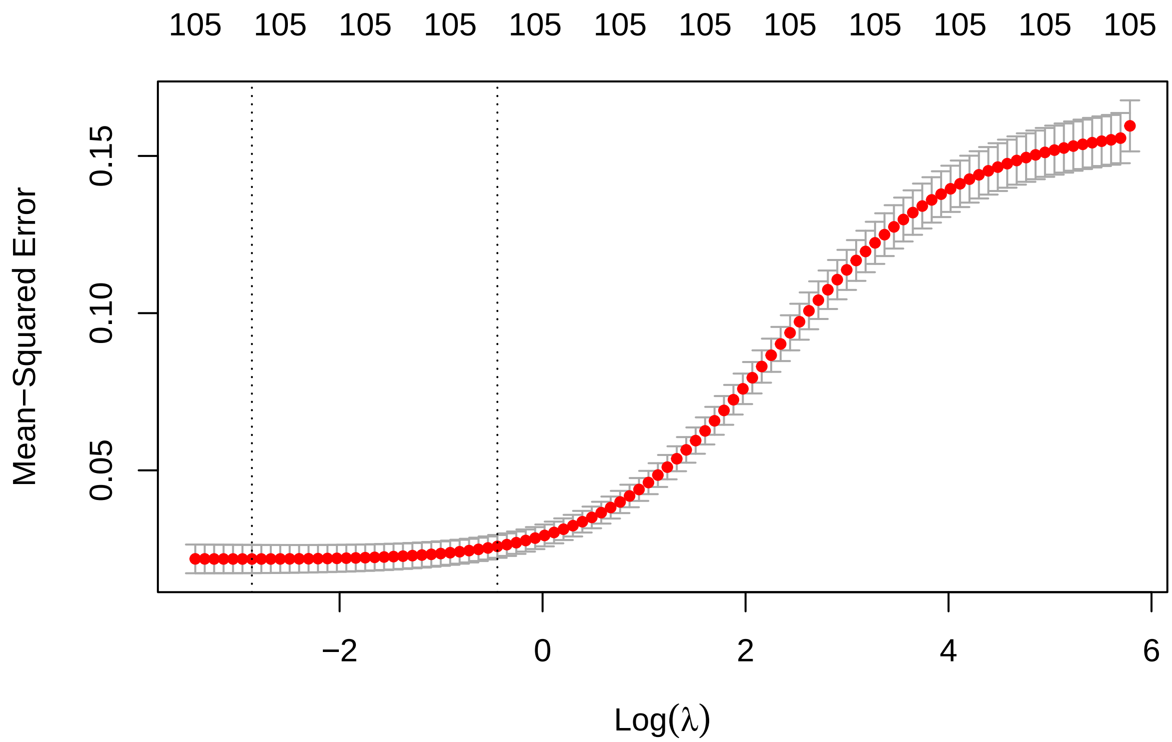
Ridge

```
grid <- 10^ seq(10, -2, length = 100)
ridge.mod <- glmnet(x, y, alpha = 0, lambda = grid)
plot(ridge.mod, main="ridge")
```



Cross Validation:

```
cv.out <- cv.glmnet(x,y, alpha = 0)
plot(cv.out)
```




```
cv.out$lambda.min
```

```
## [1] 0.0570243
```

```
min(cv.out$cvm)
```

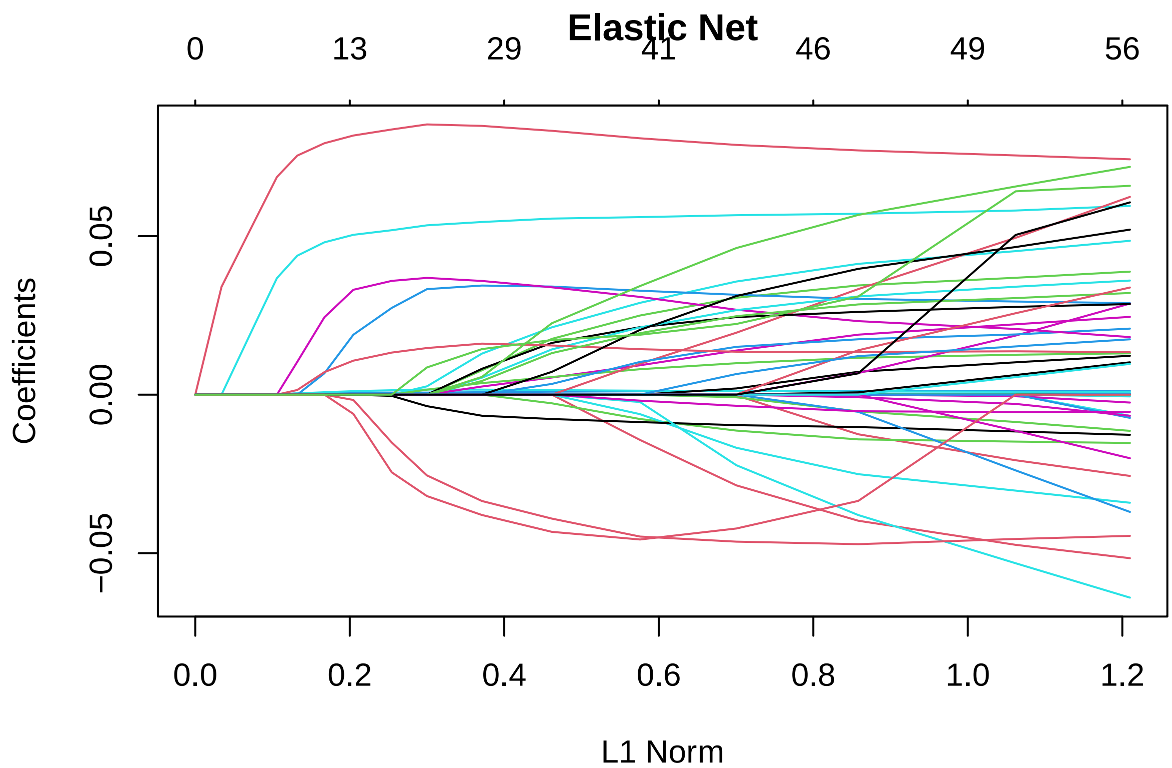
```
## [1] 0.02179968
```

Using a ridge regression, the lowest mean Cross-Validated error is 0.02179968, with lambda of 0.0570243.

Elastic Net

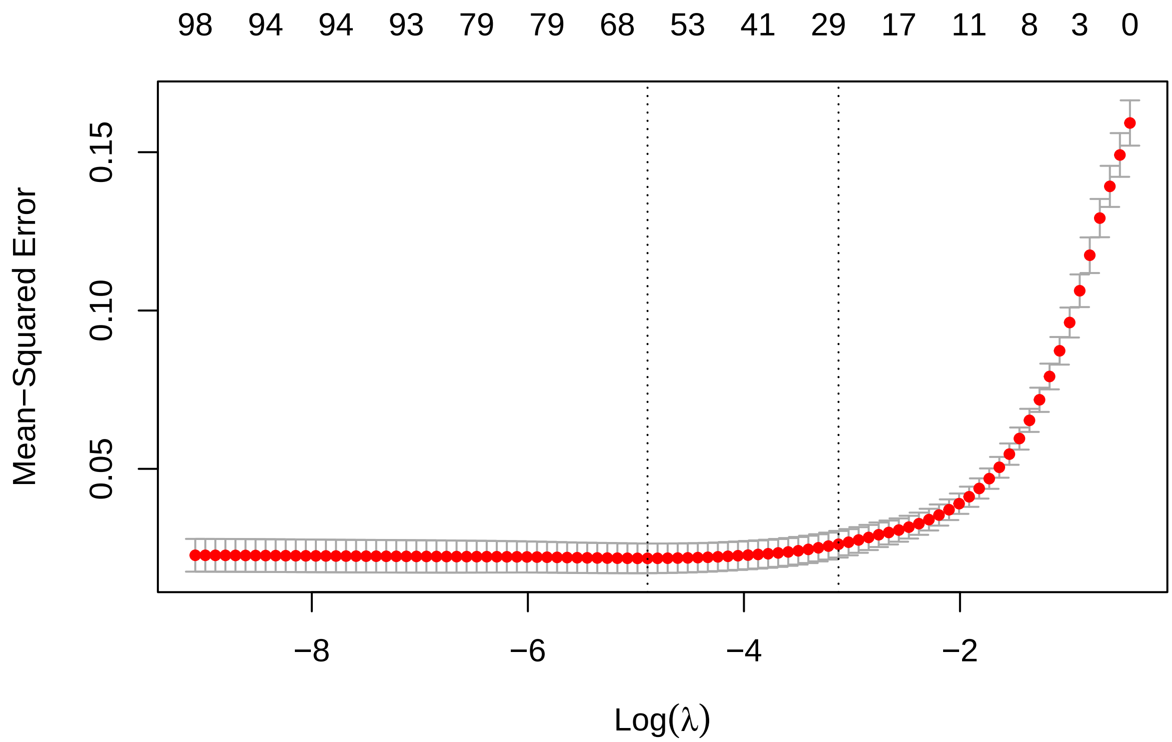
```
grid <- 10^ seq(10, -2, length = 100)
ridge.mod <- glmnet(x, y, alpha = .5, lambda = grid)
plot(ridge.mod, main="Elastic Net")
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```



Cross Validation:

```
cv.out <- cv.glmnet(x,y, alpha = .5)
plot(cv.out)
```




```r
cv.out$lambda.min
```

```
## [1] 0.007503637
```

```r
min(cv.out$cvm)
```

```
## [1] 0.02173416
```

Using an elastic net with alpha of 0.5, the lowest mean Cross-Validated error is 0.02173416, with lambda of 0.007503637.

PCR and PLS - Ari

Beginning with PCR:

```r
library(pls)
```

```
##
## Attaching package: 'pls'

## The following object is masked from 'package:stats':
##
##     loadings
```

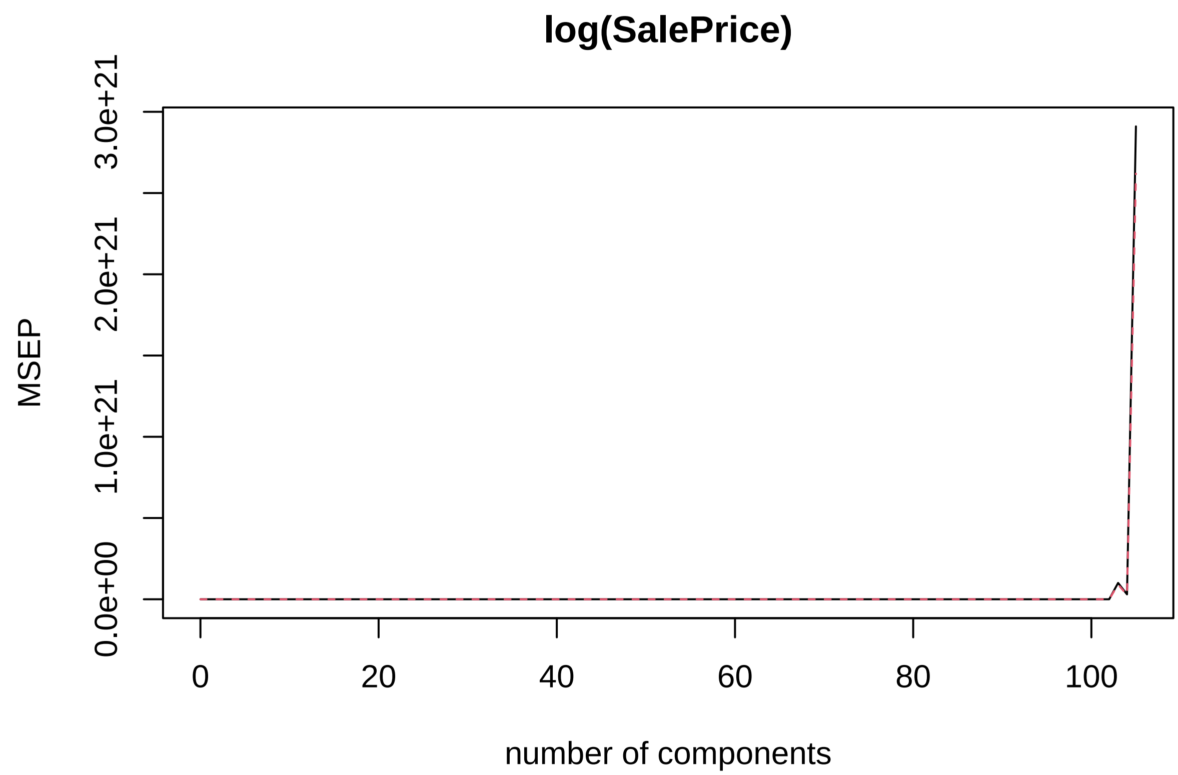
```r
pcr.fit <- pcr(log(SalePrice) ~., data = data.cl, scale = TRUE, validation = "CV")
summary(pcr.fit)
```

```
## Data:    X dimension: 1460 105
##  Y dimension: 1460 1
## Fit method: svdpc
```

```
## Number of components considered: 105
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV          0.3996   0.2063   0.2031   0.1829   0.1792   0.1769   0.1768
## adjCV       0.3996   0.2062   0.2031   0.1828   0.1791   0.1768   0.1768
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV      0.1646   0.1625   0.1586    0.1581    0.1572    0.1571    0.1570
## adjCV   0.1644   0.1623   0.1581    0.1578    0.1569    0.1568    0.1567
##        14 comps  15 comps  16 comps  17 comps  18 comps  19 comps  20 comps
## CV       0.1575    0.1578    0.1579    0.1577    0.1574    0.1572    0.1567
## adjCV    0.1572    0.1575    0.1576    0.1575    0.1571    0.1571    0.1565
##        21 comps  22 comps  23 comps  24 comps  25 comps  26 comps  27 comps
## CV       0.1567    0.1562    0.1559    0.1552    0.1550    0.1552    0.1547
## adjCV    0.1566    0.1558    0.1554    0.1547    0.1547    0.1550    0.1546
##        28 comps  29 comps  30 comps  31 comps  32 comps  33 comps  34 comps
## CV       0.1551    0.1552    0.1557    0.1564    0.1562    0.1552    0.1556
## adjCV    0.1547    0.1550    0.1555    0.1560    0.1556    0.1542    0.1548
##        35 comps  36 comps  37 comps  38 comps  39 comps  40 comps  41 comps
## CV       0.1552    0.1550    0.1550    0.1550    0.1552    0.1556    0.1553
## adjCV    0.1545    0.1545    0.1545    0.1545    0.1548    0.1552    0.1549
##        42 comps  43 comps  44 comps  45 comps  46 comps  47 comps  48 comps
## CV       0.1554    0.1551    0.1548    0.1546    0.1546    0.1552    0.1558
## adjCV    0.1551    0.1545    0.1546    0.1541    0.1542    0.1547    0.1556
##        49 comps  50 comps  51 comps  52 comps  53 comps  54 comps  55 comps
## CV       0.1558    0.1553    0.1546    0.1536    0.1537    0.1534    0.1536
## adjCV    0.1555    0.1551    0.1539    0.1530    0.1529    0.1528    0.1529
##        56 comps  57 comps  58 comps  59 comps  60 comps  61 comps  62 comps
## CV       0.1538    0.1537    0.1542    0.1541    0.1536    0.1531    0.1532
## adjCV    0.1533    0.1530    0.1536    0.1536    0.1529    0.1524    0.1526
##        63 comps  64 comps  65 comps  66 comps  67 comps  68 comps  69 comps
## CV       0.1534    0.1538    0.1519    0.1519    0.1516    0.1515    0.1519
## adjCV    0.1528    0.1533    0.1513    0.1509    0.1508    0.1508    0.1512
##        70 comps  71 comps  72 comps  73 comps  74 comps  75 comps  76 comps
## CV       0.1532    0.1534    0.1529    0.1534    0.1533    0.1540    0.1536
## adjCV    0.1524    0.1526    0.1522    0.1525    0.1525    0.1532    0.1528
##        77 comps  78 comps  79 comps  80 comps  81 comps  82 comps  83 comps
## CV       0.1534    0.1518    0.1522    0.1495    0.1498    0.1495    0.1496
## adjCV    0.1524    0.1508    0.1512    0.1485    0.1488    0.1485    0.1485
##        84 comps  85 comps  86 comps  87 comps  88 comps  89 comps  90 comps
## CV       0.1500    0.1495    0.1492    0.1494    0.1496    0.1498    0.1501
## adjCV    0.1489    0.1484    0.1481    0.1483    0.1485    0.1487    0.1490
##        91 comps  92 comps  93 comps  94 comps  95 comps  96 comps  97 comps
## CV       0.1504    0.1505    0.1504    0.1506    0.1510    0.1495    0.1486
## adjCV    0.1492    0.1494    0.1492    0.1495    0.1498    0.1478    0.1472
##        98 comps  99 comps  100 comps  101 comps  102 comps  103 comps
## CV       0.1486    0.1486     0.1492     0.1504     0.1504  1.003e+10
## adjCV    0.1474    0.1474     0.1479     0.1491     0.1491  9.516e+09
##        104 comps  105 comps
## CV     5.488e+09  5.395e+10
## adjCV  5.207e+09  5.118e+10
##
## TRAINING: % variance explained
```

```
##                  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
## X                  12.43    17.85    22.71    26.46    29.49    32.32    34.94
## log(SalePrice)     73.44    74.34    79.16    80.07    80.70    80.71    83.51
##                  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## X                  37.42    39.60     41.67     43.61     45.41     47.12
## log(SalePrice)     83.89    84.74     84.79     85.02     85.02     85.03
##                  14 comps  15 comps  16 comps  17 comps  18 comps  19 comps
## X                   48.74     50.30     51.81     53.28     54.68     56.02
## log(SalePrice)      85.06     85.07     85.09     85.12     85.25     85.29
##                  20 comps  21 comps  22 comps  23 comps  24 comps  25 comps
## X                   57.33     58.57     59.79     61.00     62.17     63.31
## log(SalePrice)      85.48     85.49     85.68     85.77     85.84     85.84
##                  26 comps  27 comps  28 comps  29 comps  30 comps  31 comps
## X                   64.45     65.56     66.65     67.72     68.76     69.78
## log(SalePrice)      85.84     85.88     85.99     86.00     86.01     86.13
##                  32 comps  33 comps  34 comps  35 comps  36 comps  37 comps
## X                   70.78     71.77     72.75     73.72     74.66     75.60
## log(SalePrice)      86.21     86.38     86.38     86.38     86.38     86.39
##                  38 comps  39 comps  40 comps  41 comps  42 comps  43 comps
## X                   76.50     77.39     78.27     79.13     79.97     80.78
## log(SalePrice)      86.41     86.42     86.42     86.48     86.48     86.57
##                  44 comps  45 comps  46 comps  47 comps  48 comps  49 comps
## X                   81.59     82.38     83.13     83.88     84.62     85.33
## log(SalePrice)      86.58     86.67     86.69     86.72     86.73     86.87
##                  50 comps  51 comps  52 comps  53 comps  54 comps  55 comps
## X                   86.03     86.72     87.37     88.03     88.65     89.26
## log(SalePrice)      86.90     87.07     87.14     87.21     87.22     87.24
##                  56 comps  57 comps  58 comps  59 comps  60 comps  61 comps
## X                   89.87     90.45     91.01     91.56     92.09     92.60
## log(SalePrice)      87.24     87.33     87.34     87.34     87.50     87.56
##                  62 comps  63 comps  64 comps  65 comps  66 comps  67 comps
## X                   93.09     93.56     94.01     94.44     94.84     95.23
## log(SalePrice)      87.58     87.59     87.59     87.87     88.01     88.03
##                  68 comps  69 comps  70 comps  71 comps  72 comps  73 comps
## X                   95.60     95.96     96.31     96.64     96.95     97.25
## log(SalePrice)      88.03     88.06     88.07     88.07     88.09     88.16
##                  74 comps  75 comps  76 comps  77 comps  78 comps  79 comps
## X                   97.52     97.79     98.05     98.27     98.49     98.69
## log(SalePrice)      88.16     88.18     88.28     88.46     88.72     88.73
##                  80 comps  81 comps  82 comps  83 comps  84 comps  85 comps
## X                   98.88     99.04     99.17     99.28     99.38     99.46
## log(SalePrice)      89.02     89.03     89.13     89.19     89.22     89.33
##                  86 comps  87 comps  88 comps  89 comps  90 comps  91 comps
## X                   99.53     99.61     99.67     99.73     99.78     99.83
## log(SalePrice)      89.35     89.36     89.37     89.38     89.38     89.38
##                  92 comps  93 comps  94 comps  95 comps  96 comps  97 comps
## X                   99.86     99.89     99.91     99.93     99.95     99.96
## log(SalePrice)      89.38     89.42     89.42     89.42     89.82     89.82
##                  98 comps  99 comps  100 comps  101 comps  102 comps  103 comps
## X                   99.97     99.98      99.99     100.00     100.00     100.00
## log(SalePrice)      89.82     89.84      89.84      89.84      89.84      89.84
##                  104 comps  105 comps
## X                   100.00     100.00
## log(SalePrice)       89.85      89.86
```

```r
validationplot(pcr.fit, val.type = "MSEP")
```



We see that the CV levels out at around 9 components, with few, if any, improvements to CV beyond that point. The CV-derived MSE associated with 9 components is $0.1586^2 = 0.02515396$, which is higher than the lasso, ridge, and elastic net regressions.

Since we have an outcome variable, supervised learning is possible. Consequently, Partial Least Squares is preferred to Principal Component Analysis.

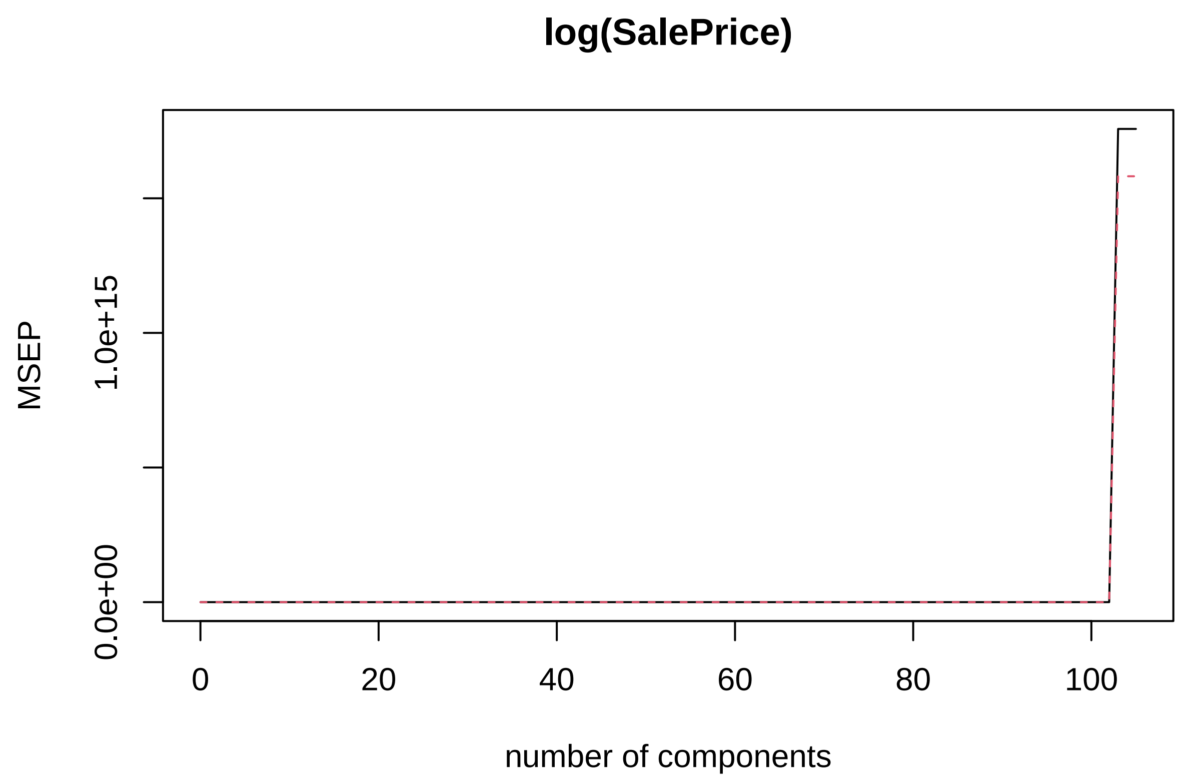
```r
pls.fit <- plsr(log(SalePrice) ~., data = data.cl, scale = TRUE, validation = "CV")
summary(pls.fit)
```

```
## Data:    X dimension: 1460 105
##  Y dimension: 1460 1
## Fit method: kernelpls
## Number of components considered: 105
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV          0.3996    0.181   0.1555   0.1521   0.1527   0.1533   0.1526
## adjCV       0.3996    0.181   0.1552   0.1515   0.1518   0.1522   0.1514
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV      0.1516   0.1511   0.1513    0.1511    0.1508    0.1510    0.1505
## adjCV   0.1505   0.1500   0.1501    0.1499    0.1496    0.1498    0.1492
##        14 comps  15 comps  16 comps  17 comps  18 comps  19 comps  20 comps
## CV       0.1502    0.1504    0.1508    0.1508    0.1507    0.1507    0.1505
```

```
## adjCV    0.1490    0.1492    0.1495    0.1495    0.1494    0.1494    0.1491
##        21 comps  22 comps  23 comps  24 comps  25 comps  26 comps  27 comps
## CV       0.1502    0.1502    0.1500    0.1501    0.1501    0.1501    0.1500
## adjCV    0.1489    0.1489    0.1488    0.1488    0.1488    0.1488    0.1487
##        28 comps  29 comps  30 comps  31 comps  32 comps  33 comps  34 comps
## CV       0.1499    0.1499    0.1499    0.1499    0.1499    0.1499    0.1499
## adjCV    0.1486    0.1486    0.1486    0.1486    0.1486    0.1486    0.1486
##        35 comps  36 comps  37 comps  38 comps  39 comps  40 comps  41 comps
## CV       0.1499    0.1499    0.1499    0.1499    0.1499    0.1500    0.1500
## adjCV    0.1486    0.1486    0.1486    0.1487    0.1487    0.1487    0.1487
##        42 comps  43 comps  44 comps  45 comps  46 comps  47 comps  48 comps
## CV       0.1500    0.1500    0.1500    0.1500    0.1500    0.1500    0.1500
## adjCV    0.1487    0.1487    0.1487    0.1487    0.1487    0.1487    0.1487
##        49 comps  50 comps  51 comps  52 comps  53 comps  54 comps  55 comps
## CV       0.1500    0.1500    0.1500    0.1500    0.1501    0.1501    0.1501
## adjCV    0.1487    0.1487    0.1487    0.1488    0.1488    0.1488    0.1488
##        56 comps  57 comps  58 comps  59 comps  60 comps  61 comps  62 comps
## CV       0.1501    0.1501    0.1501    0.1501    0.1501    0.1501    0.1501
## adjCV    0.1488    0.1488    0.1488    0.1488    0.1488    0.1488    0.1488
##        63 comps  64 comps  65 comps  66 comps  67 comps  68 comps  69 comps
## CV       0.1501    0.1501    0.1501    0.1501    0.1501    0.1501    0.1501
## adjCV    0.1488    0.1489    0.1489    0.1488    0.1488    0.1488    0.1488
##        70 comps  71 comps  72 comps  73 comps  74 comps  75 comps  76 comps
## CV       0.1501    0.1501    0.1501    0.1501    0.1501    0.1501    0.1501
## adjCV    0.1488    0.1488    0.1488    0.1488    0.1488    0.1488    0.1488
##        77 comps  78 comps  79 comps  80 comps  81 comps  82 comps  83 comps
## CV       0.1501    0.1501    0.1501    0.1501    0.1502    0.1501    0.1501
## adjCV    0.1488    0.1489    0.1489    0.1489    0.1489    0.1489    0.1489
##        84 comps  85 comps  86 comps  87 comps  88 comps  89 comps  90 comps
## CV       0.1501    0.1501    0.1501    0.1501    0.1501    0.1501    0.1501
## adjCV    0.1489    0.1489    0.1489    0.1489    0.1489    0.1489    0.1489
##        91 comps  92 comps  93 comps  94 comps  95 comps  96 comps  97 comps
## CV       0.1502    0.1501    0.1502    0.1502    0.1502    0.1502    0.1502
## adjCV    0.1489    0.1489    0.1489    0.1489    0.1489    0.1489    0.1489
##        98 comps  99 comps  100 comps  101 comps  102 comps  103 comps
## CV       0.1502    0.1502     0.1502     0.1502     0.1502   41923774
## adjCV    0.1489    0.1489     0.1489     0.1489     0.1489   39772384
##        104 comps  105 comps
## CV      41923933   41923897
## adjCV   39772535   39772501
##
## TRAINING: % variance explained
##                 1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
## X                 12.30    16.57    19.63    22.09    25.61    28.28    30.50
## log(SalePrice)    79.91    86.09    87.67    88.48    88.81    89.06    89.24
##                 8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## X                 32.38    33.98     35.87     37.07     38.33     39.68
## log(SalePrice)    89.35    89.43     89.47     89.53     89.57     89.62
##                 14 comps  15 comps  16 comps  17 comps  18 comps  19 comps
## X                  41.02     42.49     43.91     45.57     46.83     47.79
## log(SalePrice)     89.66     89.69     89.73     89.76     89.79     89.81
##                 20 comps  21 comps  22 comps  23 comps  24 comps  25 comps
## X                  48.86     50.04     51.16     52.12     53.14     54.03
## log(SalePrice)     89.82     89.83     89.83     89.83     89.84     89.84
```

```
##                 26 comps  27 comps  28 comps  29 comps  30 comps  31 comps
## X                  55.04     55.97     56.83     57.78     58.67     59.48
## log(SalePrice)     89.84     89.84     89.84     89.84     89.84     89.84
##                 32 comps  33 comps  34 comps  35 comps  36 comps  37 comps
## X                  60.32     61.08     61.96     62.79     63.47     64.27
## log(SalePrice)     89.84     89.84     89.84     89.84     89.84     89.84
##                 38 comps  39 comps  40 comps  41 comps  42 comps  43 comps
## X                  65.12     65.95     66.65     67.35     68.06     68.87
## log(SalePrice)     89.84     89.84     89.84     89.84     89.84     89.84
##                 44 comps  45 comps  46 comps  47 comps  48 comps  49 comps
## X                  69.44     70.08     70.71     71.42     72.03     72.64
## log(SalePrice)     89.84     89.84     89.84     89.84     89.84     89.84
##                 50 comps  51 comps  52 comps  53 comps  54 comps  55 comps
## X                  73.34     73.98     74.52     75.26     75.87     76.38
## log(SalePrice)     89.84     89.84     89.84     89.84     89.84     89.84
##                 56 comps  57 comps  58 comps  59 comps  60 comps  61 comps
## X                  77.00     77.52     78.12     78.72     79.27     79.87
## log(SalePrice)     89.84     89.84     89.84     89.84     89.84     89.84
##                 62 comps  63 comps  64 comps  65 comps  66 comps  67 comps
## X                  80.49     81.05     81.55     82.12     82.80     83.19
## log(SalePrice)     89.84     89.84     89.84     89.84     89.84     89.84
##                 68 comps  69 comps  70 comps  71 comps  72 comps  73 comps
## X                  83.95     84.46     85.08     85.62     86.08     86.67
## log(SalePrice)     89.84     89.84     89.84     89.84     89.84     89.84
##                 74 comps  75 comps  76 comps  77 comps  78 comps  79 comps
## X                  87.24     87.73     88.19     88.77     89.26     89.73
## log(SalePrice)     89.84     89.84     89.84     89.84     89.84     89.84
##                 80 comps  81 comps  82 comps  83 comps  84 comps  85 comps
## X                  90.09     90.60     91.13     91.67     92.16     92.53
## log(SalePrice)     89.84     89.84     89.84     89.84     89.84     89.84
##                 86 comps  87 comps  88 comps  89 comps  90 comps  91 comps
## X                  93.18     93.71     94.21     94.64     95.21     95.73
## log(SalePrice)     89.84     89.84     89.84     89.84     89.84     89.84
##                 92 comps  93 comps  94 comps  95 comps  96 comps  97 comps
## X                  96.16     96.50     97.09     97.60     97.91     98.39
## log(SalePrice)     89.84     89.84     89.84     89.84     89.84     89.84
##                 98 comps  99 comps  100 comps  101 comps  102 comps  103 comps
## X                  98.93     99.15      99.37      99.97      99.99     100.00
## log(SalePrice)     89.84     89.84      89.84      89.84      89.84      89.84
##                 104 comps  105 comps
## X                  100.01     100.01
## log(SalePrice)      89.84      89.84
```

```
validationplot(pls.fit, val.type = "MSEP")
```



We see that the CV levels out at around 3 components, with few, if any, improvements to CV beyond that point. The CV-derived MSE associated with 3 components is $0.1521^2 = 0.02313441$, which is higher than the lasso, ridge, and elastic net regressions, but lower than PCR.

regression trees and random forests

```r
library(tree)
```

```
## Warning: package 'tree' was built under R version 4.4.3
```

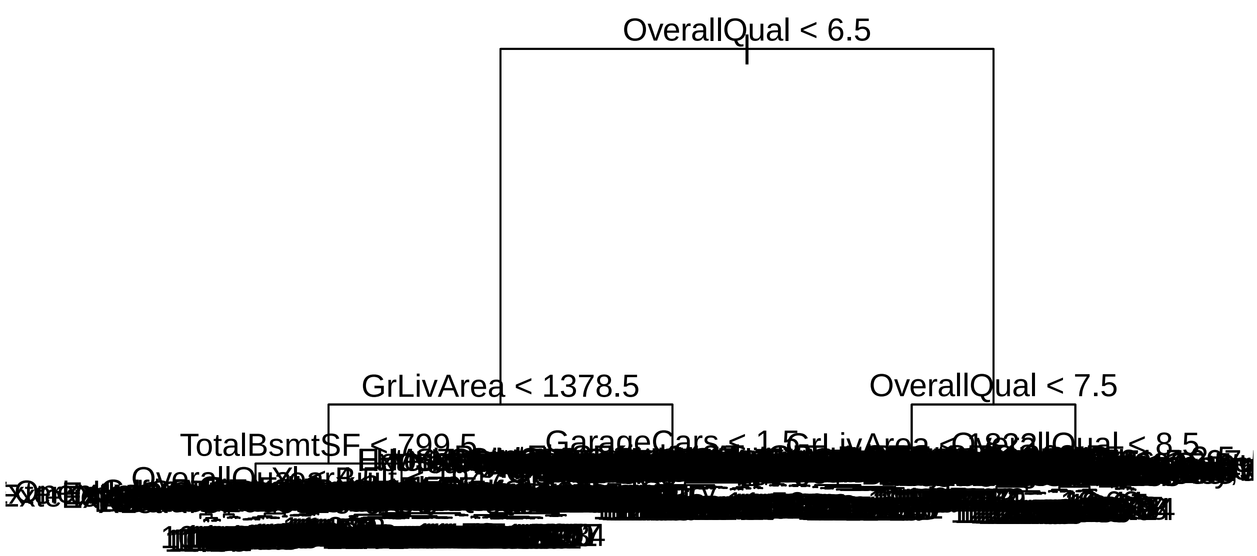
The tree

```r
tree.reg <- tree(log(SalePrice)~., data = data.cl, control = tree.control(nobs = length(data.cl[,1]), mi
summary(tree.reg)
```

```
##
## Regression tree:
## tree(formula = log(SalePrice) ~ ., data = data.cl, control = tree.control(nobs = length(data.cl[,
##     1]), mindev = 0))
## Variables actually used in tree construction:
##  [1] "OverallQual"   "GrLivArea"     "TotalBsmtSF"   "X1stFlrSF"
##  [5] "YearBuilt"     "YrSold"        "LotArea"       "Exterior1st"
##  [9] "HouseStyle"    "WoodDeckSF"    "BsmtUnfSF"     "MoSold"
## [13] "BsmtFinSF1"    "Condition1"    "LotFrontage"   "GarageYrBlt"
## [17] "Exterior2nd"   "OverallCond"   "ExterCond"     "GarageArea"
## [21] "TotRmsAbvGrd"  "Neighborhood"  "GarageCars"    "HeatingQC"
```

```
## [25] "BsmtFinSF2"    "YearRemodAdd"  "LotShape"      "MasVnrArea"
## [29] "FireplaceQu"   "FullBath"      "SaleCondition" "HalfBath"
## [33] "BsmtExposure"  "BedroomAbvGr"  "MSZoning"      "X2ndFlrSF"
## [37] "KitchenAbvGr"  "BsmtQual"      "BsmtCond"      "ExterQual"
## [41] "OpenPorchSF"   "GarageType"    "BsmtFullBath"  "SaleType"
## [45] "LotConfig"     "MasVnrType"
## Number of terminal nodes:  239
## Residual mean deviance:  0.01049 = 12.8 / 1221
## Distribution of residuals:
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.5957000 -0.0388700  0.0005882  0.0000000  0.0435600  0.4462000
```

```r
plot(tree.reg)
text(tree.reg, pretty = 0)
```



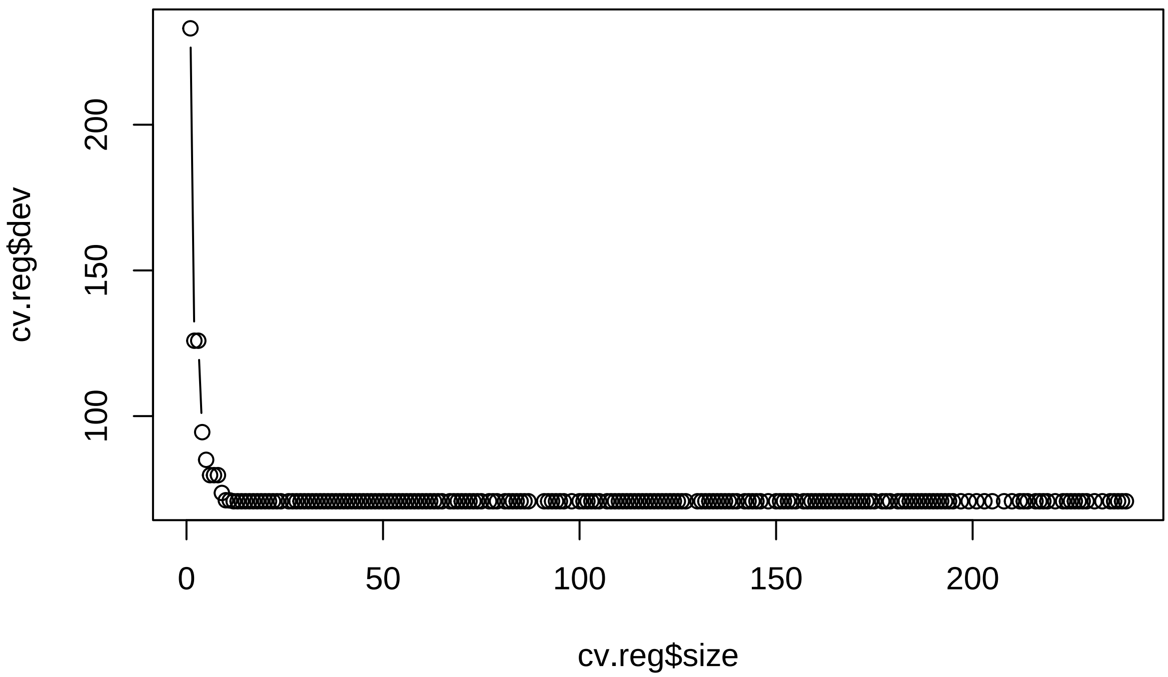

```r
cv.reg <- cv.tree(tree.reg, K=10)
print(cv.reg)
```

```
## $size
##   [1] 239 238 237 236 235 233 231 229 228 227 226 225 224 223 221 219 218 217
##  [19] 216 214 213 212 210 208 205 203 201 199 197 195 194 193 192 191 190 189
##  [37] 188 187 186 185 184 183 182 181 179 178 177 175 174 173 172 171 170 169
##  [55] 168 167 166 165 164 163 162 161 160 159 158 157 155 154 153 152 151 150
##  [73] 148 146 145 144 143 142 140 139 138 137 136 135 134 133 132 131 130 127
##  [91] 126 125 124 123 122 121 120 119 118 117 116 115 114 113 112 111 110 109
## [109] 108 107 105 104 103 102 101 100  98  96  95  94  93  92  91  87  86  85
## [127]  84  83  82  81  79  78  77  75  74  73  72  71  70  69  68  67  65  64
```

```
## [145]  63  62  61  60  59  58  57  56  55  54  53  52  51  50  49  48  47  46
## [163]  45  44  43  42  41  40  39  38  37  36  35  34  33  32  31  30  29  28
## [181]  27  26  24  23  22  21  20  19  18  17  16  15  14  13  12  11  10   9
## [199]   8   7   6   5   4   3   2   1
##
## $dev
##   [1]  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425
##   [8]  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425
##  [15]  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425
##  [22]  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425
##  [29]  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425
##  [36]  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425
##  [43]  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425
##  [50]  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425
##  [57]  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425
##  [64]  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425
##  [71]  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425
##  [78]  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425
##  [85]  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425
##  [92]  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425
##  [99]  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425
## [106]  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425
## [113]  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425
## [120]  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425
## [127]  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425
## [134]  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425
## [141]  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425
## [148]  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425
## [155]  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425
## [162]  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425
## [169]  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425
## [176]  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425
## [183]  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425
## [190]  70.78425  70.78425  70.78425  70.78425  70.78425  70.78425  71.06924
## [197]  71.13296  73.62431  79.73687  79.73687  79.73687  85.00138  94.49438
## [204] 125.87332 125.87332 233.07362
##
## $k
##   [1]          -Inf 1.229729e-03 1.918480e-03 2.369698e-03 2.739378e-03
##   [6] 3.020189e-03 3.211487e-03 3.719489e-03 5.215247e-03 6.376954e-03
##  [11] 6.485753e-03 7.541607e-03 7.906839e-03 8.150037e-03 1.062930e-02
##  [16] 1.088330e-02 1.116896e-02 1.129106e-02 1.172615e-02 1.212359e-02
##  [21] 1.223277e-02 1.363249e-02 1.491342e-02 1.511428e-02 1.589930e-02
##  [26] 1.696867e-02 1.713346e-02 1.744132e-02 1.773081e-02 1.786479e-02
##  [31] 1.924679e-02 2.021546e-02 2.109839e-02 2.159760e-02 2.215771e-02
##  [36] 2.242138e-02 2.286310e-02 2.324377e-02 2.396179e-02 2.425097e-02
##  [41] 2.574196e-02 2.647983e-02 2.668440e-02 2.700948e-02 2.729452e-02
##  [46] 2.886604e-02 2.962653e-02 3.015838e-02 3.028029e-02 3.143732e-02
##  [51] 3.183503e-02 3.249083e-02 3.284683e-02 3.327510e-02 3.449485e-02
##  [56] 3.472777e-02 3.612982e-02 3.642057e-02 3.692354e-02 3.968667e-02
##  [61] 4.077562e-02 4.095330e-02 4.303659e-02 4.466123e-02 4.572132e-02
##  [66] 4.663467e-02 4.779281e-02 4.895026e-02 4.984238e-02 4.998915e-02
##  [71] 5.052338e-02 5.070531e-02 5.269815e-02 5.461222e-02 5.608540e-02
##  [76] 5.622817e-02 5.680491e-02 5.748996e-02 5.897576e-02 6.021186e-02
```

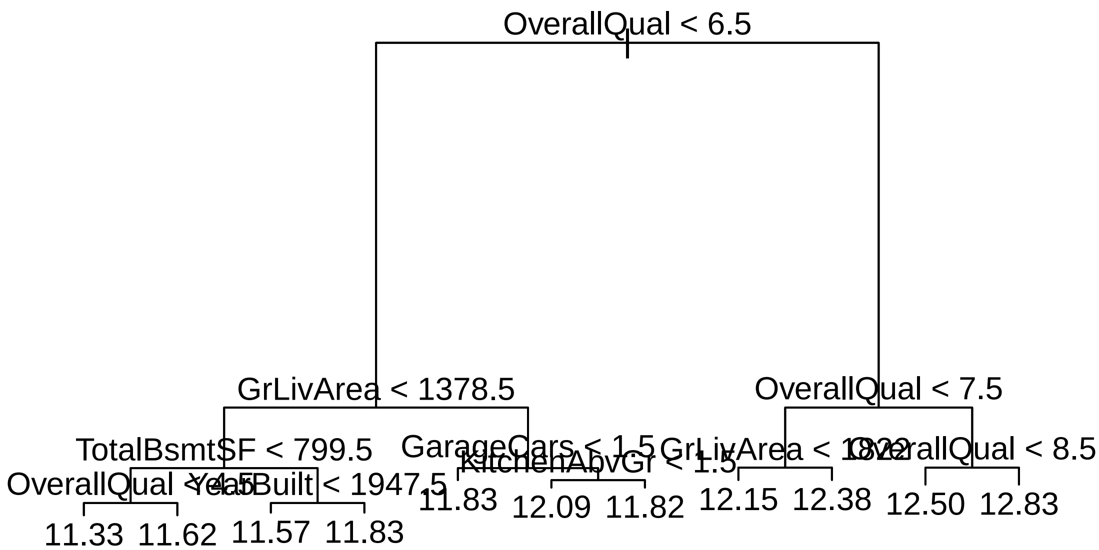
```
##  [81] 6.094426e-02 6.138097e-02 6.544574e-02 6.594105e-02 6.683070e-02
##  [86] 6.715572e-02 6.746194e-02 6.764058e-02 6.868112e-02 6.918579e-02
##  [91] 6.959999e-02 7.240075e-02 7.763235e-02 7.792474e-02 7.874782e-02
##  [96] 8.012493e-02 8.065203e-02 8.423124e-02 8.508617e-02 8.553573e-02
## [101] 8.574871e-02 9.549097e-02 9.579276e-02 9.616228e-02 1.014175e-01
## [106] 1.064005e-01 1.072514e-01 1.073900e-01 1.075169e-01 1.088210e-01
## [111] 1.111045e-01 1.119679e-01 1.132924e-01 1.148014e-01 1.168293e-01
## [116] 1.198026e-01 1.214148e-01 1.240187e-01 1.271801e-01 1.279614e-01
## [121] 1.291362e-01 1.330816e-01 1.338210e-01 1.356710e-01 1.358796e-01
## [126] 1.379535e-01 1.451799e-01 1.490491e-01 1.495800e-01 1.511168e-01
## [131] 1.531946e-01 1.605445e-01 1.608159e-01 1.630836e-01 1.655814e-01
## [136] 1.659064e-01 1.752210e-01 1.773386e-01 1.857656e-01 1.876875e-01
## [141] 1.935361e-01 1.980941e-01 2.023179e-01 2.051377e-01 2.059153e-01
## [146] 2.084160e-01 2.182847e-01 2.382129e-01 2.492212e-01 2.560572e-01
## [151] 2.628875e-01 2.653909e-01 2.746014e-01 2.784699e-01 3.042111e-01
## [156] 3.071053e-01 3.265857e-01 3.375314e-01 3.414078e-01 3.506486e-01
## [161] 3.515565e-01 3.529156e-01 3.888351e-01 3.914985e-01 4.151745e-01
## [166] 4.243287e-01 4.248165e-01 4.424588e-01 4.679549e-01 4.725113e-01
## [171] 4.778235e-01 4.807534e-01 5.011534e-01 5.029941e-01 5.151566e-01
## [176] 5.683890e-01 5.726424e-01 5.743427e-01 5.792967e-01 6.369784e-01
## [181] 6.452246e-01 7.705099e-01 7.867520e-01 8.563857e-01 9.111062e-01
## [186] 9.515659e-01 1.077821e+00 1.101731e+00 1.171509e+00 1.205652e+00
## [191] 1.341675e+00 1.400527e+00 1.475171e+00 1.838210e+00 1.893201e+00
## [196] 2.151329e+00 2.209398e+00 2.839906e+00 3.535005e+00 3.615300e+00
## [201] 3.631448e+00 4.988011e+00 1.027251e+01 1.769635e+01 1.783793e+01
## [206] 1.074531e+02
## 
## $method
## [1] "deviance"
## 
## attr(,"class")
## [1] "prune"         "tree.sequence"
```

```
plot(cv.reg$size, cv.reg$dev, type = "b")
```



The optimal tree size appears to be 11. Let's make one of th that size

```
pruned <- prune.tree(tree.reg, best = 11)
plot(pruned)
text(pruned, pretty = 0)
```



Referring back to the cross-validation, we see that the sum of squared errors for the size-eleven tree is 77.12415. To get the *Mean Squared Error* we divide by the number of observations: $\frac{71.06924}{1460} = 0.04867756$. This is considerably worse than dimension reduction or regularization approaches, as would be expected.

Random Forest

```
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.4.3
```

```
## randomForest 4.7-1.2
```

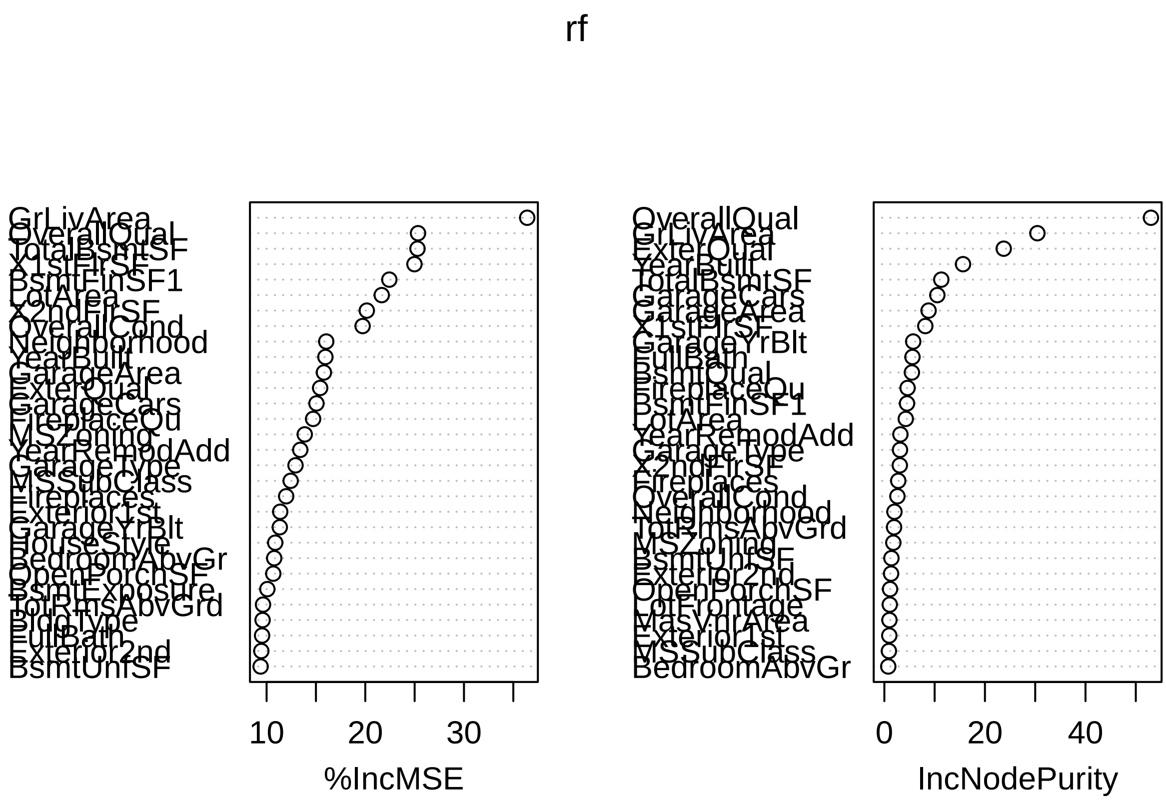
```
## Type rfNews() to see new features/changes/bug fixes.
```

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

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

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

```
rf <- randomForest(log(SalePrice)~., data = data.cl, importance = TRUE)# use default mtry values
varImpPlot(rf)
```




```r
mean(rf$mse)
```

```
## [1] 0.0202643
```

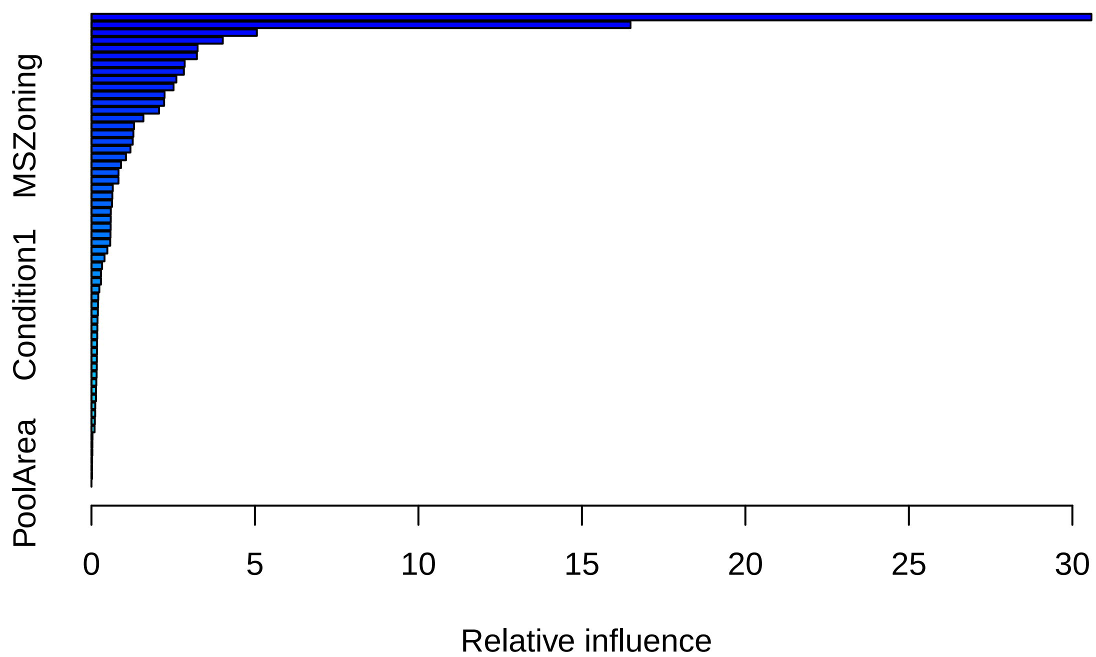
MSE appears to be based on out of bag predictions, so I use that in lieu of K-fold cross validation to get an MSE of 0.0202643, which is competitive with — if not better than — dimension reduction or regularization approaches.

Boosting

```r
library(gbm)
```

```
## Warning: package 'gbm' was built under R version 4.4.3

## Loaded gbm 2.2.2

## This version of gbm is no longer under development. Consider transitioning to gbm3, https://github.co
```

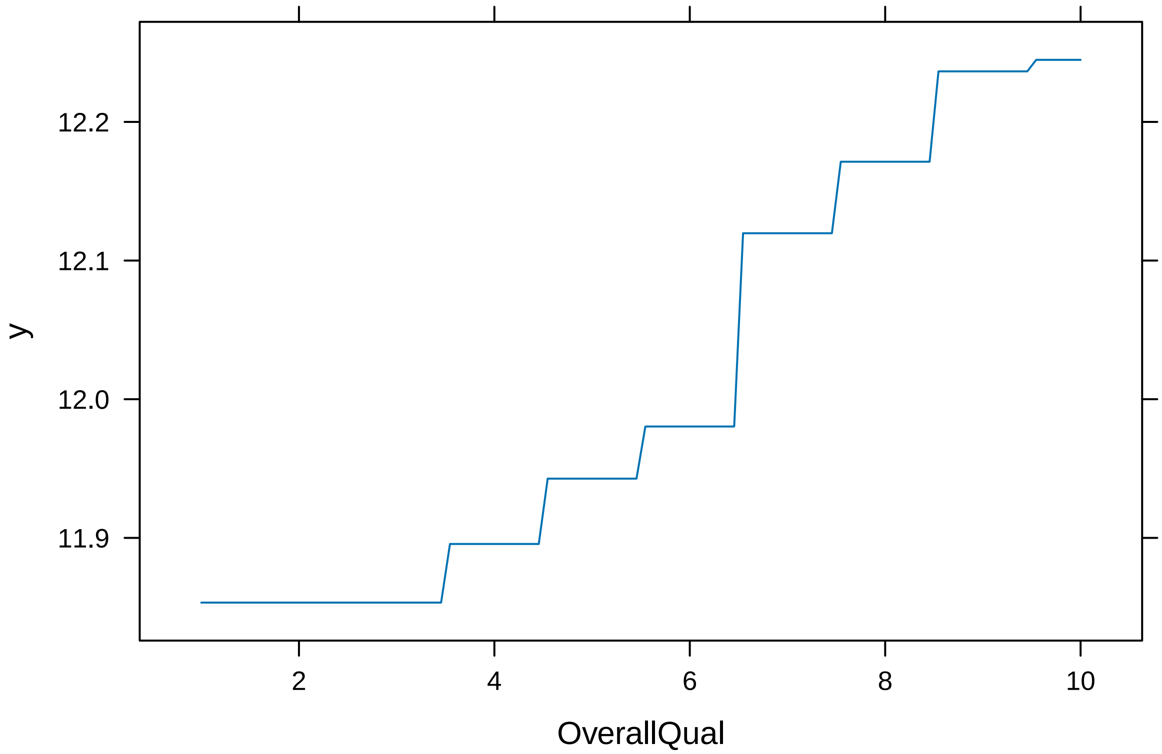
```r
boost <- gbm(log(SalePrice)~., data = data.cl, distribution = "gaussian", n.trees = 5000, interaction.d
summary(boost)
```



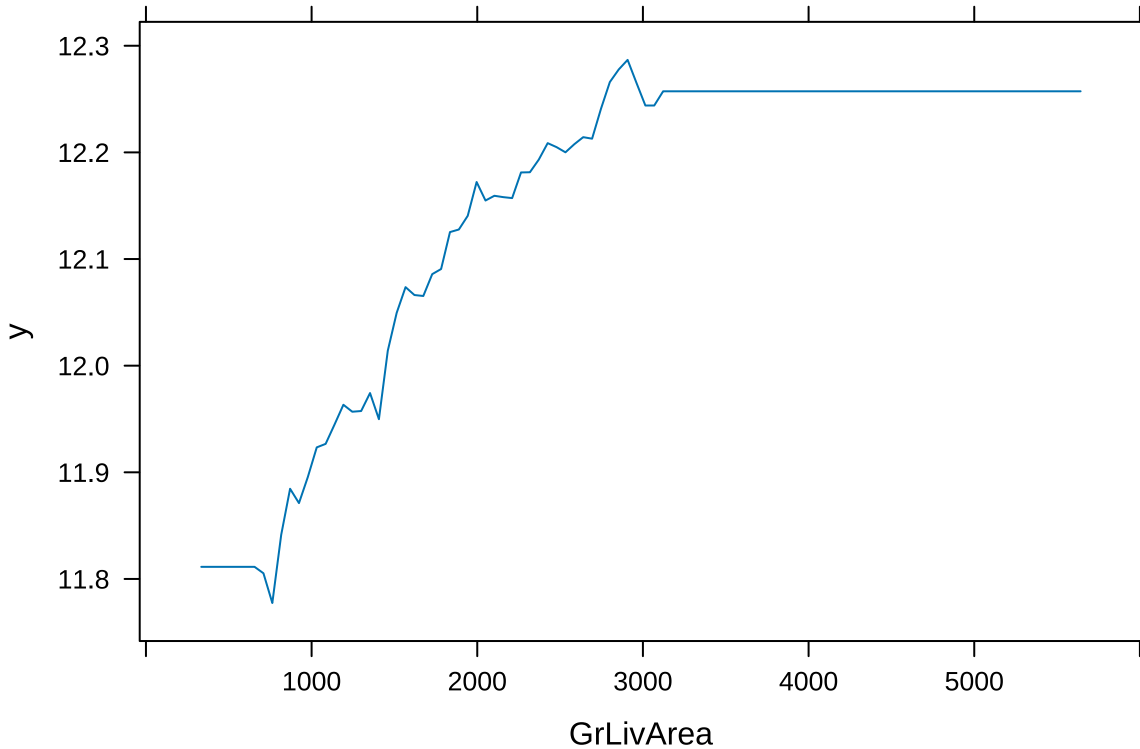

```
##                         var     rel.inf
## OverallQual     OverallQual 30.58054394
## GrLivArea         GrLivArea 16.48373112
## TotalBsmtSF     TotalBsmtSF  5.05857021
## ExterQual         ExterQual  4.01595412
## YearBuilt         YearBuilt  3.24465088
## BsmtFinSF1       BsmtFinSF1  3.22463525
## LotArea             LotArea  2.85055612
## GarageArea       GarageArea  2.82667272
## YearRemodAdd   YearRemodAdd  2.59593406
## GarageCars       GarageCars  2.50975557
## X1stFlrSF         X1stFlrSF  2.23434260
## OverallCond     OverallCond  2.22276351
## GarageType       GarageType  2.06675337
## GarageYrBlt     GarageYrBlt  1.58818729
## MSZoning           MSZoning  1.30352557
## OpenPorchSF     OpenPorchSF  1.28138073
## Neighborhood   Neighborhood  1.26183722
## BsmtUnfSF         BsmtUnfSF  1.19130236
## Exterior2nd     Exterior2nd  1.05505577
## FireplaceQu     FireplaceQu  0.90346082
## SaleCondition SaleCondition  0.82623593
## LotFrontage     LotFrontage  0.82550990
## HalfBath           HalfBath  0.65144558
## Exterior1st     Exterior1st  0.63747089
## Fireplaces       Fireplaces  0.62930101
```

```
## MasVnrArea       MasVnrArea  0.59219925
## MoSold               MoSold  0.59058497
## WoodDeckSF       WoodDeckSF  0.58690313
## X2ndFlrSF         X2ndFlrSF  0.58088997
## FullBath           FullBath  0.57361990
## EnclosedPorch EnclosedPorch  0.48516168
## BsmtExposure   BsmtExposure  0.39880603
## YrSold               YrSold  0.32513615
## BedroomAbvGr   BedroomAbvGr  0.29128383
## ExterCond         ExterCond  0.29065257
## BsmtFullBath   BsmtFullBath  0.24072698
## ScreenPorch     ScreenPorch  0.20787869
## Condition1       Condition1  0.20252907
## TotRmsAbvGrd   TotRmsAbvGrd  0.19677029
## PavedDrive       PavedDrive  0.18435270
## Foundation       Foundation  0.18052327
## SaleType           SaleType  0.17686506
## BldgType           BldgType  0.17214253
## KitchenAbvGr   KitchenAbvGr  0.17081297
## Fence                 Fence  0.16528499
## Electrical       Electrical  0.16270465
## HouseStyle       HouseStyle  0.15778849
## BsmtFinSF2       BsmtFinSF2  0.14847677
## RoofStyle         RoofStyle  0.13922323
## MSSubClass       MSSubClass  0.13855688
## MasVnrType       MasVnrType  0.11599559
## HeatingQC         HeatingQC  0.11450120
## BsmtQual           BsmtQual  0.10326046
## LotShape           LotShape  0.09548255
## BsmtHalfBath   BsmtHalfBath  0.03123435
## LotConfig         LotConfig  0.02761891
## BsmtCond           BsmtCond  0.02711442
## X3SsnPorch       X3SsnPorch  0.01993342
## LowQualFinSF   LowQualFinSF  0.01781470
## MiscVal             MiscVal  0.01759379
## PoolArea           PoolArea  0.00000000
```

```
plot(boost, i = "OverallQual")
```




```
plot(boost, i = "GrLivArea")
```




```
mean(boost$cv.error)
```

```
## [1] 0.01833493
```

With 5-fold cross-validation, the MSE was 0.01833493. This is a dramatic improvement from the single tree.

Kaggle Submission - Boosting

```
outboost <- test[1]
#exponentiate predictions
outboost$SalePrice <- exp(predict(boost ,newdata = test.cl, n.trees = 5000))
write.csv(outboost, "outboost.csv", row.names=FALSE)
```

This submission had an RMSE of 0.13944 (so an MSE of 0.01944351)

Support Vector Machines

```
library(e1071)
tune.svm.rad <- tune(svm, log(SalePrice)~., data = data.cl, kernel = "radial", ranges = list(cost = c(0
summary(tune.svm.rad)
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
```

```
##  cost
##     1
##
## - best performance: 0.01722402
##
## - Detailed performance results:
##    cost      error  dispersion
## 1 1e-04 0.15877219 0.023463245
## 2 1e-03 0.14365140 0.022504528
## 3 1e-02 0.06319821 0.015705128
## 4 1e-01 0.02429484 0.008488040
## 5 1e+00 0.01722402 0.006033719
## 6 5e+00 0.01751792 0.005964808
## 7 1e+01 0.01850665 0.006092545
```

```r
svm.radial <- svm(log(SalePrice)~., data = data.cl, kernel = "radial", scale = TRUE, cost = 1, gamma = 
summary(svm.radial)
```

```
##
## Call:
## svm(formula = log(SalePrice) ~ ., data = data.cl, kernel = "radial",
##     cost = 1, gamma = 1, scale = TRUE)
##
##
## Parameters:
##    SVM-Type:  eps-regression
##  SVM-Kernel:  radial
##        cost:  1
##       gamma:  1
##     epsilon:  0.1
##
##
## Number of Support Vectors:  1353
```

```r
tune.svm.lin <- tune(svm, log(SalePrice)~., data = data.cl, kernel = "linear", ranges = list(cost = c(0
summary(tune.svm.lin)
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  cost
##     1
##
## - best performance: 0.02182551
##
## - Detailed performance results:
##    cost      error  dispersion
## 1 1e-04 0.04728422 0.008362997
## 2 1e-03 0.02320519 0.016662837
## 3 1e-02 0.02204117 0.019224971
## 4 1e-01 0.02201796 0.018435102
## 5 1e+00 0.02182551 0.018432725
```

```
## 6 5e+00 0.02197608 0.018676907
## 7 1e+01 0.02199580 0.018711991
```

```
svm.linear <- svm(log(SalePrice)~., data = data.cl, kernel = "linear", cost = 1)
summary(svm.linear)
```

```
##
## Call:
## svm(formula = log(SalePrice) ~ ., data = data.cl, kernel = "linear",
##     cost = 1)
##
##
## Parameters:
##    SVM-Type:  eps-regression
##  SVM-Kernel:  linear
##        cost:  1
##       gamma:  0.009433962
##     epsilon:  0.1
##
##
## Number of Support Vectors:  971
```

10-fold cross validation indicated that using a cost of 1 was ideal for a Support Vector Regression with a radial kernel, having an MSE of 0.01722402. Similarly, a cost of 1 was ideal for a SVR with a linear kernel, corresponding with an MSE of 0.02182551.

Kaggle Submission - SVM (radial)

```
outsvm.rad <- test[1]
#exponentiate predictions
outsvm.rad$SalePrice <- exp(predict(svm.radial ,newdata = test.cl))
write.csv(outsvm.rad, "outsvmrad.csv", row.names=FALSE)
```

This performed MUCH worse than other submissions, with an RMSE of 0.41565 (so an MSE of 0.1727649). Perhaps the data was overfitted.

kNN

```
library(class)
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.4.3

## Loading required package: lattice

##
## Attaching package: 'lattice'

## The following object is masked from 'package:boot':
##
##     melanoma

##
## Attaching package: 'caret'

## The following object is masked from 'package:pls':
##
##     R2
```

```r
library(fastDummies)
```

```
## Warning: package 'fastDummies' was built under R version 4.4.3
```

Use k-fold KNN to identify the best model:

```
## k-Nearest Neighbors
##
## 1460 samples
##  132 predictor
##
## Pre-processing: centered (132), scaled (132)
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 1314, 1313, 1314, 1314, 1316, 1314, ...
## Resampling results across tuning parameters:
##
##   k   RMSE       Rsquared   MAE
##    1  0.2266603  0.6939428  0.1627084
##    2  0.2014251  0.7496769  0.1442350
##    3  0.1958080  0.7638003  0.1397588
##    4  0.1937021  0.7690636  0.1372853
##    5  0.1912934  0.7757728  0.1354705
##    6  0.1877445  0.7861866  0.1323649
##    7  0.1857135  0.7933602  0.1308519
##    8  0.1833490  0.8007593  0.1299392
##    9  0.1827747  0.8035640  0.1293270
##   10  0.1828497  0.8052067  0.1294865
##   11  0.1823653  0.8076992  0.1289399
##   12  0.1826037  0.8082850  0.1290635
##   13  0.1827501  0.8092271  0.1291270
##   14  0.1832968  0.8089408  0.1293299
##   15  0.1840594  0.8081424  0.1298129
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was k = 11.
```

The optimal value for K was 11 (when testing a range from 1 to 15), with a 10-fold CV RMSE of 0.1823653, notably much higher than other types of models.