Final Project

Gaeun Lee

12/4/2019

Introduction to the data an dproblem

```
train <- read.csv("train.csv")
test <- read.csv("test.csv")
```

Explain the raw data

```
glimpse(train)
```

```
## Observations: 1,460
## Variables: 81
## $ Id            <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, ...
## $ MSSubClass    <int> 60, 20, 60, 70, 60, 50, 20, 60, 50, 190, 20, 60, 20, 20...
## $ MSZoning      <fct> RL, RL, RL, RL, RL, RL, RL, RL, RM, RL, RL, RL, RL, RL,...
## $ LotFrontage   <int> 65, 80, 68, 60, 84, 85, 75, NA, 51, 50, 70, 85, NA, 91,...
## $ LotArea       <int> 8450, 9600, 11250, 9550, 14260, 14115, 10084, 10382, 61...
## $ Street        <fct> Pave, Pave, Pave, Pave, Pave, Pave, Pave, Pave, Pave, P...
## $ Alley         <fct> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ LotShape      <fct> Reg, Reg, IR1, IR1, IR1, IR1, Reg, IR1, Reg, Reg, Reg, ...
## $ LandContour   <fct> Lvl, Lvl, Lvl, Lvl, Lvl, Lvl, Lvl, Lvl, Lvl, Lvl, Lvl, ...
## $ Utilities     <fct> AllPub, AllPub, AllPub, AllPub, AllPub, AllPub, AllPub,...
## $ LotConfig     <fct> Inside, FR2, Inside, Corner, FR2, Inside, Inside, Corne...
## $ LandSlope     <fct> Gtl, Gtl, Gtl, Gtl, Gtl, Gtl, Gtl, Gtl, Gtl, Gtl, Gtl, ...
## $ Neighborhood  <fct> CollgCr, Veenker, CollgCr, Crawfor, NoRidge, Mitchel, S...
## $ Condition1    <fct> Norm, Feedr, Norm, Norm, Norm, Norm, Norm, PosN, Artery...
## $ Condition2    <fct> Norm, Norm, Norm, Norm, Norm, Norm, Norm, Norm, Norm, A...
## $ BldgType      <fct> 1Fam, 1Fam, 1Fam, 1Fam, 1Fam, 1Fam, 1Fam, 1Fam, 1Fam, 2...
## $ HouseStyle    <fct> 2Story, 1Story, 2Story, 2Story, 2Story, 1.5Fin, 1Story,...
## $ OverallQual   <int> 7, 6, 7, 7, 8, 5, 8, 7, 7, 5, 5, 9, 5, 7, 6, 7, 6, 4, 5...
## $ OverallCond   <int> 5, 8, 5, 5, 5, 5, 5, 6, 5, 6, 5, 5, 6, 5, 5, 8, 7, 5, 5...
## $ YearBuilt     <int> 2003, 1976, 2001, 1915, 2000, 1993, 2004, 1973, 1931, 1...
## $ YearRemodAdd  <int> 2003, 1976, 2002, 1970, 2000, 1995, 2005, 1973, 1950, 1...
## $ RoofStyle     <fct> Gable, Gable, Gable, Gable, Gable, Gable, Gable, Gable,...
## $ RoofMatl      <fct> CompShg, CompShg, CompShg, CompShg, CompShg, CompShg, C...
## $ Exterior1st   <fct> VinylSd, MetalSd, VinylSd, Wd Sdng, VinylSd, VinylSd, V...
## $ Exterior2nd   <fct> VinylSd, MetalSd, VinylSd, Wd Shng, VinylSd, VinylSd, V...
## $ MasVnrType    <fct> BrkFace, None, BrkFace, None, BrkFace, None, Stone, Sto...
## $ MasVnrArea    <int> 196, 0, 162, 0, 350, 0, 186, 240, 0, 0, 0, 286, 0, 306,...
## $ ExterQual     <fct> Gd, TA, Gd, TA, Gd, TA, Gd, TA, TA, TA, TA, Ex, TA, Gd,...
## $ ExterCond     <fct> TA, TA, TA, TA, TA, TA, TA, TA, TA, TA, TA, TA, TA, TA,...
## $ Foundation    <fct> PConc, CBlock, PConc, BrkTil, PConc, Wood, PConc, CBloc...
## $ BsmtQual      <fct> Gd, Gd, Gd, TA, Gd, Gd, Ex, Gd, TA, TA, TA, Ex, TA, Gd,...
## $ BsmtCond      <fct> TA, TA, TA, Gd, TA, TA, TA, TA, TA, TA, TA, TA, TA, TA,...
## $ BsmtExposure  <fct> No, Gd, Mn, No, Av, No, Av, Mn, No, No, No, No, No, Av,...
## $ BsmtFinType1  <fct> GLQ, ALQ, GLQ, ALQ, GLQ, GLQ, GLQ, ALQ, Unf, GLQ, Rec, ...
## $ BsmtFinSF1    <int> 706, 978, 486, 216, 655, 732, 1369, 859, 0, 851, 906, 9...
## $ BsmtFinType2  <fct> Unf, Unf, Unf, Unf, Unf, Unf, Unf, BLQ, Unf, Unf, Unf, ...
## $ BsmtFinSF2    <int> 0, 0, 0, 0, 0, 0, 0, 32, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
## $ BsmtUnfSF     <int> 150, 284, 434, 540, 490, 64, 317, 216, 952, 140, 134, 1...
## $ TotalBsmtSF   <int> 856, 1262, 920, 756, 1145, 796, 1686, 1107, 952, 991, 1...
## $ Heating       <fct> GasA, GasA, GasA, GasA, GasA, GasA, GasA, GasA, GasA, G...
## $ HeatingQC     <fct> Ex, Ex, Ex, Gd, Ex, Ex, Ex, Ex, Gd, Ex, Ex, Ex, TA, Ex,...
## $ CentralAir    <fct> Y, Y, Y, Y, Y, Y, Y, Y, Y, Y, Y, Y, Y, Y, Y, Y, Y, Y, Y...
## $ Electrical    <fct> SBrkr, SBrkr, SBrkr, SBrkr, SBrkr, SBrkr, SBrkr, SBrkr,...
## $ X1stFlrSF     <int> 856, 1262, 920, 961, 1145, 796, 1694, 1107, 1022, 1077,...
## $ X2ndFlrSF     <int> 854, 0, 866, 756, 1053, 566, 0, 983, 752, 0, 0, 1142, 0...
## $ LowQualFinSF  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ GrLivArea     <int> 1710, 1262, 1786, 1717, 2198, 1362, 1694, 2090, 1774, 1...
## $ BsmtFullBath  <int> 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1...
## $ BsmtHalfBath  <int> 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ FullBath      <int> 2, 2, 2, 1, 2, 1, 2, 2, 2, 1, 1, 3, 1, 2, 1, 1, 1, 2, 1...
## $ HalfBath      <int> 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1...
## $ BedroomAbvGr  <int> 3, 3, 3, 3, 4, 1, 3, 3, 2, 2, 3, 4, 2, 3, 2, 2, 2, 2, 3...
## $ KitchenAbvGr  <int> 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 1, 1, 1, 1, 1, 1, 1, 2, 1...
## $ KitchenQual   <fct> Gd, TA, Gd, Gd, Gd, TA, Gd, TA, TA, TA, TA, Ex, TA, Gd,...
## $ TotRmsAbvGrd  <int> 8, 6, 6, 7, 9, 5, 7, 7, 8, 5, 5, 11, 4, 7, 5, 5, 5, 6, ...
## $ Functional    <fct> Typ, Typ, Typ, Typ, Typ, Typ, Typ, Typ, Min1, Typ, Typ,...
## $ Fireplaces    <int> 0, 1, 1, 1, 1, 0, 1, 2, 2, 2, 0, 2, 0, 1, 1, 0, 1, 0, 0...
## $ FireplaceQu   <fct> NA, TA, TA, Gd, TA, NA, Gd, TA, TA, TA, NA, Gd, NA, Gd,...
## $ GarageType    <fct> Attchd, Attchd, Attchd, Detchd, Attchd, Attchd, Attchd,...
## $ GarageYrBlt   <int> 2003, 1976, 2001, 1998, 2000, 1993, 2004, 1973, 1931, 1...
## $ GarageFinish  <fct> RFn, RFn, RFn, Unf, RFn, Unf, RFn, RFn, Unf, RFn, Unf, ...
## $ GarageCars    <int> 2, 2, 2, 3, 3, 2, 2, 2, 2, 1, 1, 3, 1, 3, 1, 2, 2, 2, 2...
## $ GarageArea    <int> 548, 460, 608, 642, 836, 480, 636, 484, 468, 205, 384, ...
## $ GarageQual    <fct> TA, TA, TA, TA, TA, TA, TA, TA, Fa, Gd, TA, TA, TA, TA,...
## $ GarageCond    <fct> TA, TA, TA, TA, TA, TA, TA, TA, TA, TA, TA, TA, TA, TA,...
## $ PavedDrive    <fct> Y, Y, Y, Y, Y, Y, Y, Y, Y, Y, Y, Y, Y, Y, Y, Y, Y, Y, Y...
## $ WoodDeckSF    <int> 0, 298, 0, 0, 192, 40, 255, 235, 90, 0, 0, 147, 140, 16...
## $ OpenPorchSF   <int> 61, 0, 42, 35, 84, 30, 57, 204, 0, 4, 0, 21, 0, 33, 213...
## $ EnclosedPorch <int> 0, 0, 0, 272, 0, 0, 0, 228, 205, 0, 0, 0, 0, 0, 176, 0,...
## $ X3SsnPorch    <int> 0, 0, 0, 0, 0, 320, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ScreenPorch   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 176, 0, 0, 0, 0, 0,...
## $ PoolArea      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ PoolQC        <fct> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ Fence         <fct> NA, NA, NA, NA, NA, MnPrv, NA, NA, NA, NA, NA, NA, NA, ...
## $ MiscFeature   <fct> NA, NA, NA, NA, NA, Shed, NA, Shed, NA, NA, NA, NA, NA,...
## $ MiscVal       <int> 0, 0, 0, 0, 0, 700, 0, 350, 0, 0, 0, 0, 0, 0, 0, 0, 700...
## $ MoSold        <int> 2, 5, 9, 2, 12, 10, 8, 11, 4, 1, 2, 7, 9, 8, 5, 7, 3, 1...
## $ YrSold        <int> 2008, 2007, 2008, 2006, 2008, 2009, 2007, 2009, 2008, 2...
## $ SaleType      <fct> WD, WD, WD, WD, WD, WD, WD, WD, WD, WD, WD, New, WD, Ne...
## $ SaleCondition <fct> Normal, Normal, Normal, Abnorml, Normal, Normal, Normal...
## $ SalePrice     <int> 208500, 181500, 223500, 140000, 250000, 143000, 307000,...
```

```r
colSums(is.na(train))
```

```
##            Id    MSSubClass      MSZoning   LotFrontage       LotArea 
##             0             0             0           259             0 
##        Street         Alley      LotShape   LandContour     Utilities
##             0          1369             0             0             0
##     LotConfig     LandSlope  Neighborhood    Condition1    Condition2
##             0             0             0             0             0
##      BldgType    HouseStyle   OverallQual   OverallCond     YearBuilt
##             0             0             0             0             0
##  YearRemodAdd     RoofStyle      RoofMatl   Exterior1st   Exterior2nd
##             0             0             0             0             0
##    MasVnrType    MasVnrArea     ExterQual     ExterCond    Foundation
##             8             8             0             0             0
##      BsmtQual      BsmtCond  BsmtExposure  BsmtFinType1    BsmtFinSF1
##            37            37            38            37             0
##  BsmtFinType2    BsmtFinSF2     BsmtUnfSF   TotalBsmtSF       Heating
##            38             0             0             0             0
##     HeatingQC    CentralAir    Electrical     X1stFlrSF     X2ndFlrSF
##             0             0             1             0             0
##  LowQualFinSF     GrLivArea  BsmtFullBath  BsmtHalfBath      FullBath
##             0             0             0             0             0
##      HalfBath  BedroomAbvGr  KitchenAbvGr   KitchenQual  TotRmsAbvGrd
##             0             0             0             0             0
##    Functional    Fireplaces   FireplaceQu    GarageType   GarageYrBlt
##             0             0           690            81            81
##  GarageFinish    GarageCars    GarageArea    GarageQual    GarageCond
##            81             0             0            81            81
##    PavedDrive    WoodDeckSF   OpenPorchSF EnclosedPorch    X3SsnPorch
##             0             0             0             0             0
##   ScreenPorch      PoolArea        PoolQC         Fence   MiscFeature
##             0             0          1453          1179          1406
##       MiscVal        MoSold        YrSold      SaleType SaleCondition
##             0             0             0             0             0
##     SalePrice
##             0
```

```
colSums(is.na(test))
```

```
##            Id    MSSubClass      MSZoning   LotFrontage       LotArea
##             0             0             4           227             0
##        Street         Alley      LotShape   LandContour     Utilities
##             0          1352             0             0             2
##     LotConfig     LandSlope  Neighborhood    Condition1    Condition2
##             0             0             0             0             0
##      BldgType    HouseStyle   OverallQual   OverallCond     YearBuilt
##             0             0             0             0             0
##  YearRemodAdd     RoofStyle      RoofMatl   Exterior1st   Exterior2nd
##             0             0             0             1             1
##    MasVnrType    MasVnrArea     ExterQual     ExterCond    Foundation
##            16            15             0             0             0
##      BsmtQual      BsmtCond  BsmtExposure  BsmtFinType1    BsmtFinSF1
##            44            45            44            42             1
##  BsmtFinType2    BsmtFinSF2     BsmtUnfSF   TotalBsmtSF       Heating
##            42             1             1             1             0
##     HeatingQC    CentralAir    Electrical     X1stFlrSF     X2ndFlrSF
##             0             0             0             0             0
##  LowQualFinSF     GrLivArea  BsmtFullBath  BsmtHalfBath      FullBath
##             0             0             2             2             0
##      HalfBath  BedroomAbvGr  KitchenAbvGr   KitchenQual  TotRmsAbvGrd
##             0             0             0             1             0
##    Functional    Fireplaces   FireplaceQu    GarageType   GarageYrBlt
##             2             0           730            76            78
##  GarageFinish    GarageCars    GarageArea    GarageQual    GarageCond
##            78             1             1            78            78
##    PavedDrive    WoodDeckSF   OpenPorchSF EnclosedPorch    X3SsnPorch
##             0             0             0             0             0
##   ScreenPorch      PoolArea        PoolQC         Fence   MiscFeature
##             0             0          1456          1169          1408
##       MiscVal        MoSold        YrSold      SaleType SaleCondition
##             0             0             0             1             0
##     SalePrice
##             0
```

Explain how you clean the data

train data

```
#MSSubClass to factor
train$MSSubClass <-as.factor(train$MSSubClass)

#Change NA to No
train$Alley <- na.to.no(thevec=train$Alley)
train$BsmtQual <- na.to.no(thevec=train$BsmtQual)
train$BsmtCond <- na.to.no(thevec=train$BsmtCond)
train$BsmtExposure <- as.factor(ifelse(is.na(train$BsmtExposure),
                                       "NoB",
                                       as.character(train$BsmtExposure)))
train$BsmtFinType1 <- na.to.no(train$BsmtFinType1)
train$BsmtFinType2 <- na.to.no(train$BsmtFinType2)
train$FireplaceQu <- na.to.no(train$FireplaceQu)
train$GarageType <- na.to.no(train$GarageType)
train$GarageFinish <- na.to.no(train$GarageFinish)
train$GarageQual <- na.to.no(train$GarageQual)
train$GarageCond <- na.to.no(train$GarageCond)
train$PoolQC <- na.to.no(train$PoolQC)
train$Fence <- na.to.no(train$Fence)
train$MiscFeature <- na.to.no(train$MiscFeature)

train <- subset(train, select = - GarageYrBlt)
train <- subset(train, select = - LotFrontage)

# Change to 1-5
train$ExterQual <- qual.to.no(train$ExterQual)
train$ExterCond <- qual.to.no(train$ExterCond)
train$BsmtQual <- qual.to.no(train$BsmtQual)
train$BsmtCond <- qual.to.no(train$BsmtCond)
train$HeatingQC <- qual.to.no(train$HeatingQC)
train$KitchenQual <- qual.to.no(train$KitchenQual)
train$FireplaceQu <- qual.to.no(train$FireplaceQu)
train$GarageQual <- qual.to.no(train$GarageQual)
train$GarageCond <- qual.to.no(train$GarageCond)
train$PoolQC <- qual.to.no(train$PoolQC)
```

```
#Remove NAs
train <- na.omit(train)

#exclude Id
train <- train[,-1]

#Exclude Utlilities because not in test
train <- subset(train,select = - Utilities)
colSums(is.na(train))
```

```
##    MSSubClass      MSZoning       LotArea        Street         Alley 
##             0             0             0             0             0 
##      LotShape   LandContour     LotConfig     LandSlope  Neighborhood 
##             0             0             0             0             0 
##    Condition1    Condition2      BldgType    HouseStyle   OverallQual 
##             0             0             0             0             0 
##   OverallCond     YearBuilt  YearRemodAdd     RoofStyle      RoofMatl 
##             0             0             0             0             0 
##   Exterior1st   Exterior2nd    MasVnrType    MasVnrArea     ExterQual 
##             0             0             0             0             0 
##     ExterCond    Foundation      BsmtQual      BsmtCond  BsmtExposure 
##             0             0             0             0             0 
##  BsmtFinType1    BsmtFinSF1  BsmtFinType2    BsmtFinSF2     BsmtUnfSF 
##             0             0             0             0             0 
##   TotalBsmtSF       Heating     HeatingQC    CentralAir    Electrical 
##             0             0             0             0             0 
##     X1stFlrSF     X2ndFlrSF  LowQualFinSF     GrLivArea  BsmtFullBath 
##             0             0             0             0             0 
##  BsmtHalfBath      FullBath      HalfBath  BedroomAbvGr  KitchenAbvGr 
##             0             0             0             0             0 
##   KitchenQual  TotRmsAbvGrd    Functional    Fireplaces   FireplaceQu 
##             0             0             0             0             0 
##    GarageType  GarageFinish    GarageCars    GarageArea    GarageQual 
##             0             0             0             0             0 
##    GarageCond    PavedDrive    WoodDeckSF   OpenPorchSF EnclosedPorch 
##             0             0             0             0             0 
##    X3SsnPorch   ScreenPorch      PoolArea        PoolQC         Fence 
##             0             0             0             0             0 
##   MiscFeature       MiscVal        MoSold        YrSold      SaleType 
##             0             0             0             0             0 
## SaleCondition     SalePrice 
##             0             0
```

test data

```
#MSSubClass to factor
test$MSSubClass <-as.factor(test$MSSubClass)

#Change NA to No
test$Alley <- na.to.no(thevec=test$Alley)
test$BsmtQual <- na.to.no(thevec=test$BsmtQual)
test$BsmtCond <- na.to.no(thevec=test$BsmtCond)
test$BsmtExposure <- as.factor(ifelse(is.na(test$BsmtExposure),
                                      "NoB",
                                      as.character(test$BsmtExposure)))
test$BsmtFinType1 <- na.to.no(test$BsmtFinType1)
test$BsmtFinType2 <- na.to.no(test$BsmtFinType2)
test$FireplaceQu <- na.to.no(test$FireplaceQu)
test$GarageType <- na.to.no(test$GarageType)
test$GarageFinish <- na.to.no(test$GarageFinish)
test$GarageQual <- na.to.no(test$GarageQual)
test$GarageCond <- na.to.no(test$GarageCond)
test$PoolQC <- na.to.no(test$PoolQC)
test$Fence <- na.to.no(test$Fence)
test$MiscFeature <- na.to.no(test$MiscFeature)

# Because GarageYrBlt is highly correlated with YearBuilt, I decided to remove GarageYrBlt vector
test <- subset(test, select = - GarageYrBlt)
# So I decided to drop LotFrontage
test <- subset(test, select = - LotFrontage)

# Change to 1-5
test$ExterQual <- qual.to.no(test$ExterQual)
test$ExterCond <- qual.to.no(test$ExterCond)
test$BsmtQual <- qual.to.no(test$BsmtQual)
test$BsmtCond <- qual.to.no(test$BsmtCond)
test$HeatingQC <- qual.to.no(test$HeatingQC)
test$KitchenQual <- qual.to.no(test$KitchenQual)
test$FireplaceQu <- qual.to.no(test$FireplaceQu)
test$GarageQual <- qual.to.no(test$GarageQual)
test$GarageCond <- qual.to.no(test$GarageCond)
test$PoolQC <- qual.to.no(test$PoolQC)

#exclude Id
test <- test[,-1]
#exclude Utilities because test data only have 1 level
test <- subset(test,select = - Utilities)

#random fill
test$MSZoning <- random.fill(test$MSZoning)
test$Exterior1st <- random.fill(test$Exterior1st)
test$Exterior2nd <- random.fill(test$Exterior2nd)
test$MasVnrType <- random.fill(test$MasVnrType)
test$MasVnrArea <- random.fill.num(test$MasVnrArea)
test$BsmtFinSF1 <- random.fill.num(test$BsmtFinSF1)
test$BsmtFinSF2 <- random.fill.num(test$BsmtFinSF2)
test$BsmtUnfSF <- random.fill.num(test$BsmtUnfSF)
test$TotalBsmtSF <- random.fill.num(test$TotalBsmtSF)
test$BsmtFullBath <- random.fill.num(test$BsmtFullBath)
test$BsmtHalfBath <- random.fill.num(test$BsmtHalfBath)
test$KitchenQual <- random.fill.num(test$KitchenQual)
test$Functional <- random.fill(test$Functional)
test$GarageCars <- random.fill.num(test$GarageCars)
test$GarageArea <- random.fill.num(test$GarageArea)
test$SaleType <- random.fill(test$SaleType)
```

```
colSums(is.na(test))
```

```
##    MSSubClass      MSZoning       LotArea        Street         Alley
##             0             0             0             0             0
##      LotShape   LandContour     LotConfig     LandSlope  Neighborhood
##             0             0             0             0             0
##    Condition1    Condition2      BldgType    HouseStyle   OverallQual
##             0             0             0             0             0
##   OverallCond     YearBuilt  YearRemodAdd     RoofStyle      RoofMatl
##             0             0             0             0             0
##   Exterior1st   Exterior2nd    MasVnrType    MasVnrArea     ExterQual
##             0             0             0             0             0
##     ExterCond    Foundation      BsmtQual      BsmtCond  BsmtExposure
##             0             0             0             0             0
##  BsmtFinType1    BsmtFinSF1  BsmtFinType2    BsmtFinSF2     BsmtUnfSF
##             0             0             0             0             0
##   TotalBsmtSF       Heating     HeatingQC    CentralAir    Electrical
##             0             0             0             0             0
##     X1stFlrSF     X2ndFlrSF  LowQualFinSF     GrLivArea  BsmtFullBath
##             0             0             0             0             0
##  BsmtHalfBath      FullBath      HalfBath  BedroomAbvGr  KitchenAbvGr
##             0             0             0             0             0
##   KitchenQual  TotRmsAbvGrd    Functional    Fireplaces   FireplaceQu
##             0             0             0             0             0
##    GarageType  GarageFinish    GarageCars    GarageArea    GarageQual
##             0             0             0             0             0
##    GarageCond    PavedDrive    WoodDeckSF   OpenPorchSF EnclosedPorch
##             0             0             0             0             0
##    X3SsnPorch   ScreenPorch      PoolArea        PoolQC         Fence
##             0             0             0             0             0
##   MiscFeature       MiscVal        MoSold        YrSold      SaleType
##             0             0             0             0             0
## SaleCondition     SalePrice
##             0             0
```

Equal levels

```
# indicator
test$data <- rep("test",length=nrow(test))
train$data <- rep("train",length=nrow(train))

# join
train.test <- rbind(test,train)

# split
test1 <- subset(train.test,subset=data=="test")
test1 <- test1[,-ncol(test1)]
train1 <- subset(train.test,subset=data=="train")
train1 <- train1[,-ncol(train1)]
```

model matrix

```r
test.X <- model.matrix(SalePrice~.,test1)[,-1]
train.X <- model.matrix(SalePrice~.,train1)[,-1]
test.y <- test1$SalePrice
train.y <- train1$SalePrice
dim(test.X)
```

```
## [1] 1459  240
```

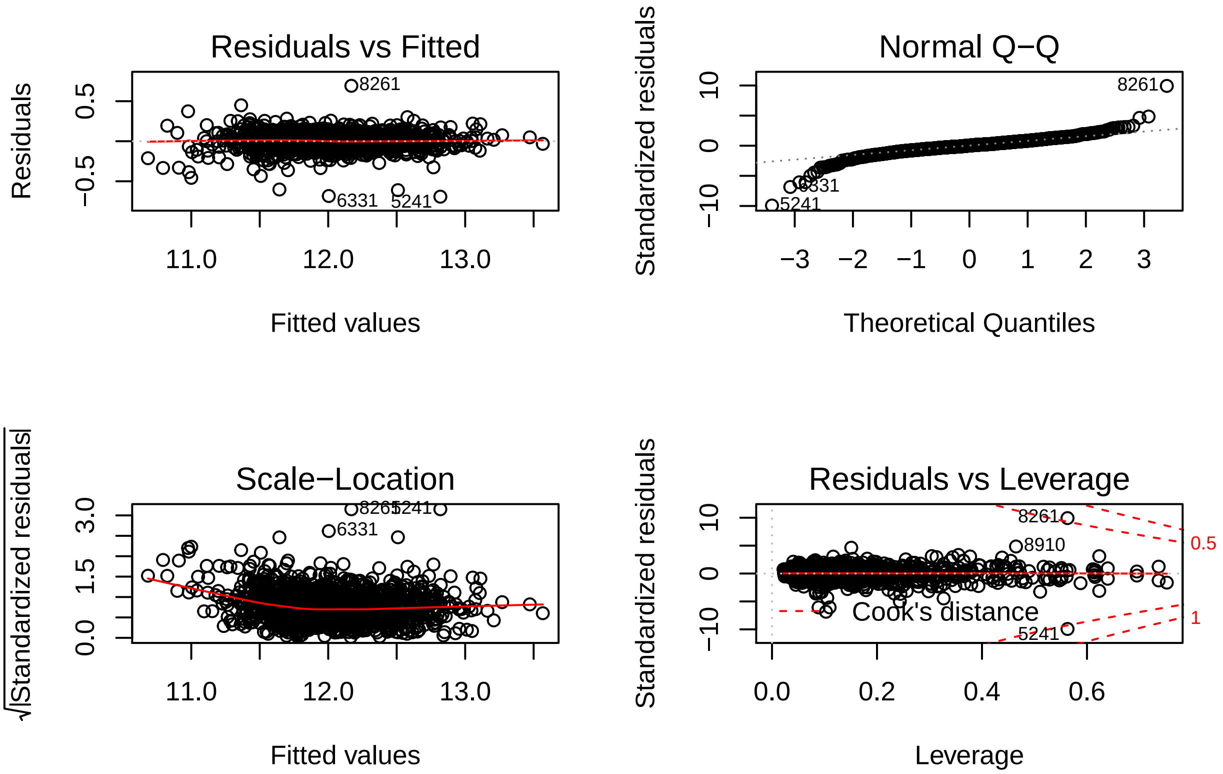
```r
dim(train.X)
```

```
## [1] 1451  240
```

Statistical learning method:

Linear regression

```r
#model
train1.lm <- subset(train1,select=-MSSubClass)
test1.lm <- subset(test1,select=-MSSubClass)
lm.pred <- lm(log(SalePrice)~.,data=train1.lm)
pred.lm <- exp(predict(lm.pred,test1.lm))
par(mfrow=c(2,2))
plot(lm.pred)
```



```r
pred.lm <- as.vector(pred.lm)

csv.function(pred.lm,name="lm.csv")
```

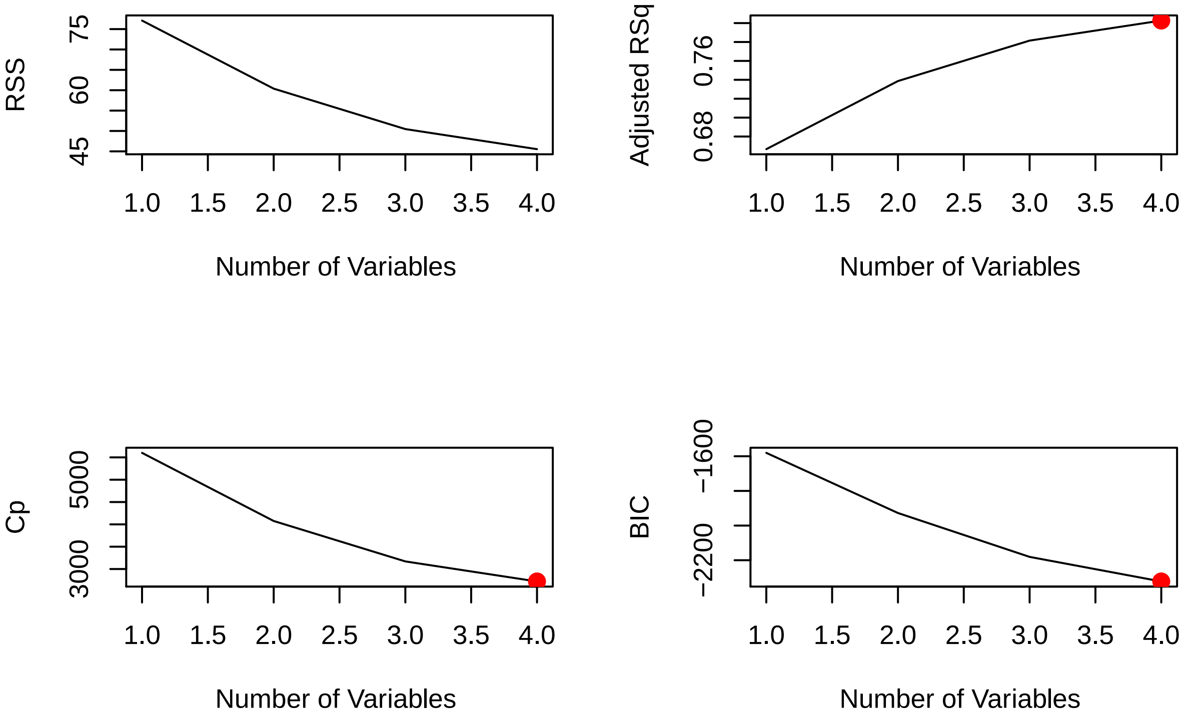
Subset selection methods

best subset

```r
best.subset <- regsubsets(log(SalePrice)~.,data=train1,nvmax=3,really.big=TRUE)
```

```
## Reordering variables and trying again:
```

```r
best.summary <- summary(best.subset)

par(mfrow=c(2,2))

# RSS plot
plot(best.summary$rss ,xlab="Number of Variables ", ylab="RSS", type="l")

# adjr2 plot
adjr2.best <- which.max(best.summary$adjr2)
plot(best.summary$adjr2 ,xlab="Number of Variables ", ylab="Adjusted RSq", type="l")
points(adjr2.best,best.summary$adjr2[adjr2.best], col = "red", cex = 2, pch = 20)

pred.best <- exp(predict.regsubsets(best.subset,test1,id=adjr2.best))
csv.function(pred.best,name="best.csv")

# Cp plot
cp.best <- which.min(best.summary$cp)
plot(best.summary$cp ,xlab="Number of Variables ", ylab="Cp", type="l")
points(cp.best,best.summary$cp[cp.best], col = "red", cex = 2, pch = 20)

# BIC plot
bic.best <- which.min(best.summary$bic)
plot(best.summary$bic ,xlab="Number of Variables ", ylab="BIC", type="l")
points(bic.best,best.summary$bic[bic.best], col = "red", cex = 2, pch = 20)
```

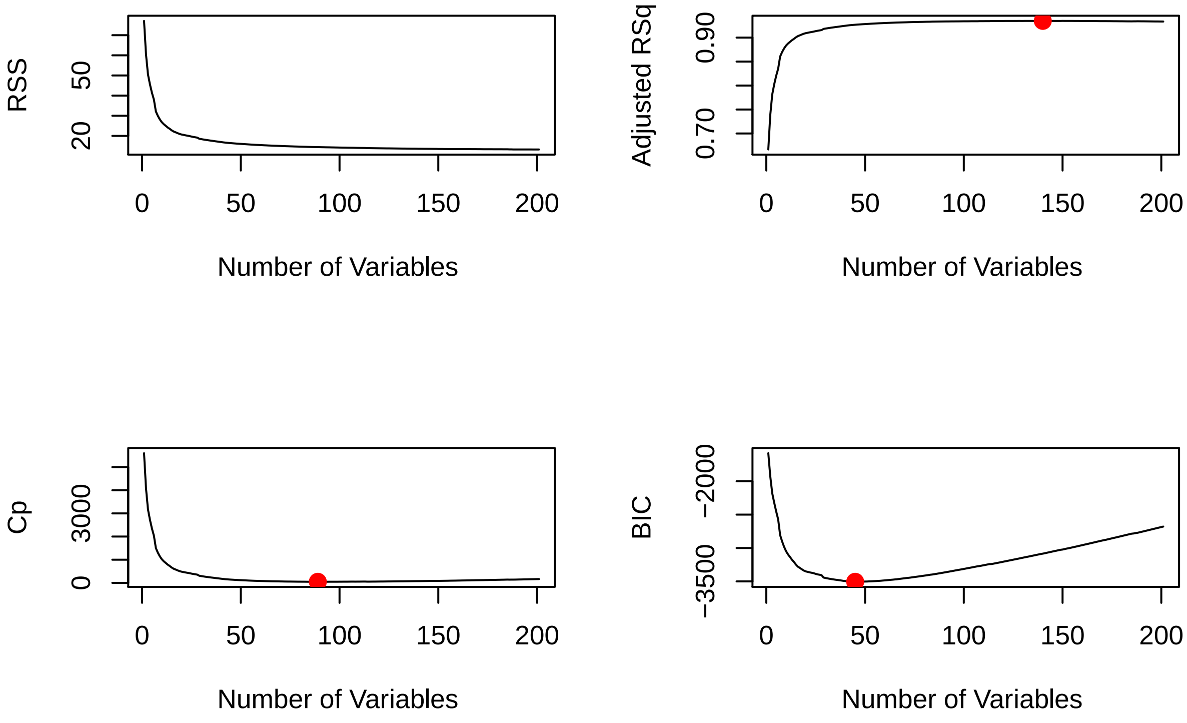


forward subset

```
forward.subset <- regsubsets(log(SalePrice)~.,data=train1,nvmax=200,method="forward",really.big=TRUE)
```

```
## Reordering variables and trying again:
```

```
for.summary <- summary(forward.subset)

par(mfrow=c(2,2))

# RSS plot
plot(for.summary$rss ,xlab="Number of Variables ", ylab="RSS", type="l")

# adjr2 plot
adjr2.for <- which.max(for.summary$adjr2)
plot(for.summary$adjr2 ,xlab="Number of Variables ", ylab="Adjusted RSq", type="l")
points(adjr2.for,for.summary$adjr2[adjr2.for], col = "red", cex = 2, pch = 20)

# Cp plot
cp.for <- which.min(for.summary$cp)
plot(for.summary$cp ,xlab="Number of Variables ", ylab="Cp", type="l")
points(cp.for,for.summary$cp[cp.for], col = "red", cex = 2, pch = 20)

# BIC plot
bic.for <- which.min(for.summary$bic)
plot(for.summary$bic ,xlab="Number of Variables ", ylab="BIC", type="l")
points(bic.for,for.summary$bic[bic.for], col = "red", cex = 2, pch = 20)
```



```
pred.forward <- exp(predict.regsubsets(forward.subset,test1,id=bic.for))
csv.function(pred.forward,name="forward.csv")

#No of variables
```

```r
c(adjr2.for,cp.for,bic.for)
```

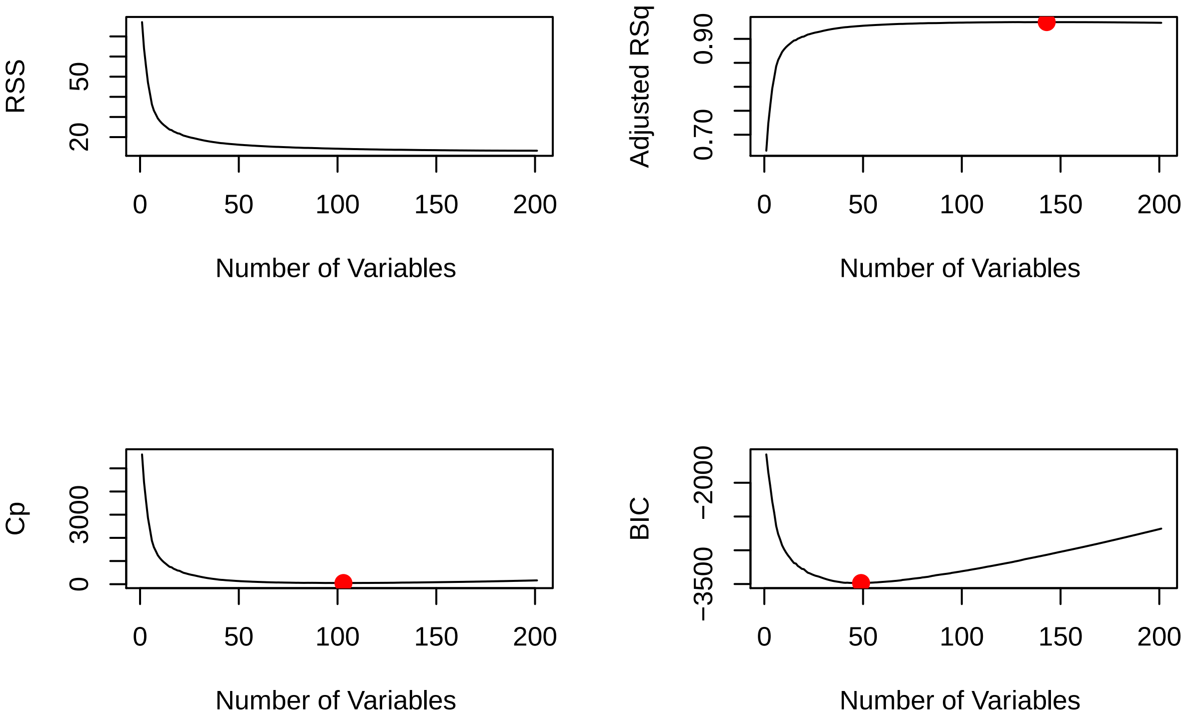
```
## [1] 140  89  45
```

backward subset

```r
back.subset <- regsubsets(log(SalePrice)~.,data=train1,nvmax=200,method="backward",really.big=TRUE)
```

```
## Reordering variables and trying again:
```

```r
back.summary <- summary(back.subset)

par(mfrow=c(2,2))

# RSS plot
plot(back.summary$rss ,xlab="Number of Variables ", ylab="RSS", type="l")

# adjr2 plot
adjr2.back <- which.max(back.summary$adjr2)
plot(back.summary$adjr2 ,xlab="Number of Variables ", ylab="Adjusted RSq", type="l")
points(adjr2.back,back.summary$adjr2[adjr2.back], col = "red", cex = 2, pch = 20)

# Cp plot
cp.back <- which.min(back.summary$cp)
plot(back.summary$cp ,xlab="Number of Variables ", ylab="Cp", type="l")
points(cp.back,back.summary$cp[cp.back], col = "red", cex = 2, pch = 20)

# BIC plot
bic.back <- which.min(back.summary$bic)
plot(back.summary$bic ,xlab="Number of Variables ", ylab="BIC", type="l")
points(bic.back,back.summary$bic[bic.back], col = "red", cex = 2, pch = 20)
```



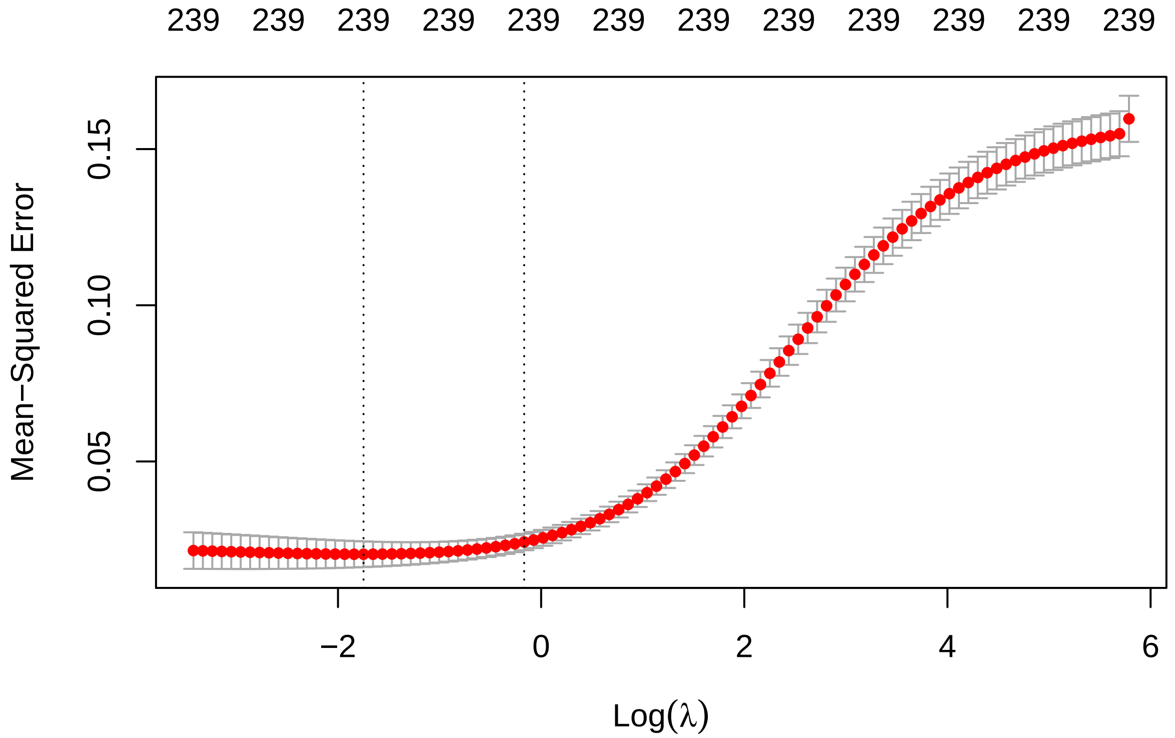
```
pred.back <- exp(predict.regsubsets(back.subset,test1,id=bic.back))
csv.function(pred.back,name="back.csv")

#No of variables
c(adjr2.back,cp.back,bic.back)
```

```
## [1] 143 103  49
```

Shrinkage methods

Ridge

```
grid <- 10^seq(10,-2,length=100)

###model
ridge.mod <- glmnet(train.X,log(train.y),alpha=0,lambda=grid)

###cross-validation
cv.ridge <- cv.glmnet(train.X,log(train.y),alpha=0,nfolds = 10)
plot(cv.ridge)
```



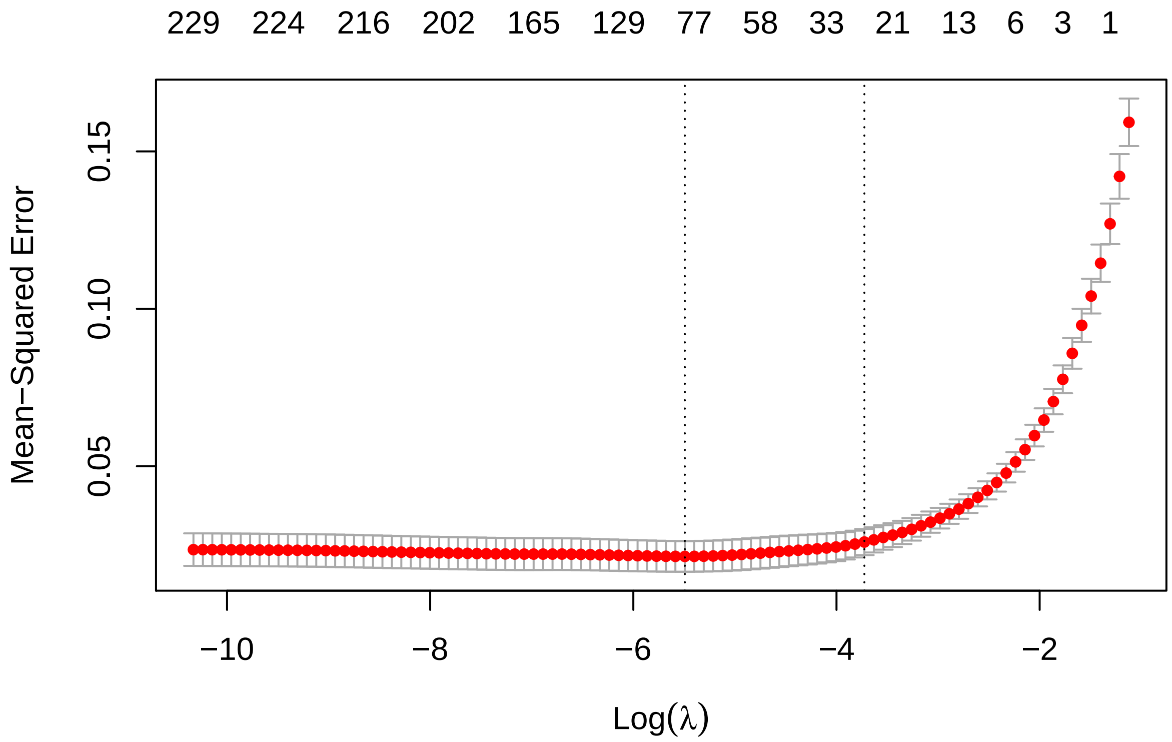
```
bestlam.ridge <- cv.ridge$lambda.min
coef.ridge <- coef(ridge.mod,s=bestlam.ridge)

### extract coef
length(colnames(train.X)[which(coef(cv.ridge, s = bestlam.ridge) != 0)])
```

```
## [1] 240
```

```
pred.ridge <- exp(predict(ridge.mod,s=bestlam.ridge,newx=test.X,type="response"))
csv.function(pred.ridge,name="ridge.csv")
```

lasso

```
###model
lasso.mod <- glmnet(train.X,log(train.y),alpha=1,lambda=grid)

###cross-validation
cv.lasso <- cv.glmnet(train.X,log(train.y),alpha=1)
plot(cv.lasso)
```



```
bestlam.lasso <- cv.lasso$lambda.min
coef.lasso <- coef(lasso.mod,s=bestlam.lasso)

### no of variables
length(colnames(train.X)[which(coef(cv.lasso, s = bestlam.lasso) != 0)])
```

```
## [1] 81
```

```
pred.lasso <- exp(predict(lasso.mod,s=bestlam.lasso,newx=test.X,type="response"))
csv.function(pred.lasso,name="lasso.csv")
```

Generalized additive model

Regression tree

Tree model

```
library(tree)
```

```
## Registered S3 method overwritten by 'tree':
##   method     from
##   print.tree cli
```

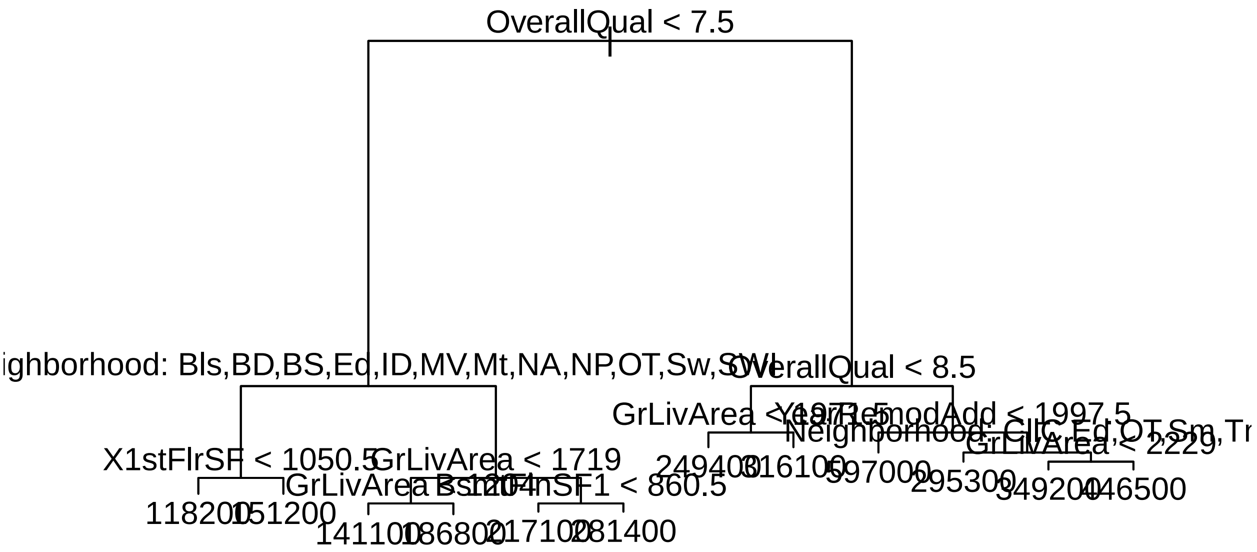
```
tree.fit <- tree(SalePrice~.,data=train1)
summary(tree.fit)
```

```
##
## Regression tree:
## tree(formula = SalePrice ~ ., data = train1)
## Variables actually used in tree construction:
## [1] "OverallQual"  "Neighborhood" "X1stFlrSF"    "GrLivArea"    "BsmtFinSF1"
## [6] "YearRemodAdd"
## Number of terminal nodes:  12
## Residual mean deviance:  1.378e+09 = 1.982e+12 / 1439
## Distribution of residuals:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -212000  -19390   -1289       0   17460  221900
```

```r
tree.fit
```

```
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 1451 9.121e+12 180600
##    2) OverallQual < 7.5 1224 2.977e+12 157600
##      4) Neighborhood: Blueste,BrDale,BrkSide,Edwards,IDOTRR,MeadowV,Mitchel,NAmes,NPkVill,OldTown,Sa
##        8) X1stFlrSF < 1050.5 410 3.254e+11 118200 *
##        9) X1stFlrSF > 1050.5 303 3.588e+11 151200 *
##      5) Neighborhood: Blmngtn,ClearCr,CollgCr,Crawfor,Gilbert,NoRidge,NridgHt,NWAmes,SawyerW,Somerst
##       10) GrLivArea < 1719 345 3.754e+11 176100
##         20) GrLivArea < 1204 81 3.704e+10 141100 *
##         21) GrLivArea > 1204 264 2.092e+11 186800 *
##       11) GrLivArea > 1719 166 3.204e+11 228300
##         22) BsmtFinSF1 < 860.5 137 1.683e+11 217100 *
##         23) BsmtFinSF1 > 860.5 29 5.306e+10 281400 *
##    3) OverallQual > 7.5 227 2.006e+12 304600
##      6) OverallQual < 8.5 167 6.804e+11 275000
##       12) GrLivArea < 1971.5 103 2.402e+11 249400 *
##       13) GrLivArea > 1971.5 64 2.645e+11 316100 *
##      7) OverallQual > 8.5 60 7.696e+11 387200
##       14) YearRemodAdd < 1997.5 5 1.075e+11 597000 *
##       15) YearRemodAdd > 1997.5 55 4.221e+11 368100
##         30) Neighborhood: CollgCr,Edwards,OldTown,Somerst,Timber 15 5.197e+10 295300 *
##         31) Neighborhood: Gilbert,NoRidge,NridgHt,StoneBr 40 2.607e+11 395400
##           62) GrLivArea < 2229 21 3.305e+10 349200 *
##           63) GrLivArea > 2229 19 1.332e+11 446500 *
```

```r
par(mfrow=c(1,1))
plot(tree.fit)
text(tree.fit, pretty = 2)
```



```r
pred.tree <- predict(tree.fit,newdata=test1)
csv.function(pred.tree,name="tree.csv")
```

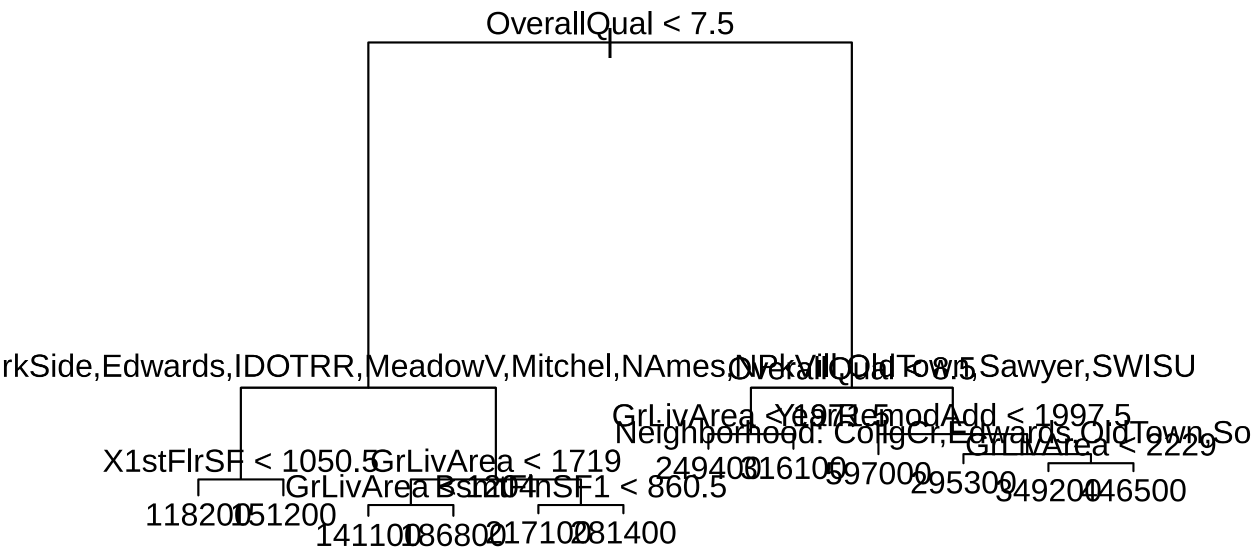
cross-validate tree

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

best.size <- tree.cv$size[which.min(tree.cv$dev)]
best.size
```

```
## [1] 12
```

prune tree

```r
pruned <- prune.tree(tree.fit,
                     best=best.size)
par(mfrow=c(1,1))
plot(pruned)
text(pruned,pretty=0)
```



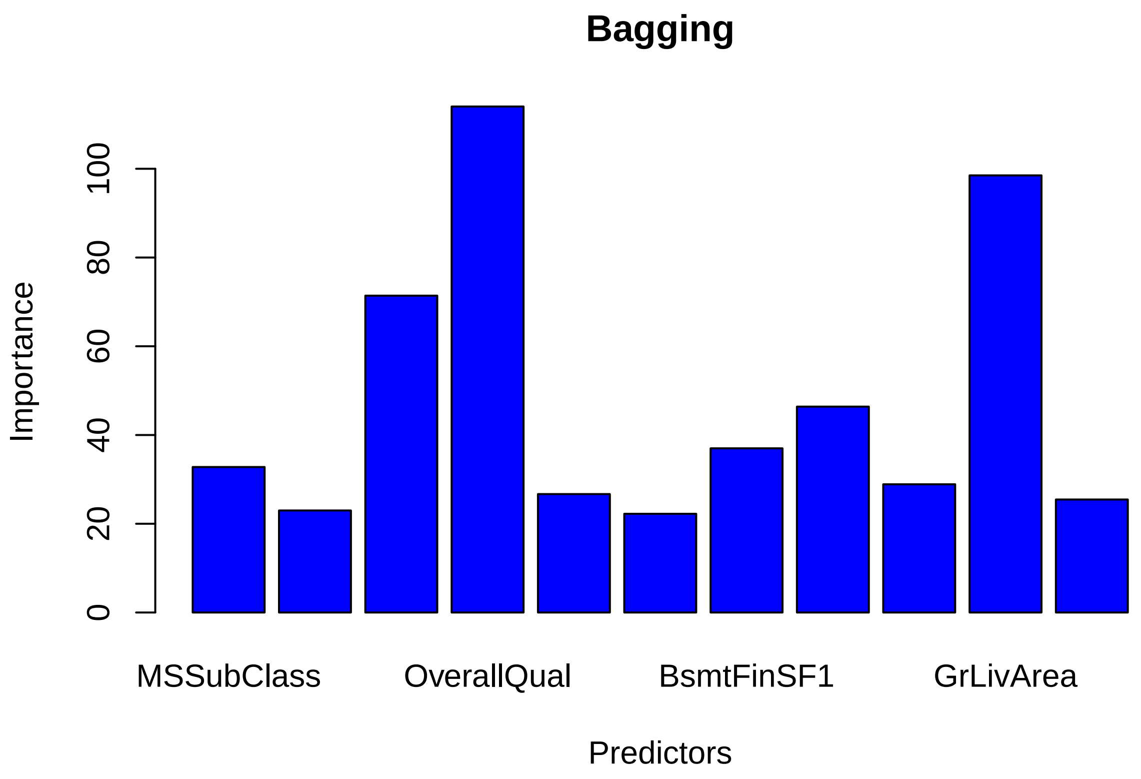
```r
pred.pruned <- predict(pruned,test1)
csv.function(pred.pruned,name="pruned.csv")
```

Bagging

```r
library(randomForest)
```

```
## randomForest 4.6-14

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

##
## Attaching package: 'randomForest'

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

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

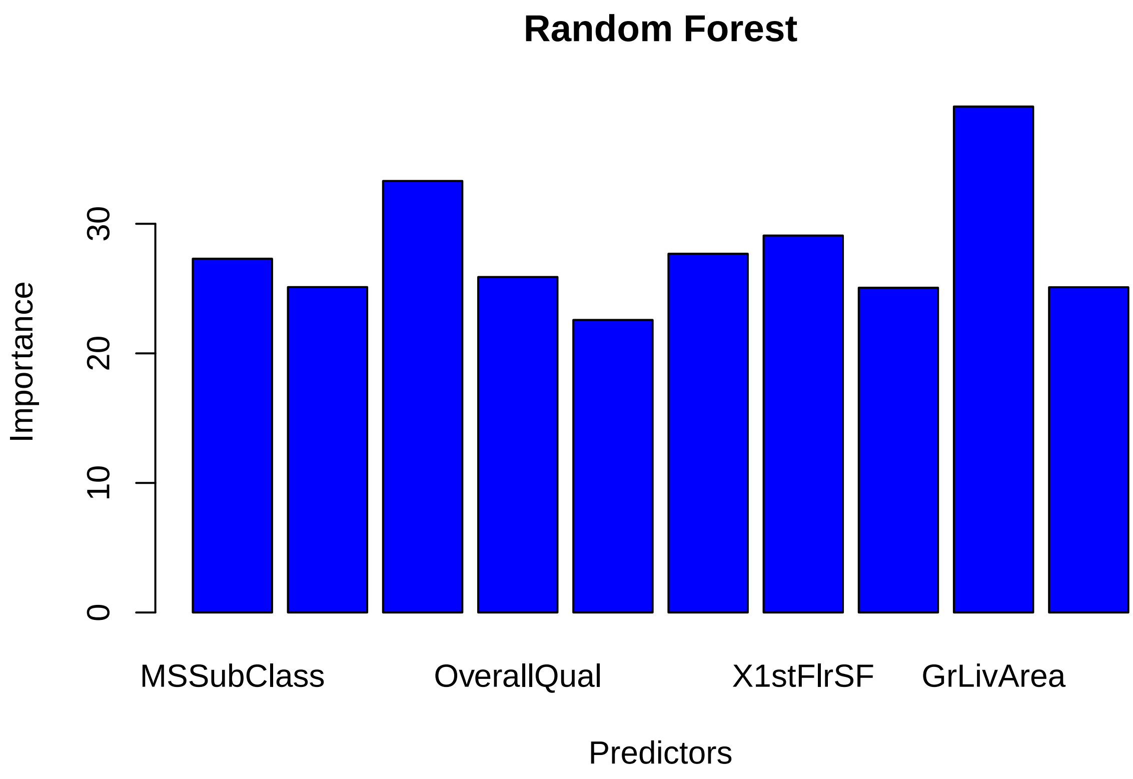
```r
bag <- randomForest(log(SalePrice)~.,
                           data=train1,
                           mtry=76,
                           importance=TRUE,
                           ntree=1000)

important <- importance(bag)[importance(bag)[,1]>22,1]

barplot(importance(bag)[names(important),1], col="blue",
        xlab="Predictors",ylab="Importance",main="Bagging")
```



```
pred.bag <- exp(predict(bag,newdata=test1))
csv.function(pred.bag,name="bag.csv")
```

Random Forest

```
forest <- randomForest(log(SalePrice)~.,
                          data=train1,
                          mtry=9,
                          importance=TRUE,
                          ntree=1000)

important <- importance(forest)[importance(forest)[,1]>22,1]

barplot(importance(forest)[names(important),1], col="blue",
        xlab="Predictors",ylab="Importance",main="Random Forest")
```



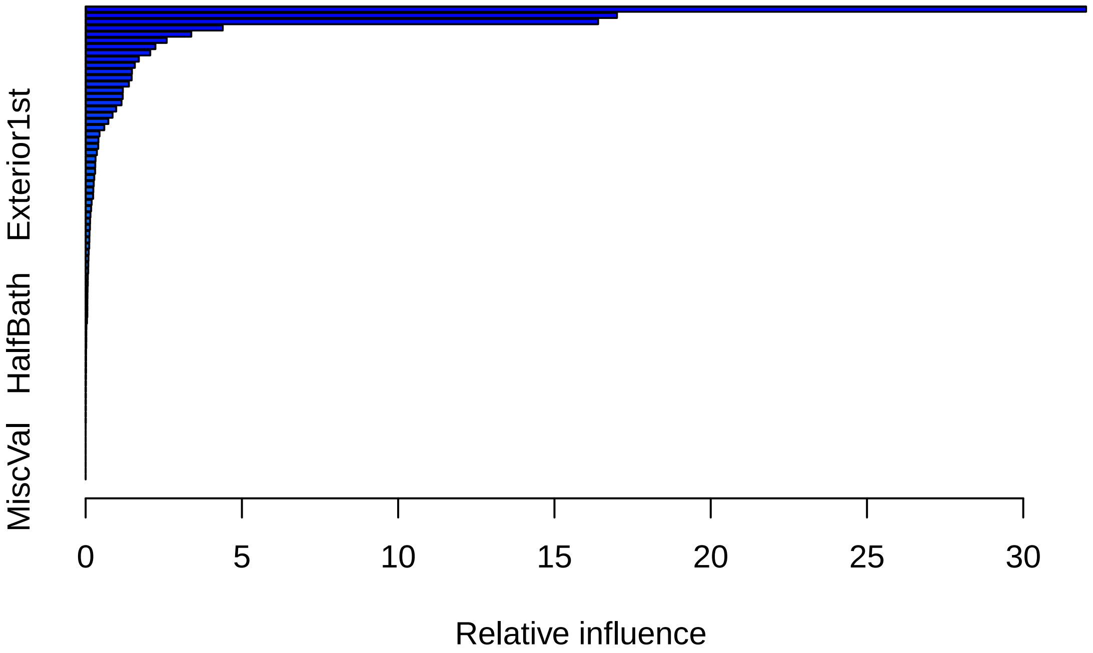
```
pred.forest <- exp(predict(forest,test1))
csv.function(pred.forest,name="forest.csv")
```

Boosting

```
library(gbm)
```

```
## Loaded gbm 2.1.5
```

```
gbm.cv <- gbm(log(SalePrice)~., data=train1,
                     distribution = "gaussian",
                     shrinkage = 0.01,
                     n.tree=1000,
                     interaction.depth = 4,
                     cv.folds=10)


summary(gbm.cv)
```



```
##                         var      rel.inf
## OverallQual     OverallQual 32.009199937
## GrLivArea         GrLivArea 16.999334715
## Neighborhood   Neighborhood 16.396180073
## TotalBsmtSF     TotalBsmtSF  4.385676475
## KitchenQual     KitchenQual  3.378557691
## GarageCars       GarageCars  2.592446846
## X1stFlrSF         X1stFlrSF  2.231770094
## BsmtFinSF1       BsmtFinSF1  2.066172315
## MSSubClass       MSSubClass  1.701162173
## OverallCond     OverallCond  1.573581905
## GarageArea       GarageArea  1.480094933
## ExterQual         ExterQual  1.471910710
## CentralAir       CentralAir  1.382504232
## GarageFinish   GarageFinish  1.189513944
## YearRemodAdd   YearRemodAdd  1.185879184
## LotArea             LotArea  1.146659070
## FireplaceQu     FireplaceQu  0.976366241
## GarageType       GarageType  0.859427181
## SaleCondition SaleCondition  0.725589572
## BsmtQual           BsmtQual  0.594718335
## BsmtFinType1   BsmtFinType1  0.443360555
## YearBuilt         YearBuilt  0.408301717
## X2ndFlrSF         X2ndFlrSF  0.401672099
## FullBath           FullBath  0.358519294
## MSZoning           MSZoning  0.315333258
## Exterior1st     Exterior1st  0.306821318
## OpenPorchSF     OpenPorchSF  0.302640391
## Functional       Functional  0.274340739
## BsmtExposure   BsmtExposure  0.257945695
## GarageCond       GarageCond  0.244260793
## GarageQual       GarageQual  0.239209337
## Condition1       Condition1  0.191410854
## Exterior2nd     Exterior2nd  0.174655234
## ScreenPorch     ScreenPorch  0.152530188
## WoodDeckSF       WoodDeckSF  0.141404509
## ExterCond         ExterCond  0.136251291
## PavedDrive       PavedDrive  0.122468953
## HeatingQC         HeatingQC  0.118627491
## TotRmsAbvGrd   TotRmsAbvGrd  0.113270532
## BsmtFullBath   BsmtFullBath  0.097429076
## LandContour     LandContour  0.088942619
## SaleType           SaleType  0.083363959
## BsmtCond           BsmtCond  0.080785196
## LotConfig         LotConfig  0.068492133
## Fireplaces       Fireplaces  0.067482629
## YrSold               YrSold  0.060086942
## MasVnrArea       MasVnrArea  0.056018117
## BsmtUnfSF         BsmtUnfSF  0.053704954
## MoSold               MoSold  0.053380982
## RoofMatl           RoofMatl  0.052928285
## Fence                 Fence  0.042096422
## EnclosedPorch EnclosedPorch  0.022206791
## HalfBath           HalfBath  0.019242177
## BsmtFinType2   BsmtFinType2  0.019225603
## LotShape           LotShape  0.017930487
## Electrical       Electrical  0.011434656
## HouseStyle       HouseStyle  0.011244026
## BedroomAbvGr   BedroomAbvGr  0.010373215
## Foundation       Foundation  0.008322379
## RoofStyle         RoofStyle  0.005722109
## BsmtFinSF2       BsmtFinSF2  0.004991812
## LowQualFinSF   LowQualFinSF  0.003605689
## Condition2       Condition2  0.003575140
## Alley                 Alley  0.002217883
## MasVnrType       MasVnrType  0.002211507
## LandSlope         LandSlope  0.001692332
## MiscFeature     MiscFeature  0.001523010
## Street               Street  0.000000000
## BldgType           BldgType  0.000000000
## Heating             Heating  0.000000000
## BsmtHalfBath   BsmtHalfBath  0.000000000
## KitchenAbvGr   KitchenAbvGr  0.000000000
## X3SsnPorch       X3SsnPorch  0.000000000
## PoolArea           PoolArea  0.000000000
## PoolQC               PoolQC  0.000000000
## MiscVal             MiscVal  0.000000000
```

```r
which.min(gbm.cv$cv.error)
```

```
## [1] 997
```

```r
pred.boost <- exp(predict(gbm.cv,test1,n.trees = which.min(gbm.cv$cv.error)))
csv.function(pred.boost,name="boost.csv")
```

KNN

```r
fold.index <- cut(sample(1:nrow(train.X)),
                  breaks=10, labels=FALSE)

K <- c(1,5,10,15,20,25,30)
mse.df <- rep(NA,length=7)
mse.k <- rep(NA,length=10)
n <- 1
for (k in K){
  for (i in 1:10){
    cvknn <- knn.reg(train.X[fold.index!=i,],
                     train.X[fold.index==i,],
                     train.y[fold.index!=i],
                     k=k)
    pred <- cvknn$pred
    mse <- mean((pred-train.y[fold.index==i])^2)
    mse.k[i] <- mse
  }
  mse.df[n] <- mean(mse.k)
  n <- n+1
}
mse.df <- data.frame(mse.df)
row.names(mse.df) <- c(1,5,10,15,20,25,30)
which.min(mse.df$mse.df) # K=10 is the best
```

```
## [1] 3
```

```r
knn.fit <- knn.reg(train.X,
                  test.X,
                  train.y,k=10)

pred.knn <- knn.fit$pred


csv.function(pred.knn,"knn.csv")
```

TRUE TEST ERROR : 0.24294

Estimated Test Error

KNN

```r
fold.index <- cut(sample(1:nrow(train.X)),
                  breaks=10, labels=FALSE)

K <- c(1,5,10,15,20,25,30)
mse.df <- rep(NA,length=7)
mse.k <- rep(NA,length=10)
n <- 1
for (k in K){
  for (i in 1:10){
    cvknn <- knn.reg(train.X[fold.index!=i,],
                     train.X[fold.index==i,],
                     log(train.y[fold.index!=i]),
                     k=k)
    pred <- cvknn$pred
    mse <- mean((pred-log(train.y[fold.index==i]))^2)
    mse.k[i] <- mse
  }
  mse.df[n] <- mean(mse.k)
  n <- n+1
}
mse.df <- data.frame(mse.df)
row.names(mse.df) <- c(1,5,10,15,20,25,30)
min(mse.df$mse.df) # K=10
```

```
## [1] 0.05078415
```

The least mse for tuning paramater K cross-validation is K=10 and K=5. This agrees with true test error

Linear model

```
# train1.lm <- subset(train1,select=-c(MSSubClass,
#                                      BldgType,
#                                      Exterior2nd,
#                                      TotalBsmtSF,
#                                      GrLivArea,
#                                      GarageFinish))
#
# lm.pred <- lm(SalePrice~.,data=train1.lm[fold.index!=i,])
# pred.lm <- data.frame(predict(lm.pred,
#                               train1.lm[fold.index==i,]))
#
# error.vec <- rep(NA,length=10)
#
# for (i in 1:10){
#   glm.fit <- lm(SalePrice~.,
#                  data=train1[fold.index!=i,])
#   predict(glm.fit,train1[fold.index==i,])
# }
#glm.fit <- glm(SalePrice~.,data=train1)
#cv.error <- cv.glm(train,glm.fit,K=10)$delta[1]
```

Subset selection

Best Subset

```
# fold.index <- cut(sample(1:nrow(train1)), breaks=10, labels=FALSE)
#
# for (i in 1:adjr2.best){
#   cat("i=", i,"\n")
#   error <- rep(0,10)
#   for(k in 1:10){
#     train1.train <- train1[fold.index!=k,]
#     train1.test <- train1[fold.index==k,]
#     true.y <- train1.test[,"SalePrice"]
#     best.fit <- regsubsets(SalePrice~.,data=train1.train,
#                            nvmax=3,really.big = TRUE)
#     pred <- predict(best.fit,train1.test,id=i)
#     error[k] <- mean((pred-true.y)^2)
#   }
#   print(mean(error))
#   cv.error.best.fit[i] <- mean(error)
# }
```

It takes too long to do cross-validation

Forward Subset

```
fold.index <- cut(sample(1:nrow(train1)), breaks=10, labels=FALSE)

cv.error.best.fit <- rep(0,50)

for (i in 1:50){
  cat("i=", i,"\n")
  error <- rep(0,10)
  for(k in 1:10){
    train1.train <- train1[fold.index!=k,]
    train1.test <- train1[fold.index==k,]
    true.y <- train1.test[,"SalePrice"]
    best.fit <- regsubsets(log(SalePrice)~.,data=train1.train,
                           nvmax=50,really.big = TRUE,
                           method="forward")
    pred <- predict(best.fit,train1.test,id=i)
    error[k] <- mean((pred-log(true.y))^2)
  }
  #print(mean(error))
  cv.error.best.fit[i] <- mean(error)
}
```

```
c(which.min(cv.error.best.fit),cv.error.best.fit[which.min(cv.error.best.fit)])
```

```
## [1] 50.00000000  0.03433853
```

lowest CV estimated test error for forward is with 50 predictors (nvmax=50)

Backward Subset

```
fold.index <- cut(sample(1:nrow(train1)), breaks=10, labels=FALSE)

cv.error.best.fit <- rep(0,50)

for (i in 1:50){
  cat("i=", i,"\n")
  error <- rep(0,10)
  for(k in 1:10){
    train1.train <- train1[fold.index!=k,]
    train1.test <- train1[fold.index==k,]
    true.y <- train1.test[,"SalePrice"]
    best.fit <- regsubsets(log(SalePrice)~.,data=train1.train,
                           nvmax=50,really.big = TRUE,
                           method="backward")
    pred <- predict(best.fit,train1.test,id=i)
    error[k] <- mean((pred-log(true.y))^2)
  }
  print(mean(error))
  cv.error.best.fit[i] <- mean(error)
}
```

```
c(which.min(cv.error.best.fit),cv.error.best.fit[which.min(cv.error.best.fit)])
```

```
## [1] 50.00000000  0.03145151
```

lowest CV estimated test error for backward is with 44 predictors (nvmax=50)

Shrinkage Method

Ridge Regression

```
###model
ridge.mod <- glmnet(train.X,log(train.y),alpha=0,lambda=grid)

###cross-validation
cv.ridge <- cv.glmnet(train.X,log(train.y),alpha=0,nfolds = 10)
c(which.min(cv.ridge$cvm),cv.ridge$cvm[which.min(cv.ridge$cvm)])
```

```
## [1] 86.00000000  0.01981642
```

```
#train.y before transformation
```

lowest mse is when there are 82 predictors

Lasso Regression

```
###model
lasso.mod <- glmnet(train.X,log(train.y),alpha=1,lambda=grid)

###cross-validation
cv.lasso <- cv.glmnet(train.X,log(train.y),alpha=1)
c(which.min(cv.lasso$cvm),cv.lasso$cvm[which.min(cv.lasso$cvm)])
```

```
## [1] 50.00000000  0.02076761
```

```
# Use train.y before log transformation
```

lowest mse is when there are 41 predictors

Estimated Test Error

```
Model <- c("knn","Forward Subset(45 predictors)",
          "Backward Subset(50 predictors)","Ridge Regression",
          "Lasso Regression(49 predictors)")

est.error <- c(0.05043,0.02973,0.03348,0.02050,0.02125)

est.df <- data.frame(Model,est.error)
est.df
```

```
##                             Model est.error
## 1                             knn   0.05043
## 2   Forward Subset(45 predictors)   0.02973
## 3  Backward Subset(50 predictors)   0.03348
## 4                Ridge Regression   0.02050
## 5 Lasso Regression(49 predictors)   0.02125
```

true test error

```
Model <- c("knn","linear model","Best Subset(nvmax=3)","Forward Subset(adjr2,140 predictors)",
          "Backward Subset(adjr2,143 predictors)","Ridge Regression(lambda=0.1585827)",
          "Lasso Regression(lambda=0.004115261)")

true.error <- c(0.24094,0.13704,0.27518,0.16819,0.16689,0.13225,0.13156)

true.df <- data.frame(Model,true.error)
true.df
```

```
##                                    Model true.error
## 1                                    knn    0.24094
## 2                           linear model    0.13704
## 3                   Best Subset(nvmax=3)    0.27518
## 4  Forward Subset(adjr2,140 predictors)    0.16819
## 5 Backward Subset(adjr2,143 predictors)    0.16689
## 6    Ridge Regression(lambda=0.1585827)    0.13225
## 7  Lasso Regression(lambda=0.004115261)    0.13156
```