

Final Code

Ashton Pallottini, Carson Fick, Ben Wurst, Zongheng Zhang

December 6, 2018

```
library(leaps)
library(glmnet)

## Loading required package: Matrix
## Loading required package: foreach
## Loaded glmnet 2.0-16
library(class)
library(DMwR)

## Loading required package: lattice
## Loading required package: grid
library(pls)

##
## Attaching package: 'pls'
## The following object is masked from 'package:stats':
##
##     loadings
library(boot)

##
## Attaching package: 'boot'
## The following object is masked from 'package:lattice':
##
##     melanoma
library(plyr)

##
## Attaching package: 'plyr'
## The following object is masked from 'package:DMwR':
##
##     join
library(FNN)

##
## Attaching package: 'FNN'
## The following objects are masked from 'package:class':
##
##     knn, knn.cv
library(gam)

## Loading required package: splines
```

```
## Loaded gam 1.16
library(tree)
library(randomForest)

## randomForest 4.6-14

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

## Loaded gbm 2.1.4
```

Data Cleaning

```
#Read in test and train data, accounting for headers and NA values
train <- read.csv("train.csv", header = TRUE, na.strings = "NA")
test <- read.csv("test.csv", header = TRUE, na.strings = "NA")

#How many NAs and a look at Lot Frontage NAs
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
min(train$LotFrontage, na.rm = T)

## [1] 21
#Remove columns with large numbers of NA and utilities which has no variance in test data

#Lot Frontage NA to 0.
train$LotFrontage[is.na(train$LotFrontage)==TRUE] <- 0

#Fix columns where NA should be None
levels(train$Alley) <- c("Grv1", "Pave", "None")
train$Alley[is.na(train$Alley)==TRUE] <- "None"

levels(train$FireplaceQu) <- c("Ex", "Fa", "Gd", "Po", "TA", "None")
train$FireplaceQu[is.na(train$FireplaceQu)==TRUE] <- "None"

levels(train$BsmtQual) <- c(levels(train$BsmtQual), "None")
train$BsmtQual[is.na(train$BsmtQual)==TRUE] <- "None"

levels(train$BsmtCond) <- c(levels(train$BsmtCond), "None")
train$BsmtCond[is.na(train$BsmtCond)==TRUE] <- "None"

levels(train$BsmtExposure) <- c(levels(train$BsmtExposure), "None")
train$BsmtExposure[is.na(train$BsmtExposure)==TRUE] <- "None"

levels(train$BsmtFinType1) <- c(levels(train$BsmtFinType1), "None")
train$BsmtFinType1[is.na(train$BsmtFinType1)==TRUE] <- "None"

levels(train$BsmtFinType2) <- c(levels(train$BsmtFinType2), "None")
train$BsmtFinType2[is.na(train$BsmtFinType2)==TRUE] <- "None"

levels(train$GarageType) <- c(levels(train$GarageType), "None")
train$GarageType[is.na(train$GarageType)==TRUE] <- "None"

levels(train$GarageFinish) <- c(levels(train$GarageFinish), "None")
train$GarageFinish[is.na(train$GarageFinish)==TRUE] <- "None"

levels(train$BsmtQual) <- c(levels(train$BsmtQual), "None")
train$BsmtQual[is.na(train$BsmtQual)==TRUE] <- "None"

levels(train$GarageQual) <- c(levels(train$GarageQual), "None")
train$GarageQual[is.na(train$GarageQual)==TRUE] <- "None"

levels(train$GarageCond) <- c(levels(train$GarageCond), "None")
train$GarageCond[is.na(train$GarageCond)==TRUE] <- "None"

levels(train$PoolQC) <- c(levels(train$PoolQC), "None")
train$PoolQC[is.na(train$PoolQC)==TRUE] <- "None"

levels(train$Fence) <- c(levels(train$Fence), "None")
train$Fence[is.na(train$Fence)==TRUE] <- "None"

levels(train$MiscFeature) <- c(levels(train$MiscFeature), "None")
```

```
train$MiscFeature[is.na(train$MiscFeature)==TRUE] <- "None"
```

```
#How many NAs left
colSums(is.na(train))
```

```
##          Id    MSSubClass    MSZoning    LotFrontage    LotArea
##          0          0          0          0          0
##      Street      Alley      LotShape    LandContour    Utilities
##          0          0          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
##          0          0          0          0          0
##    BsmtFinType2    BsmtFinSF2    BsmtUnfSF    TotalBsmtSF    Heating
##          0          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          0          0          81
##    GarageFinish    GarageCars    GarageArea    GarageQual    GarageCond
##          0          0          0          0          0
##      PavedDrive    WoodDeckSF    OpenPorchSF    EnclosedPorch    X3SsnPorch
##          0          0          0          0          0
##      ScreenPorch    PoolArea    PoolQC          Fence    MiscFeature
##          0          0          0          0          0
##      MiscVal      MoSold      YrSold      SaleType    SaleCondition
##          0          0          0          0          0
##      SalePrice
##          0
```

```
#IRemove ID, Utilities, linearly dependent columns, save IDs to include in submission
test.ids <- test$Id
train <- train[,-c(1,10,39,47)]
test <- test[,-c(1,10,39,47)]
```

```
#Save colnames of df
df.names <- colnames(train)
```

```
#KNN imputation for rest of NAs. Fix column names
train <- data.frame(knnImputation(train[,1:76]),train[,77])
colnames(train) <- df.names
```

```
#Fix NAs in test data
```

```
#Lot Frontage NA to 0.
```

```

test$LotFrontage[is.na(test$LotFrontage)==TRUE] <- 0

#Fix columns where NA should be None
levels(test$Alley) <- c(levels(test$Alley), "None")
test$Alley[is.na(test$Alley)==TRUE] <- "None"

levels(test$FireplaceQu) <- c(levels(test$FireplaceQu), "None")
test$FireplaceQu[is.na(test$FireplaceQu)==TRUE] <- "None"

levels(test$BsmtQual) <- c(levels(test$BsmtQual), "None")
test$BsmtQual[is.na(test$BsmtQual)==TRUE] <- "None"

levels(test$BsmtCond) <- c(levels(test$BsmtCond), "None")
test$BsmtCond[is.na(test$BsmtCond)==TRUE] <- "None"

levels(test$BsmtExposure) <- c(levels(test$BsmtExposure), "None")
test$BsmtExposure[is.na(test$BsmtExposure)==TRUE] <- "None"

levels(test$BsmtFinType1) <- c(levels(test$BsmtFinType1), "None")
test$BsmtFinType1[is.na(test$BsmtFinType1)==TRUE] <- "None"

levels(test$BsmtFinType2) <- c(levels(test$BsmtFinType2), "None")
test$BsmtFinType2[is.na(test$BsmtFinType2)==TRUE] <- "None"

levels(test$GarageType) <- c(levels(test$GarageType), "None")
test$GarageType[is.na(test$GarageType)==TRUE] <- "None"

levels(test$GarageFinish) <- c(levels(test$GarageFinish), "None")
test$GarageFinish[is.na(test$GarageFinish)==TRUE] <- "None"

levels(test$BsmtQual) <- c(levels(test$BsmtQual), "None")
test$BsmtQual[is.na(test$BsmtQual)==TRUE] <- "None"

levels(test$GarageQual) <- c(levels(test$GarageQual), "None")
test$GarageQual[is.na(test$GarageQual)==TRUE] <- "None"

levels(test$GarageCond) <- c(levels(test$GarageCond), "None")
test$GarageCond[is.na(test$GarageCond)==TRUE] <- "None"

levels(test$PoolQC) <- c(levels(test$PoolQC), "None")
test$PoolQC[is.na(test$PoolQC)==TRUE] <- "None"

levels(test$Fence) <- c(levels(test$Fence), "None")
test$Fence[is.na(test$Fence)==TRUE] <- "None"

levels(test$MiscFeature) <- c(levels(test$MiscFeature), "None")
test$MiscFeature[is.na(test$MiscFeature)==TRUE] <- "None"

#How many NAs left
colSums(is.na(test))

```

```

##      MSSubClass      MSZoning  LotFrontage      LotArea      Street
##           0           4           0           0           0
##       Alley      LotShape  LandContour      LotConfig      LandSlope

```

```
##           0           0           0           0           0
## Neighborhood Condition1 Condition2 BldgType HouseStyle
##           0           0           0           0           0
## OverallQual OverallCond YearBuilt YearRemodAdd RoofStyle
##           0           0           0           0           0
## RoofMatl Exterior1st Exterior2nd MasVnrType MasVnrArea
##           0           1           1           16          15
## ExterQual ExterCond Foundation BsmtQual BsmtCond
##           0           0           0           0           0
## BsmtExposure BsmtFinType1 BsmtFinSF1 BsmtFinType2 BsmtFinSF2
##           0           0           1           0           1
## BsmtUnfSF Heating HeatingQC CentralAir Electrical
##           1           0           0           0           0
## X1stFlrSF X2ndFlrSF LowQualFinSF BsmtFullBath BsmtHalfBath
##           0           0           0           2           2
## FullBath HalfBath BedroomAbvGr KitchenAbvGr KitchenQual
##           0           0           0           0           1
## TotRmsAbvGrd Functional Fireplaces FireplaceQu GarageType
##           0           2           0           0           0
## GarageYrBlt GarageFinish GarageCars GarageArea GarageQual
##           78           0           1           1           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           1
## SaleCondition
##           0
```

```
#Use Knn imputation with k = 10 to estimate missing values. Initialize SalePrice
knnOutput <- knnImputation(test)
knnOutput$SalePrice <- rep(0,1459)
test <- knnOutput
```

```
#Put all into one df
all.df <- as.data.frame(matrix(nrow = 2919, ncol = 77))
all.df[1:1460,] <- train
all.df[1461:2919,] <- test
colnames(all.df) <- df.names
```

```
#Log transform Sale Price
all.df$SalePrice[1:1460] <- log(all.df$SalePrice[1:1460])
```

Linear Regression

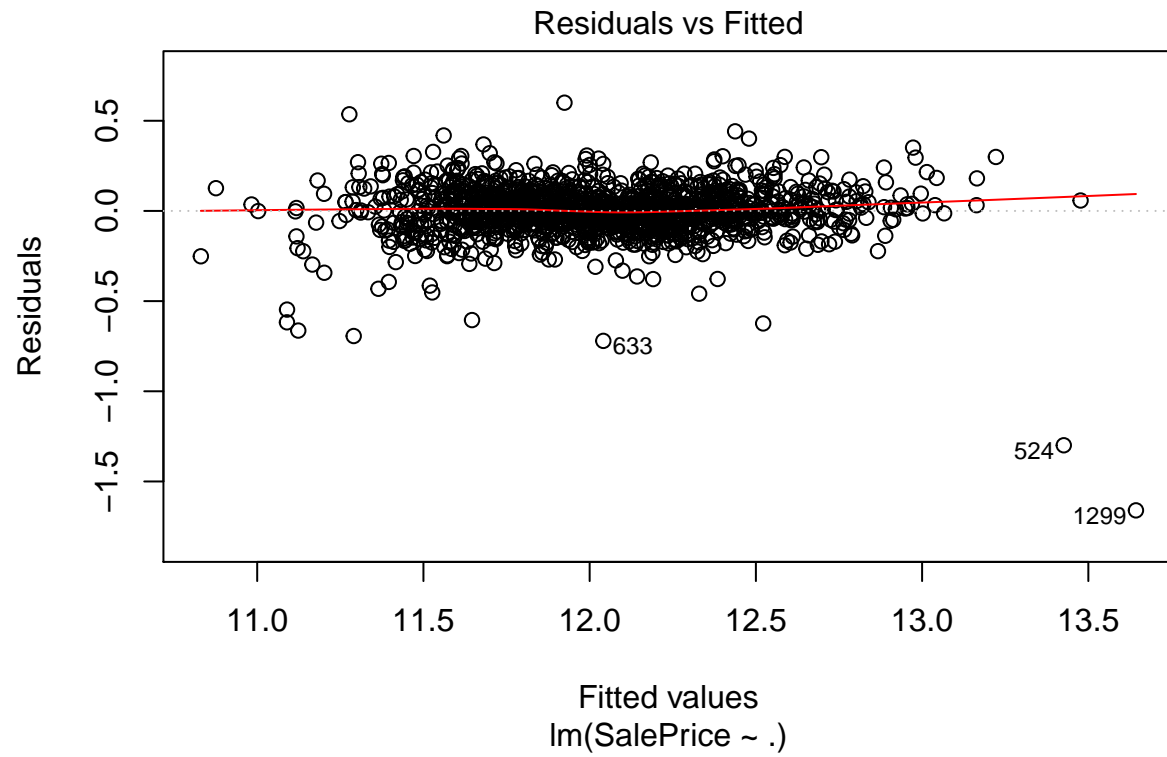
```
#Linear model without removing outliers or high leverage points
lin.model <- lm(SalePrice~., data = all.df[1:1460,])
lmpreds <- predict(lin.model, newdata = all.df[1461:2919,])
e <- exp(1)
lmpreds <- e^lmpreds

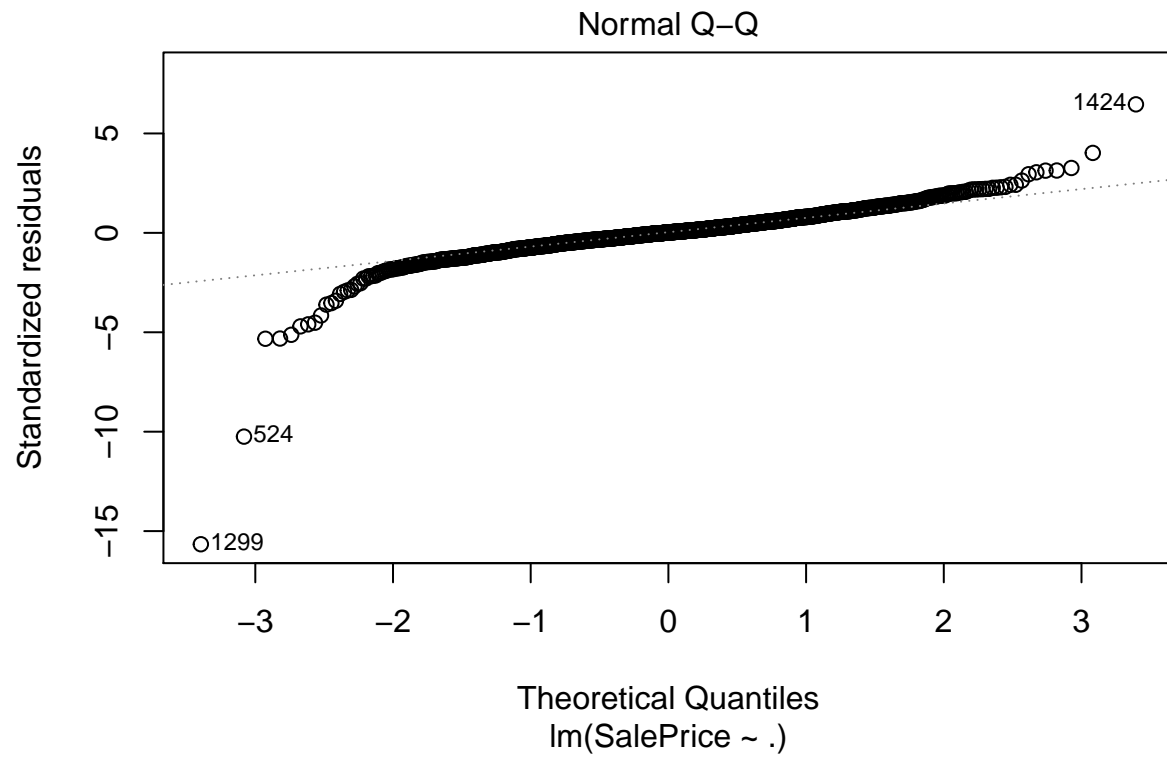
lm.sub.submit <- data.frame(test.ids, lmpreds)
colnames(lm.sub.submit) <- c("Id", "SalePrice")
```

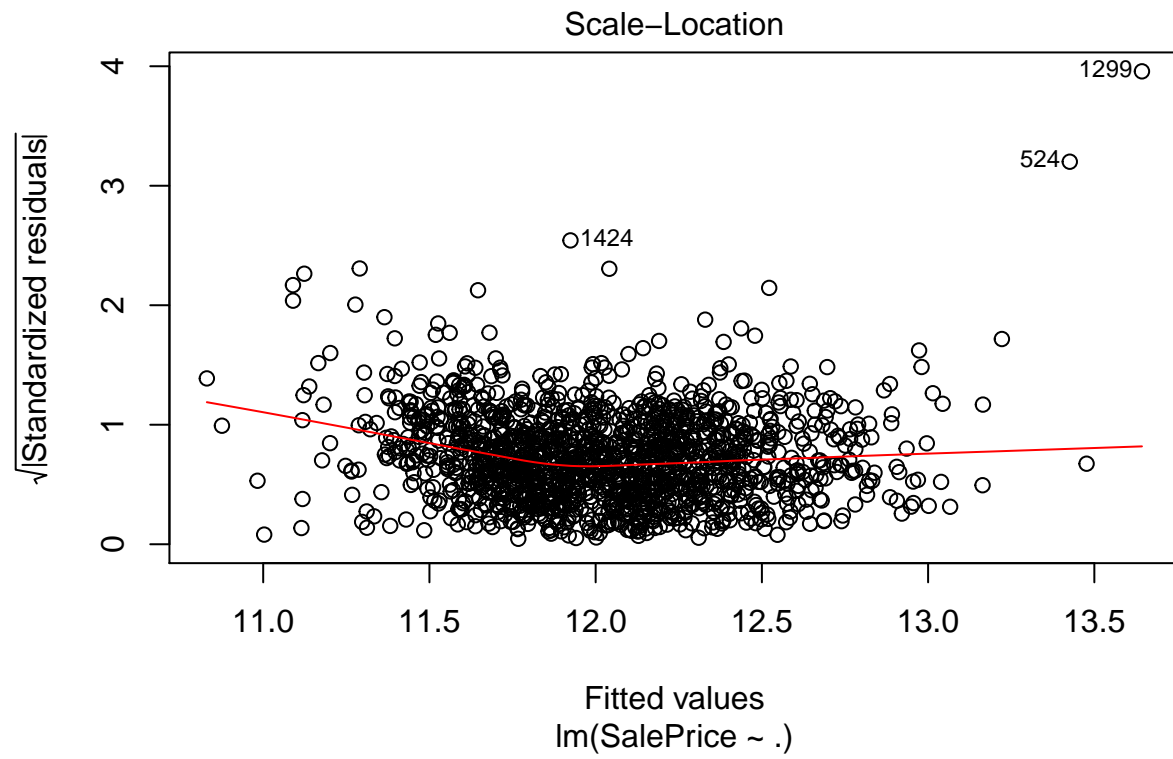
```
#write.csv(lm.sub.submit, "lm.csv")
```

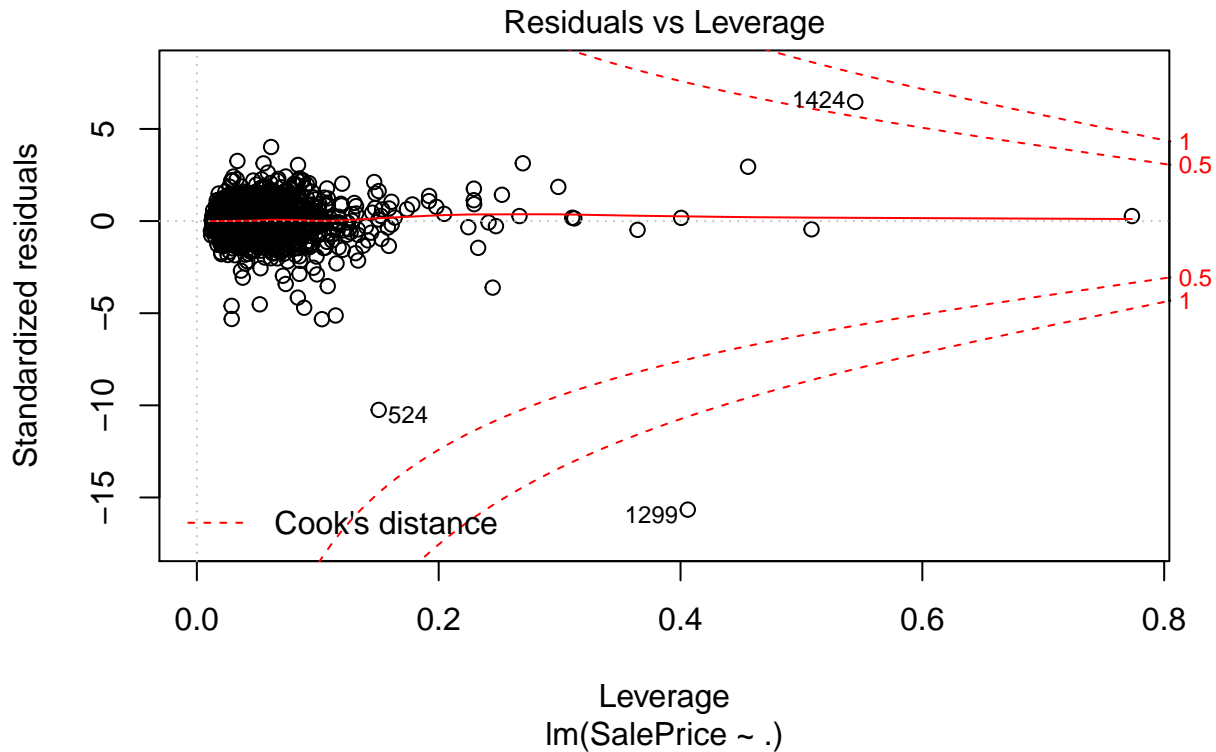
```
#Check for and remove high leverage and outliers
```

```
plot(lin.model)
```









```
all.df.nolvg <- all.df[-c(1424, 1299, 524),]

#No leverage or outliers model
lin.model.nolvg <- lm(SalePrice~., data = all.df.nolvg[1:1457,])
lmpreds <- predict(lin.model.nolvg, newdata = all.df.nolvg[1458:2916,])
lmpreds <- e^lmpreds

lm.sub.submit <- data.frame(test.ids, lmpreds)
colnames(lm.sub.submit) <- c("Id", "SalePrice")
#write.csv(lm.sub.submit, "lm.csv.nolvg.csv")

#Data Exploration
cor(all.df$OverallCond, all.df$OverallQual)

## [1] -0.09384734
cor(all.df$ExterCond, all.df$ExterQual)

## [1] -0.03780843
cor(all.df$BsmtCond, all.df$BsmtQual)

## [1] 0.0954153
cor(all.df$LotFrontage, all.df$LotArea)

## [1] 0.1352459
```

```
cor(all.df$BldgType, all.df$HouseStyle)

## [1] 0.06308723

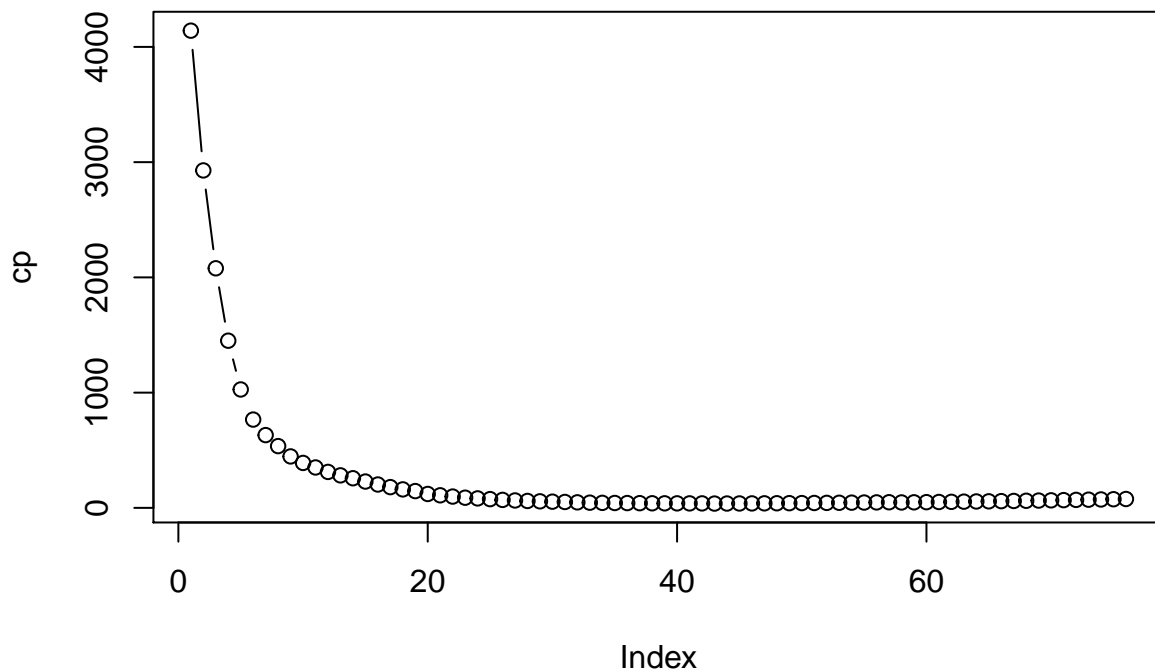
cor(all.df$Exterior1st, all.df$Exterior2nd)

## [1] 0.8432983
```

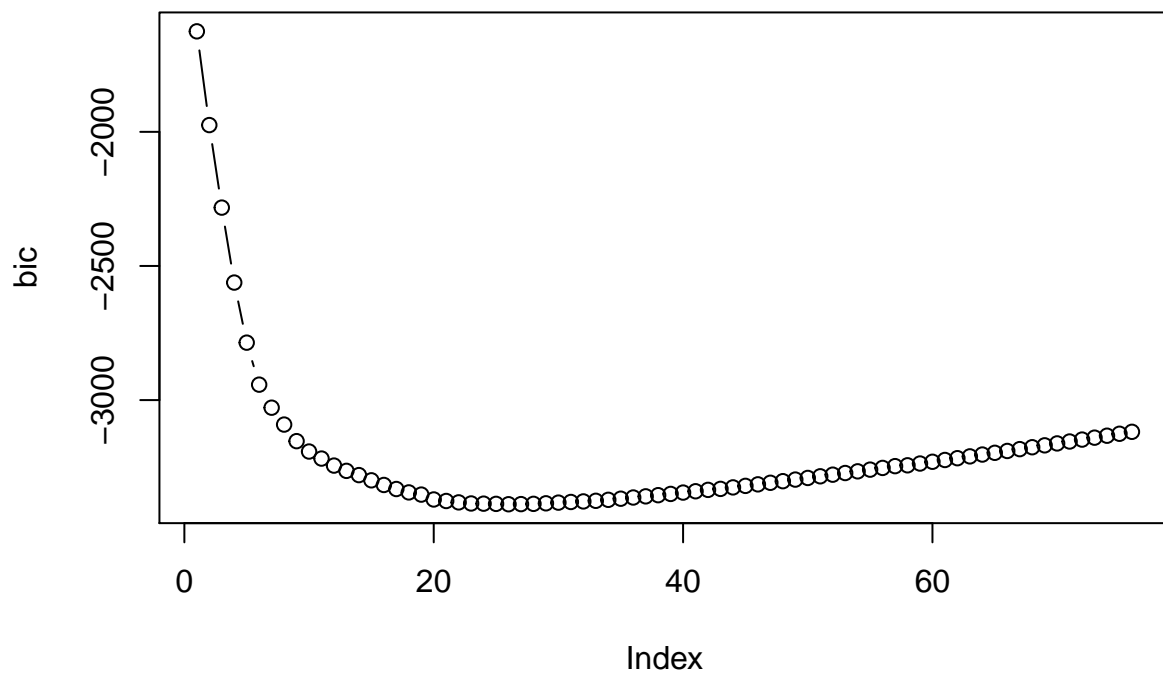
Subset Selection

```
predict.regsubsets <- function(object, newdata , id, ...){
  form <- as.formula(object$call[[2]])
  mat <- model.matrix(form, newdata)
  coefi <- coef(object, id = id)
  xvars <- names(coefi)
  return(mat[,xvars] %*% coefi)
}

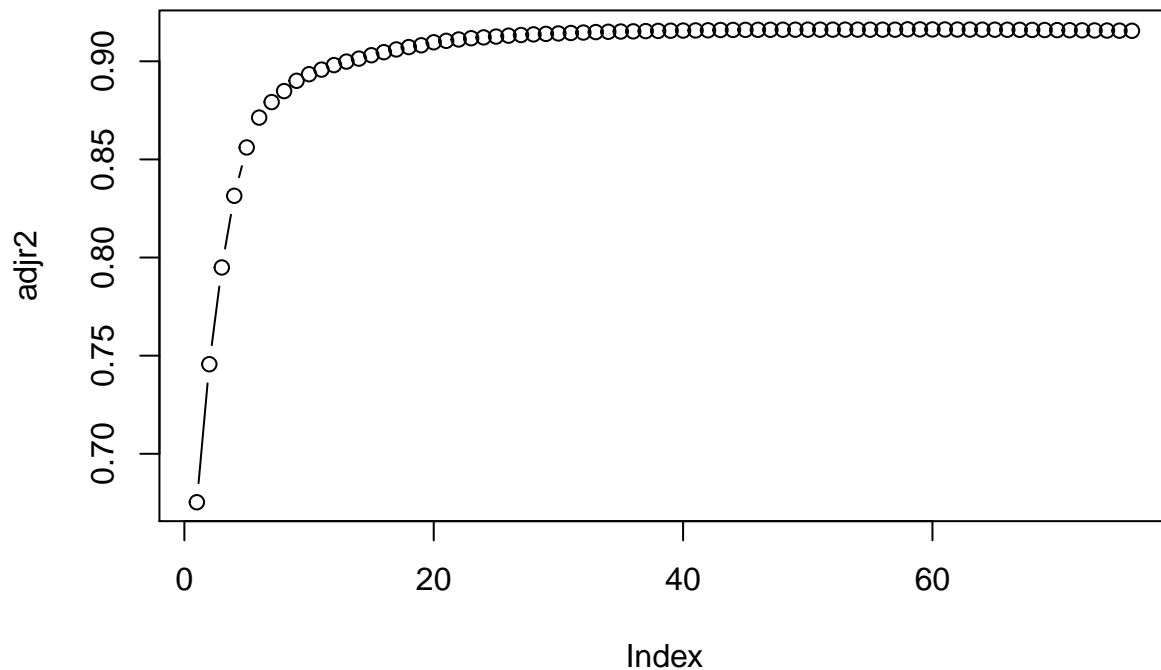
#Forward subset selection with no outliers or leverage pts
forw.sub <- regsubsets(SalePrice~., data = all.df.nolvg[1:1457,], nvmax = 77, method = "forward")
forw.sub.sum <- summary(forw.sub)
#forw.sub.sum
plot(forw.sub.sum$cp, type = "b", ylab = "cp")
```



```
plot(forw.sub.sum$bic, type = "b", ylab = "bic")
```



```
plot(forw.sub.sum$adjr2, type = "b", ylab = "adjr2")
```



```
which.min(forw.sub.sum$bic)
```

```
## [1] 26
```

```
coef(forw.sub, 26)
```

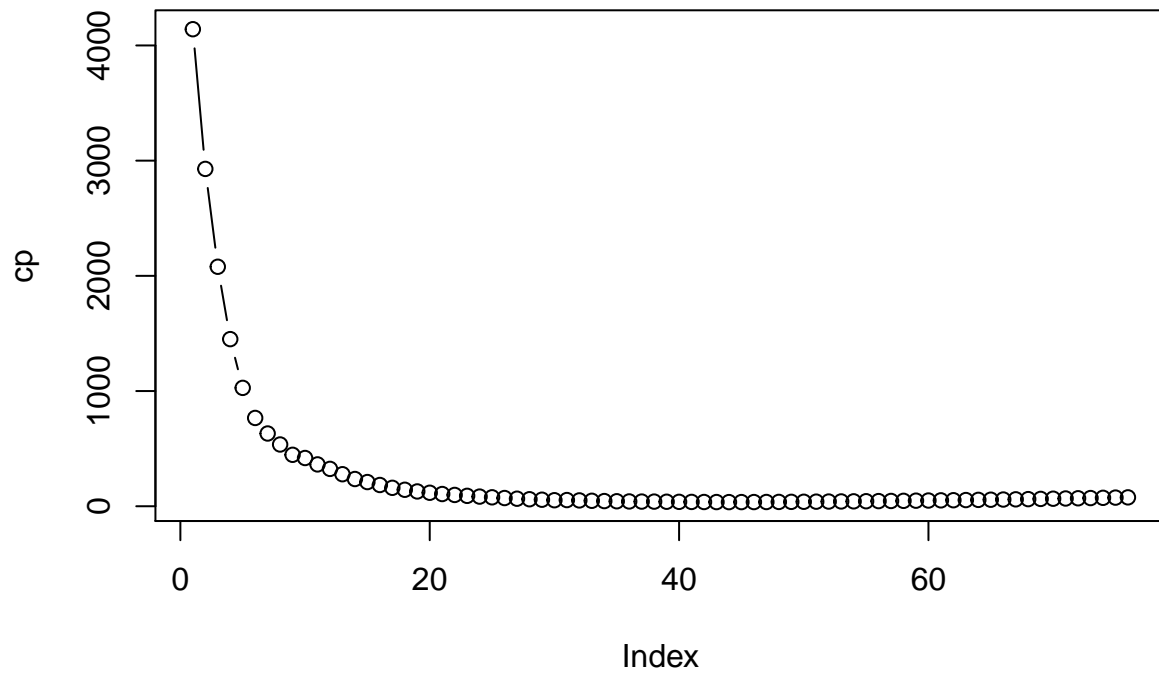
```
##      (Intercept)      MSZoning      LotArea      Street      BldgType
## 4.300799e+00 -1.859880e-02 2.752828e-06 1.952257e-01 -1.466519e-02
## OverallQual OverallCond YearBuilt YearRemodAdd ExterCond
## 6.280007e-02 4.204364e-02 2.212196e-03 7.353405e-04 1.269971e-02
## BsmtQual BsmtFinSF1 BsmtFinSF2 BsmtUnfSF HeatingQC
## -1.646464e-02 1.587461e-04 1.235607e-04 7.641819e-05 -8.619712e-03
## CentralAir X1stFlrSF X2ndFlrSF BsmtFullBath KitchenAbvGr
## 7.354042e-02 3.315885e-04 2.827067e-04 2.782892e-02 -4.729417e-02
## KitchenQual Functional FireplaceQu GarageCars PavedDrive
## -2.074175e-02 1.871924e-02 -1.287291e-02 5.214347e-02 2.133373e-02
## ScreenPorch SaleCondition
## 2.663956e-04 2.573918e-02
```

```
forw.sub.preds <- predict.regsbsets(forw.sub, newdata = all.df.nolvg[1458:2916,], id = 26)
forw.sub.preds <- e~forw.sub.preds
forw.sub.submit <- data.frame(test.ids, forw.sub.preds)
colnames(forw.sub.submit) <- c("Id", "SalePrice")
#write.csv(forw.sub.submit, "Forward.Subset.nolvg.csv")
```

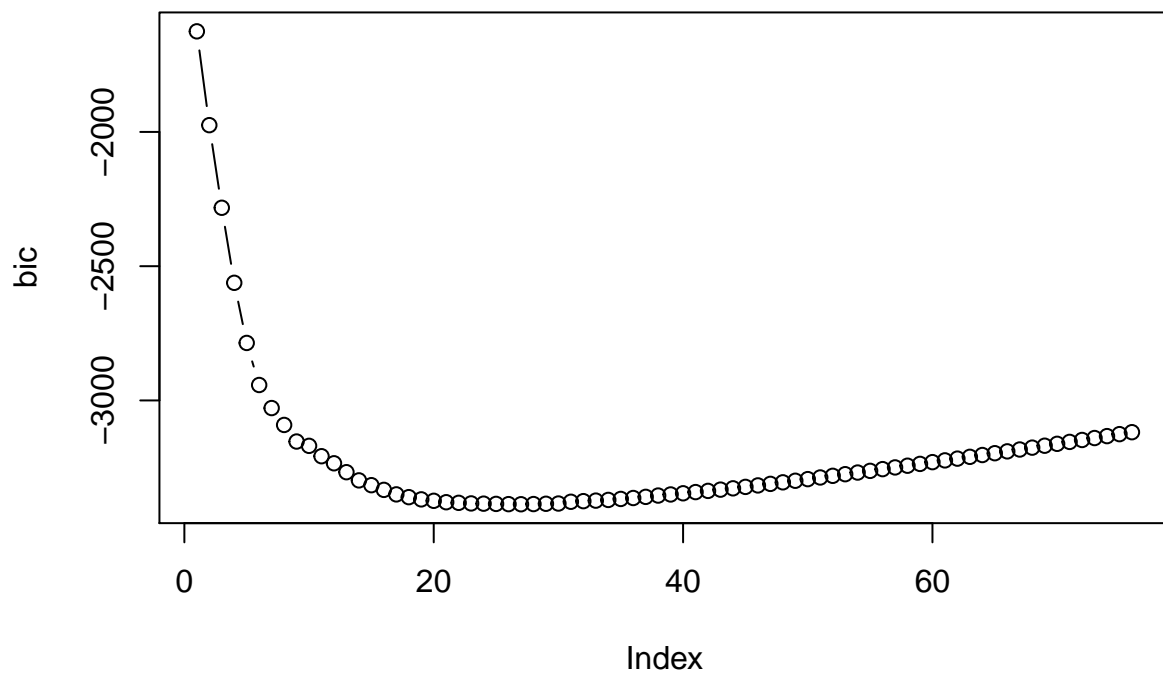
```
#Backward Subset Selection without outliers/leverage
```

```
back.sub <- regsbsets(SalePrice~., data = all.df.nolvg[1:1457,], nvmax = 77, method = "backward")
back.sub.sum <- summary(back.sub)
```

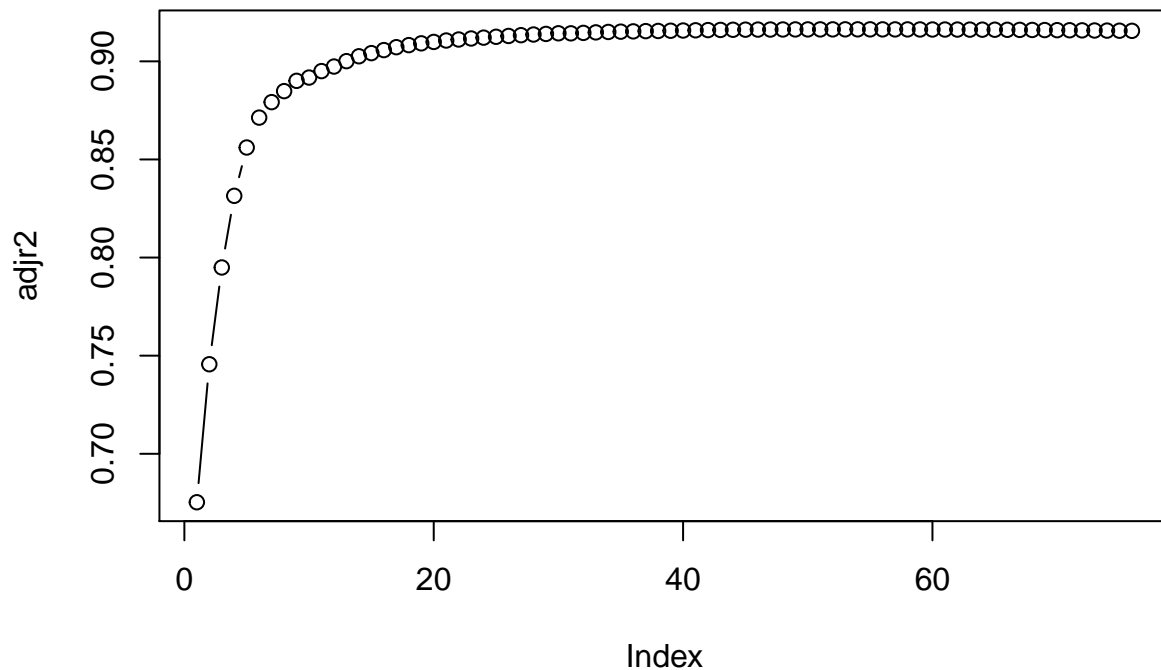
```
#back.sub.sum  
plot(back.sub.sum$cp, type = "b", ylab = "cp")
```



```
plot(back.sub.sum$bic, type = "b", ylab = "bic")
```



```
plot(back.sub.sum$adjr2, type = "b", ylab = "adjr2")
```



```
which.min(back.sub.sum$bic)
```

```
## [1] 27
```

```
coef(back.sub, 27)
```

```
##      (Intercept)      MSZoning      LotArea      Street      BldgType
## 4.146417e+00 -1.801323e-02 2.610436e-06 1.969786e-01 -1.410242e-02
## OverallQual    OverallCond    YearBuilt    YearRemodAdd    MasVnrType
## 6.299799e-02 4.182951e-02 2.223670e-03 7.393258e-04 1.423821e-02
## ExterCond      BsmtQual      BsmtFinSF1      BsmtFinSF2      BsmtUnfSF
## 1.265530e-02 -1.656281e-02 1.590889e-04 1.247363e-04 7.847722e-05
## HeatingQC      CentralAir      X1stFlrSF      X2ndFlrSF      BsmtFullBath
## -8.616867e-03 7.288525e-02 3.302104e-04 2.795593e-04 2.524375e-02
## KitchenAbvGr    KitchenQual    Functional    Fireplaces    GarageCars
## -4.873816e-02 -2.166535e-02 2.021194e-02 2.823062e-02 5.252269e-02
## PavedDrive      ScreenPorch    SaleCondition
## 2.137493e-02 2.509782e-04 2.541942e-02
```

```
back.sub.preds <- predict.regsbsets(back.sub, newdata = all.df.nolvg[1458:2916,], id = 27)
```

```
back.sub.preds <- e~back.sub.preds
```

```
back.sub.submit <- data.frame(test.ids, back.sub.preds)
```

```
colnames(back.sub.submit) <- c("Id", "SalePrice")
```

```
#write.csv(back.sub.submit, "Backward.Subset.nolvg.csv")
```

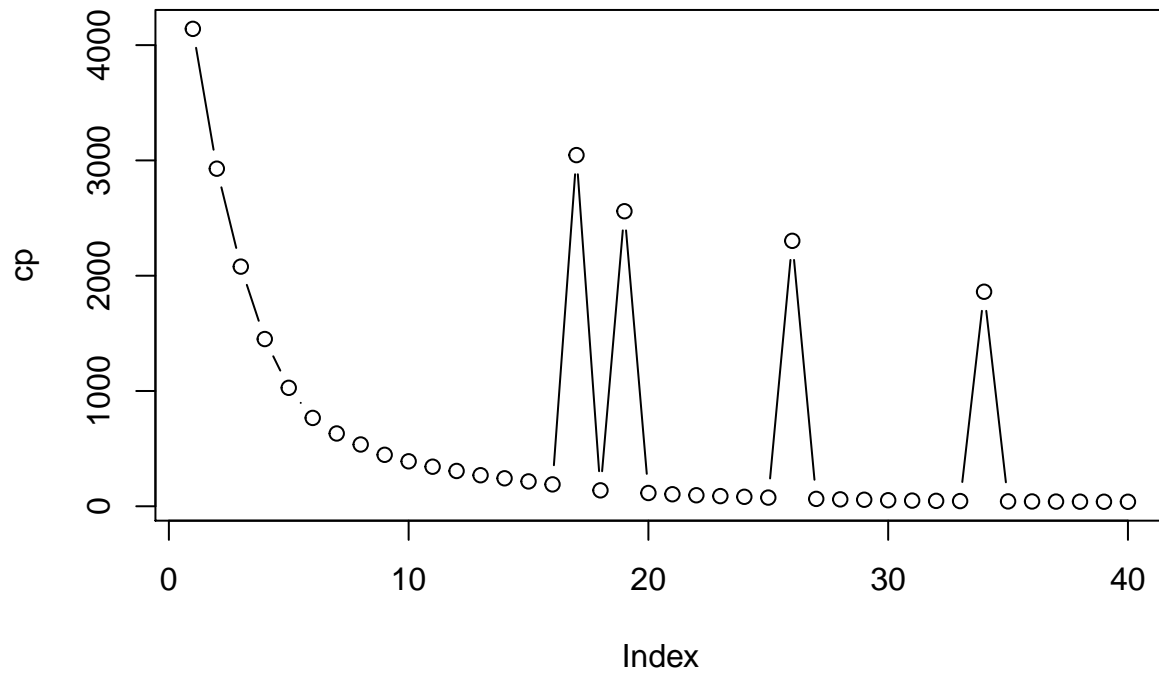
```
#Mixed subsets with no outliers
```

```
mix.sub <- regsbsets(SalePrice~., data = all.df.nolvg[1:1457,], nvmax = 40, method = "seqrep")
```

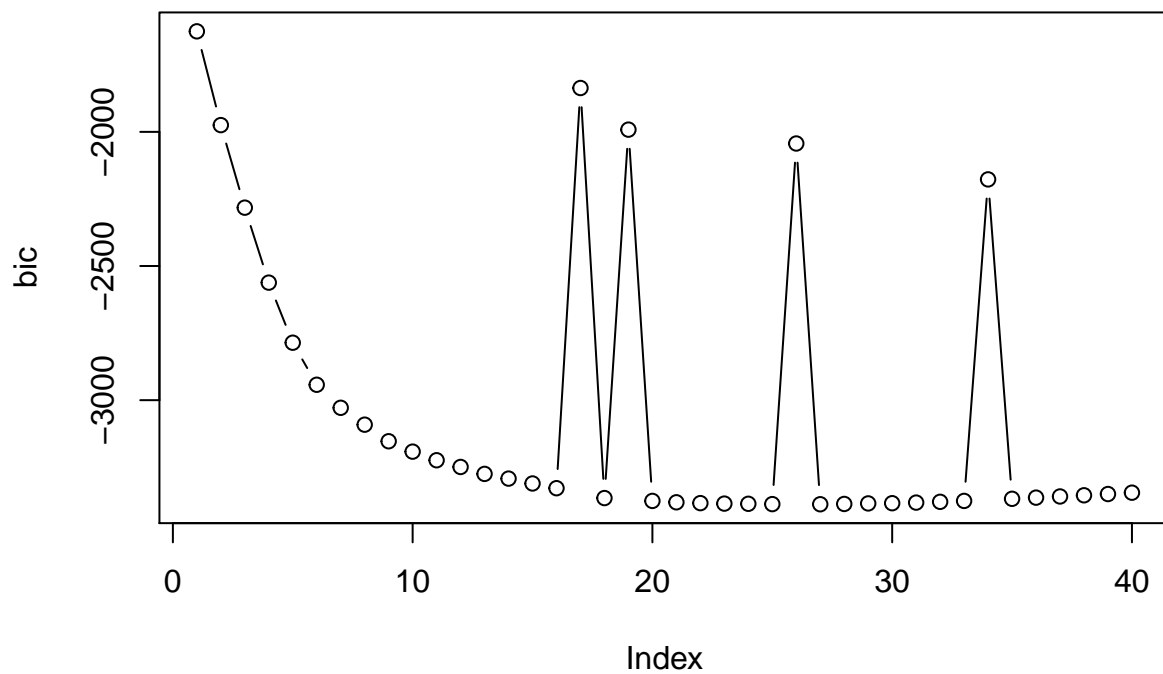
```
mix.sub.sum <- summary(mix.sub)
```



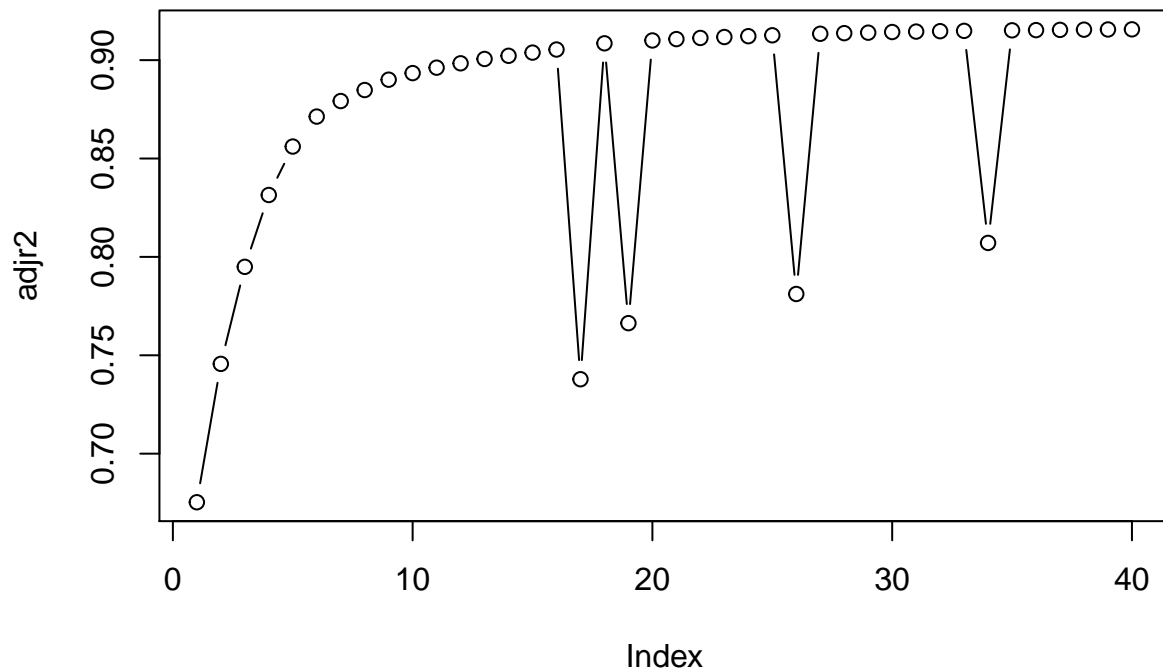
```
plot(mix.sub.sum$cp, type = "b", ylab = "cp")
```



```
plot(mix.sub.sum$bic, type = "b", ylab = "bic")
```



```
plot(mix.sub.sum$adjr2, type = "b", ylab = "adjr2")
```



```
which.min(mix.sub.sum$bic)
```

```
## [1] 27
```

```
coef(mix.sub, 27)
```

```
##      (Intercept)      MSZoning      LotArea      Street      BldgType
## 4.748404e+00 -1.821245e-02 2.767292e-06 1.926123e-01 -1.456543e-02
## OverallQual    OverallCond    YearBuilt    YearRemodAdd    ExterCond
## 6.279768e-02 4.292570e-02 1.998854e-03 7.019910e-04 1.281384e-02
## Foundation      BsmtQual      BsmtFinSF1      BsmtFinSF2      BsmtUnfSF
## 1.531053e-02 -1.663428e-02 1.663813e-04 1.322147e-04 8.437926e-05
## HeatingQC      CentralAir      X1stFlrSF      X2ndFlrSF      BsmtFullBath
## -8.069153e-03 7.443263e-02 3.247582e-04 2.813146e-04 2.828080e-02
## KitchenAbvGr    KitchenQual    Functional    FireplaceQu    GarageCars
## -5.039486e-02 -1.969084e-02 1.819833e-02 -1.310323e-02 5.177468e-02
## PavedDrive      ScreenPorch    SaleCondition
## 2.242308e-02 2.690391e-04 2.565312e-02
```

```
mix.sub.preds <- predict.regsubsets(mix.sub, newdata = all.df.nolvg[1458:2916,], id = 27)
mix.sub.preds <- e~mix.sub.preds
mix.sub.submit <- data.frame(test.ids, mix.sub.preds)
colnames(mix.sub.submit) <- c("Id", "SalePrice")
#write.csv(mix.sub.submit, "Mixed.Subset.nolvg.csv")
```

Shrinkage Methods

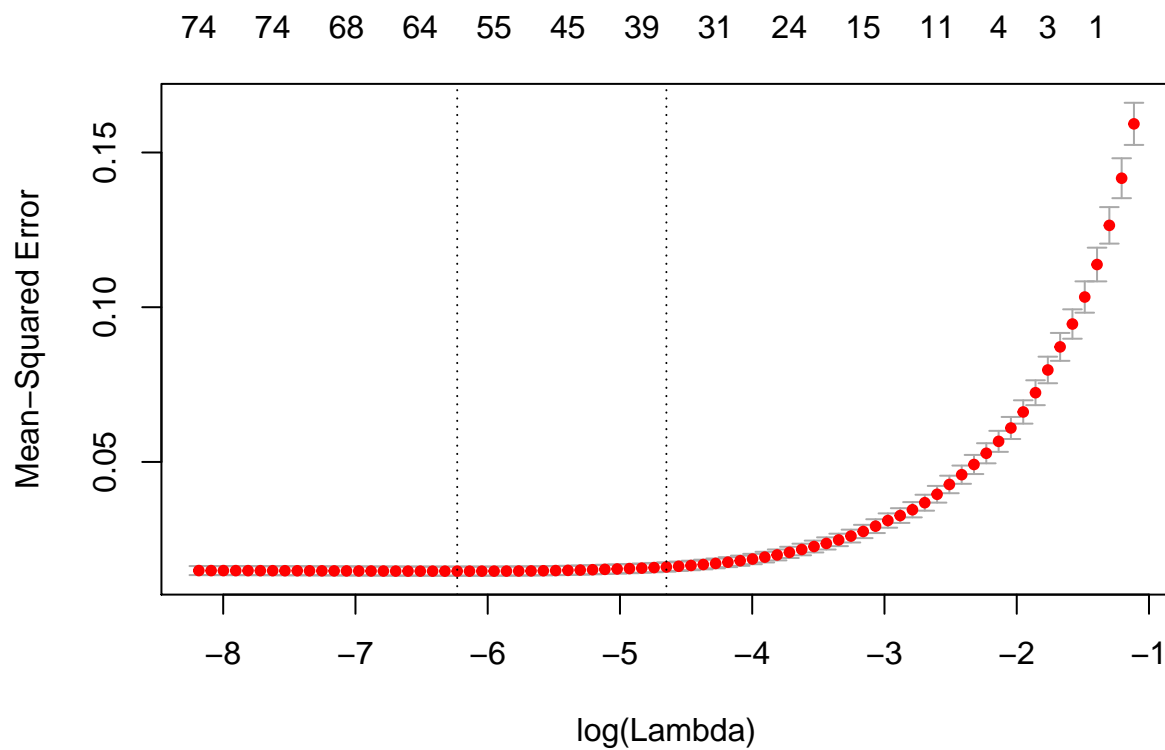
```
#Lasso Method without outliers
mod.mat <- model.matrix(SalePrice~., data = all.df.nolvg)[-1]

lasso.model <- glmnet(mod.mat[1:1457,],all.df.nolvg$SalePrice[1:1457], alpha = 1)

cv.out <- cv.glmnet(mod.mat[1:1457,],all.df.nolvg$SalePrice[1:1457], alpha = 1, nfolds = 10)
cv.out$lambda.min

## [1] 0.001968419

plot(cv.out)
```



```
lasso.pred <- predict(lasso.model, s = cv.out$lambda.min, newx = mod.mat[1458:2916,])
lasso.pred <- e^lasso.pred

lasso.pred.coef <- predict(lasso.model, s = cv.out$lambda.min, newx = mod.mat[1458:2916,], type = "coef")
lasso.pred.coef

## 77 x 1 sparse Matrix of class "dgCMatrix"
##              1
## (Intercept)  1.353332e+01
## MSSubClass   -5.514656e-05
## MSZoning     -1.298365e-02
## LotFrontage  4.289997e-05
## LotArea      2.416764e-06
```

## Street	1.701493e-01
## Alley	2.383405e-04
## LotShape	-3.855912e-03
## LandContour	-4.274479e-03
## LotConfig	-1.253532e-03
## LandSlope	.
## Neighborhood	.
## Condition1	1.625177e-04
## Condition2	-6.398460e-03
## BldgType	-1.073767e-02
## HouseStyle	.
## OverallQual	6.406108e-02
## OverallCond	3.923384e-02
## YearBuilt	1.699725e-03
## YearRemodAdd	7.071759e-04
## RoofStyle	.
## RoofMatl	-3.518907e-04
## Exterior1st	.
## Exterior2nd	3.108448e-04
## MasVnrType	1.082768e-02
## MasVnrArea	1.307620e-05
## ExterQual	-1.102433e-02
## ExterCond	1.050415e-02
## Foundation	1.098649e-02
## BsmtQual	-1.353493e-02
## BsmtCond	3.183254e-03
## BsmtExposure	-3.701239e-03
## BsmtFinType1	-1.342846e-03
## BsmtFinSF1	1.341418e-04
## BsmtFinType2	.
## BsmtFinSF2	9.047567e-05
## BsmtUnfSF	5.639437e-05
## Heating	.
## HeatingQC	-7.595133e-03
## CentralAir	7.480337e-02
## Electrical	.
## X1stFlrSF	3.015939e-04
## X2ndFlrSF	2.311540e-04
## LowQualFinSF	7.939557e-05
## BsmtFullBath	2.478487e-02
## BsmtHalfBath	.
## FullBath	1.655957e-02
## HalfBath	1.813691e-02
## BedroomAbvGr	.
## KitchenAbvGr	-4.145806e-02
## KitchenQual	-1.592542e-02
## TotRmsAbvGrd	4.052615e-03
## Functional	1.737779e-02
## Fireplaces	1.863002e-02
## FireplaceQu	-5.678028e-03
## GarageType	-7.323659e-04
## GarageYrBlt	.
## GarageFinish	-7.832782e-03
## GarageCars	3.273919e-02

```
## GarageArea      6.543230e-05
## GarageQual      .
## GarageCond      4.034506e-04
## PavedDrive      2.183570e-02
## WoodDeckSF      5.786669e-05
## OpenPorchSF     4.150208e-05
## EnclosedPorch   4.903780e-05
## X3SsnPorch      6.508460e-05
## ScreenPorch     2.444505e-04
## PoolArea        .
## PoolQC          4.800378e-03
## Fence           1.832998e-03
## MiscFeature     8.008188e-03
## MiscVal         .
## MoSold          .
## YrSold          -4.075854e-03
## SaleType        -5.017632e-04
## SaleCondition   2.351057e-02
```

```
lasso.sub.submit <- data.frame(test.ids, lasso.pred)
colnames(lasso.sub.submit) <- c("Id", "SalePrice")
#write.csv(lasso.sub.submit, "Lasso.nolvg.csv")
```

```
#Ridge Regression with no lvg
```

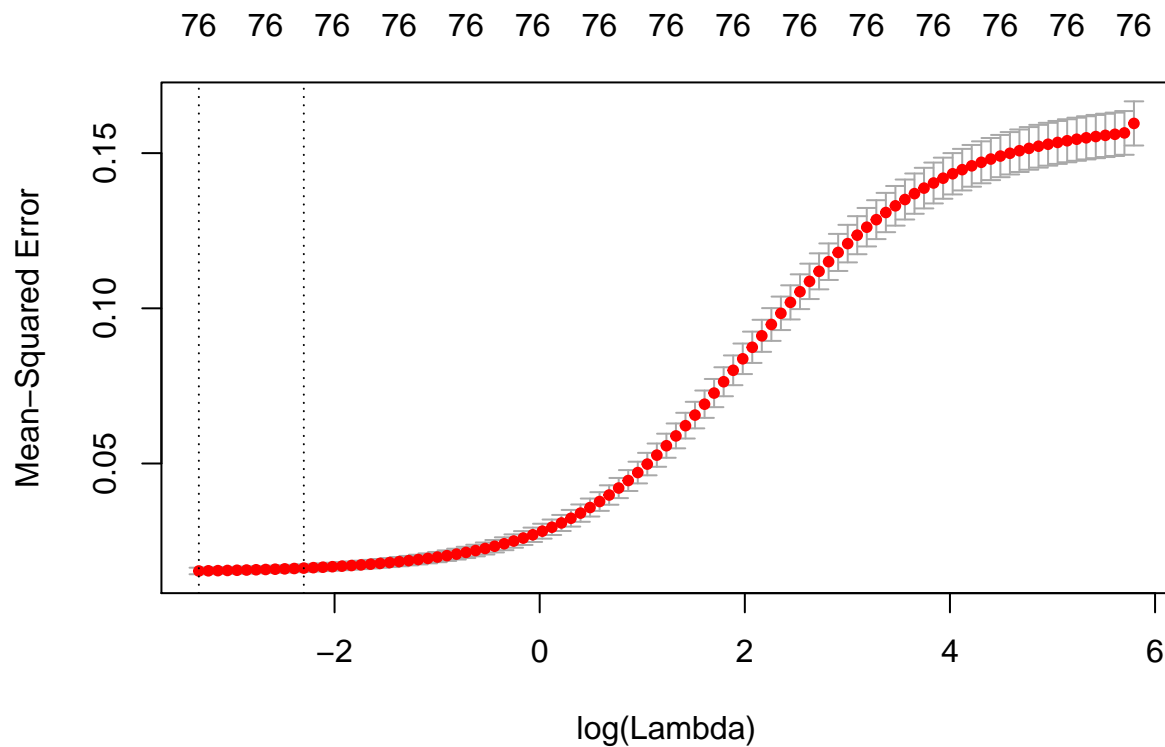
```
mod.mat <- model.matrix(SalePrice~., data = all.df.nolvg)[,-1]
```

```
ridge.model <- glmnet(mod.mat[1:1457,],all.df.nolvg$SalePrice[1:1457], alpha = 0)
```

```
cv.out <- cv.glmnet(mod.mat[1:1457,],all.df.nolvg$SalePrice[1:1457], alpha = 0, nfolds = 10)
cv.out$lambda.min
```

```
## [1] 0.0360366
```

```
plot(cv.out)
```



```
ridge.pred <- predict(ridge.model, s = cv.out$lambda.min, newx = mod.mat[1458:2916,])
ridge.pred <- e^ridge.pred
```

```
ridge.pred.coef <- predict(ridge.model, s = cv.out$lambda.min, newx = mod.mat[1458:2916,], type = "coef")
ridge.pred.coef
```

```
## 77 x 1 sparse Matrix of class "dgCMatrix"
```

```
##              1
## (Intercept)  1.704182e+01
## MSSubClass   -8.877228e-05
## MSZoning     -1.436319e-02
## LotFrontage   1.057223e-04
## LotArea       2.484238e-06
## Street       2.051126e-01
## Alley        5.033736e-03
## LotShape     -4.967935e-03
## LandContour  -7.043125e-03
## LotConfig    -2.584322e-03
## LandSlope     9.767080e-04
## Neighborhood -2.620827e-04
## Condition1   1.973383e-03
## Condition2   -1.330404e-02
## BldgType     -8.492943e-03
## HouseStyle    1.347081e-04
## OverallQual   5.587803e-02
## OverallCond   3.550927e-02
```

## YearBuilt	1.014677e-03
## YearRemodAdd	9.228143e-04
## RoofStyle	4.281405e-03
## RoofMatl	1.077880e-03
## Exterior1st	-2.913221e-03
## Exterior2nd	2.566696e-03
## MasVnrType	1.339879e-02
## MasVnrArea	5.100649e-05
## ExterQual	-1.647275e-02
## ExterCond	1.248597e-02
## Foundation	1.739220e-02
## BsmtQual	-1.691262e-02
## BsmtCond	5.885779e-03
## BsmtExposure	-5.762806e-03
## BsmtFinType1	-2.983505e-03
## BsmtFinSF1	1.235663e-04
## BsmtFinType2	1.416457e-03
## BsmtFinSF2	9.967845e-05
## BsmtUnfSF	6.118715e-05
## Heating	-2.272464e-03
## HeatingQC	-8.717511e-03
## CentralAir	8.035479e-02
## Electrical	-1.406062e-04
## X1stFlrSF	2.050090e-04
## X2ndFlrSF	1.467838e-04
## LowQualFinSF	8.531364e-05
## BsmtFullBath	3.392880e-02
## BsmtHalfBath	8.707211e-03
## FullBath	4.181557e-02
## HalfBath	3.450478e-02
## BedroomAbvGr	4.061847e-03
## KitchenAbvGr	-5.291882e-02
## KitchenQual	-1.925444e-02
## TotRmsAbvGrd	1.502259e-02
## Functional	1.680196e-02
## Fireplaces	2.745537e-02
## FireplaceQu	-5.954970e-03
## GarageType	-2.972242e-03
## GarageYrBlt	-2.822121e-05
## GarageFinish	-9.913755e-03
## GarageCars	3.028137e-02
## GarageArea	1.013484e-04
## GarageQual	-3.238982e-03
## GarageCond	7.214812e-03
## PavedDrive	2.722856e-02
## WoodDeckSF	8.505257e-05
## OpenPorchSF	9.959643e-05
## EnclosedPorch	1.014291e-04
## X3SsnPorch	1.201164e-04
## ScreenPorch	2.624127e-04
## PoolArea	3.168582e-05
## PoolQC	7.568543e-03
## Fence	3.471121e-03
## MiscFeature	1.717240e-02


```
## MiscVal      5.026172e-06
## MoSold       2.399476e-04
## YrSold       -5.362944e-03
## SaleType     -1.990245e-03
## SaleCondition 2.250869e-02

ridge.sub.submit <- data.frame(test.ids, ridge.pred)
colnames(ridge.sub.submit) <- c("Id", "SalePrice")
#write.csv(ridge.sub.submit, "Ridge.nolvg.csv")
```

Dimension Reduction Methods

```
#PCR without lvg
mod.mat <- model.matrix(SalePrice~., data = all.df.nolvg)[-1]
pcr.fit <- pcr(SalePrice~., data = all.df.nolvg[1:1457,], scale = TRUE, validation = "CV")

pcr.cv.rmse <- RMSEP(pcr.fit)
which.min(pcr.cv.rmse$val[1,1,-1])
```

```
## 76 comps
##      76
```

```
pcr.preds <- predict(pcr.fit, newdata = mod.mat[1458:2916,], ncomp = which.min(pcr.cv.rmse$val[1,1,-1]))
pcr.preds <- e~pcr.preds
```

```
pcr.sub.submit <- data.frame(test.ids, pcr.preds)
colnames(pcr.sub.submit) <- c("Id", "SalePrice")
#write.csv(pcr.sub.submit, "pcr.nolvg.csv")
```

```
#PLS without lvg
mod.mat <- model.matrix(SalePrice~., data = all.df.nolvg)[-1]
pls.fit <- pls(SalePrice~., data = all.df.nolvg[1:1457,], scale = TRUE, validation = "CV")

pls.cv.rmse <- RMSEP(pls.fit)
which.min(pls.cv.rmse$val[1,1,-1])
```

```
## 37 comps
##      37
```

```
pls.preds <- predict(pls.fit, newdata = mod.mat[1458:2916,], ncomp = which.min(pls.cv.rmse$val[1,1,-1]))
pls.preds <- e~pls.preds
```

```
pls.sub.submit <- data.frame(test.ids, pls.preds)
colnames(pls.sub.submit) <- c("Id", "SalePrice")
#write.csv(pls.sub.submit, "pls.nolvg.csv")
```

```
#AVG PCR and PLS without lvg
avg.preds <- (pls.preds + pcr.preds)/2
avg.sub.submit <- data.frame(test.ids, avg.preds)
colnames(avg.sub.submit) <- c("Id", "SalePrice")
#write.csv(avg.sub.submit, "avg.nolvg.csv")
```

KNN Method

```
knn.cv.errs <- rep(0,45)
for(k in 1:45){
  knn.pred.cv <- knn.reg(train=all.df[1:1460,1:76], y=all.df[1:1460,77], k=k)
  knn.cv.errs[k] <- knn.pred.cv$R2Pred
}
knn.cv.errs
```

```
## [1] 0.5293874 0.6252251 0.6432279 0.6549161 0.6641704 0.6652731 0.6712946
## [8] 0.6656542 0.6618129 0.6576714 0.6554114 0.6520779 0.6497247 0.6446271
## [15] 0.6422286 0.6388551 0.6361311 0.6298162 0.6279316 0.6256842 0.6199400
## [22] 0.6165615 0.6132717 0.6090084 0.6035331 0.5986864 0.5952055 0.5906836
## [29] 0.5877867 0.5847024 0.5823999 0.5801287 0.5778674 0.5756501 0.5719079
## [36] 0.5692908 0.5656224 0.5624919 0.5595485 0.5575187 0.5551391 0.5515529
## [43] 0.5501435 0.5473380 0.5452276
```

```
which.max(knn.cv.errs)
```

```
## [1] 7
```

```
#Predict and export
```

```
knn.pred <- knn.reg(all.df[1:1460,1:76], all.df[1461:2919,1:76], all.df[1:1460,77], k=7)
knn.preds <- e^knn.pred$pred
knn.sub.submit <- data.frame(test.ids, knn.preds)
colnames(knn.sub.submit) <- c("Id", "SalePrice")
#write.csv(knn.sub.submit, "knn2.csv")
```

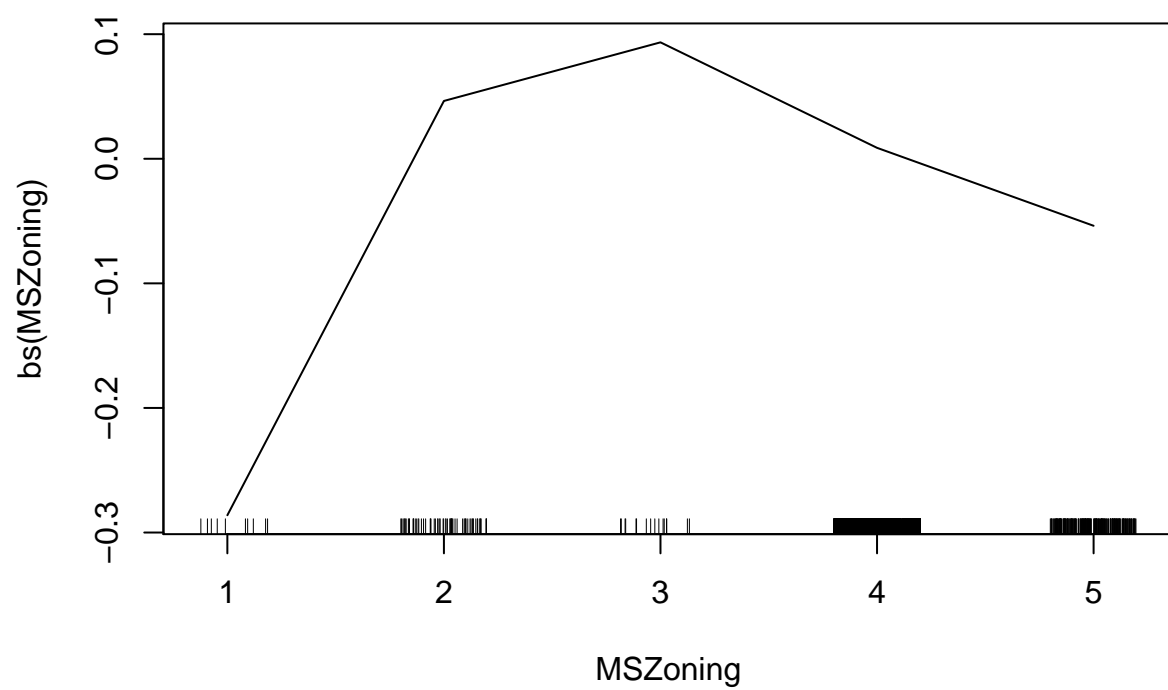
Nonlinear Regression

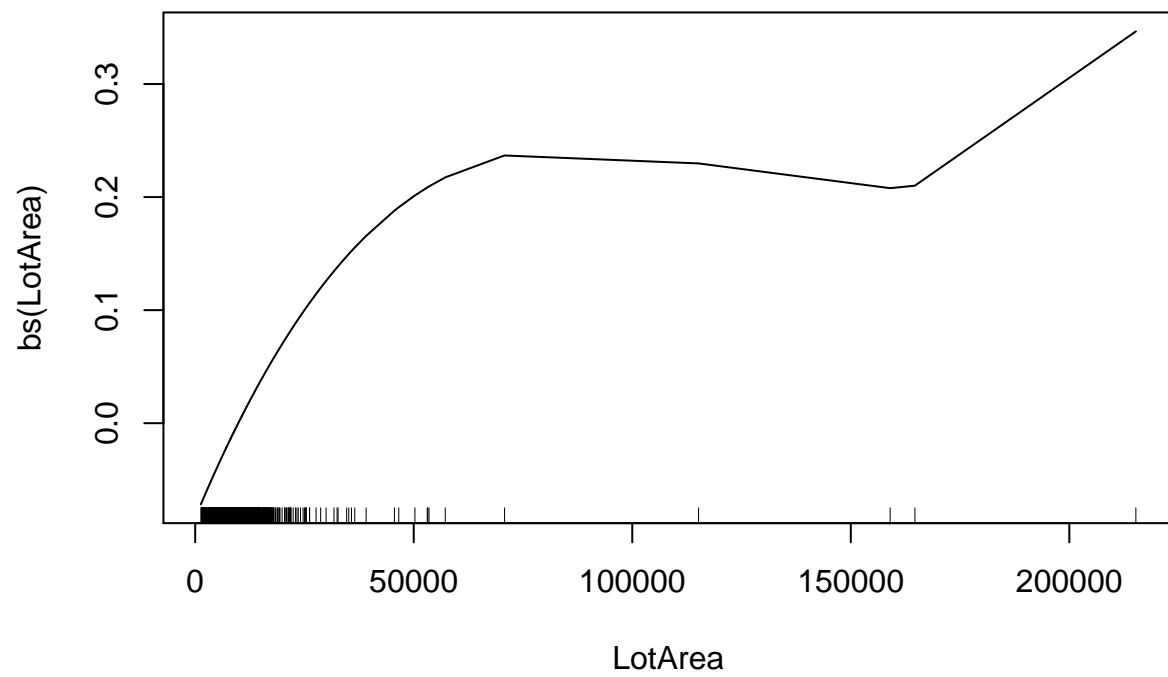
```
#Mixed subset natural splines
```

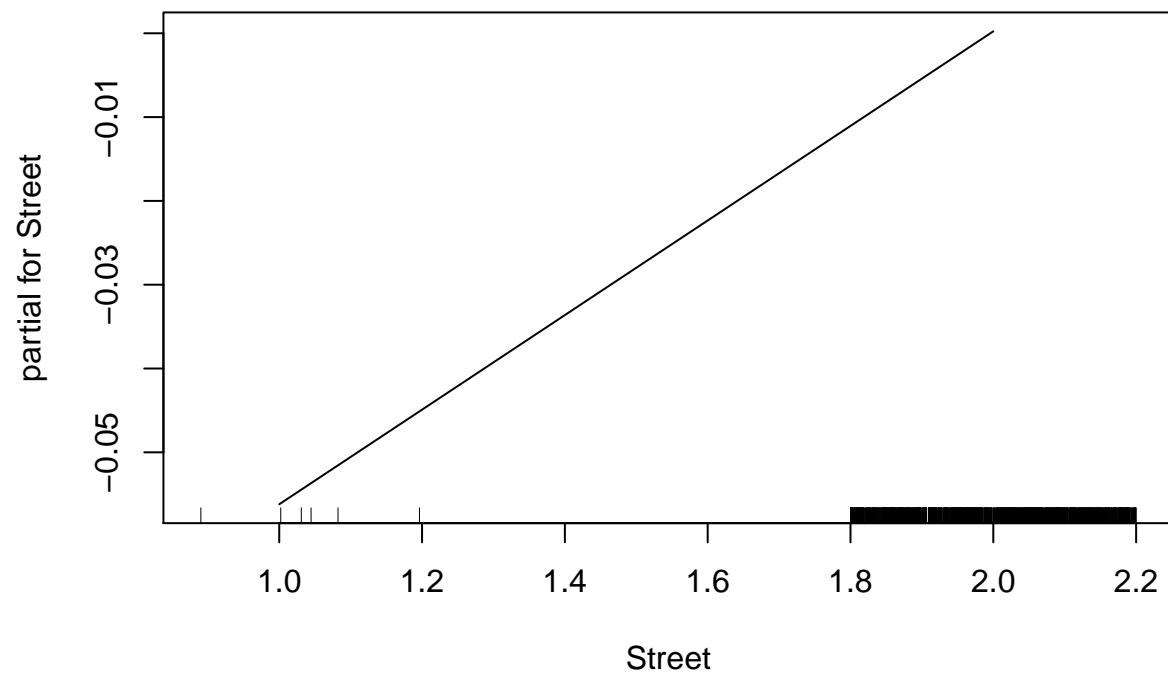
```
coef(mix.sub, 27)
```

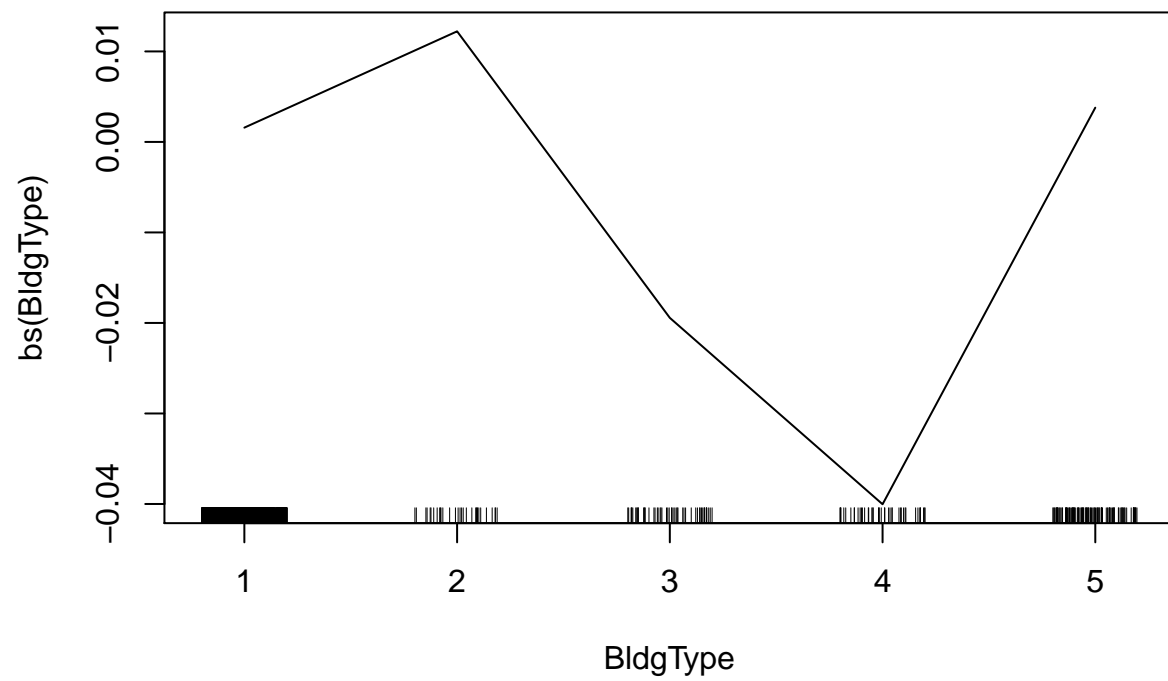
```
## (Intercept) MSZoning LotArea Street BldgType
## 4.748404e+00 -1.821245e-02 2.767292e-06 1.926123e-01 -1.456543e-02
## OverallQual OverallCond YearBuilt YearRemodAdd ExterCond
## 6.279768e-02 4.292570e-02 1.998854e-03 7.019910e-04 1.281384e-02
## Foundation BsmtQual BsmtFinSF1 BsmtFinSF2 BsmtUnfSF
## 1.531053e-02 -1.663428e-02 1.663813e-04 1.322147e-04 8.437926e-05
## HeatingQC CentralAir X1stFlrSF X2ndFlrSF BsmtFullBath
## -8.069153e-03 7.443263e-02 3.247582e-04 2.813146e-04 2.828080e-02
## KitchenAbvGr KitchenQual Functional FireplaceQu GarageCars
## -5.039486e-02 -1.969084e-02 1.819833e-02 -1.310323e-02 5.177468e-02
## PavedDrive ScreenPorch SaleCondition
## 2.242308e-02 2.690391e-04 2.565312e-02
```

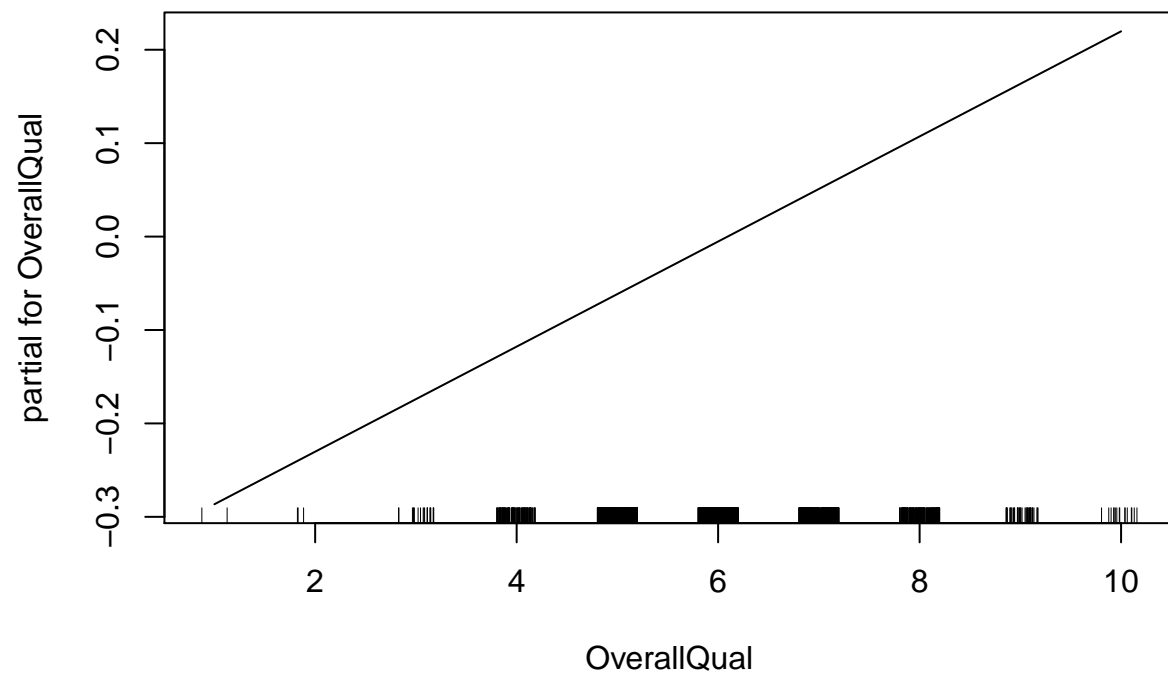
```
gam.lm <- lm(SalePrice~bs(MSZoning)+bs(LotArea)+Street+bs(BldgType)+OverallQual+bs(OverallCond)+bs(YearBuilt))
#summary(gam.lm)
plot.Gam(gam.lm)
```

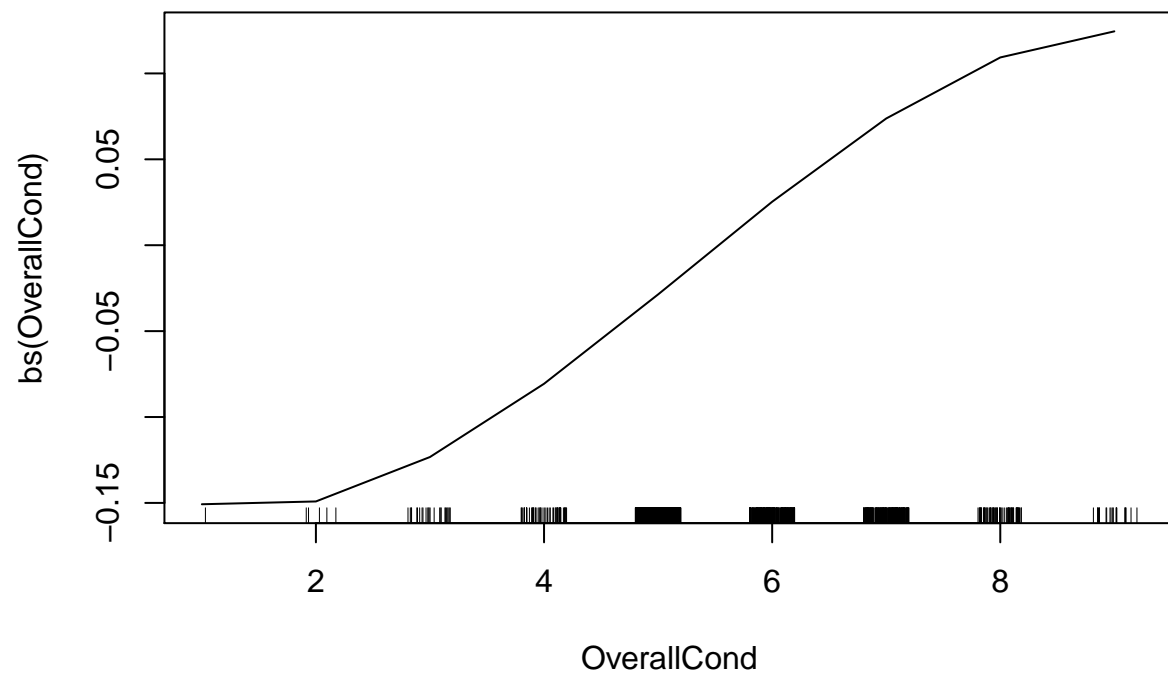


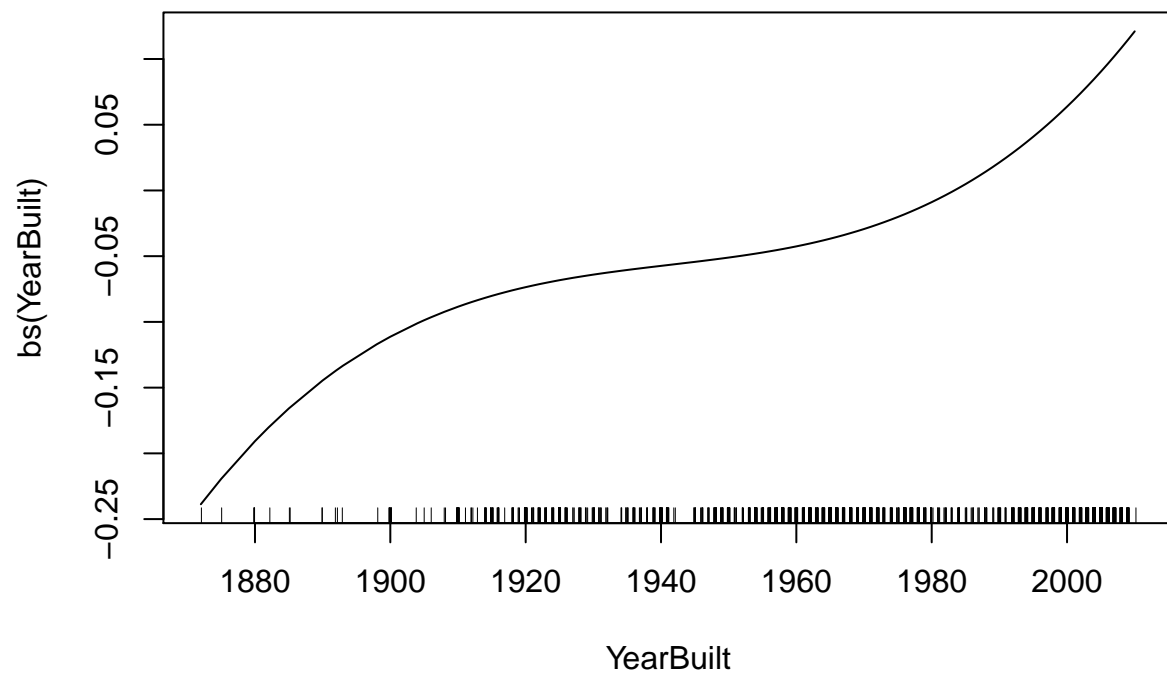


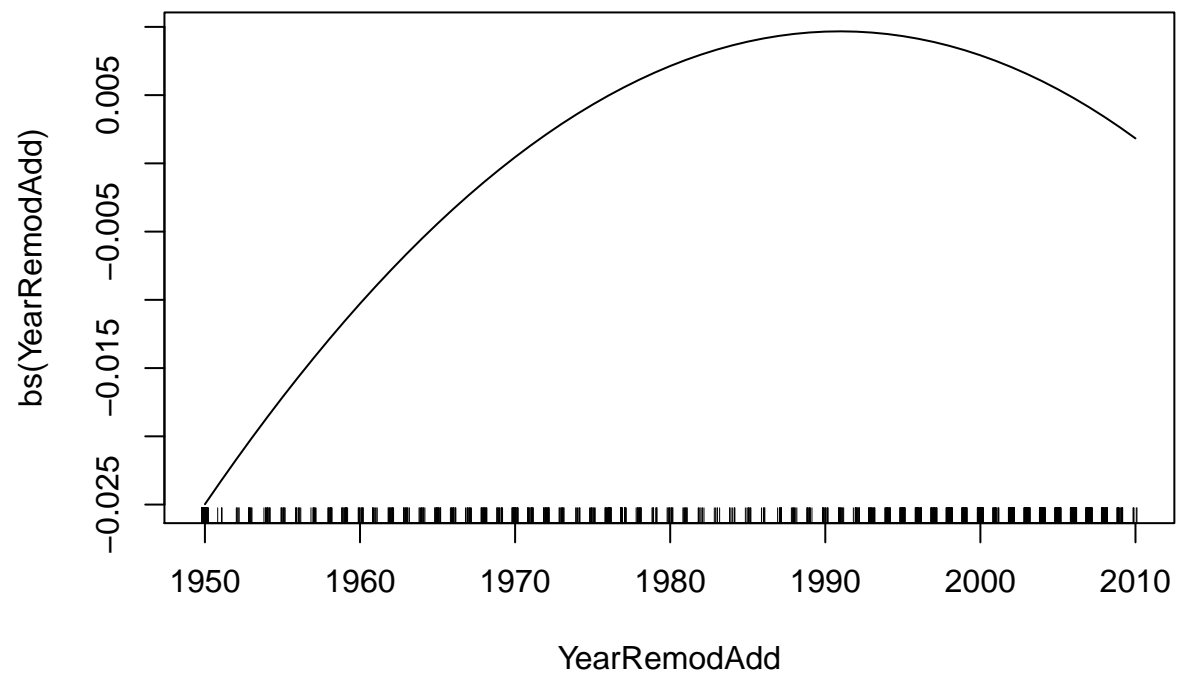


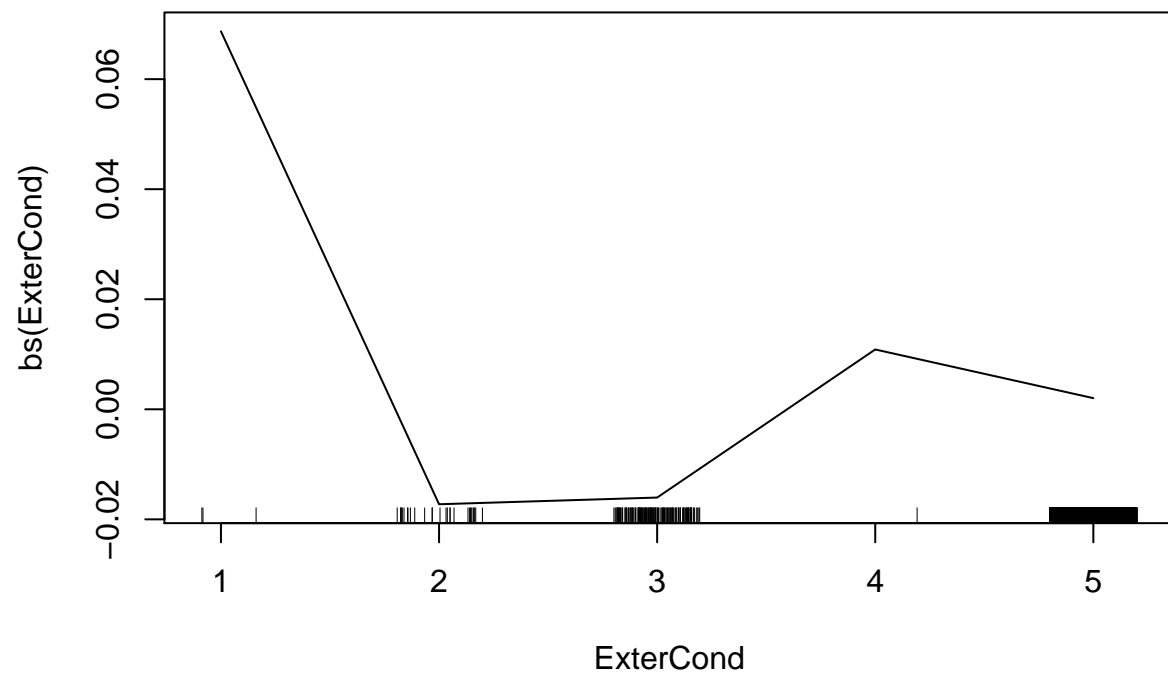


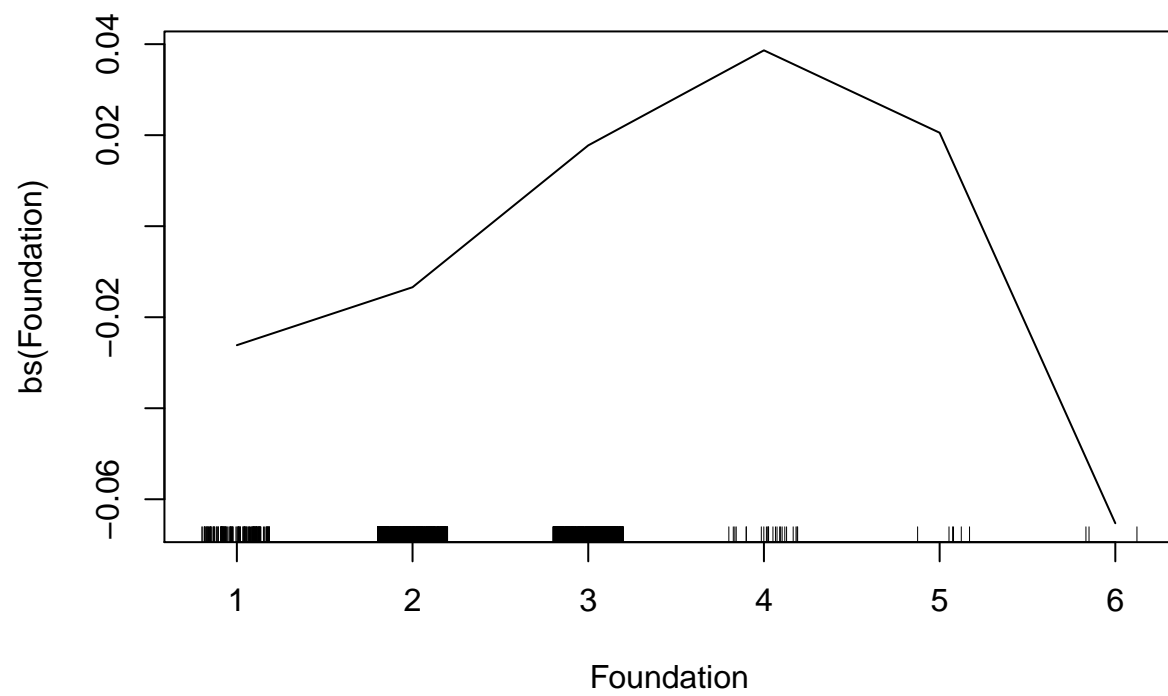


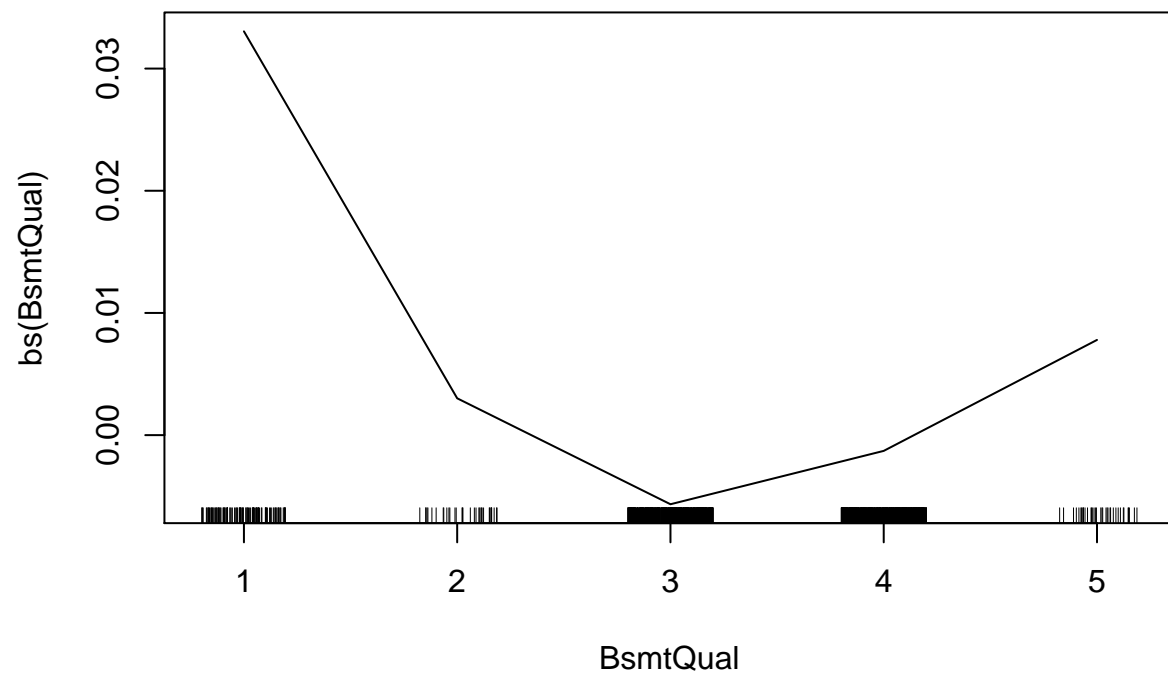


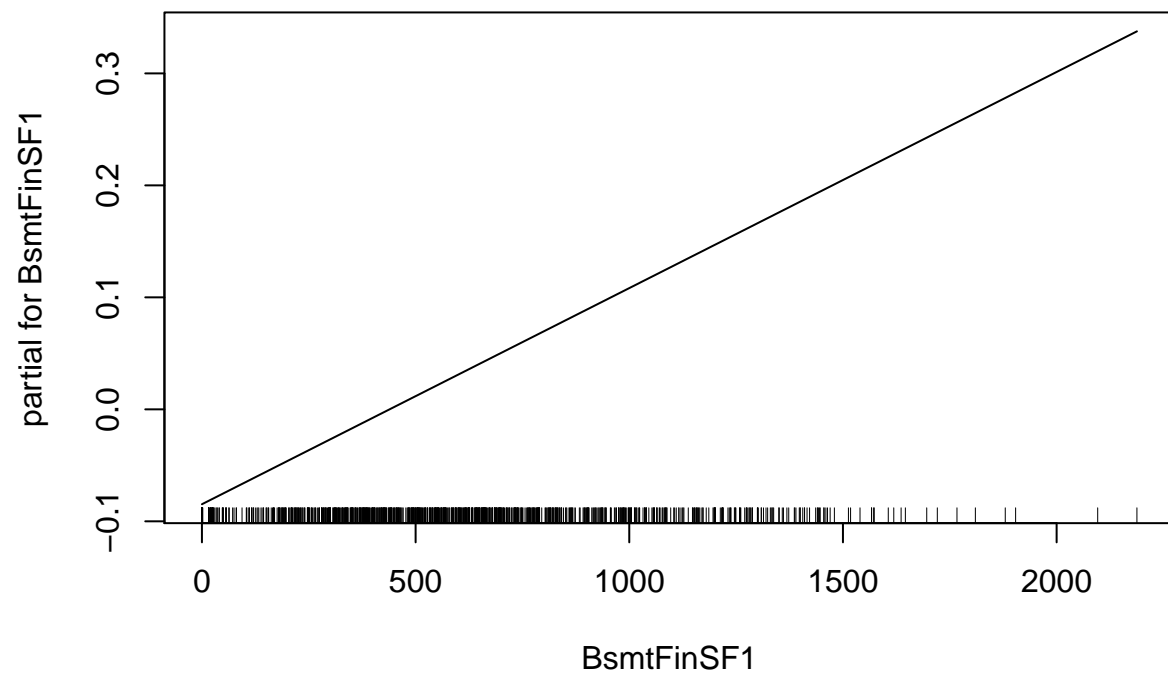


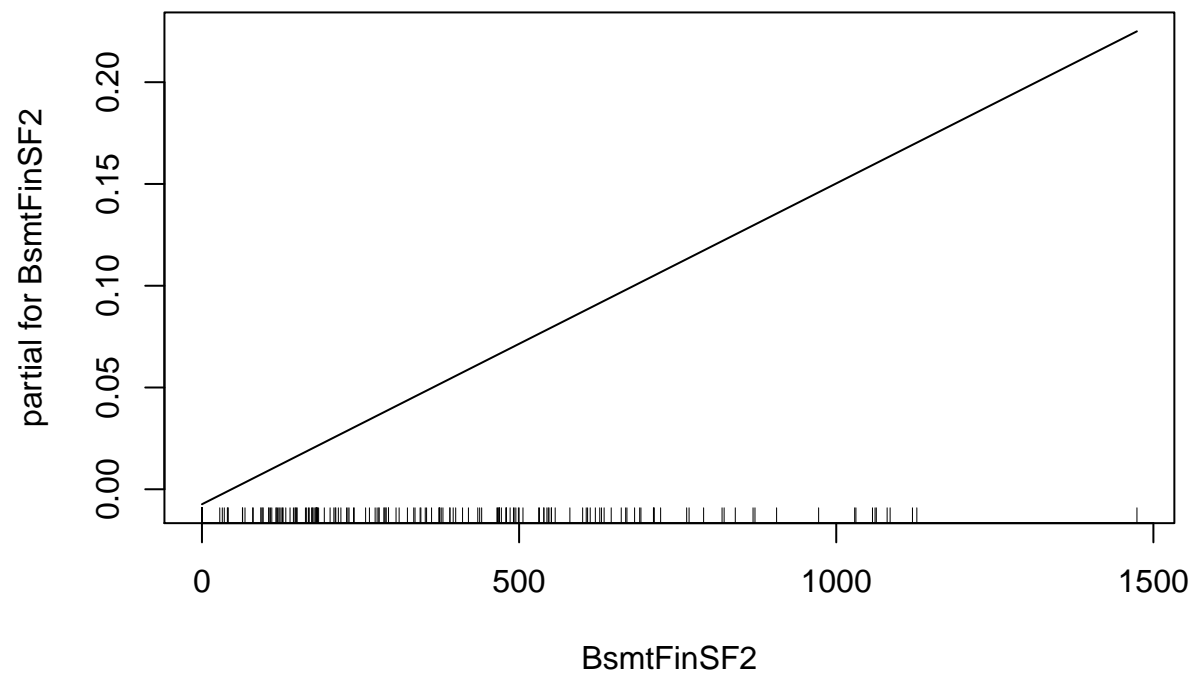


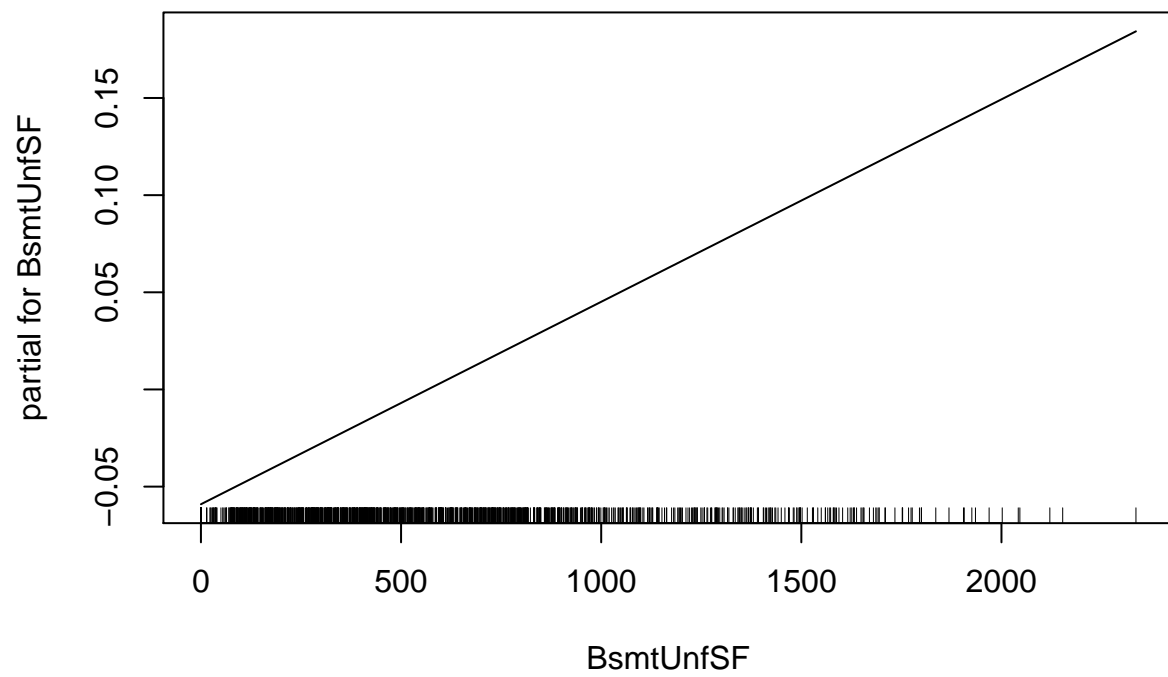


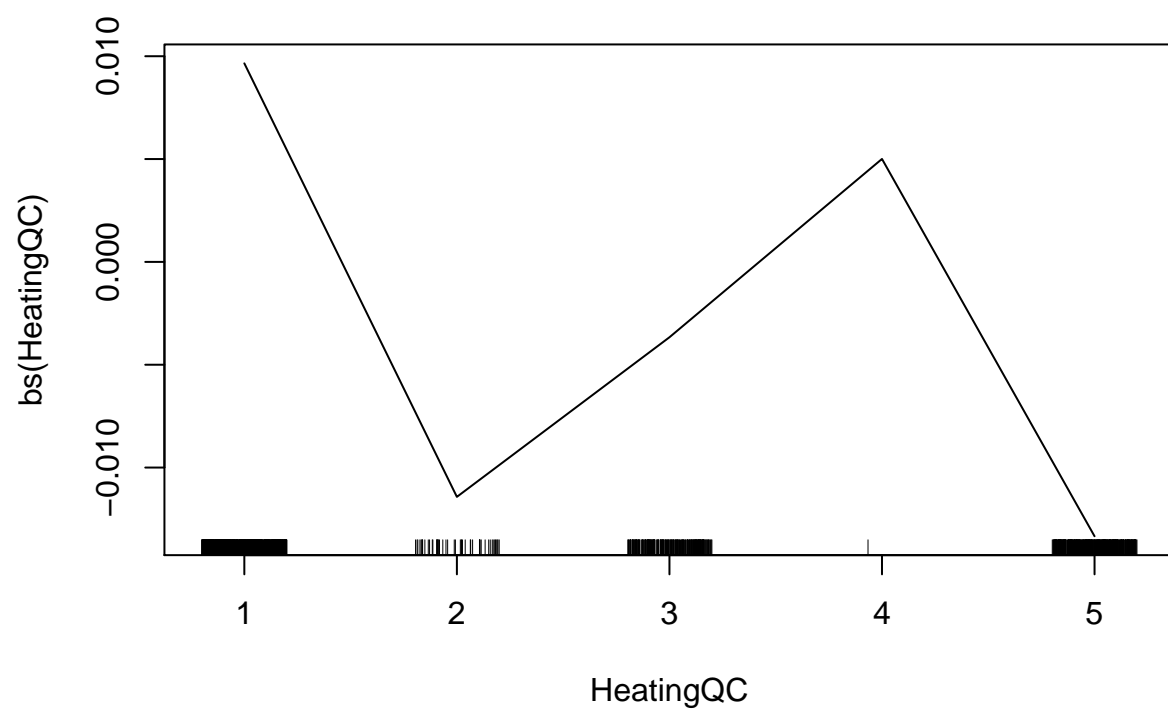


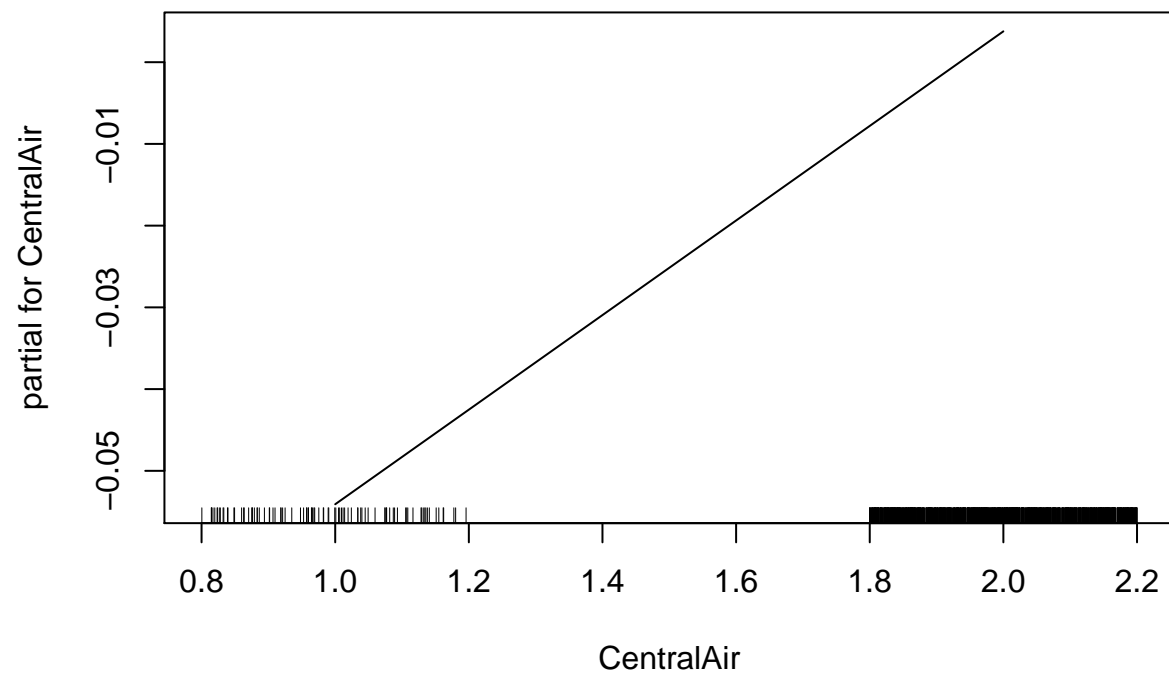


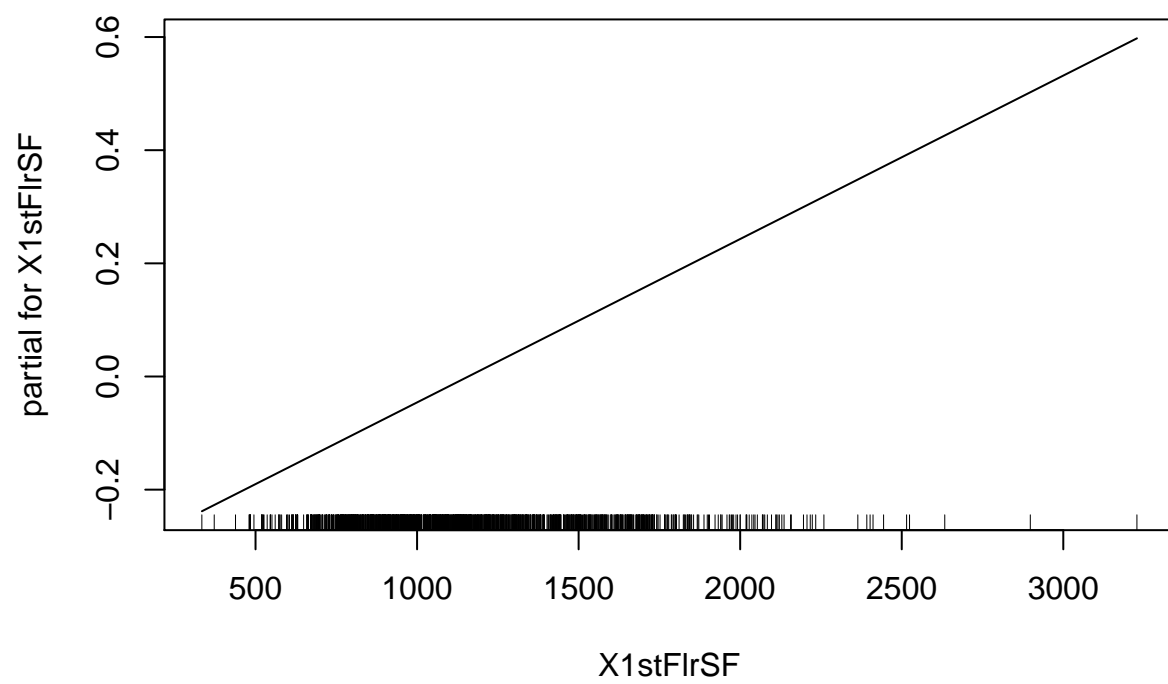


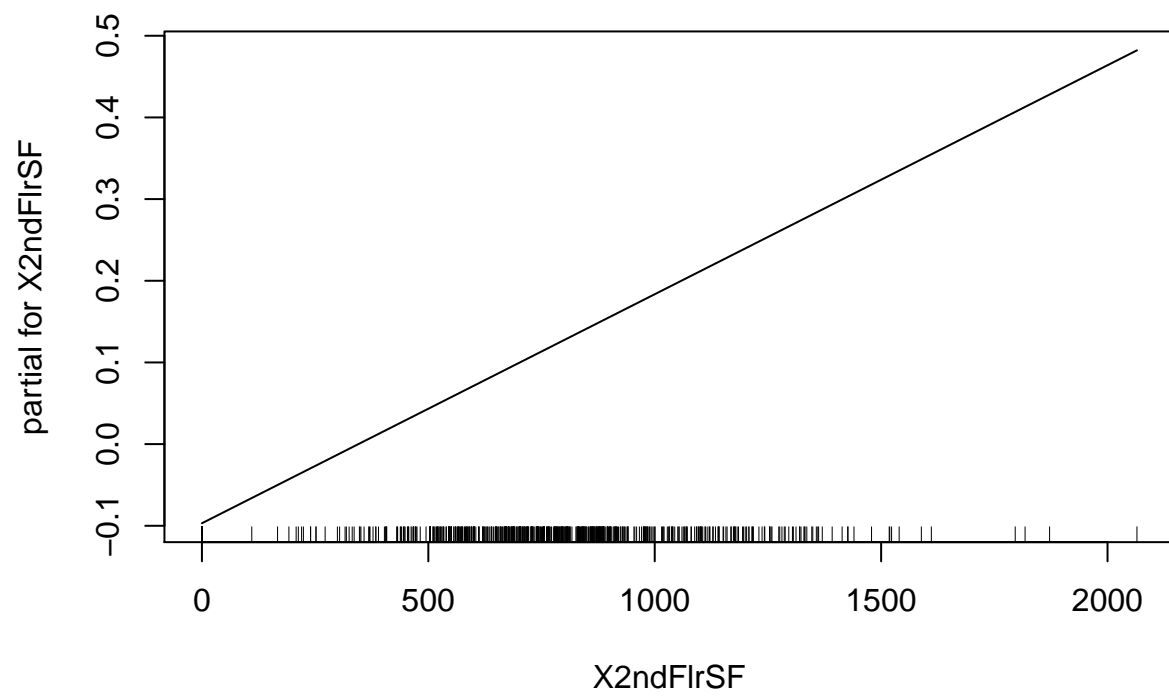


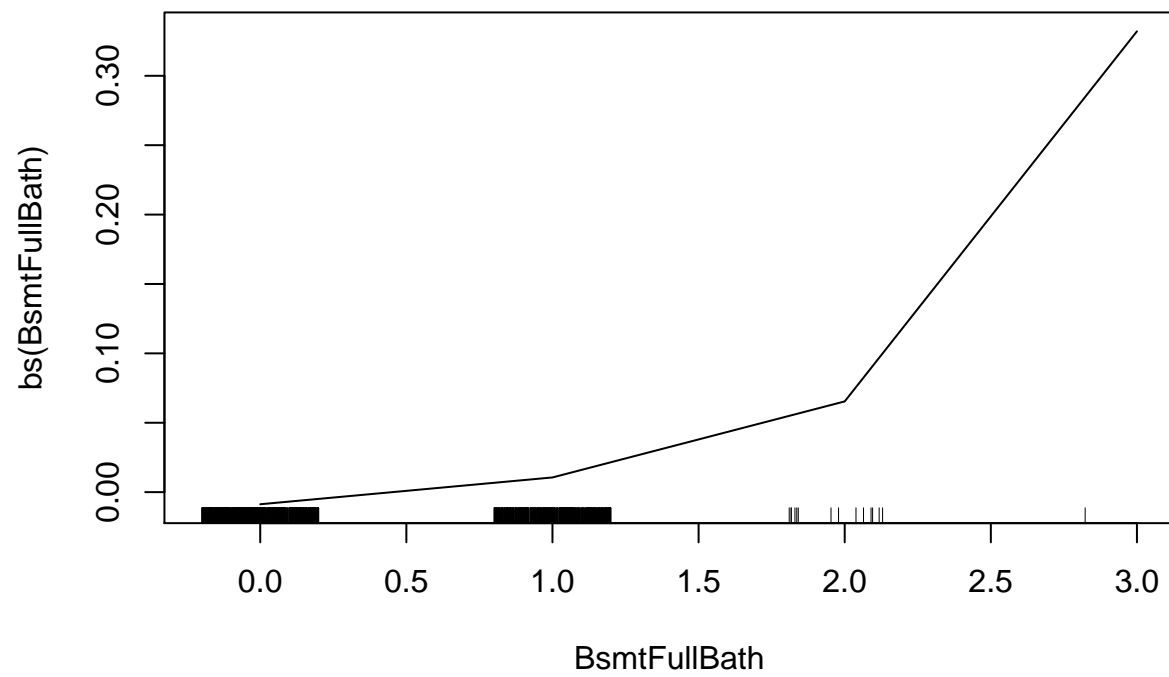


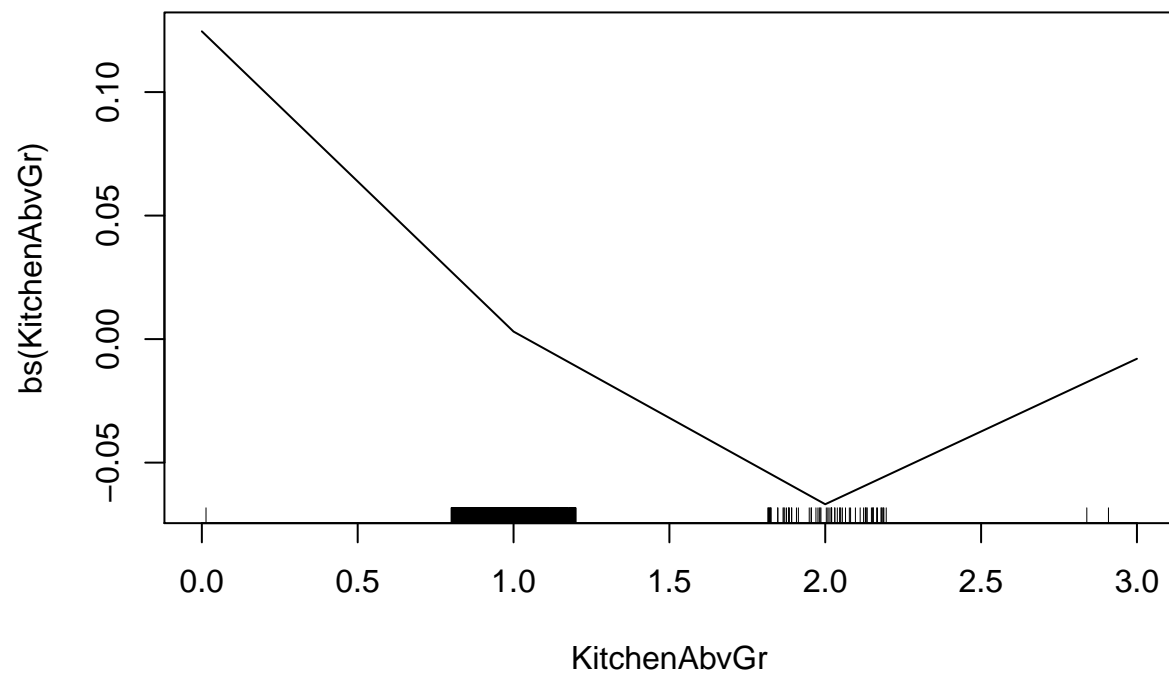


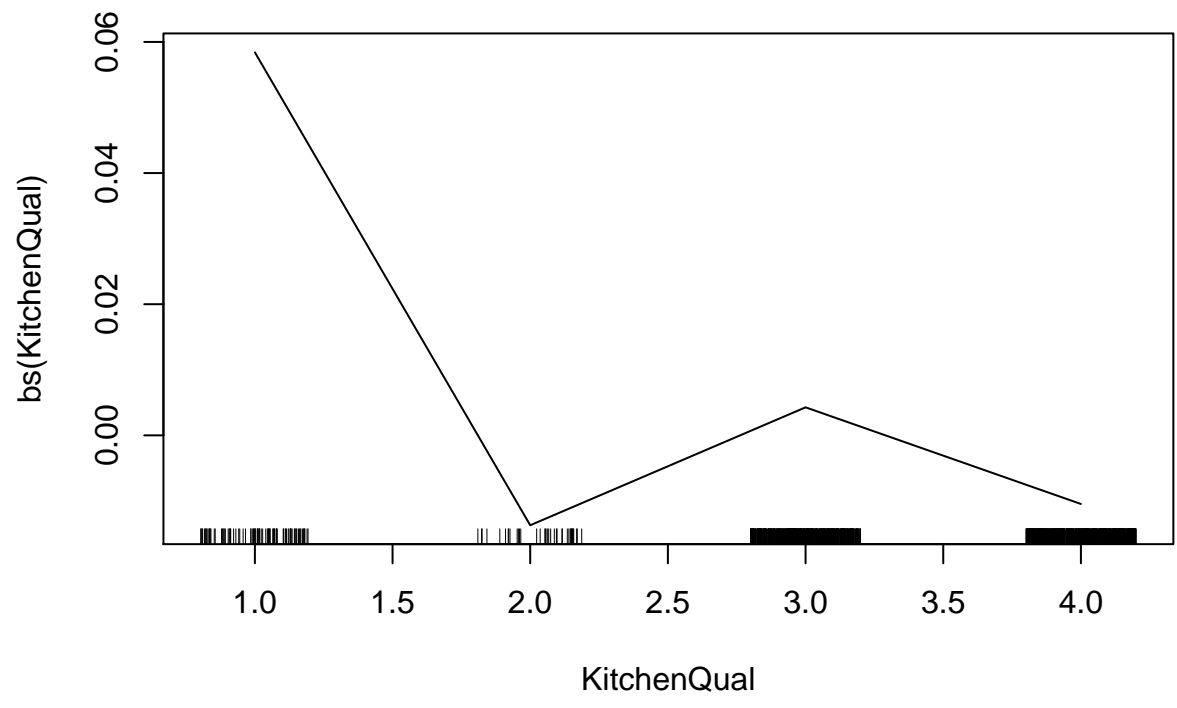


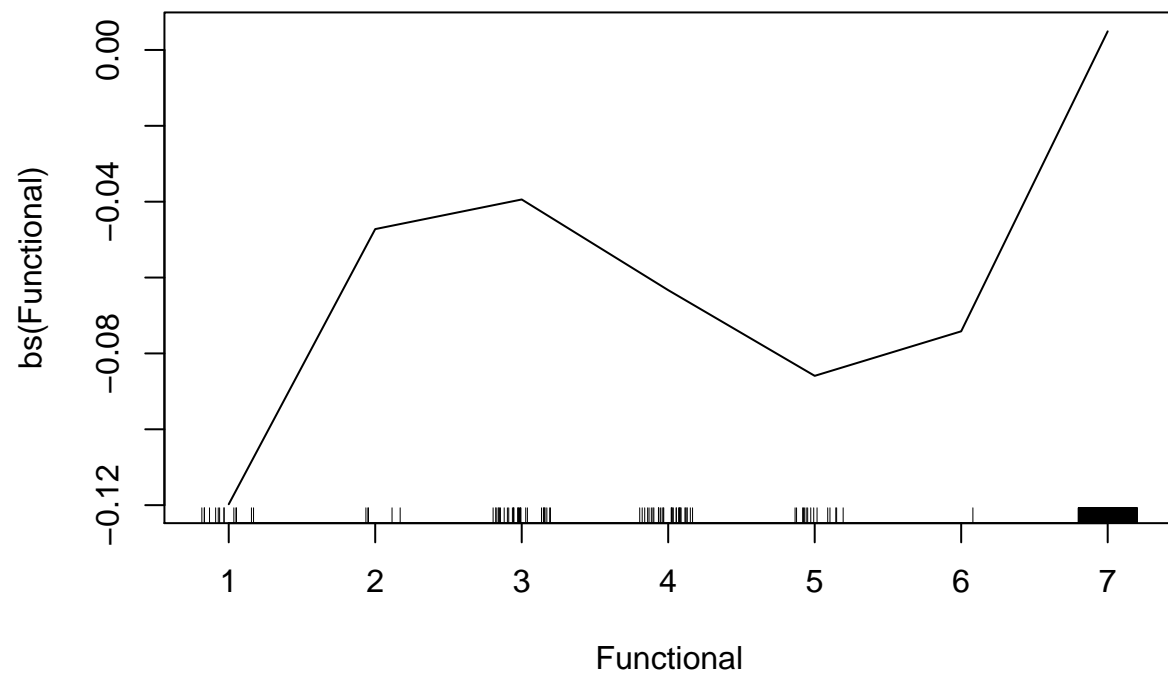


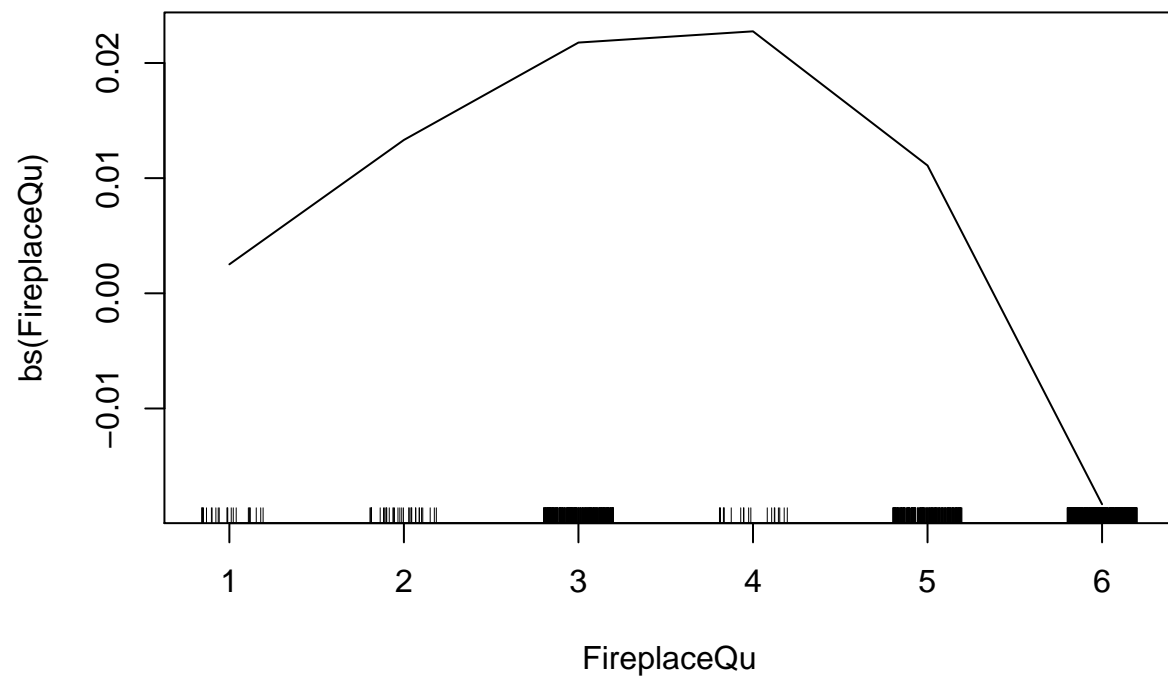


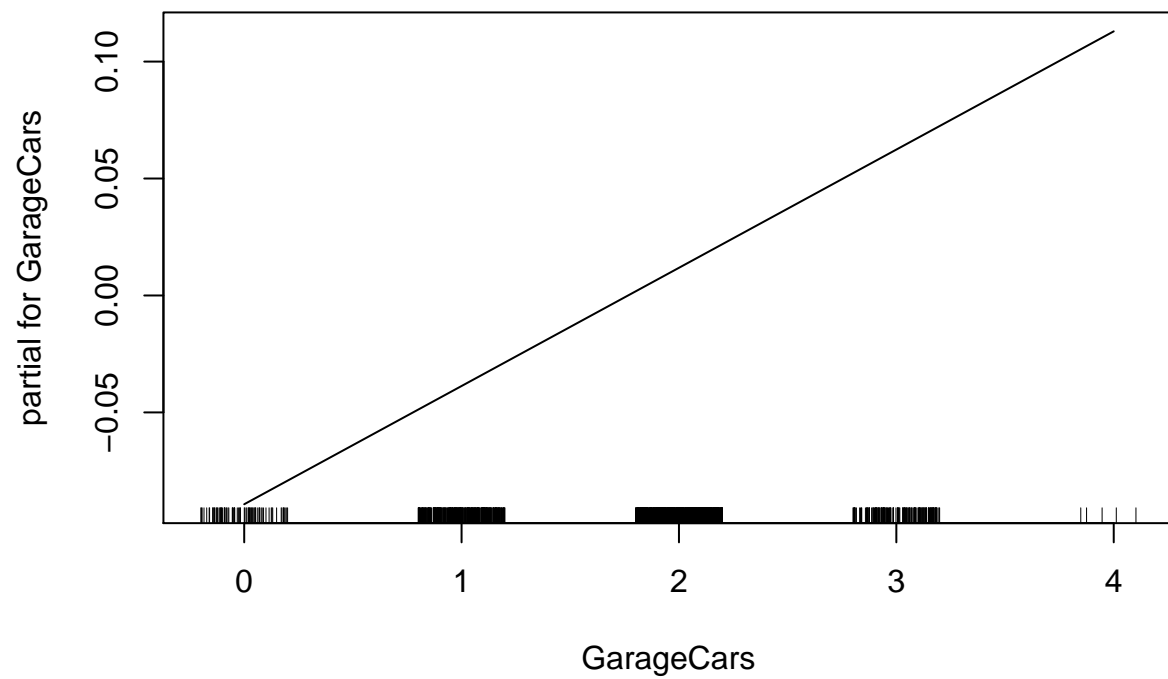


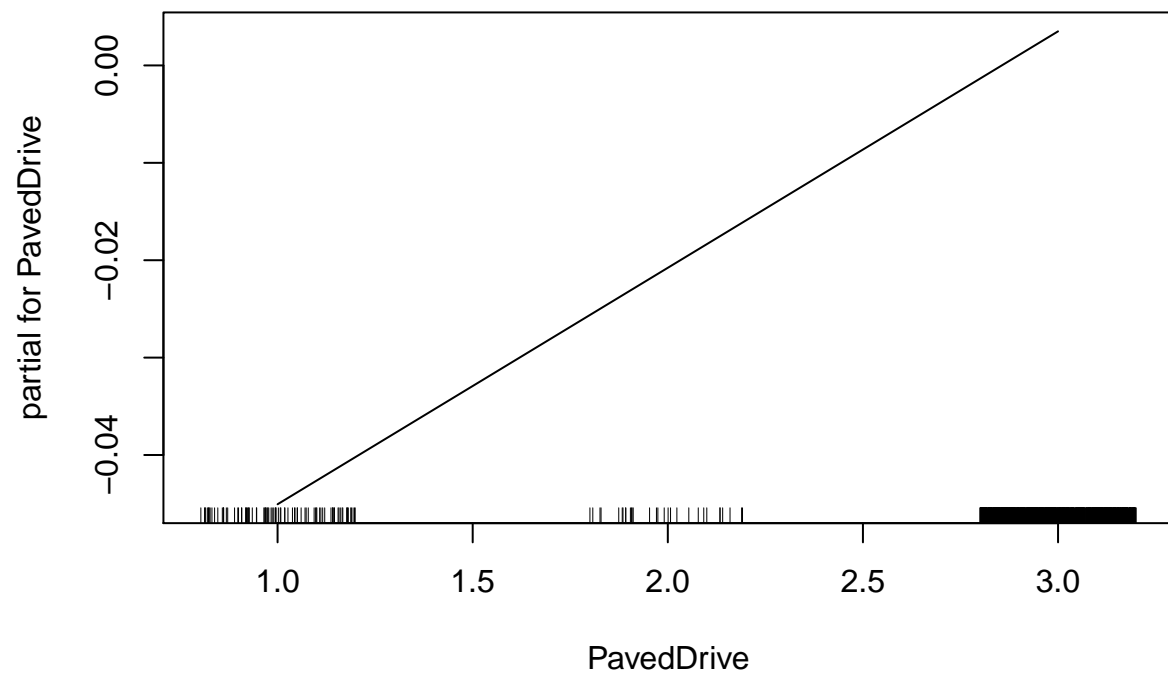


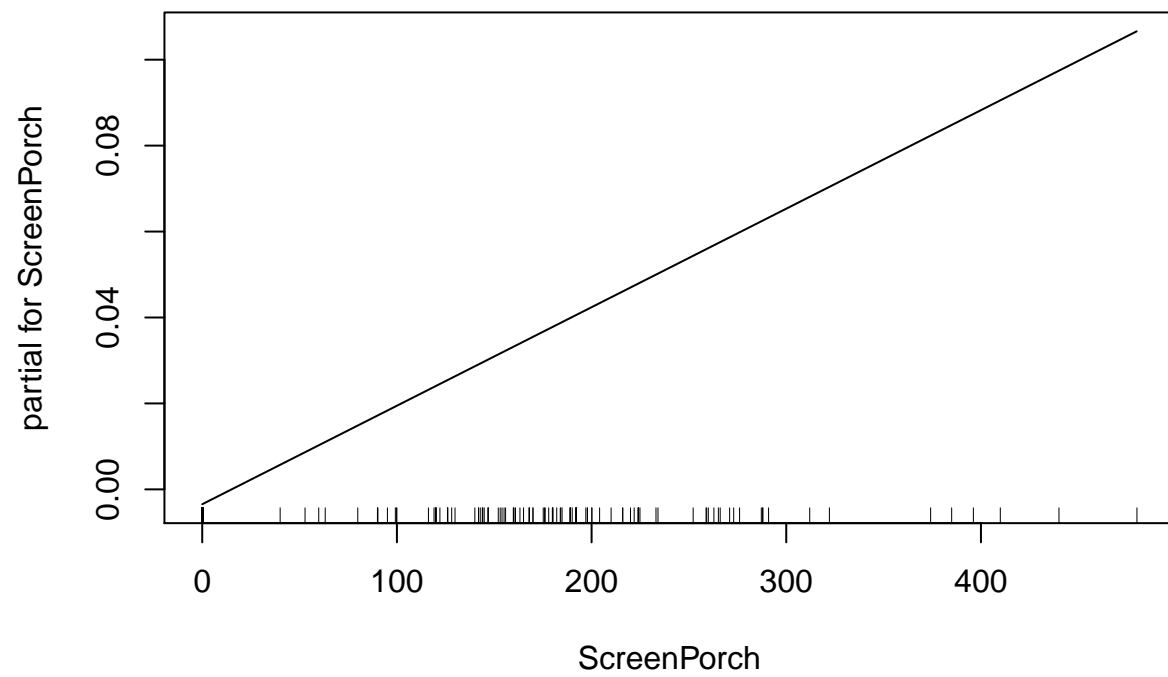


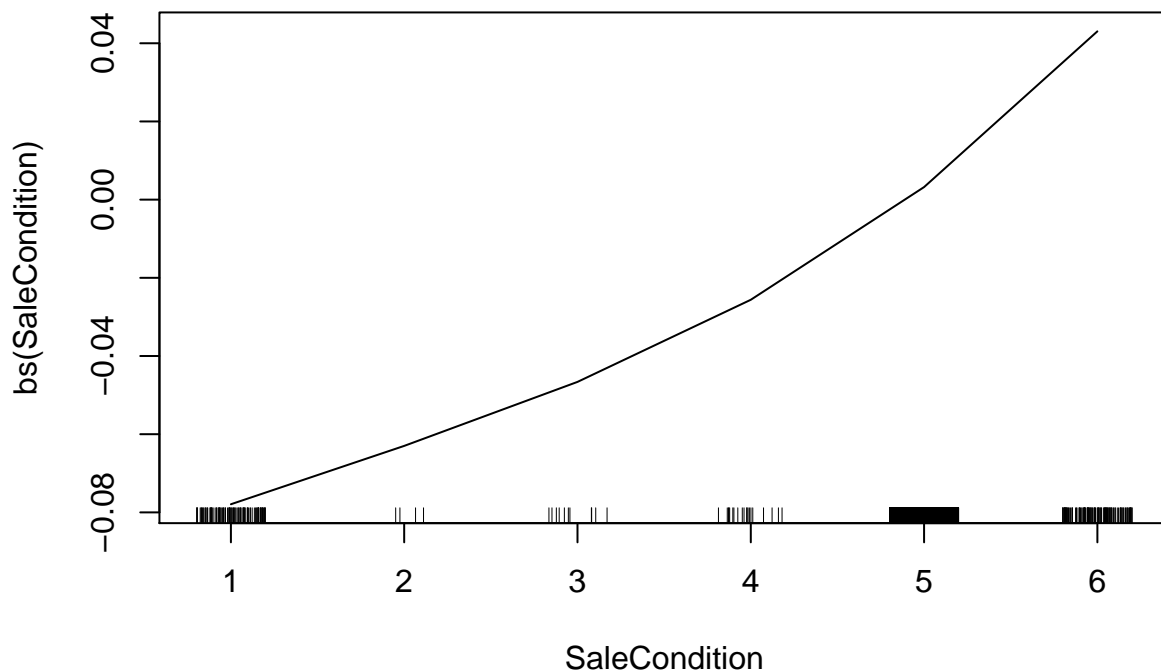










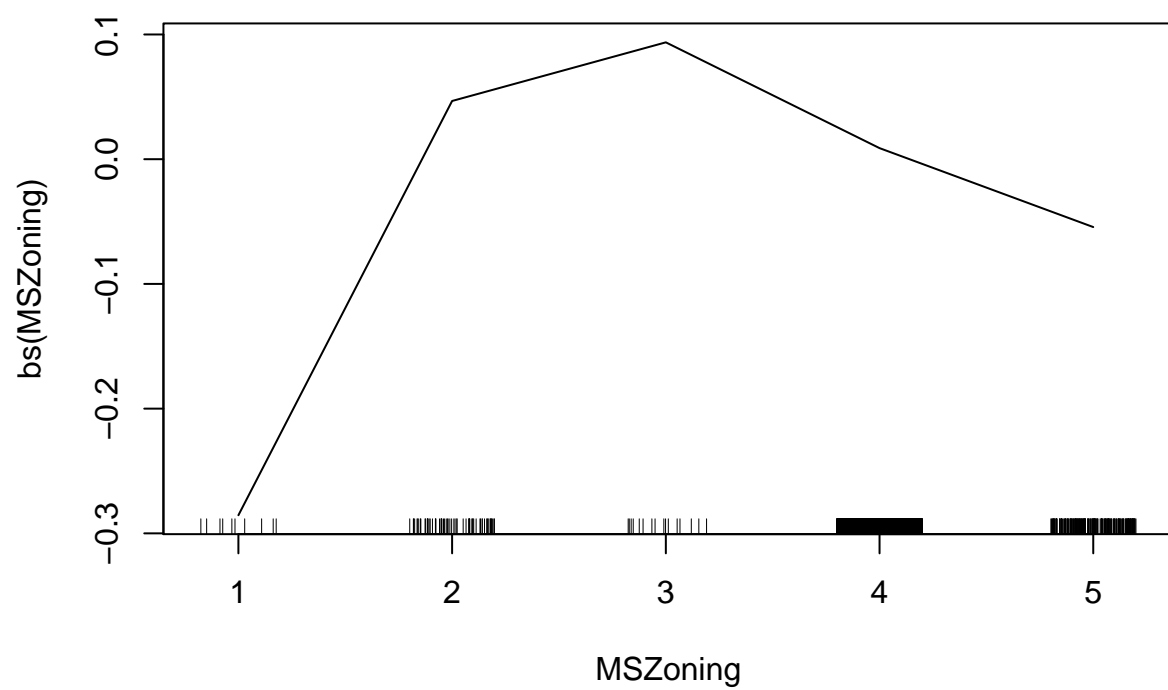


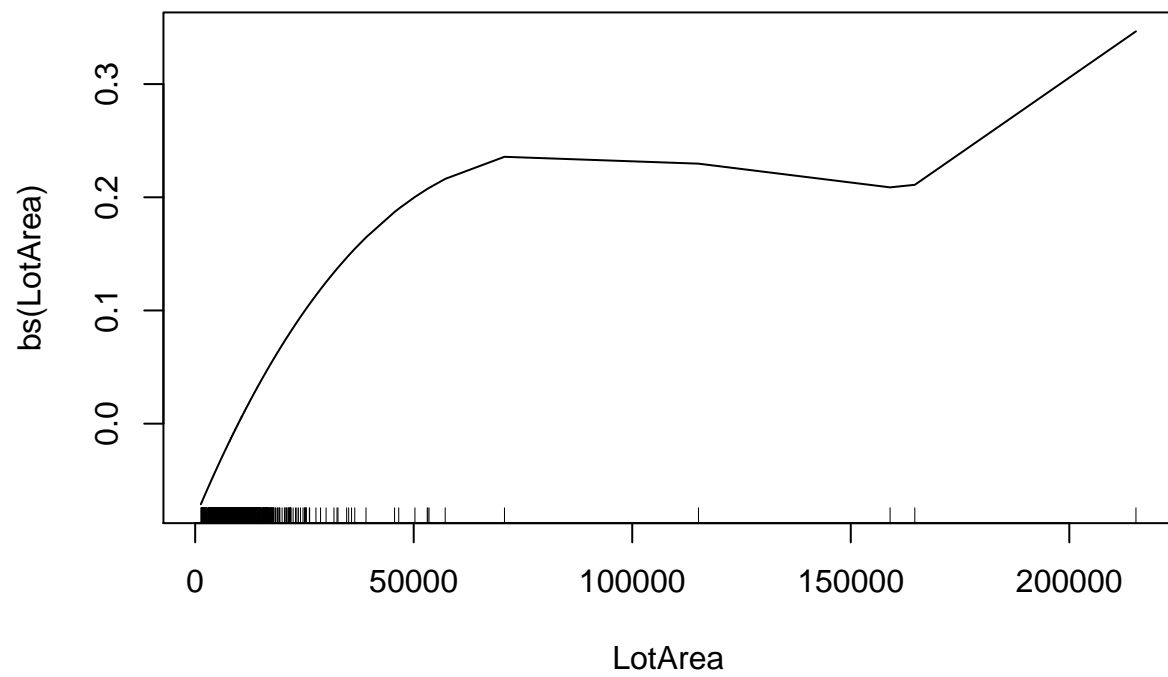
```
gam.preds <- newdata.predict.Gam(gam.lm, newdata = all.df.nolvg[1458:2916,])
gam.preds <- e^gam.preds
gam.submit <- data.frame(test.ids, gam.preds)
colnames(gam.submit) <- c("Id", "SalePrice")
#write.csv(gam.submit, "gam.csv")
```

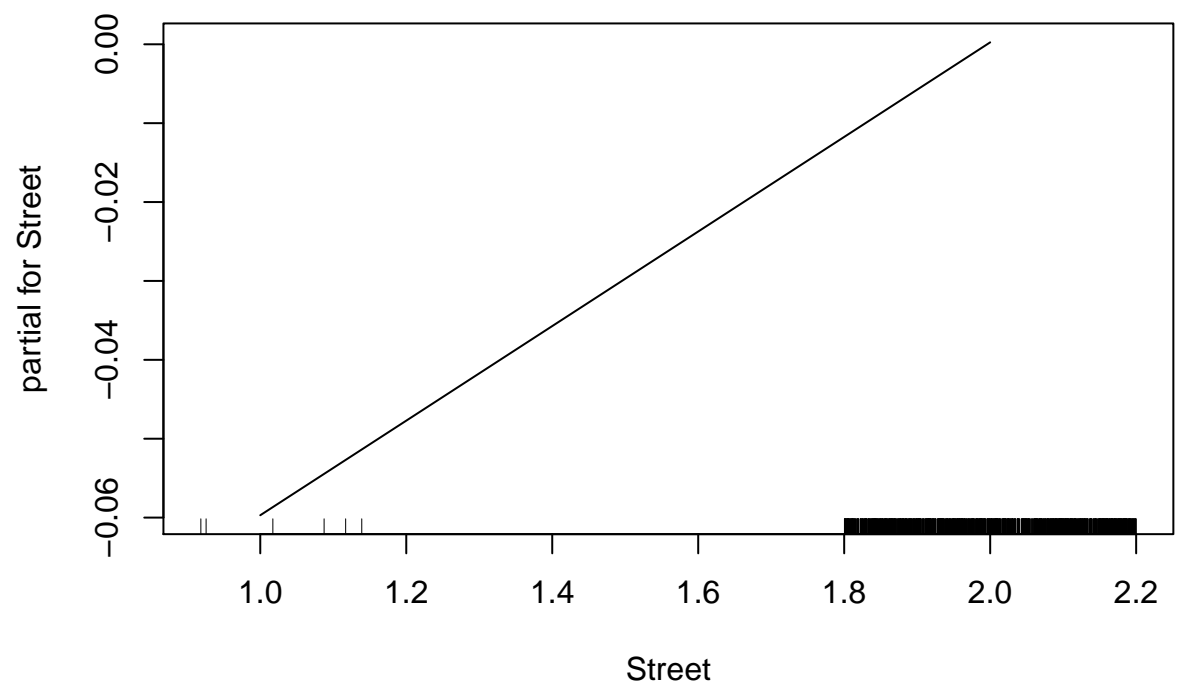
```
#Forward subset selection natural splines
coef(forw.sub, 26)
```

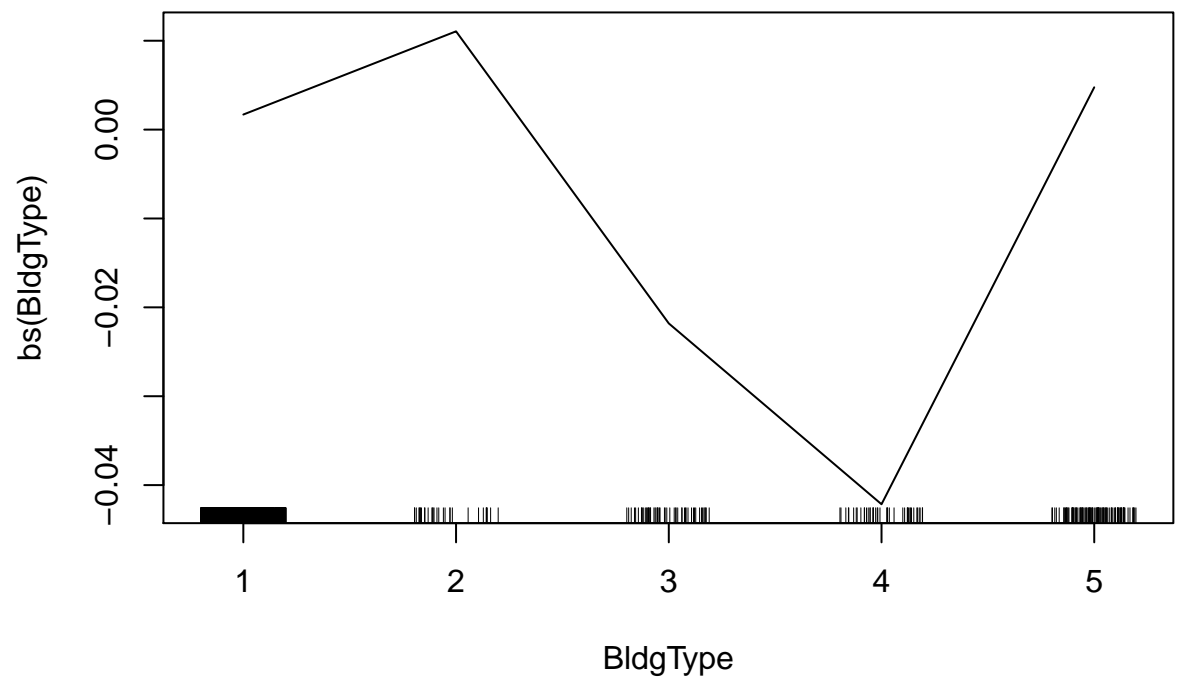
```
## (Intercept)      MSZoning      LotArea      Street      BldgType
## 4.300799e+00 -1.859880e-02  2.752828e-06  1.952257e-01 -1.466519e-02
## OverallQual   OverallCond   YearBuilt   YearRemodAdd   ExterCond
## 6.280007e-02  4.204364e-02  2.212196e-03  7.353405e-04  1.269971e-02
## BsmtQual      BsmtFinSF1     BsmtFinSF2     BsmtUnfSF      HeatingQC
## -1.646464e-02  1.587461e-04  1.235607e-04  7.641819e-05  -8.619712e-03
## CentralAir    X1stFlrSF      X2ndFlrSF      BsmtFullBath   KitchenAbvGr
## 7.354042e-02  3.315885e-04  2.827067e-04  2.782892e-02  -4.729417e-02
## KitchenQual   Functional     FireplaceQu     GarageCars     PavedDrive
## -2.074175e-02  1.871924e-02  -1.287291e-02  5.214347e-02  2.133373e-02
## ScreenPorch   SaleCondition
## 2.663956e-04  2.573918e-02
```

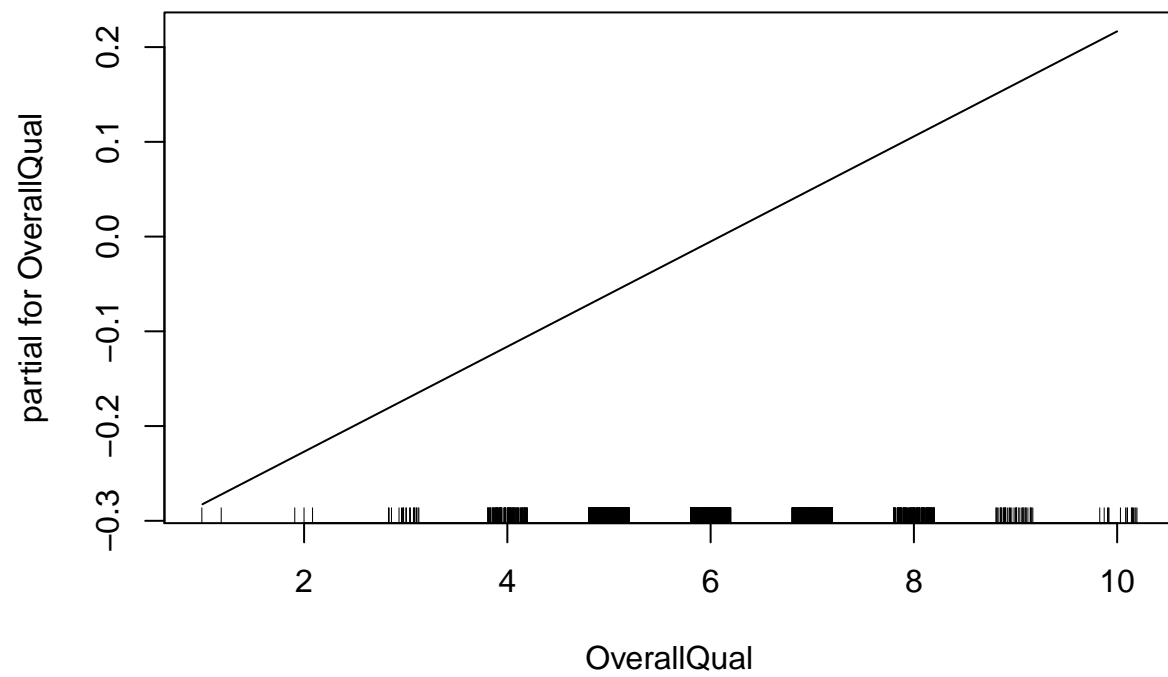
```
gam.lm <- lm(SalePrice~bs(MSZoning)+bs(LotArea)+Street+bs(BldgType)+OverallQual+bs(OverallCond)+bs(YearBuilt))
#summary(gam.lm)
plot.Gam(gam.lm)
```

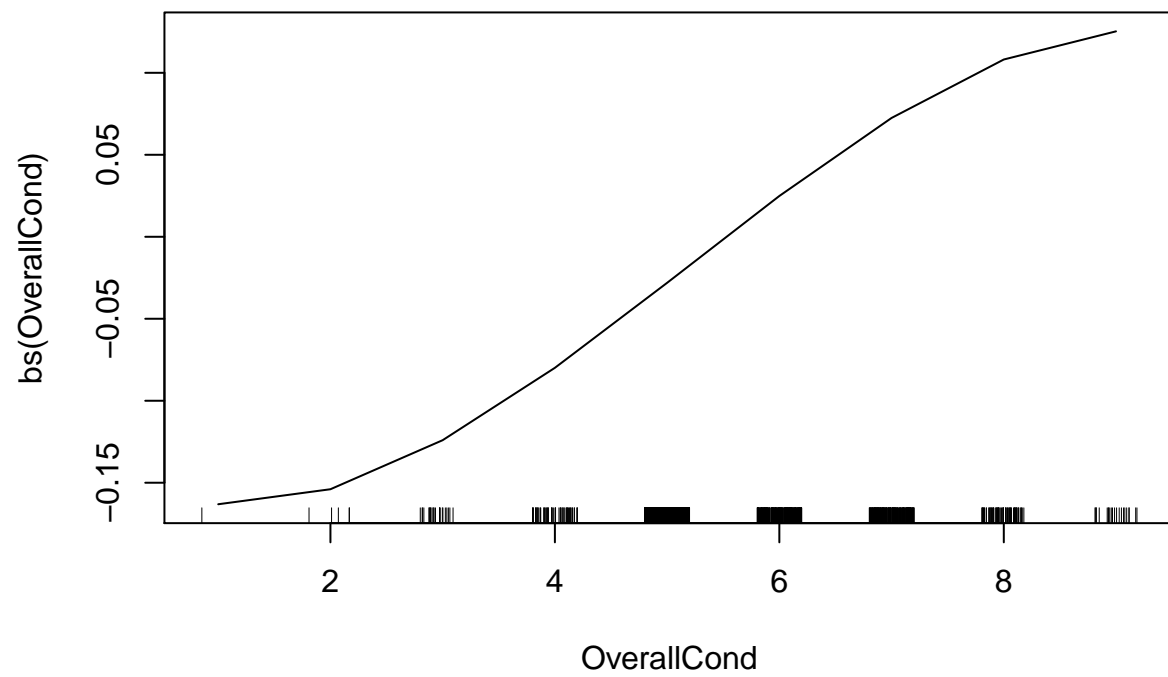


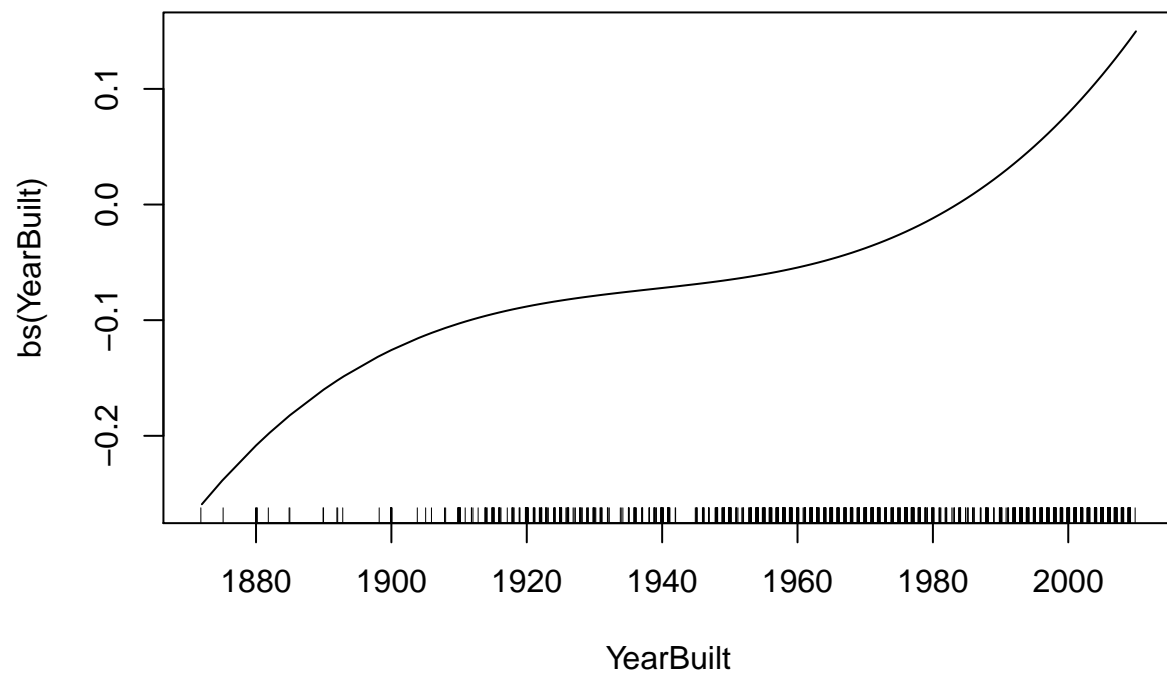


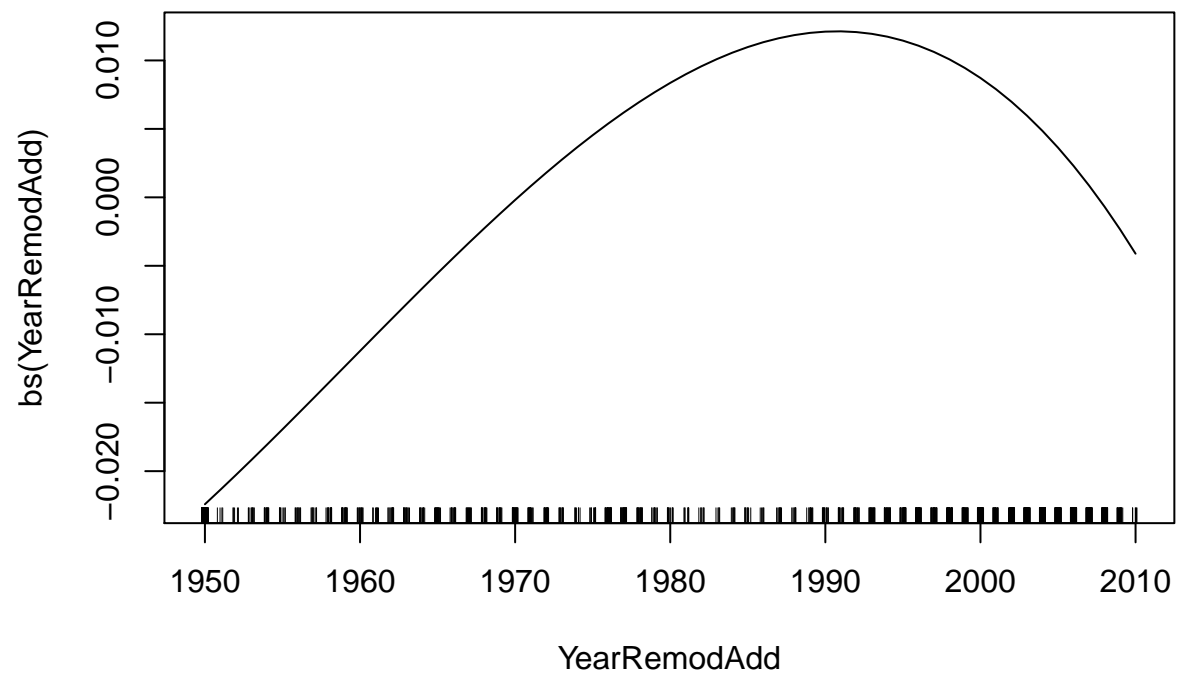


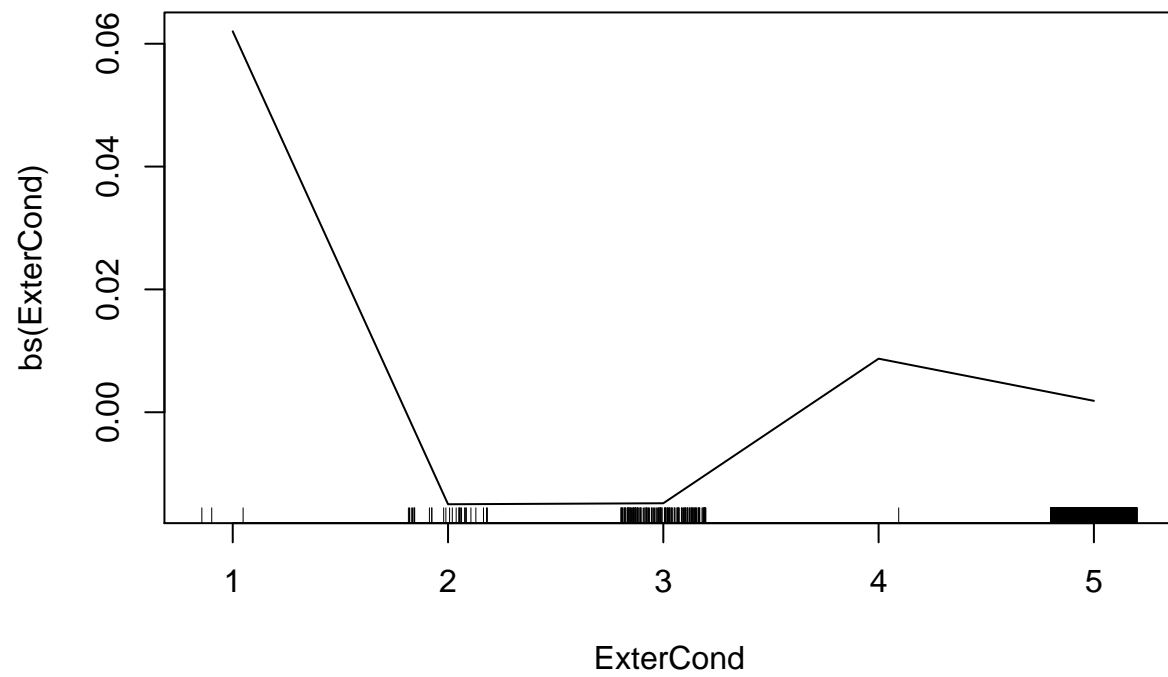


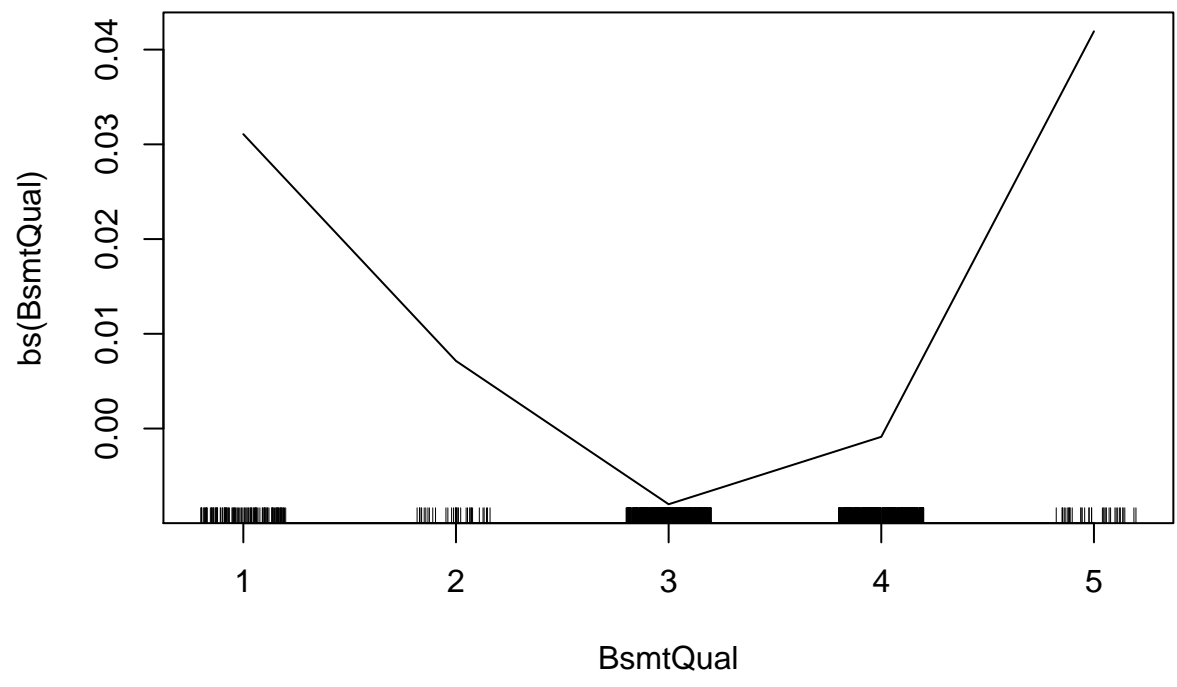


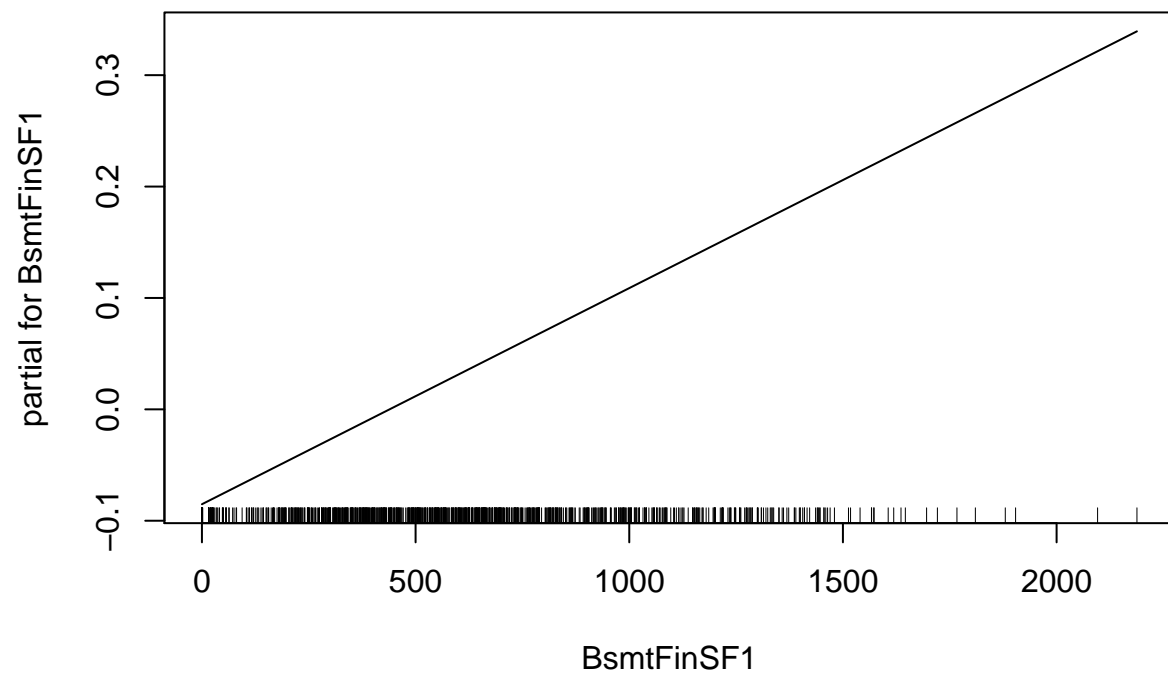


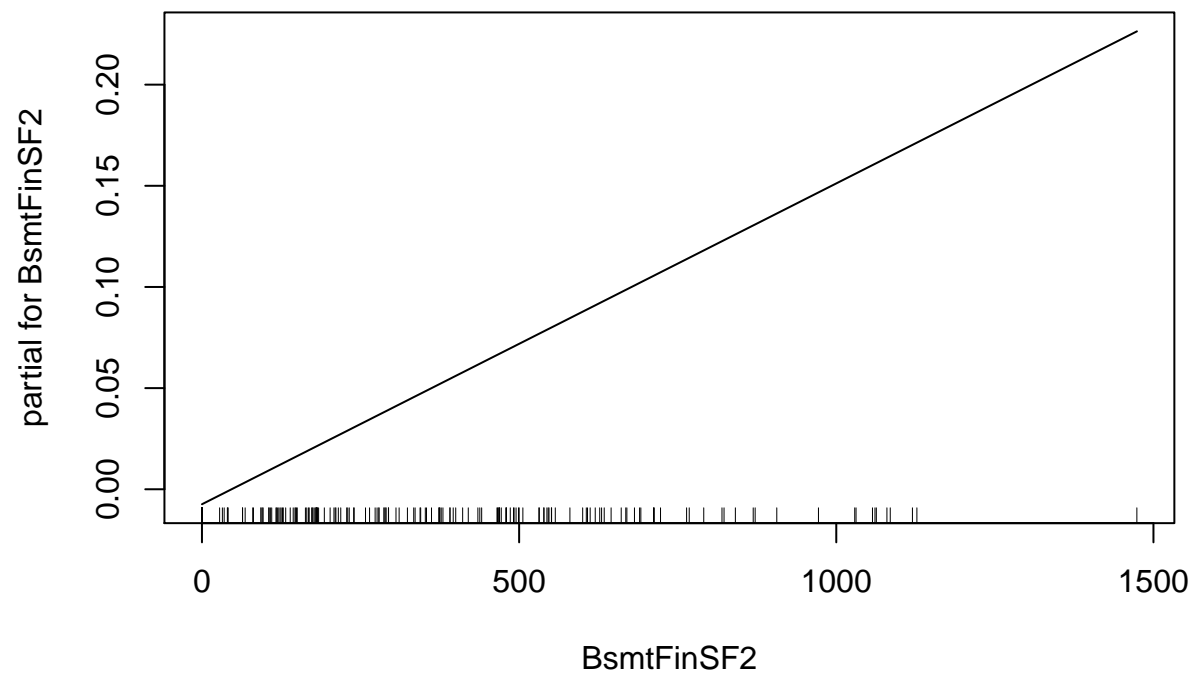


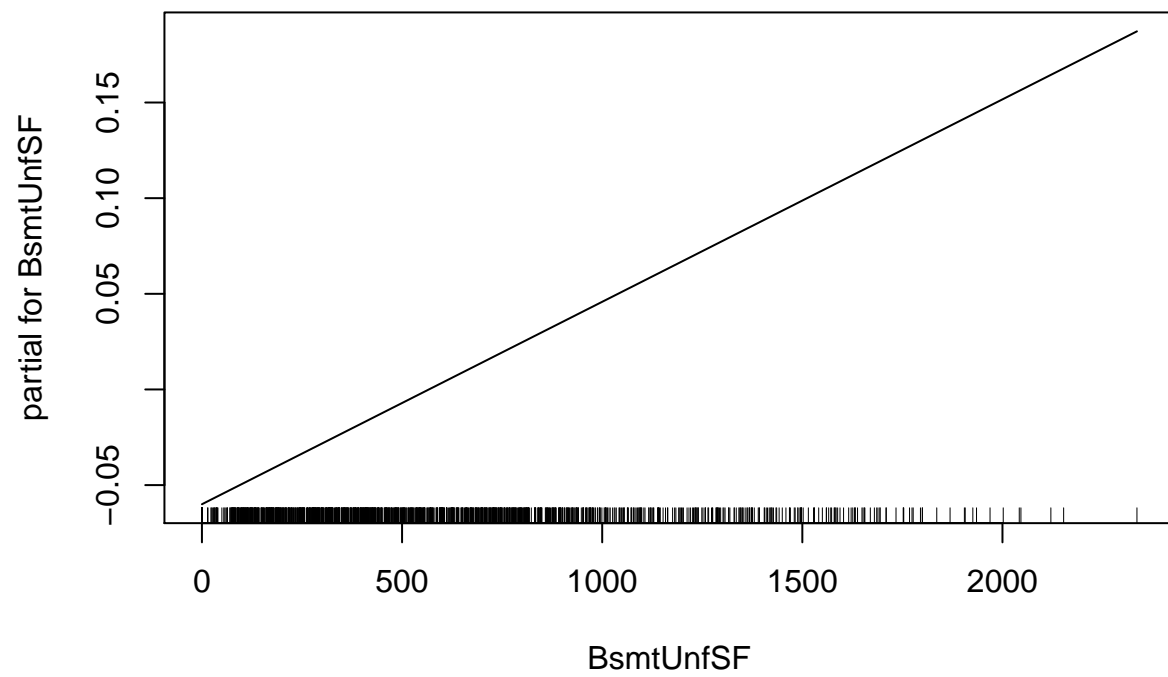


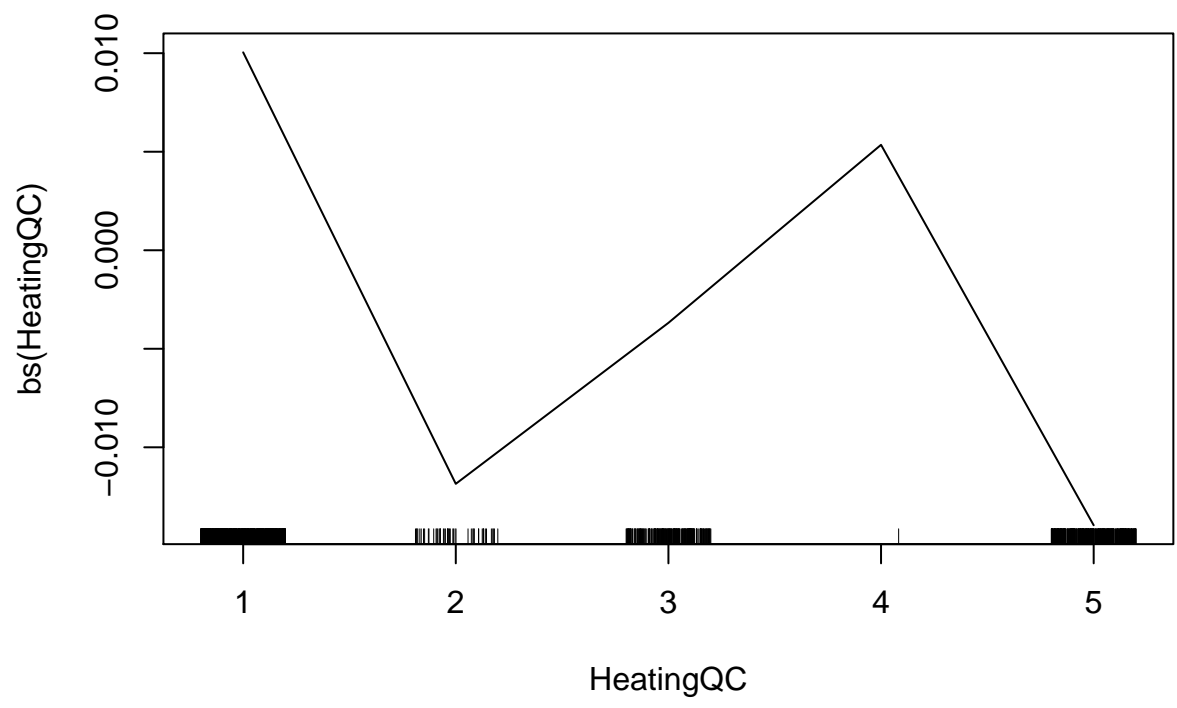


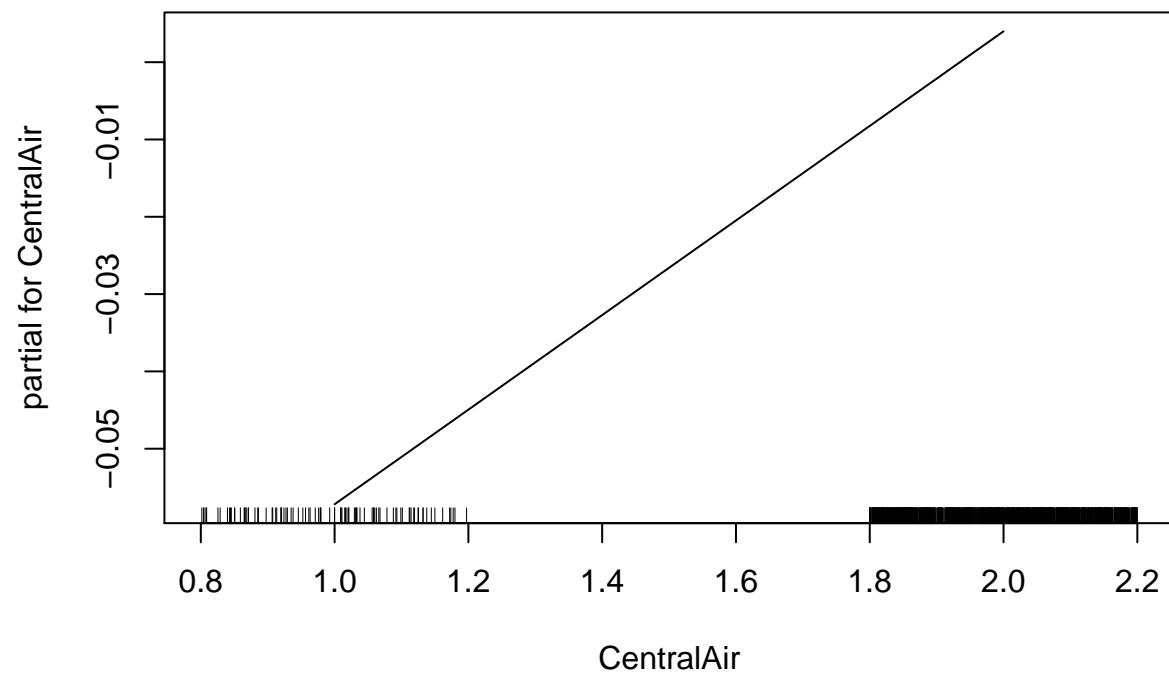


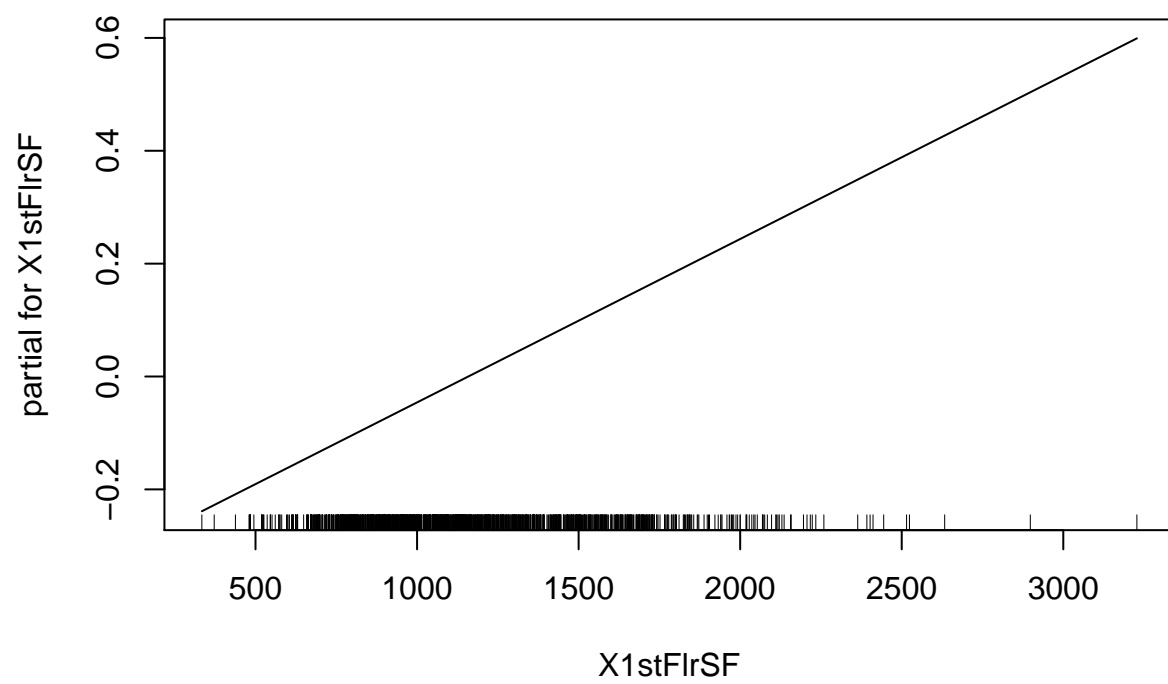


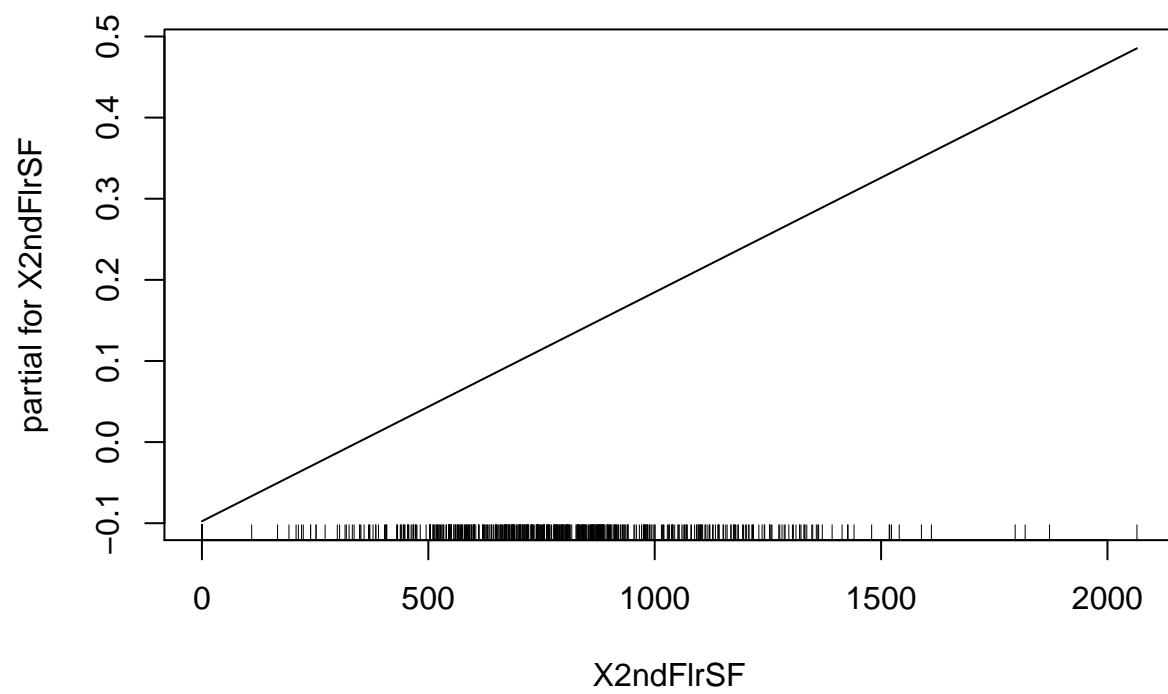


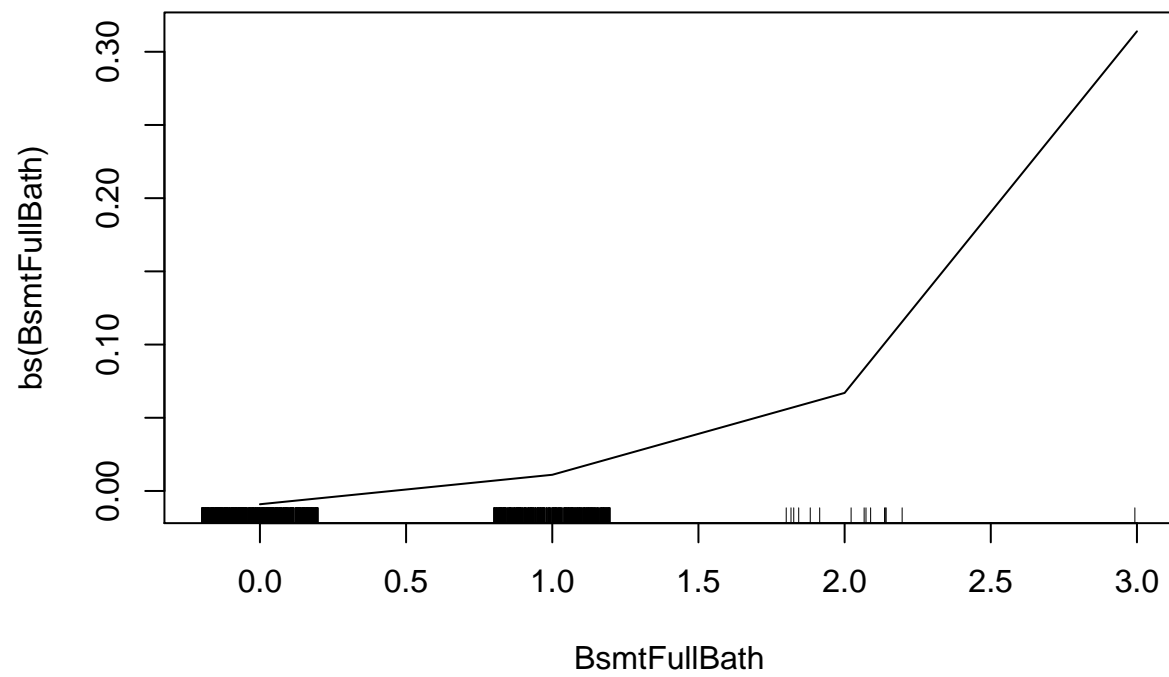


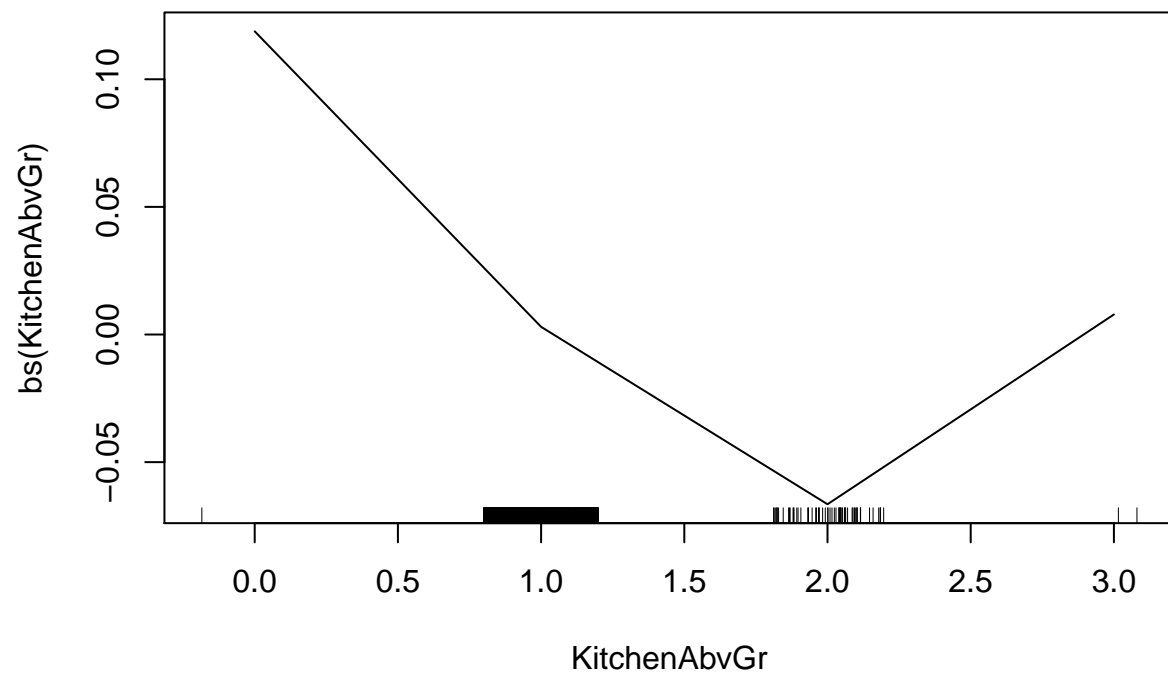


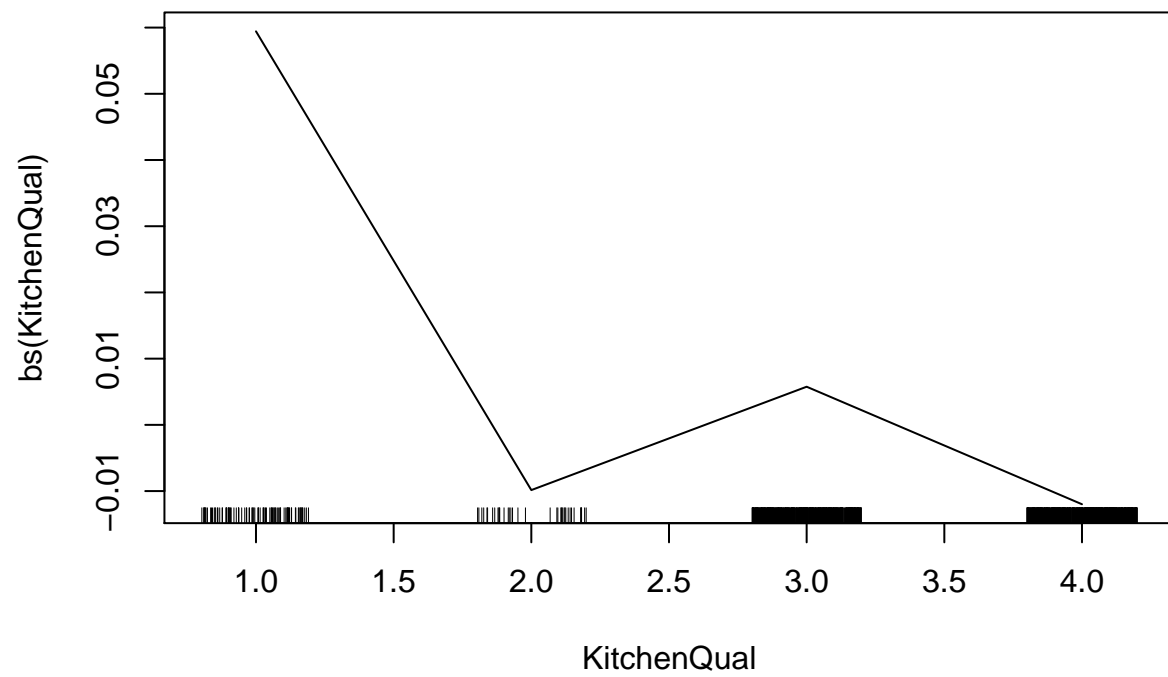


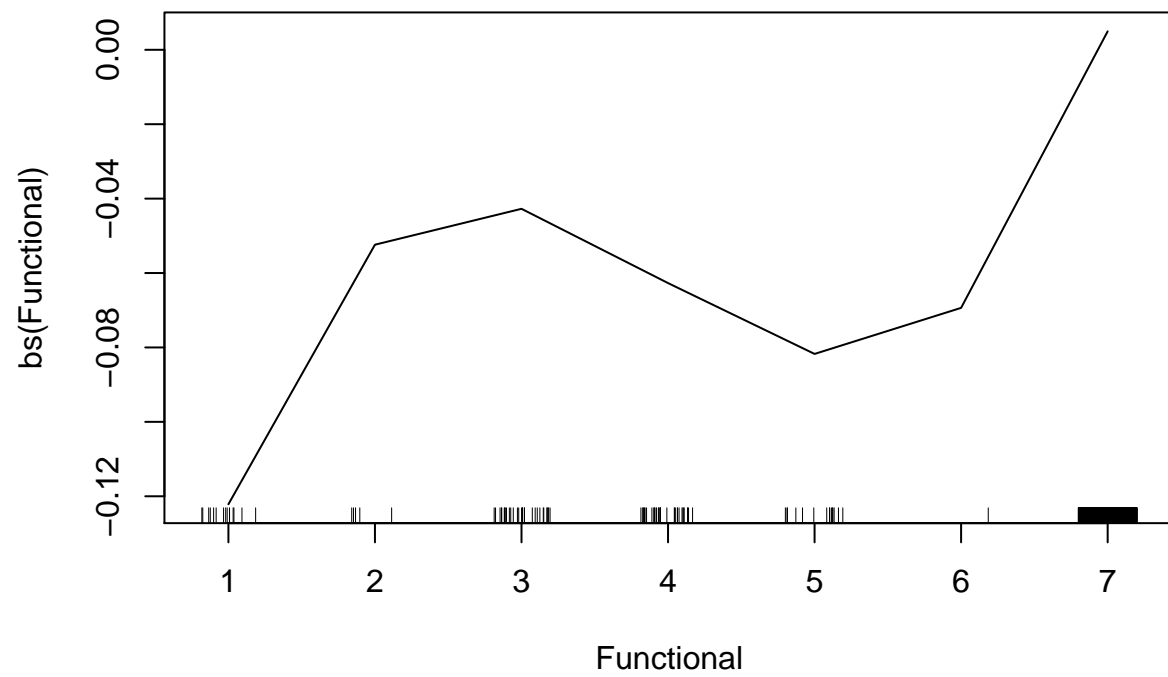


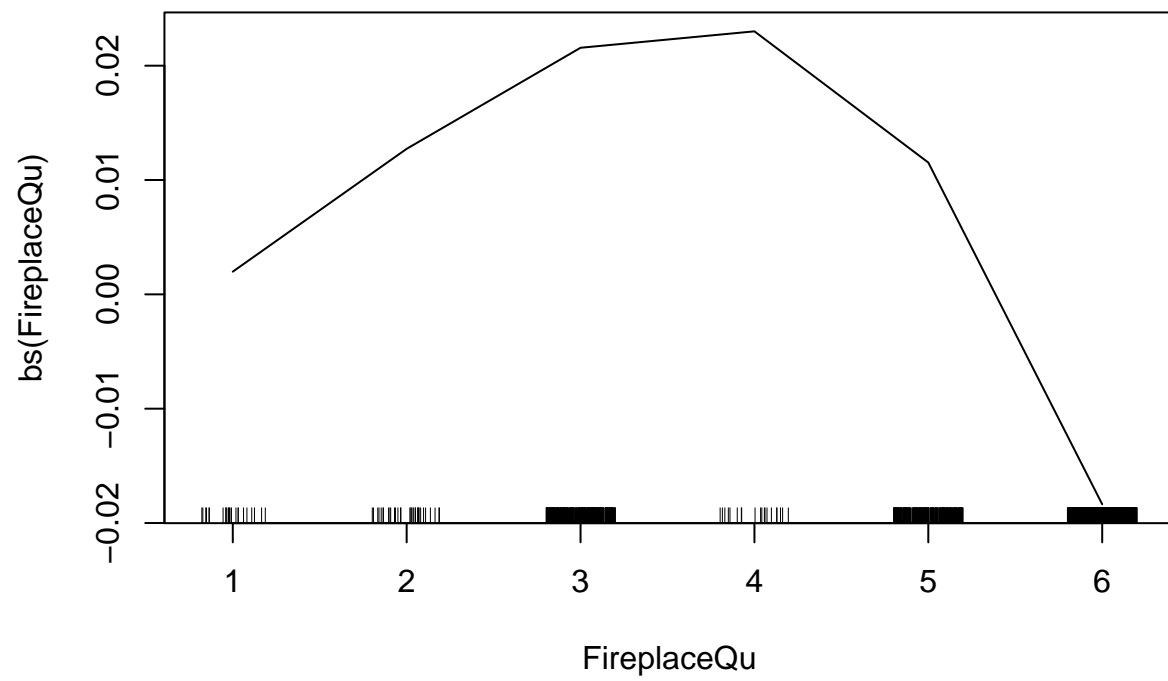


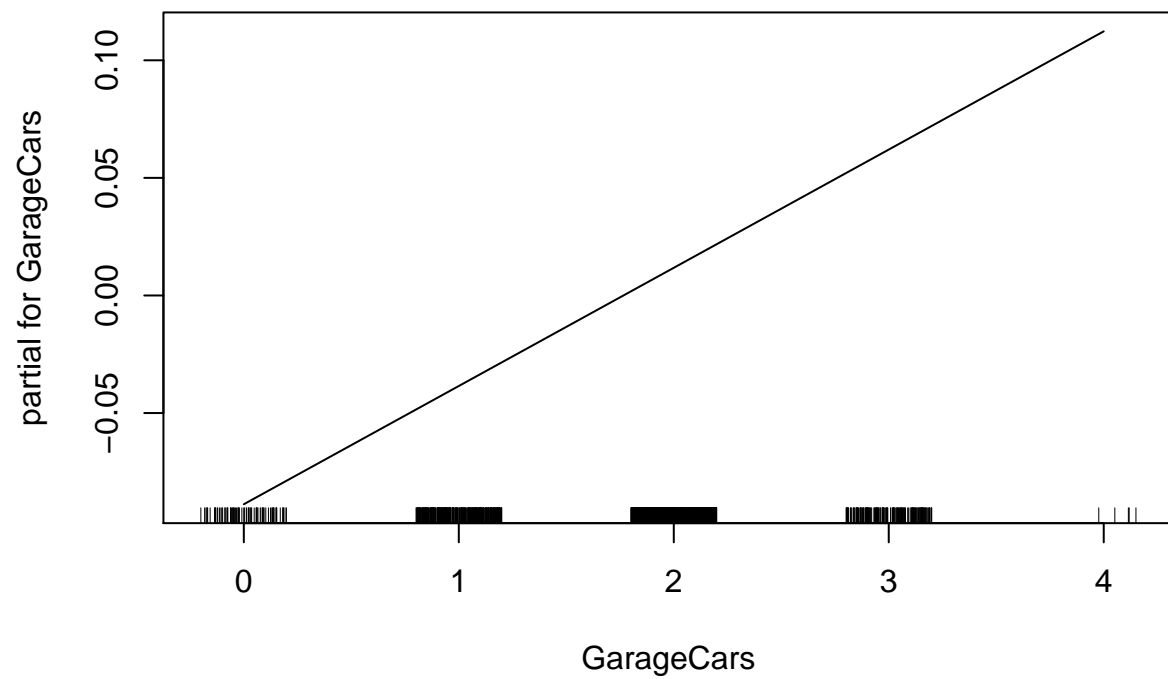


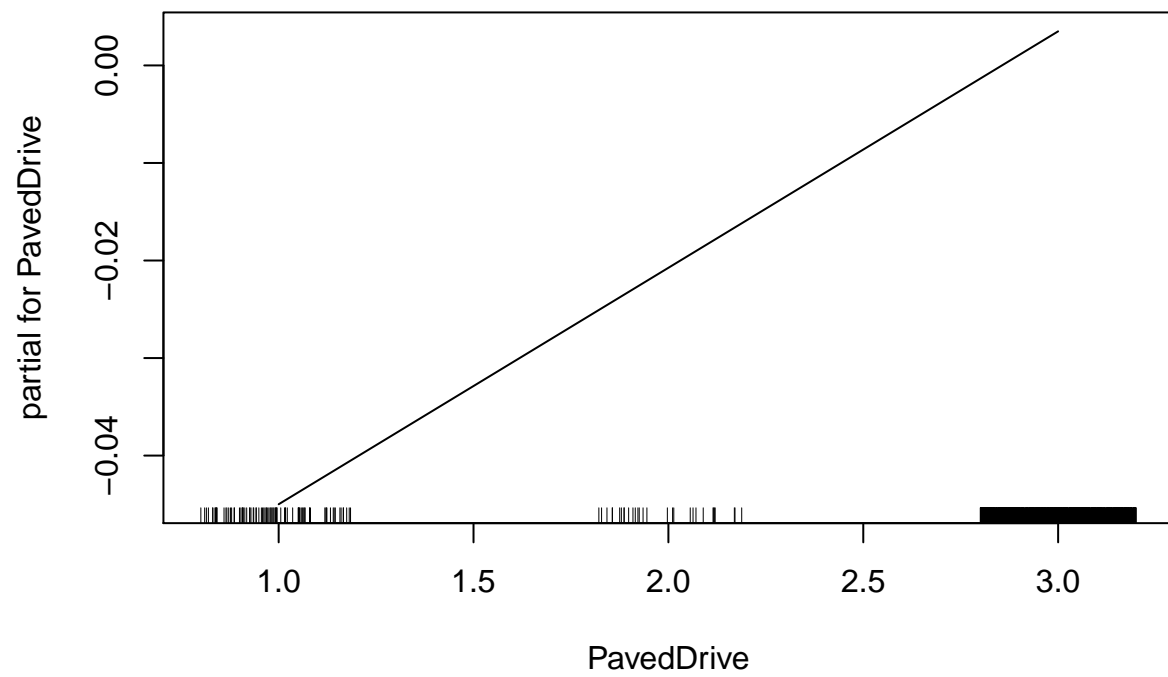


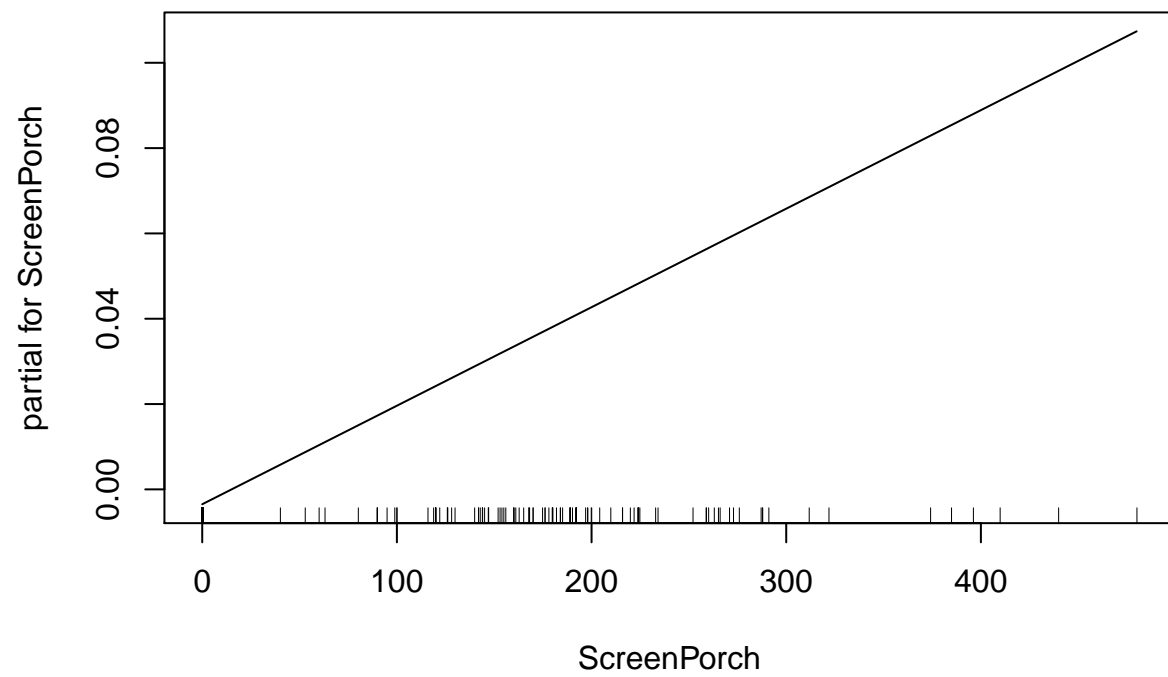


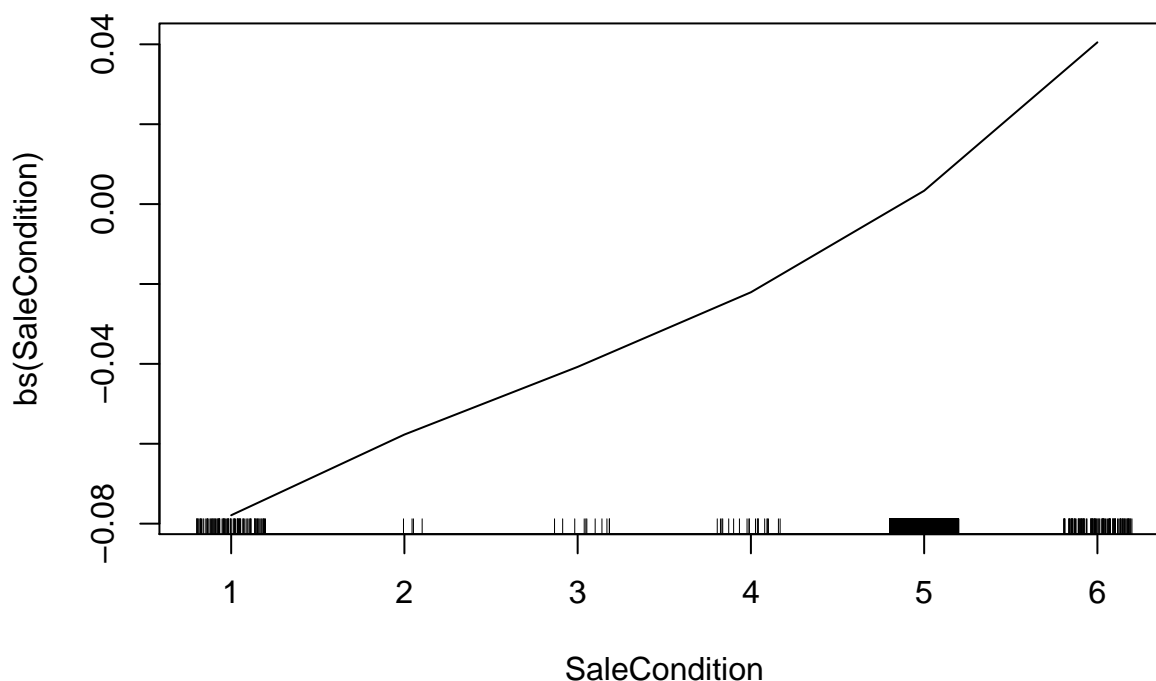










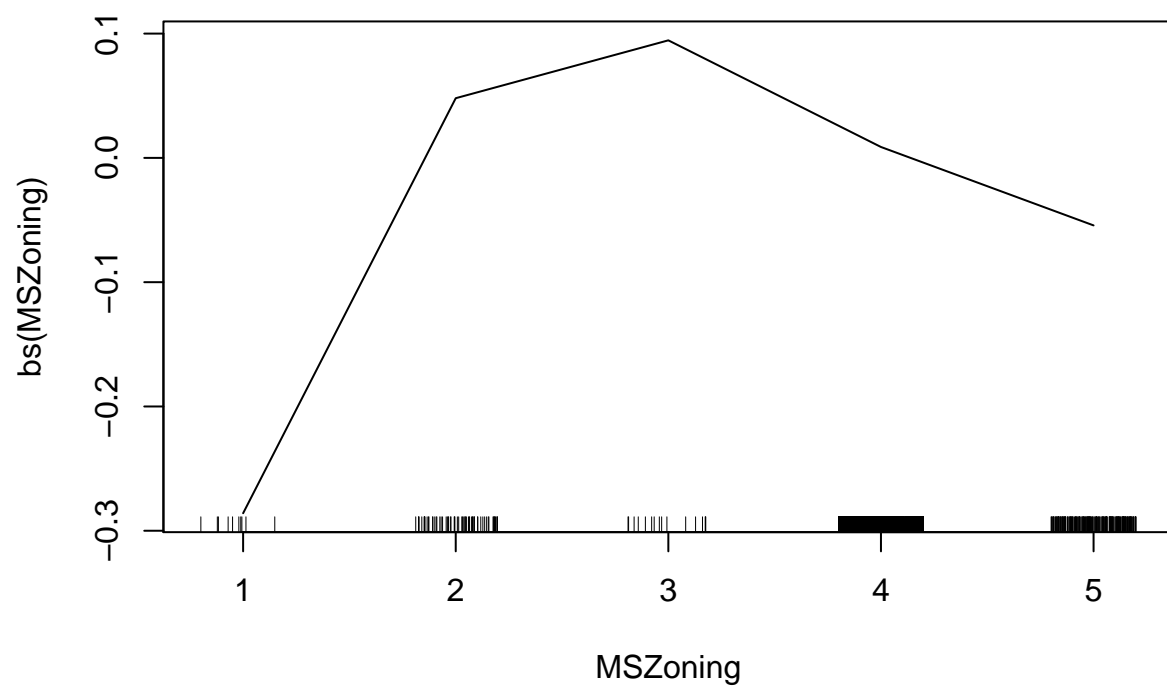


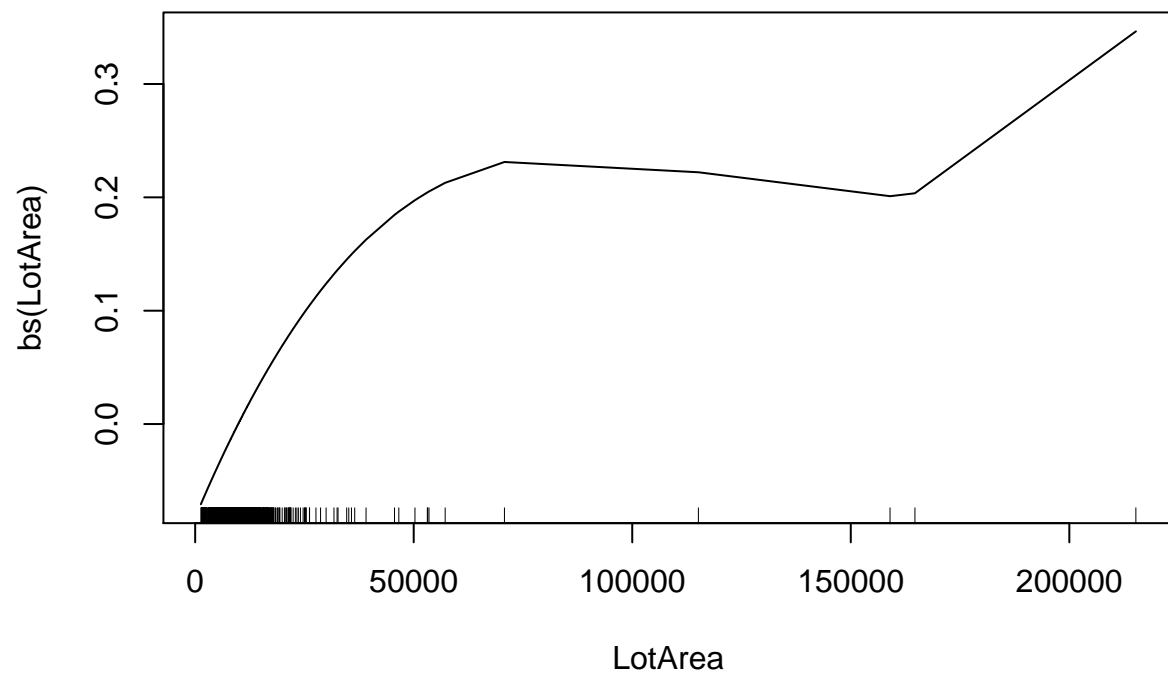
```
gam.preds <- newdata.predict.Gam(gam.lm, newdata = all.df.nolvg[1458:2916,])
gam.preds <- e^gam.preds
gam.submit <- data.frame(test.ids,gam.preds)
colnames(gam.submit) <- c("Id", "SalePrice")
#write.csv(gam.submit, "gam.forw1.csv")
```

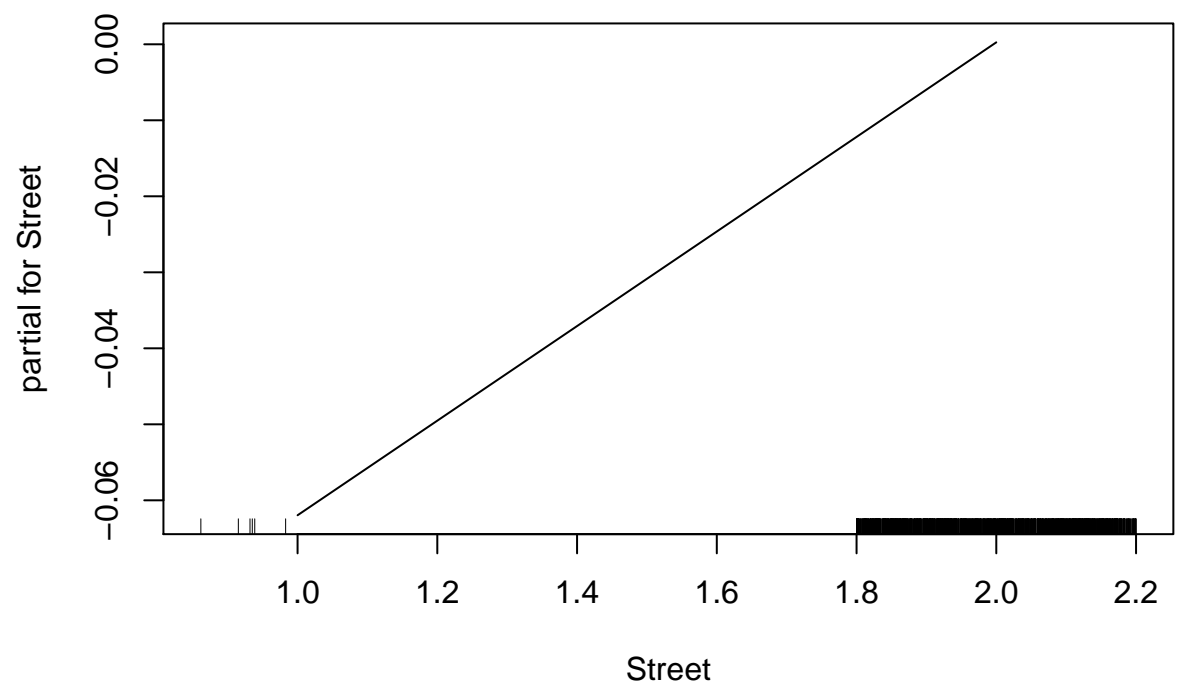
```
#Mixed subset natural splines
coef(back.sub, 27)
```

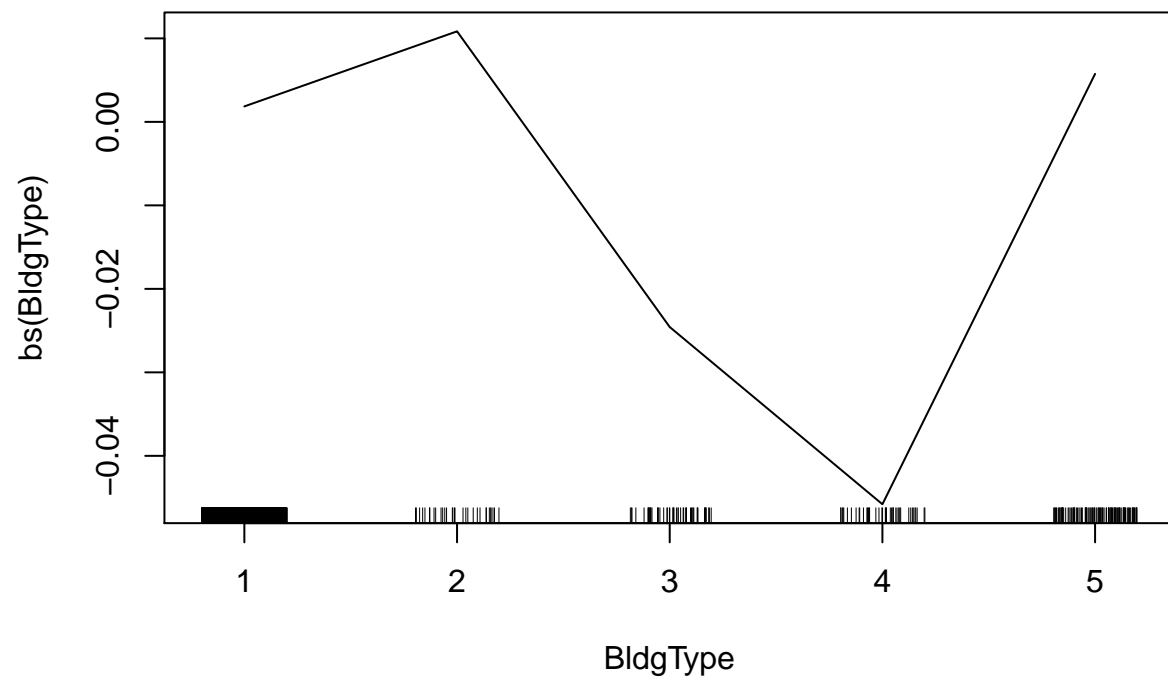
```
## (Intercept)      MSZoning      LotArea      Street      BldgType
## 4.146417e+00 -1.801323e-02 2.610436e-06 1.969786e-01 -1.410242e-02
## OverallQual OverallCond YearBuilt YearRemodAdd MasVnrType
## 6.299799e-02 4.182951e-02 2.223670e-03 7.393258e-04 1.423821e-02
## ExterCond BsmtQual BsmtFinSF1 BsmtFinSF2 BsmtUnfSF
## 1.265530e-02 -1.656281e-02 1.590889e-04 1.247363e-04 7.847722e-05
## HeatingQC CentralAir X1stFlrSF X2ndFlrSF BsmtFullBath
## -8.616867e-03 7.288525e-02 3.302104e-04 2.795593e-04 2.524375e-02
## KitchenAbvGr KitchenQual Functional Fireplaces GarageCars
## -4.873816e-02 -2.166535e-02 2.021194e-02 2.823062e-02 5.252269e-02
## PavedDrive ScreenPorch SaleCondition
## 2.137493e-02 2.509782e-04 2.541942e-02
```

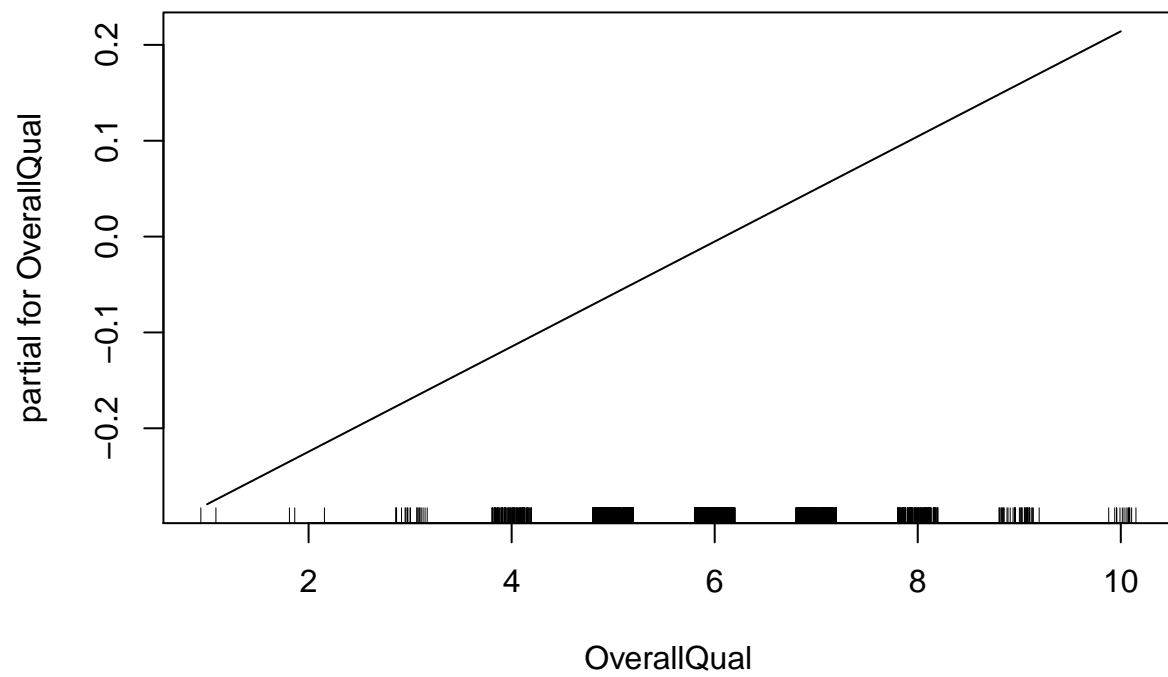
```
gam.lm <- lm(SalePrice~bs(MSZoning)+bs(LotArea)+Street+bs(BldgType)+OverallQual+bs(OverallCond)+bs(YearBuilt))
#summary(gam.lm)
plot.Gam(gam.lm)
```

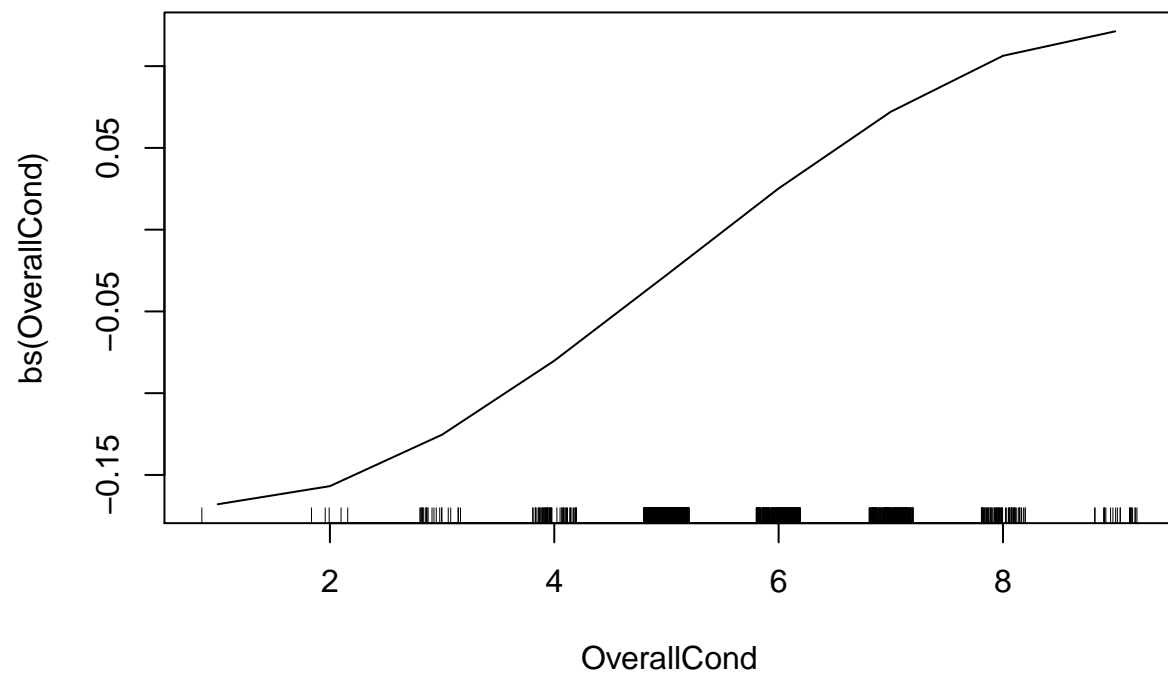


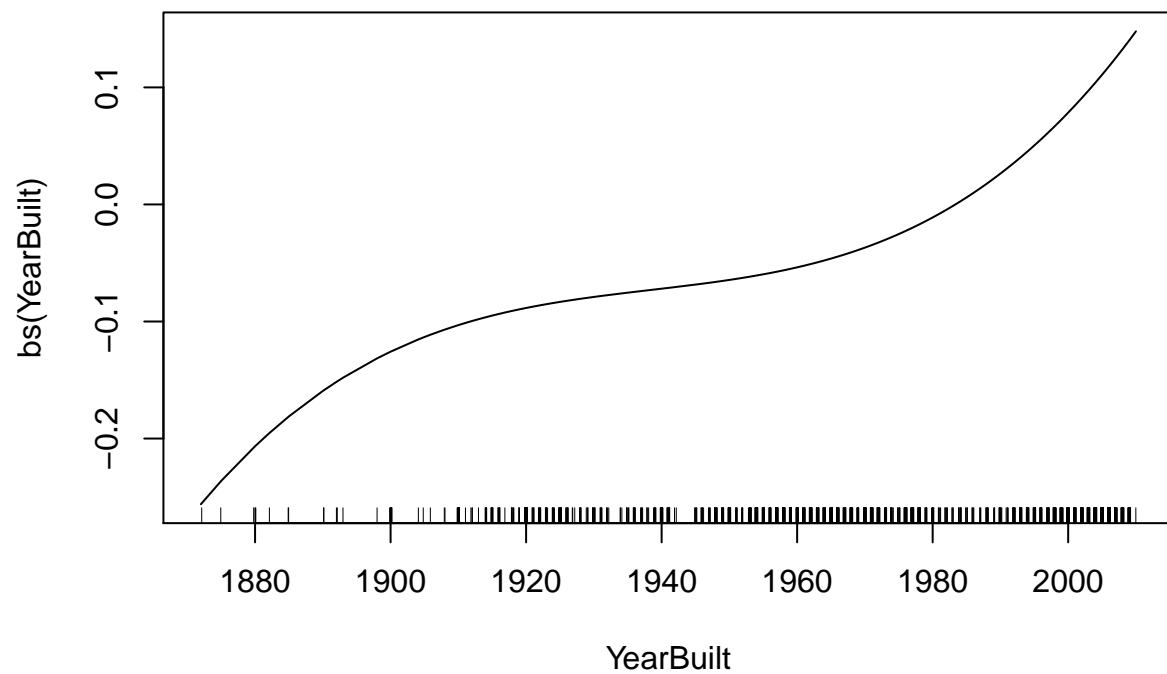


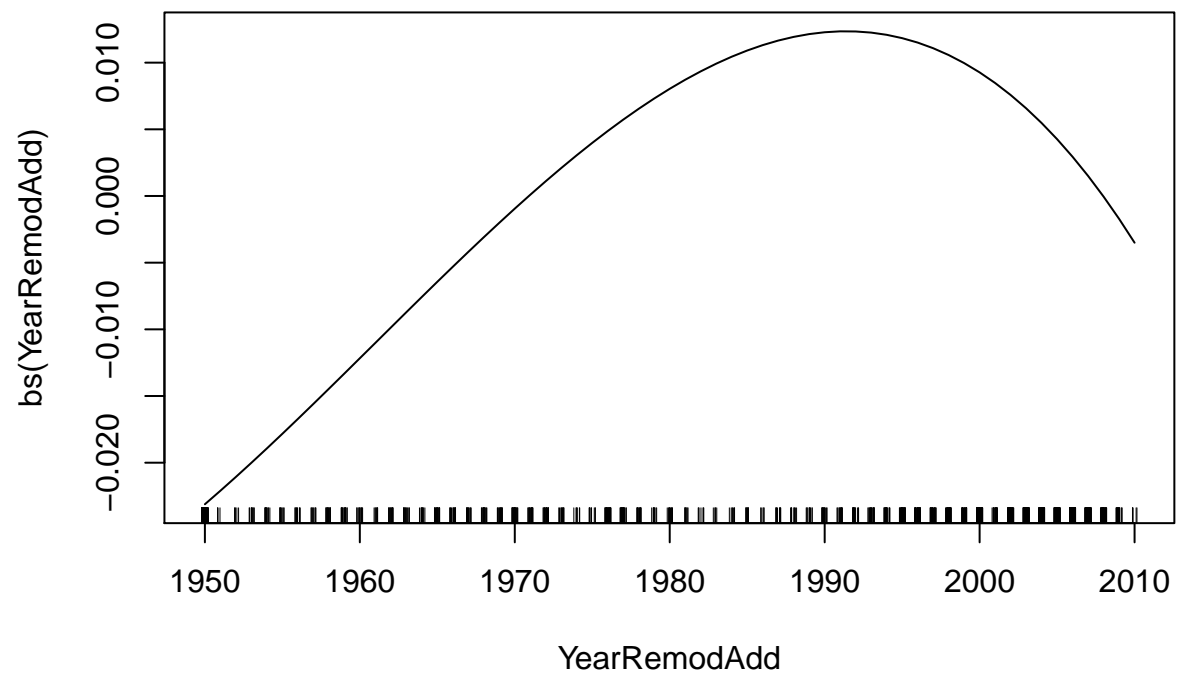


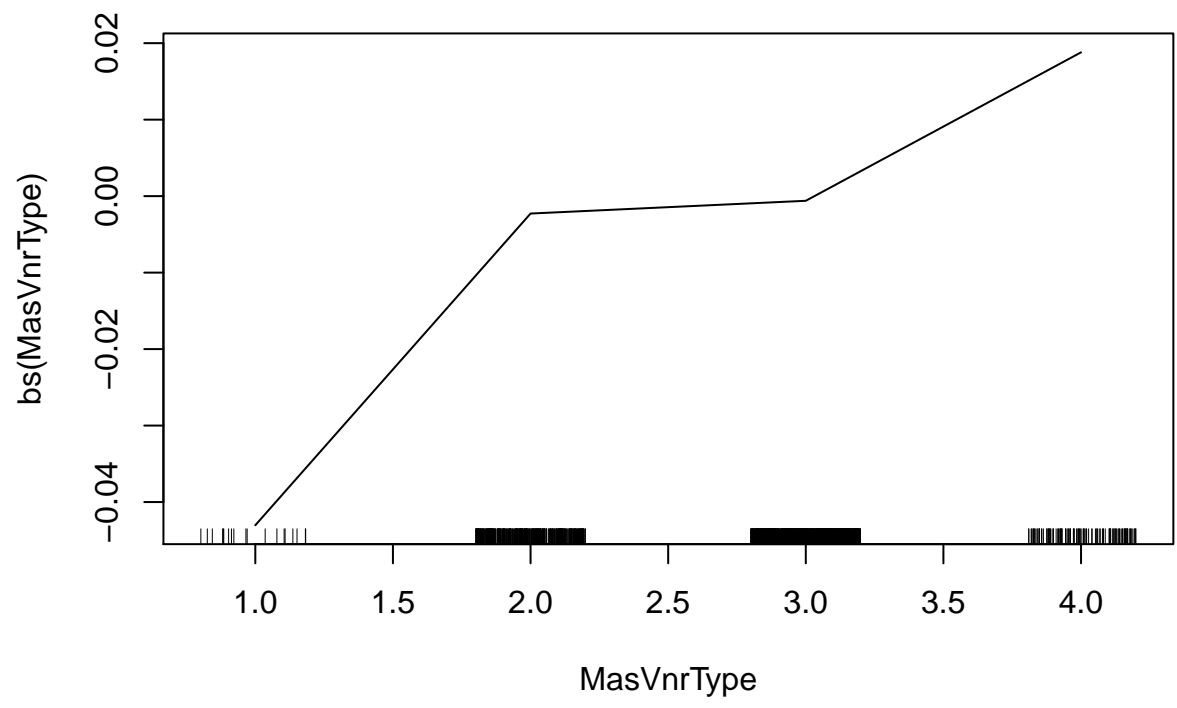


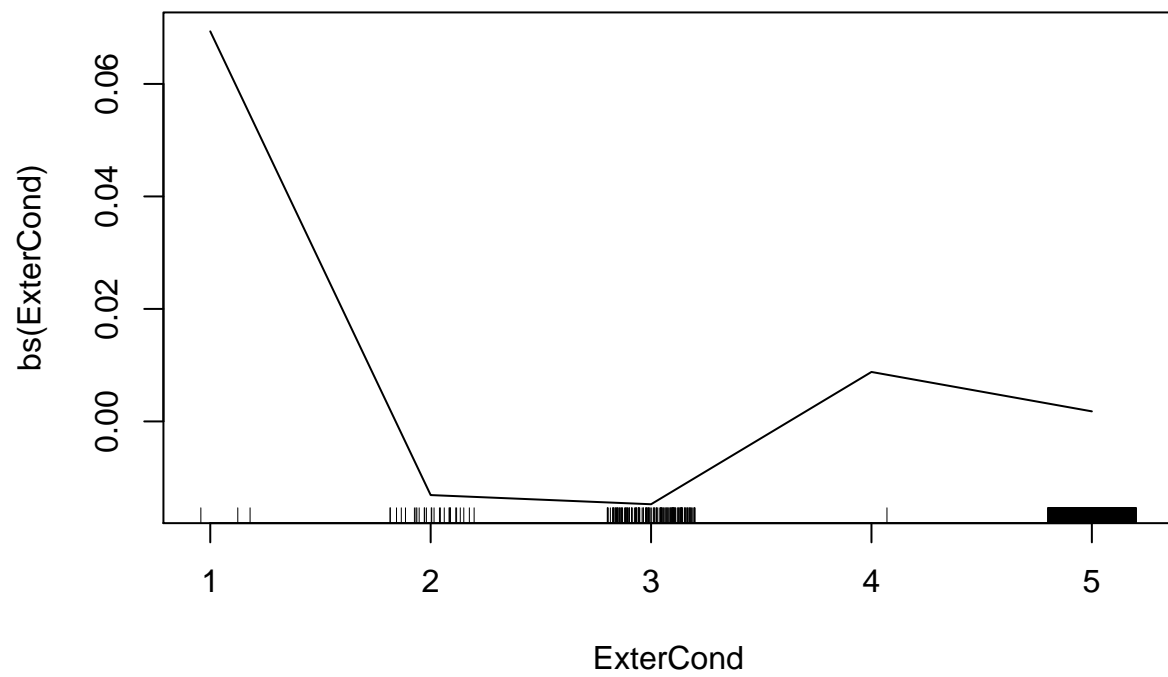


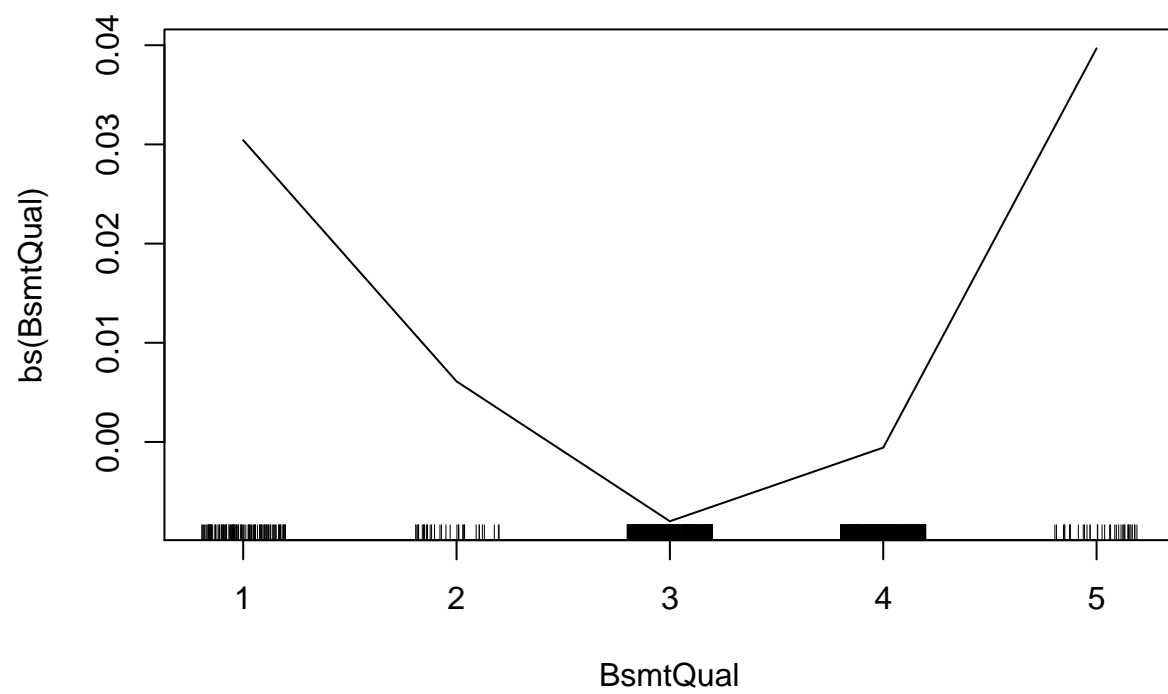


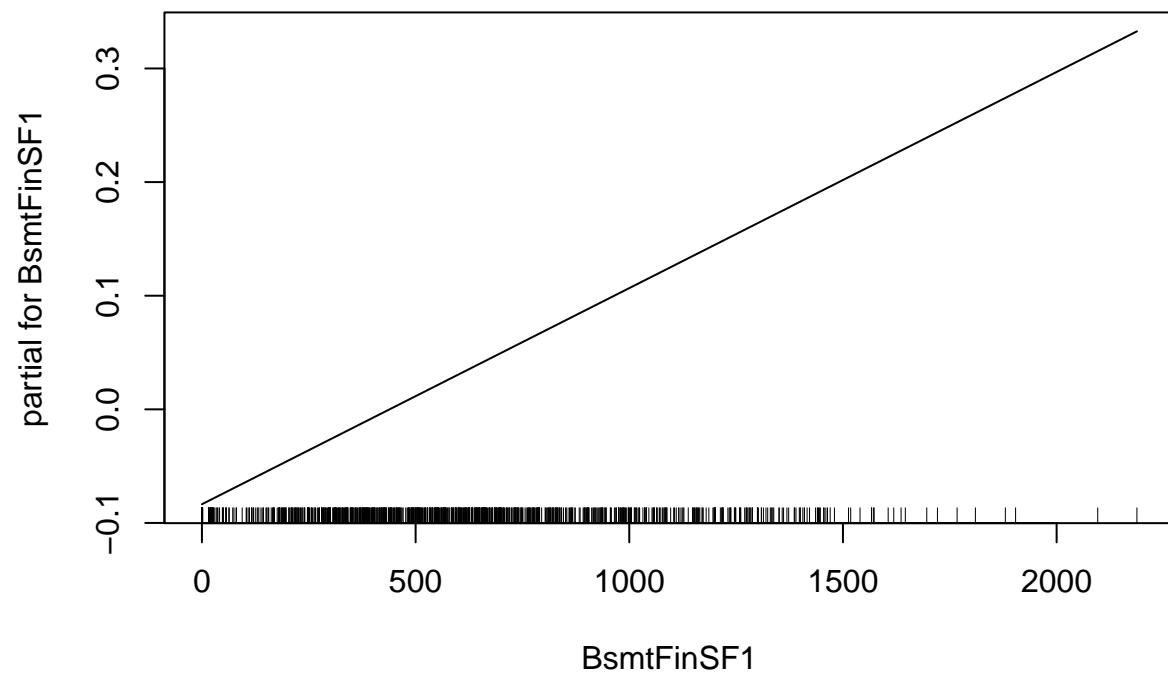


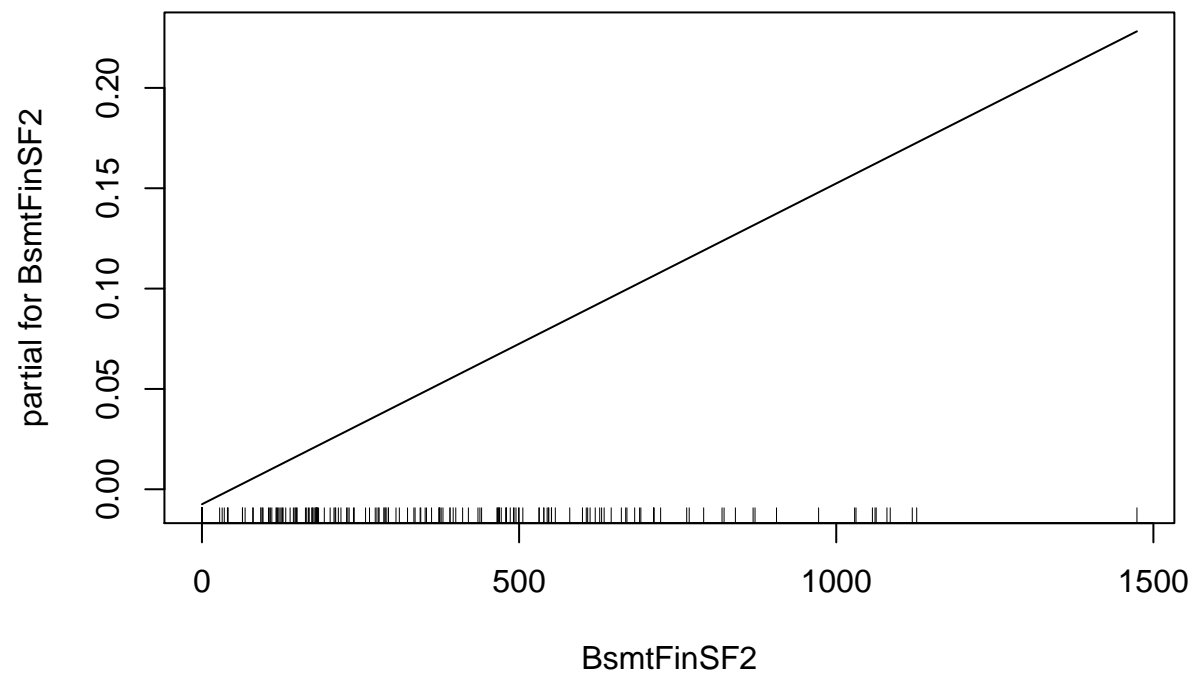


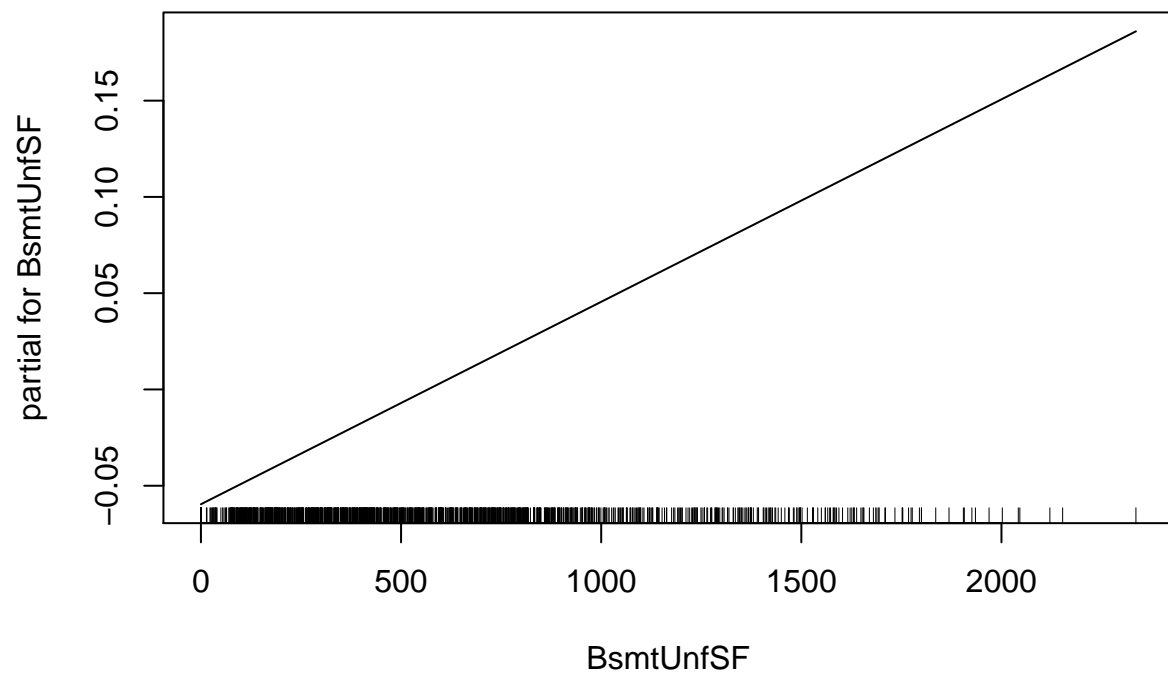


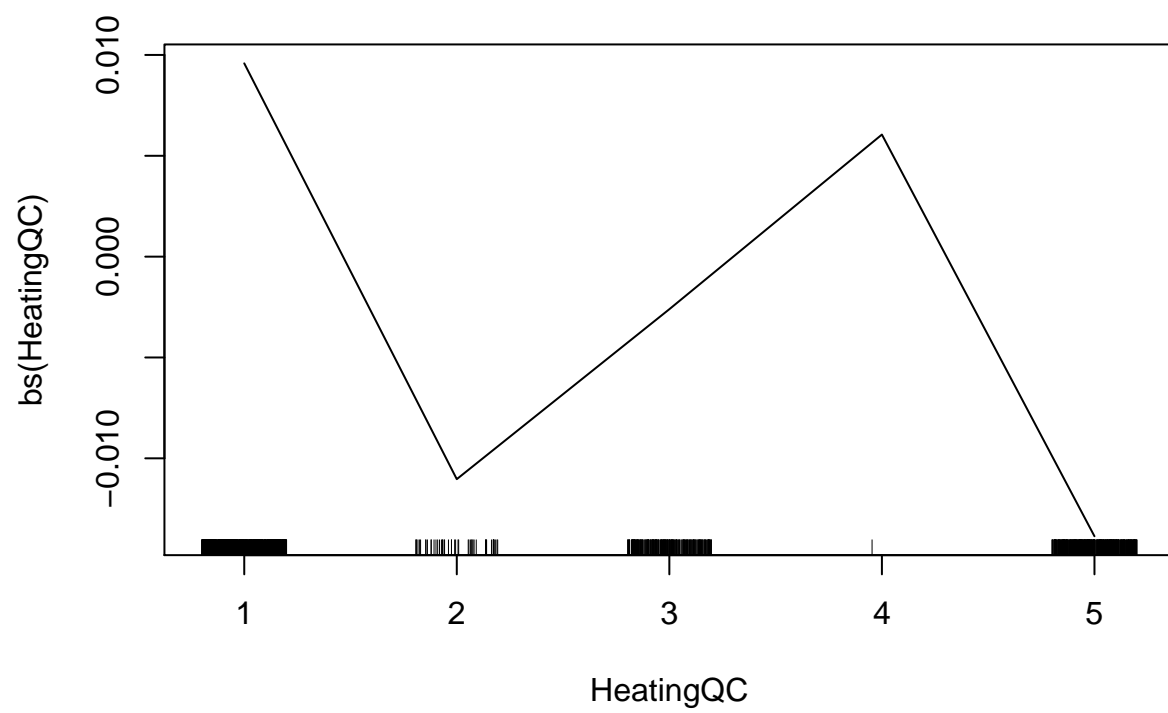


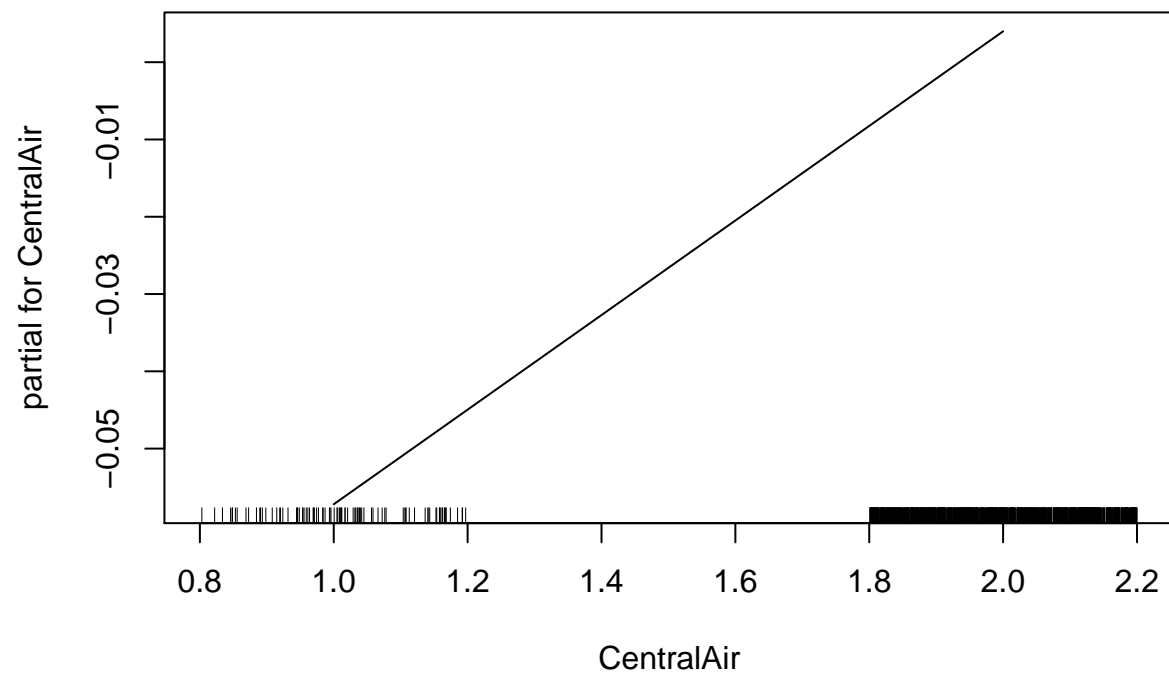


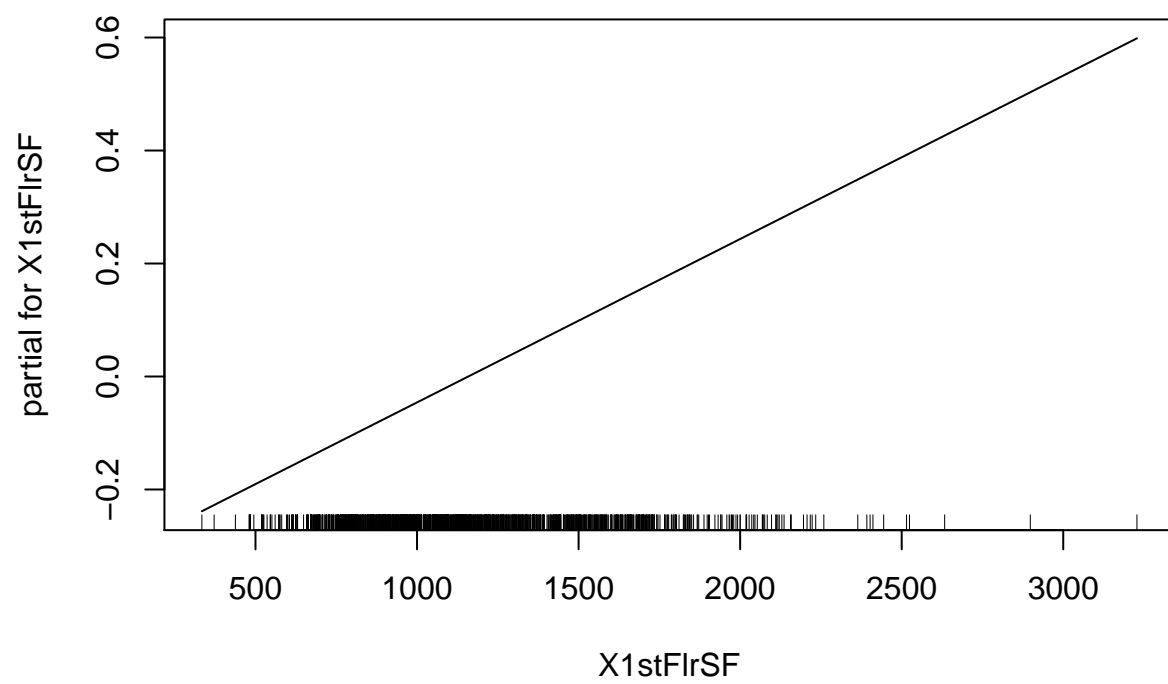


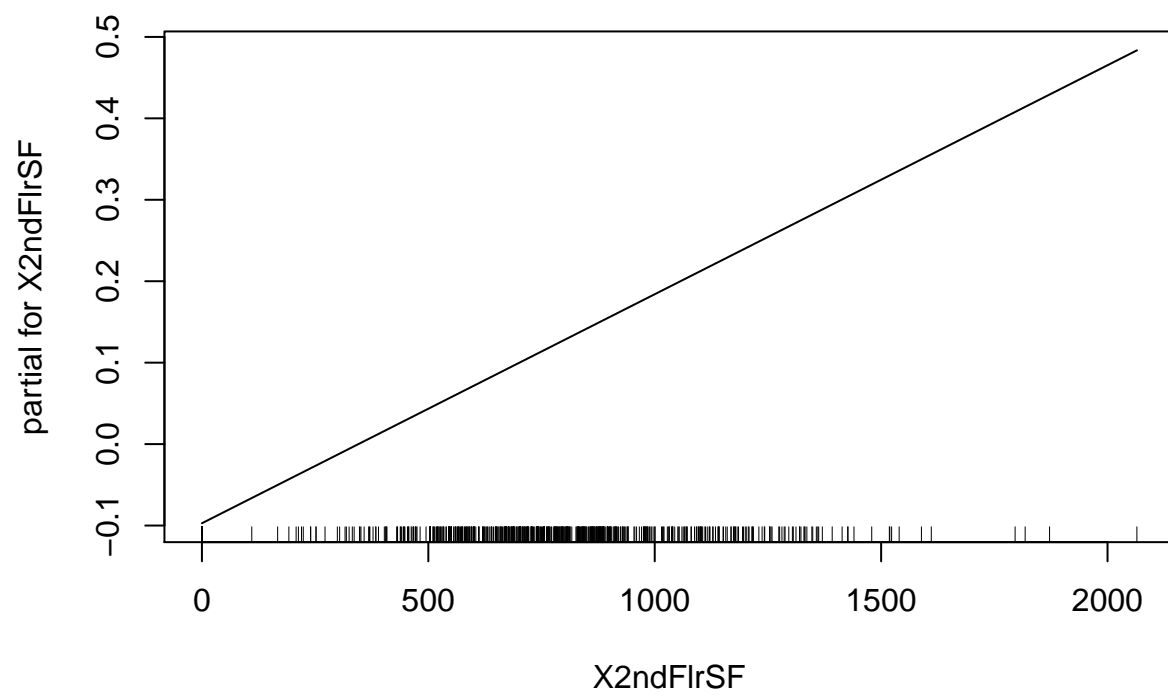


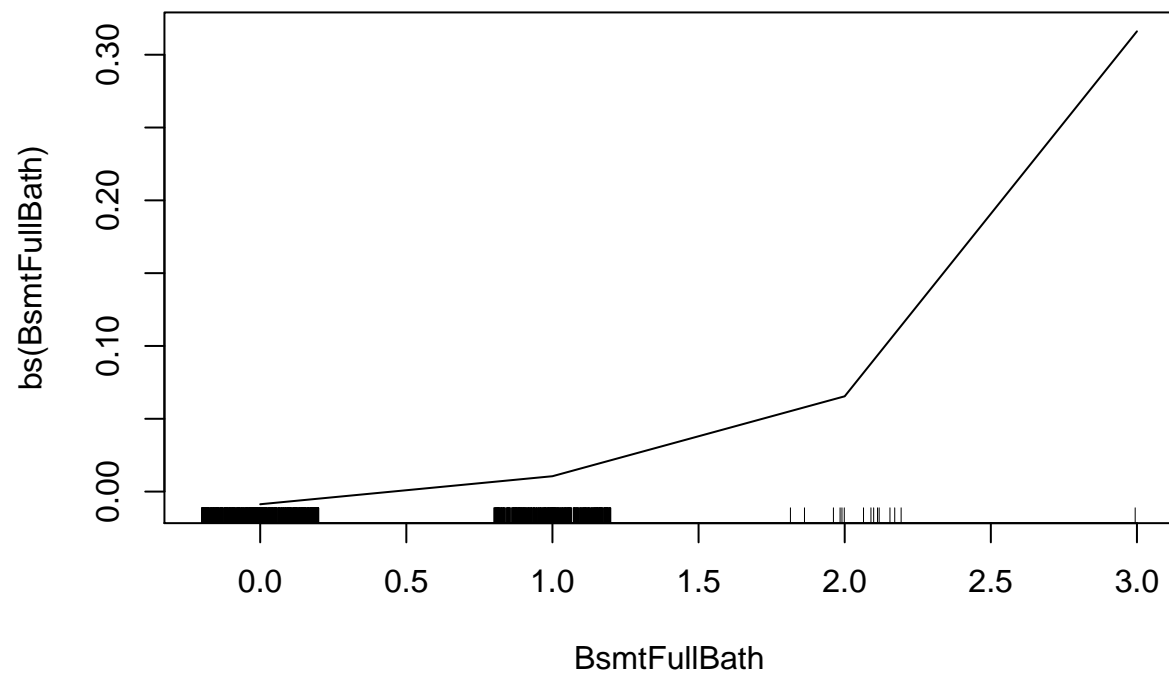


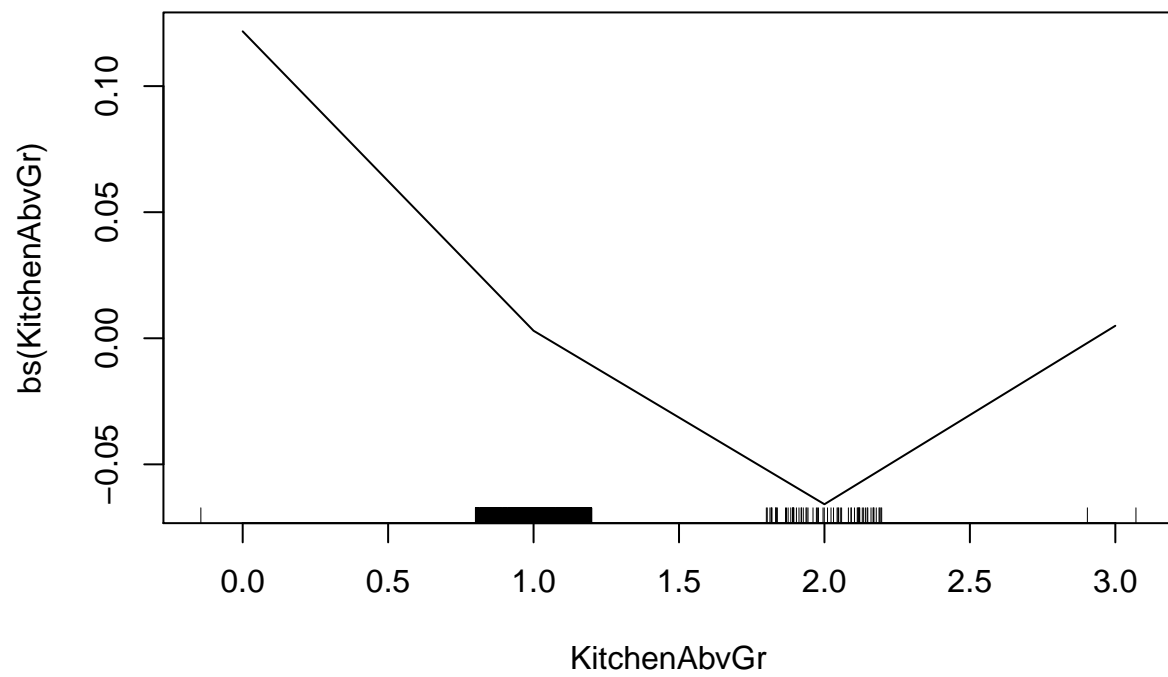


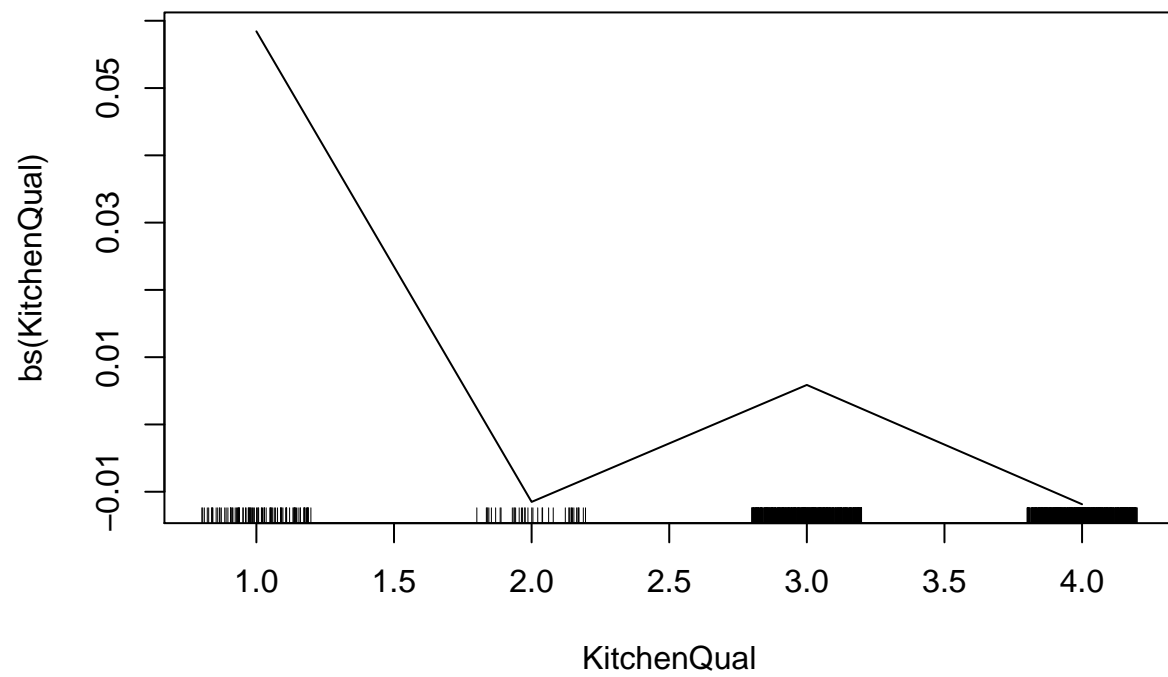


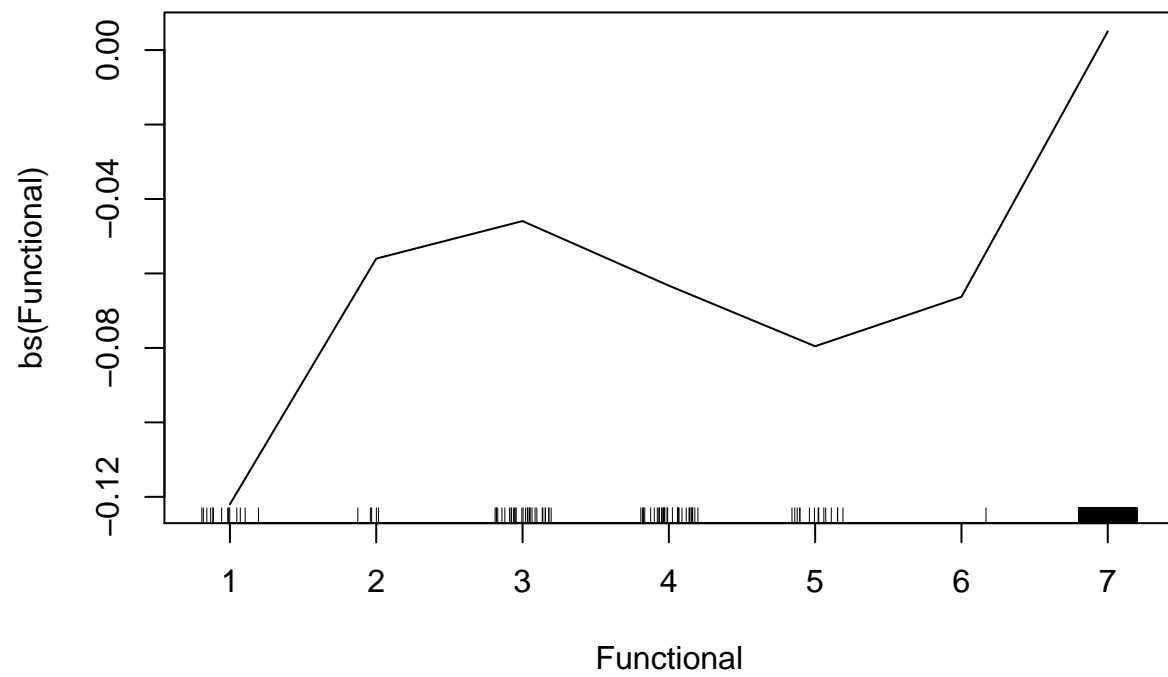


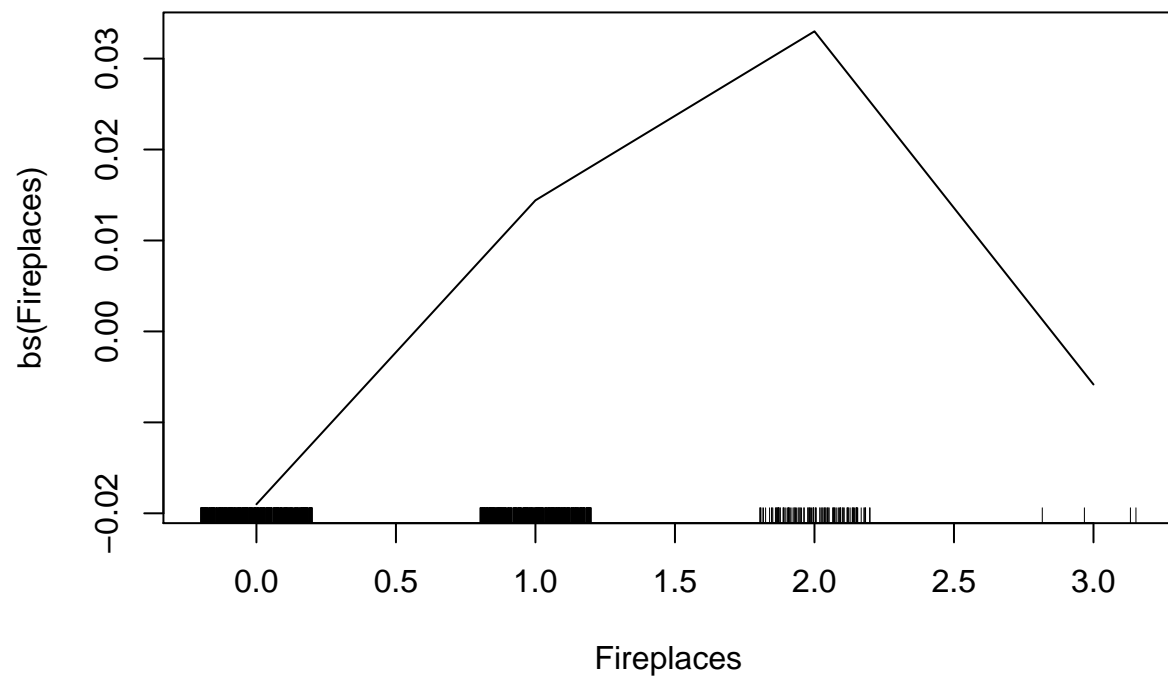


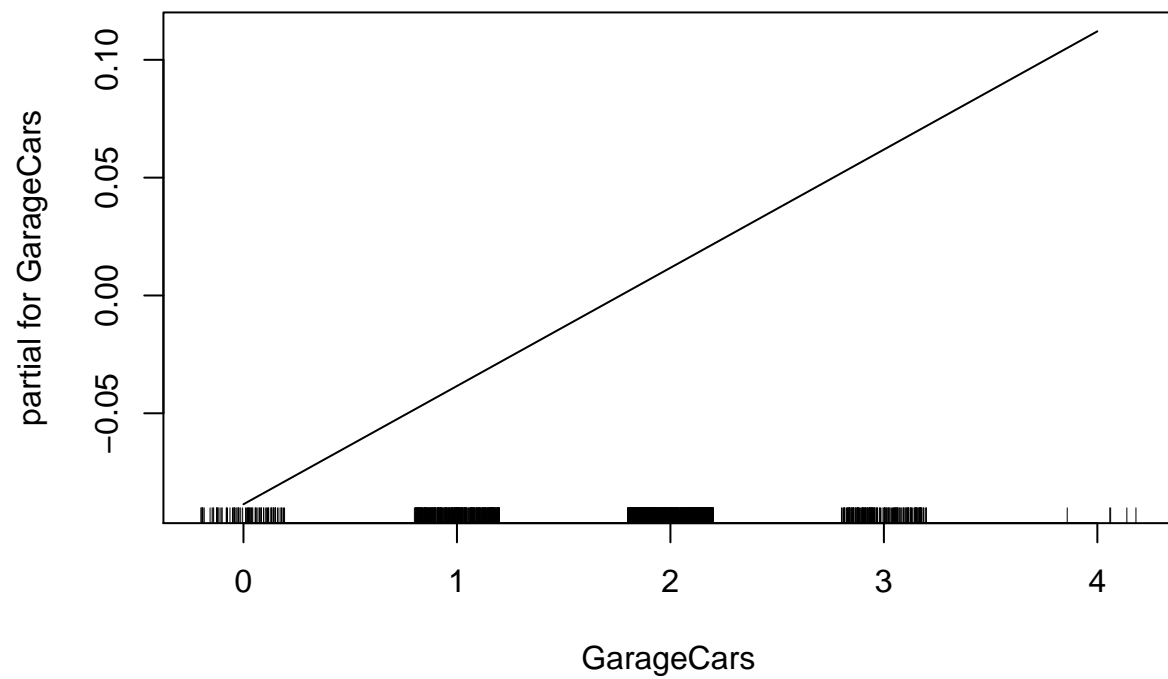


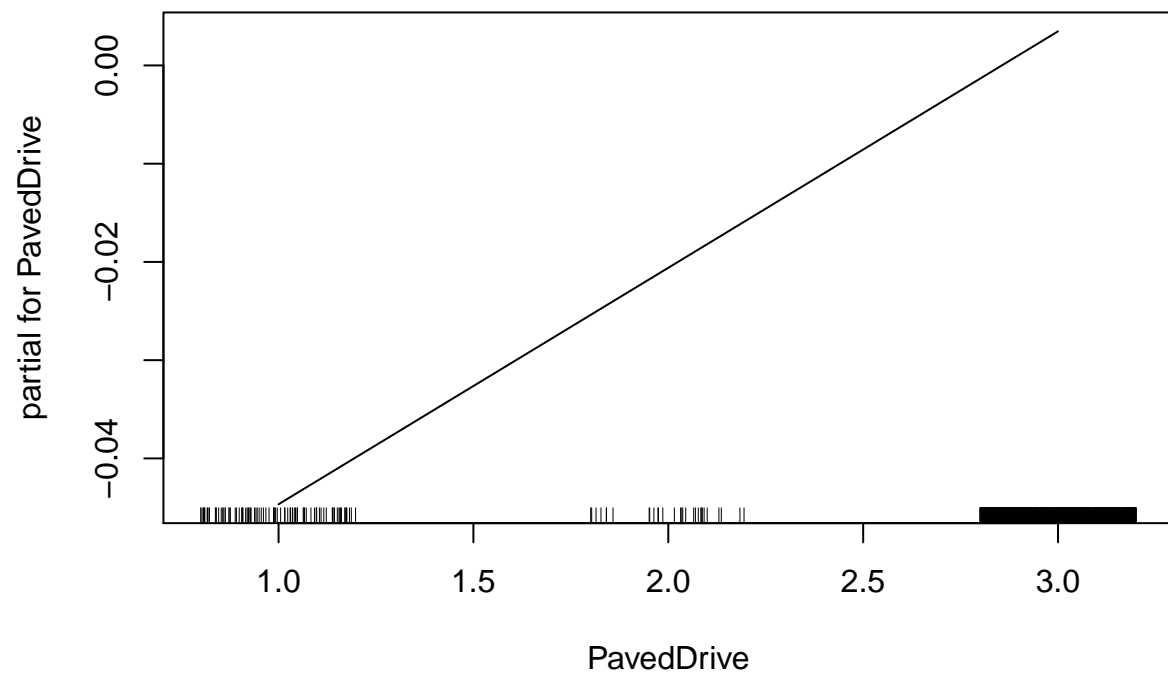


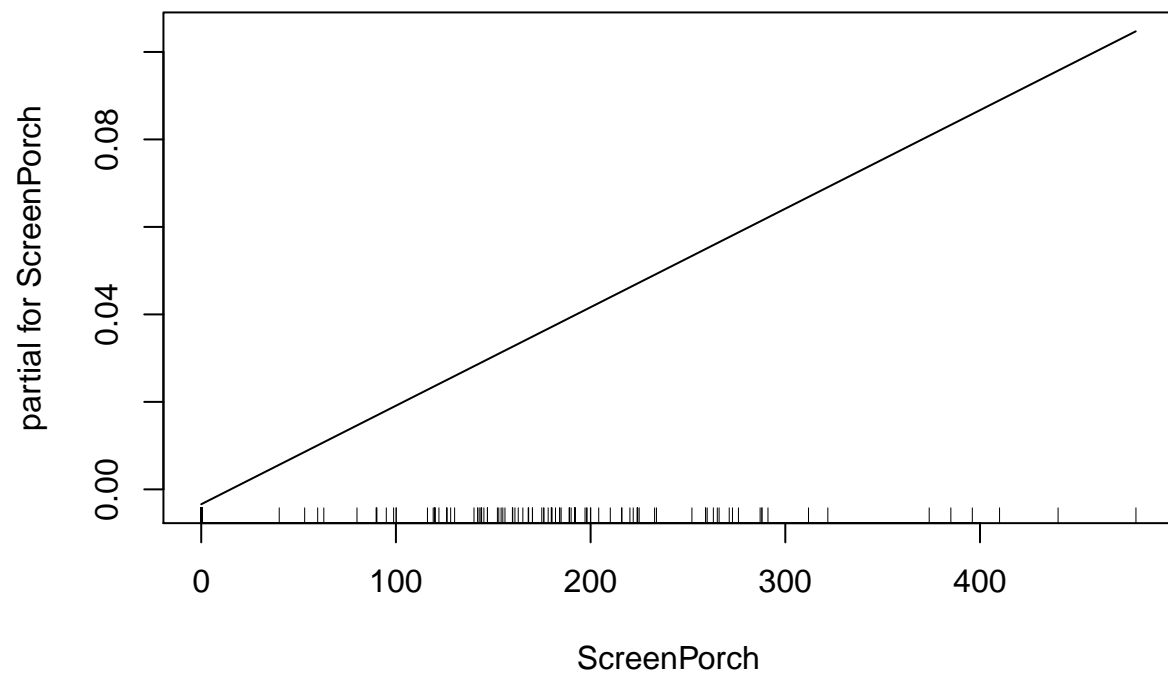


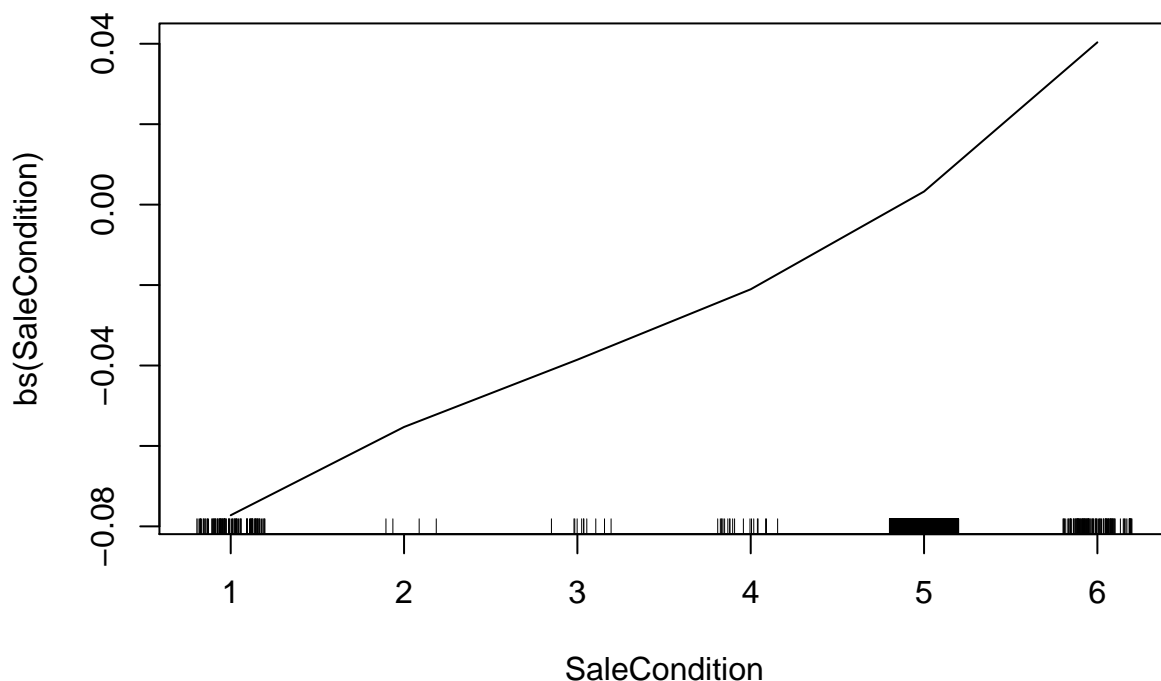












```
gam.preds <- newdata.predict.Gam(gam.lm, newdata = all.df.nolvg[1458:2916,])
```

```
## Warning in bs(Fireplaces, degree = 3L, knots = numeric(0), Boundary.knots
## = c(0L, : some 'x' values beyond boundary knots may cause ill-conditioned
## bases
```

```
gam.preds <- e^gam.preds
gam.submit <- data.frame(test.ids,gam.preds)
colnames(gam.submit) <- c("Id", "SalePrice")
#write.csv(gam.submit, "gam.back.csv")
```

Tree models

```
#Unpruned tree
```

```
tree.fit <- tree(SalePrice~.,data=all.df.nolvg[1:1457,])
summary(tree.fit)
```

```
##
## Regression tree:
## tree(formula = SalePrice ~ ., data = all.df.nolvg[1:1457, ])
## Variables actually used in tree construction:
## [1] "OverallQual" "X1stFlrSF" "GarageCars" "FullBath" "GarageType"
## Number of terminal nodes: 10
## Residual mean deviance: 0.04231 = 61.22 / 1447
## Distribution of residuals:
##      Min.   1st Qu.   Median     Mean   3rd Qu.    Max.
```

```
## -0.942500 -0.113500 0.007565 0.000000 0.118700 0.865000
par(mfrow=c(1,2))
plot(tree.fit)
text(tree.fit, pretty=0)

tree.pred <- predict(tree.fit, newdata = all.df.nolvg[1458:2916,-77])
tree.pred <- unname(tree.pred)
tree.pred <- e^tree.pred
tree.submit <- data.frame(test.ids,tree.pred)
colnames(tree.submit) <- c("Id", "SalePrice")
#write.csv(tree.submit, "tree.csv")

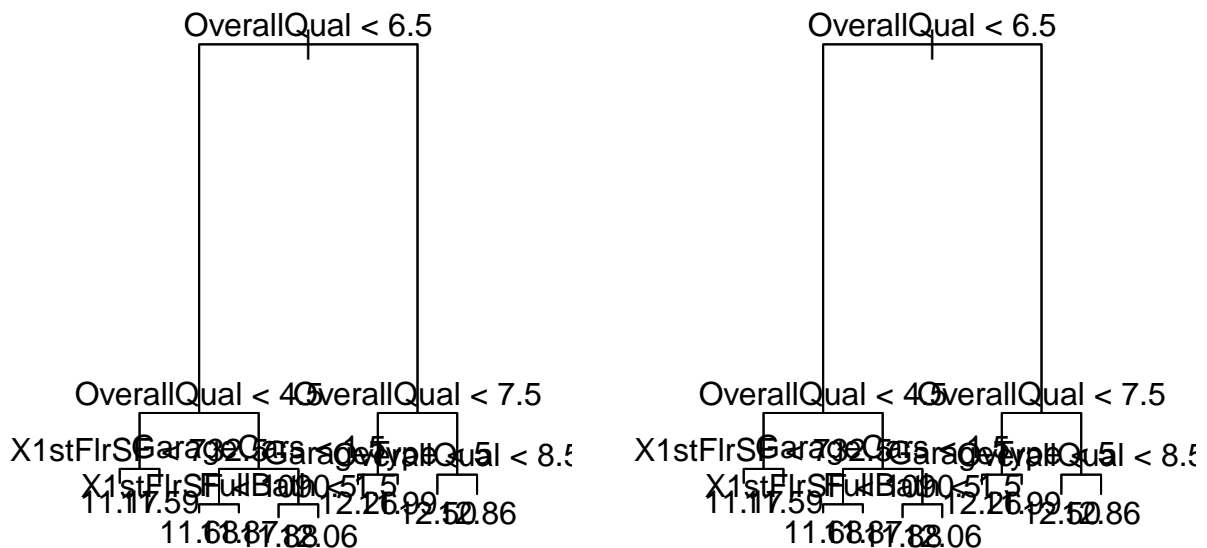
#Pruned tree
cv.tree1 <- cv.tree(tree.fit, K = 10)
best.size <- cv.tree1$size[which.min(cv.tree1$dev)]
best.size

## [1] 10

prune.tree1 <- prune.tree(tree.fit, best = best.size)
summary(prune.tree1)

##
## Regression tree:
## tree(formula = SalePrice ~ ., data = all.df.nolvg[1:1457, ])
## Variables actually used in tree construction:
## [1] "OverallQual" "X1stFlrSF" "GarageCars" "FullBath" "GarageType"
## Number of terminal nodes: 10
## Residual mean deviance: 0.04231 = 61.22 / 1447
## Distribution of residuals:
##      Min.   1st Qu.   Median     Mean   3rd Qu.    Max.
## -0.942500 -0.113500 0.007565 0.000000 0.118700 0.865000

plot(prune.tree1)
text(prune.tree1, pretty=0)
```



```
tree.pred1 <- predict(prune.tree1, newdata = all.df.nolvg[1458:2916,-77])
tree.pred1 <- unname(tree.pred1)
tree.pred1 <- e^tree.pred1
tree.submit <- data.frame(test.ids,tree.pred1)
colnames(tree.submit) <- c("Id", "SalePrice")
#write.csv(tree.submit, "tree1.csv")
```

#Bagging, no outliers

```
bag.tree <- randomForest(SalePrice~., data = all.df.nolvg[1:1457,], mtry = 76, importance = TRUE, ntree = 500)
bag.tree
```

##

Call:

randomForest(formula = SalePrice ~ ., data = all.df.nolvg[1:1457,], mtry = 76, importance = TRUE, ntree = 500)

Type of random forest: regression

Number of trees: 500

No. of variables tried at each split: 76

##

Mean of squared residuals: 0.01985942

% Var explained: 87.56

```
importance(bag.tree)
```

```
##           %IncMSE IncNodePurity
## MSSubClass 14.37738110 8.517067e-01
## MSZoning   13.32170349 1.784989e+00
## LotFrontage 8.76975839 1.065460e+00
```

## LotArea	31.41713513	5.911439e+00
## Street	-2.48231702	3.747098e-02
## Alley	4.16461190	1.983328e-01
## LotShape	6.20145310	2.898136e-01
## LandContour	-0.76329548	2.101485e-01
## LotConfig	0.08126844	1.621686e-01
## LandSlope	3.86010919	1.381023e-01
## Neighborhood	17.57508953	1.150261e+00
## Condition1	1.90808377	1.946950e-01
## Condition2	-0.90826861	1.721621e-02
## BldgType	7.06976693	1.746968e-01
## HouseStyle	5.99641765	2.491231e-01
## OverallQual	88.29401596	1.364818e+02
## OverallCond	18.00509336	2.081477e+00
## YearBuilt	17.66965352	2.019836e+00
## YearRemodAdd	13.61715312	1.876806e+00
## RoofStyle	2.94873382	1.967411e-01
## RoofMatl	-0.63741980	2.302393e-02
## Exterior1st	4.81824538	4.935919e-01
## Exterior2nd	3.75944201	4.222843e-01
## MasVnrType	4.02054150	1.475300e-01
## MasVnrArea	11.06698798	1.011060e+00
## ExterQual	7.91435243	3.160123e-01
## ExterCond	2.09038642	3.996251e-01
## Foundation	4.99548225	2.228921e-01
## BsmtQual	9.30910486	6.992698e-01
## BsmtCond	2.80778869	1.935630e-01
## BsmtExposure	6.92335565	3.157826e-01
## BsmtFinType1	9.93692231	5.367875e-01
## BsmtFinSF1	32.62617684	5.653643e+00
## BsmtFinType2	-0.67300245	1.141777e-01
## BsmtFinSF2	2.40123744	1.545018e-01
## BsmtUnfSF	10.13664893	1.406934e+00
## Heating	0.23975228	5.116658e-02
## HeatingQC	5.72832427	3.635335e-01
## CentralAir	7.15669189	2.055275e+00
## Electrical	-1.70241408	1.240745e-01
## X1stFlrSF	44.00641343	1.256715e+01
## X2ndFlrSF	44.55584244	5.874256e+00
## LowQualFinSF	-0.76220590	2.663678e-02
## BsmtFullBath	6.79285124	2.405014e-01
## BsmtHalfBath	2.82899325	4.117763e-02
## FullBath	21.36745140	3.212396e+00
## HalfBath	14.21818433	4.004152e-01
## BedroomAbvGr	13.52576820	8.446248e-01
## KitchenAbvGr	3.32030210	8.569579e-02
## KitchenQual	7.86687021	5.326133e-01
## TotRmsAbvGrd	17.14560799	2.069265e+00
## Functional	2.42481431	2.137019e-01
## Fireplaces	11.56141953	1.046506e+00
## FireplaceQu	11.67409320	1.462493e+00
## GarageType	22.80700652	4.770689e+00
## GarageYrBlt	9.61524926	1.168192e+00
## GarageFinish	8.55388476	1.234250e+00

```
## GarageCars      23.93725886  1.194901e+01
## GarageArea      24.35796781  7.486064e+00
## GarageQual       0.41657628  7.621884e-02
## GarageCond       0.67281415  8.826220e-02
## PavedDrive       1.60363889  3.449534e-01
## WoodDeckSF      10.07140381  1.014487e+00
## OpenPorchSF     14.95389110  1.471975e+00
## EnclosedPorch   -0.94942345  3.963082e-01
## X3SsnPorch       0.84278186  2.930087e-02
## ScreenPorch      4.94640758  1.773122e-01
## PoolArea         1.00100150  3.356625e-03
## PoolQC           0.00000000  3.689075e-03
## Fence            0.70068517  2.213397e-01
## MiscFeature      -0.08674017  2.058781e-02
## MiscVal          0.30667873  6.299443e-02
## MoSold           -0.03510500  7.960406e-01
## YrSold            1.22878834  3.828610e-01
## SaleType         1.38028301  1.417474e-01
## SaleCondition    4.30545805  7.291967e-01
```

```
bag.preds <- predict(bag.tree, newdata = all.df.nolvg[1458:2916,-77])
bag.preds <- unname(bag.preds)
bag.preds <- e^bag.preds
bag.submit <- data.frame(test.ids,bag.preds)
colnames(bag.submit) <- c("Id", "SalePrice")
#write.csv(bag.submit, "bag.csv")
```

#Random Forest, no outliers

```
rf.tree <- randomForest(SalePrice~., data = all.df.nolvg[1:1457,], mtry = round(sqrt(76)), importance =
rf.tree
```

```
##
## Call:
## randomForest(formula = SalePrice ~ ., data = all.df.nolvg[1:1457,      ], mtry = round(sqrt(76)), i
##               Type of random forest: regression
##               Number of trees: 500
## No. of variables tried at each split: 9
##
##               Mean of squared residuals: 0.01984438
##               % Var explained: 87.57
```

```
importance(rf.tree)
```

```
##               %IncMSE IncNodePurity
## MSSubClass      17.9453380    1.50304255
## MSZoning         13.4838342    1.92028541
## LotFrontage      9.3115164    1.90494184
## LotArea          25.1440736    6.75322922
## Street           -0.7897586    0.07654688
## Alley            2.0859571    0.20782427
## LotShape         7.1267480    0.47977381
## LandContour      2.0945743    0.35923268
## LotConfig        1.6725974    0.28349563
## LandSlope        4.2825104    0.34019269
## Neighborhood    13.6238404    1.88688151
## Condition1       4.5050083    0.32859155
```

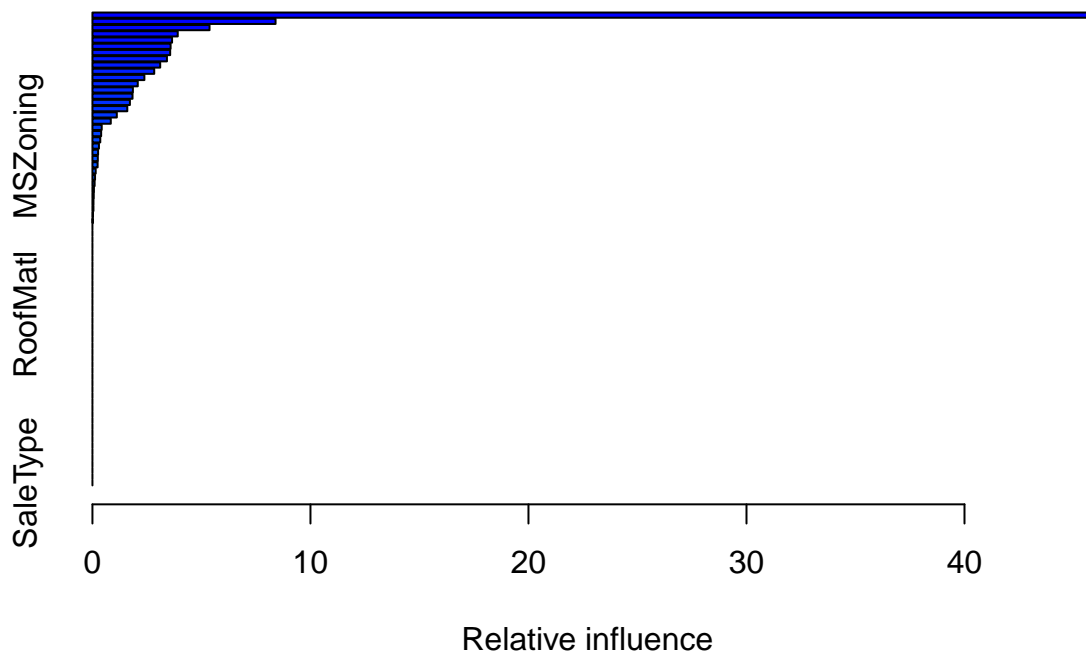
## Condition2	0.3588584	0.03825557
## BldgType	10.6883329	0.51301490
## HouseStyle	10.4101166	0.72609874
## OverallQual	22.7510516	29.11569993
## OverallCond	14.5845839	2.04228614
## YearBuilt	15.3556087	13.70501767
## YearRemodAdd	12.4410542	6.55713110
## RoofStyle	5.5237559	0.58508422
## RoofMatl	1.7955747	0.12943940
## Exterior1st	7.1343192	0.91155348
## Exterior2nd	5.3025016	0.95079603
## MasVnrType	4.5076353	0.44756493
## MasVnrArea	13.1744746	2.42803219
## ExterQual	14.0634862	11.21936064
## ExterCond	-0.1845467	0.49959443
## Foundation	6.8208455	2.08933599
## BsmtQual	10.7837002	6.51098996
## BsmtCond	3.7479919	0.44540885
## BsmtExposure	7.1362919	0.80922592
## BsmtFinType1	11.4384214	1.16843694
## BsmtFinSF1	21.8045935	5.43458004
## BsmtFinType2	2.7329245	0.33402859
## BsmtFinSF2	2.4090799	0.28037012
## BsmtUnfSF	11.5825734	1.78280467
## Heating	-1.9472969	0.17566597
## HeatingQC	6.7730444	1.65704745
## CentralAir	5.5043665	2.06967980
## Electrical	1.2276487	0.31539917
## X1stFlrSF	23.2092525	12.66762362
## X2ndFlrSF	24.1862458	5.49467138
## LowQualFinSF	-0.8056017	0.09392174
## BsmtFullBath	9.8725479	0.76282485
## BsmtHalfBath	3.3683113	0.10633312
## FullBath	14.1103727	10.95265884
## HalfBath	12.0189831	1.07712038
## BedroomAbvGr	15.0777188	2.09264075
## KitchenAbvGr	4.3832206	0.24175731
## KitchenQual	11.4863416	7.29014467
## TotRmsAbvGrd	21.7309850	5.24586459
## Functional	2.9631593	0.33012389
## Fireplaces	17.2146869	5.87384153
## FireplaceQu	15.6034869	5.59987212
## GarageType	12.9154983	6.16170289
## GarageYrBlt	13.2346487	10.79091766
## GarageFinish	13.0203246	8.09474351
## GarageCars	14.0364141	11.95394108
## GarageArea	18.6402206	12.31413828
## GarageQual	4.1446786	1.00158148
## GarageCond	4.4891347	0.89341135
## PavedDrive	3.6830192	0.74291279
## WoodDeckSF	8.6463124	1.77788745
## OpenPorchSF	11.9647049	3.08989223
## EnclosedPorch	1.6071826	0.50743697
## X3SsnPorch	0.6862661	0.06346187

```
## ScreenPorch    4.4490371    0.29190802
## PoolArea      -1.2474837    0.02899094
## PoolQC        -1.3306268    0.06588348
## Fence         2.1788658    0.42640439
## MiscFeature    0.8665385    0.11886790
## MiscVal       -2.2084644    0.15918653
## MoSold        1.4775196    1.07571013
## YrSold        -0.8894058    0.60446330
## SaleType      2.2243493    0.28010975
## SaleCondition  0.8562817    0.67464738
```

```
rf.preds <- predict(rf.tree, newdata = all.df.nolvg[1458:2916,-77])
rf.preds <- unname(rf.preds)
rf.preds <- e~rf.preds
rf.submit <- data.frame(test.ids,rf.preds)
colnames(rf.submit) <- c("Id", "SalePrice")
#write.csv(rf.submit, "rf.csv")
```

```
#Boosting, no outliers
```

```
gbm.tree <- gbm(SalePrice~., data = all.df.nolvg[1:1457,], distribution = "gaussian", shrinkage = .01, n.trees = 1000)
summary(gbm.tree)
```



```
##              var      rel.inf
## OverallQual OverallQual 45.86494410
## X1stFlrSF    X1stFlrSF  8.39923682
## GarageCars   GarageCars 5.37101786
## GarageFinish GarageFinish 3.91244509
```


## FullBath	FullBath	3.65245350
## GarageArea	GarageArea	3.58519682
## X2ndFlrSF	X2ndFlrSF	3.56598905
## LotArea	LotArea	3.41779398
## BsmtFinSF1	BsmtFinSF1	3.10799112
## GarageType	GarageType	2.83417305
## YearRemodAdd	YearRemodAdd	2.38228373
## CentralAir	CentralAir	2.07982711
## YearBuilt	YearBuilt	1.85584720
## Fireplaces	Fireplaces	1.83875906
## TotRmsAbvGrd	TotRmsAbvGrd	1.71873271
## FireplaceQu	FireplaceQu	1.59856795
## OverallCond	OverallCond	1.11786059
## KitchenQual	KitchenQual	0.84154350
## HalfBath	HalfBath	0.42544773
## BsmtQual	BsmtQual	0.39976821
## ExterQual	ExterQual	0.36014017
## MSZoning	MSZoning	0.29851709
## OpenPorchSF	OpenPorchSF	0.25944503
## PavedDrive	PavedDrive	0.24753523
## SaleCondition	SaleCondition	0.23685953
## BsmtFinType1	BsmtFinType1	0.14889106
## BedroomAbvGr	BedroomAbvGr	0.11982370
## HeatingQC	HeatingQC	0.10057834
## WoodDeckSF	WoodDeckSF	0.06800782
## BsmtFullBath	BsmtFullBath	0.05992783
## MasVnrArea	MasVnrArea	0.04783078
## BsmtUnfSF	BsmtUnfSF	0.04613058
## GarageYrBlt	GarageYrBlt	0.02284752
## ExterCond	ExterCond	0.01358613
## MSSubClass	MSSubClass	0.00000000
## LotFrontage	LotFrontage	0.00000000
## Street	Street	0.00000000
## Alley	Alley	0.00000000
## LotShape	LotShape	0.00000000
## LandContour	LandContour	0.00000000
## LotConfig	LotConfig	0.00000000
## LandSlope	LandSlope	0.00000000
## Neighborhood	Neighborhood	0.00000000
## Condition1	Condition1	0.00000000
## Condition2	Condition2	0.00000000
## BldgType	BldgType	0.00000000
## HouseStyle	HouseStyle	0.00000000
## RoofStyle	RoofStyle	0.00000000
## RoofMatl	RoofMatl	0.00000000
## Exterior1st	Exterior1st	0.00000000
## Exterior2nd	Exterior2nd	0.00000000
## MasVnrType	MasVnrType	0.00000000
## Foundation	Foundation	0.00000000
## BsmtCond	BsmtCond	0.00000000
## BsmtExposure	BsmtExposure	0.00000000
## BsmtFinType2	BsmtFinType2	0.00000000
## BsmtFinSF2	BsmtFinSF2	0.00000000
## Heating	Heating	0.00000000

```

## Electrical      Electrical  0.00000000
## LowQualFinSF    LowQualFinSF 0.00000000
## BsmtHalfBath    BsmtHalfBath 0.00000000
## KitchenAbvGr    KitchenAbvGr 0.00000000
## Functional      Functional  0.00000000
## GarageQual      GarageQual  0.00000000
## GarageCond      GarageCond  0.00000000
## EnclosedPorch   EnclosedPorch 0.00000000
## X3SsnPorch      X3SsnPorch  0.00000000
## ScreenPorch     ScreenPorch  0.00000000
## PoolArea        PoolArea     0.00000000
## PoolQC          PoolQC       0.00000000
## Fence           Fence        0.00000000
## MiscFeature     MiscFeature  0.00000000
## MiscVal         MiscVal      0.00000000
## MoSold          MoSold       0.00000000
## YrSold          YrSold       0.00000000
## SaleType        SaleType     0.00000000

boost.preds <- predict(gbm.tree, newdata = all.df.nolvg[1458:2916,-77], n.trees = 500)
boost.preds <- unname(boost.preds)
boost.preds <- e~boost.preds
boost.submit <- data.frame(test.ids,boost.preds)
colnames(boost.submit) <- c("Id", "SalePrice")
#write.csv(boost.submit, "boost.csv")

```