homework4_4

syh

June 1, 2017

```
# in this section, we are gonna try to build random forest
# also, we would like to use grid search to find optimal parameters
```

```
# read all data (train + prediction) without missing value
real_all_data <- read.csv(file = "H:/kaggle/houseprice/data/real_all_data_hybrid.csv",
                          stringsAsFactors = FALSE)[,-c(1,2)]
```

```
# transform sale price to log sale price
real_all_data[,"SalePrice"] <- log(real_all_data[,"SalePrice"])
```

```
# we would like to train a linear regression model with regulation
# 1. convert categorical ones to factors
for(i in 1:dim(real_all_data)[2]){
  if(is.character(real_all_data[,i])){
    real_all_data[,i] <- as.factor(real_all_data[,i])
  }
}
```

```
# 1. split all data into train and prediction

model_data <- real_all_data[1:1460,]

pre_x <- real_all_data[-c(1:1460),]

# 2. split model data into train and test
set.seed(1000)
train_ind <- sample(1:dim(model_data)[1], size = dim(model_data)[1] * 0.7)

train_data <- model_data[train_ind,]
test_data <- model_data[-train_ind,]
```

```
# train a random forest
library(randomForest)
```

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

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

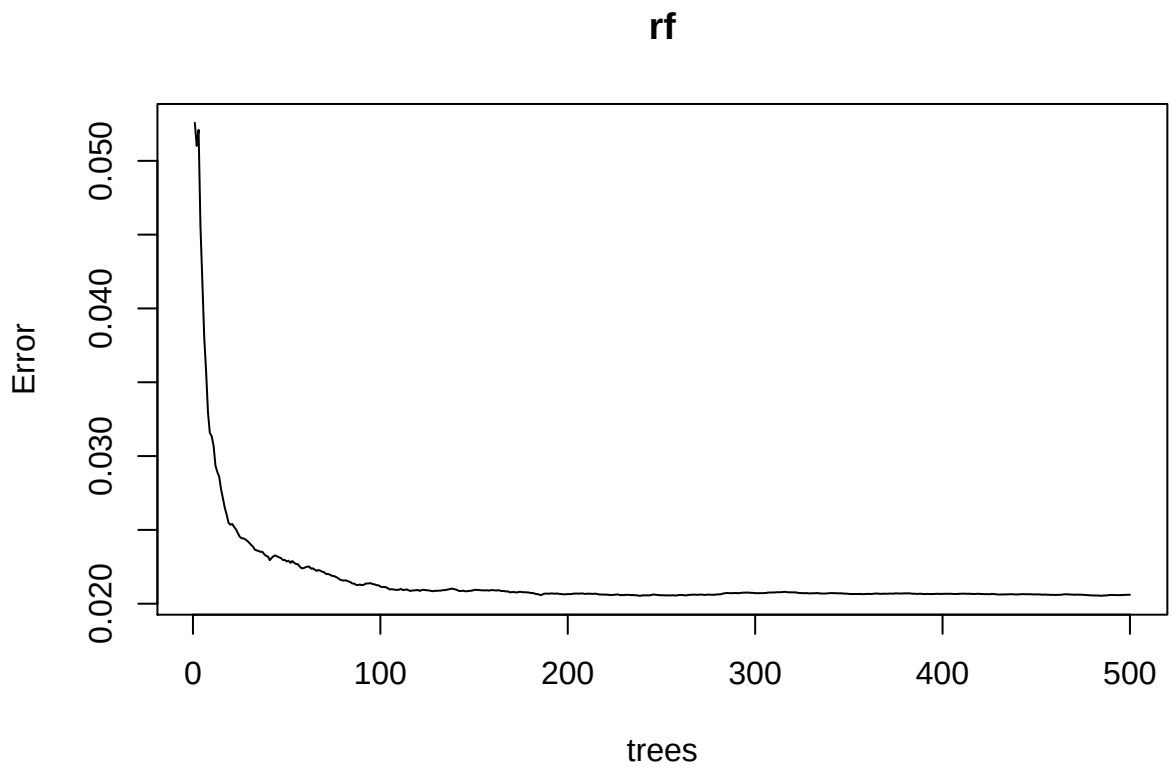
```
formula <- "SalePrice ~. - SalePrice"
set.seed(1)
rf <- randomForest(formula = as.formula(formula), data = train_data,
                   importance = TRUE, ntree = 500
                   # xtest = subset(test_data, select = -SalePrice),
                   # ytest = test_data[,"SalePrice"]
                   )
```

```
# have a look at this random forest
rf
```

```
##
## Call:
```

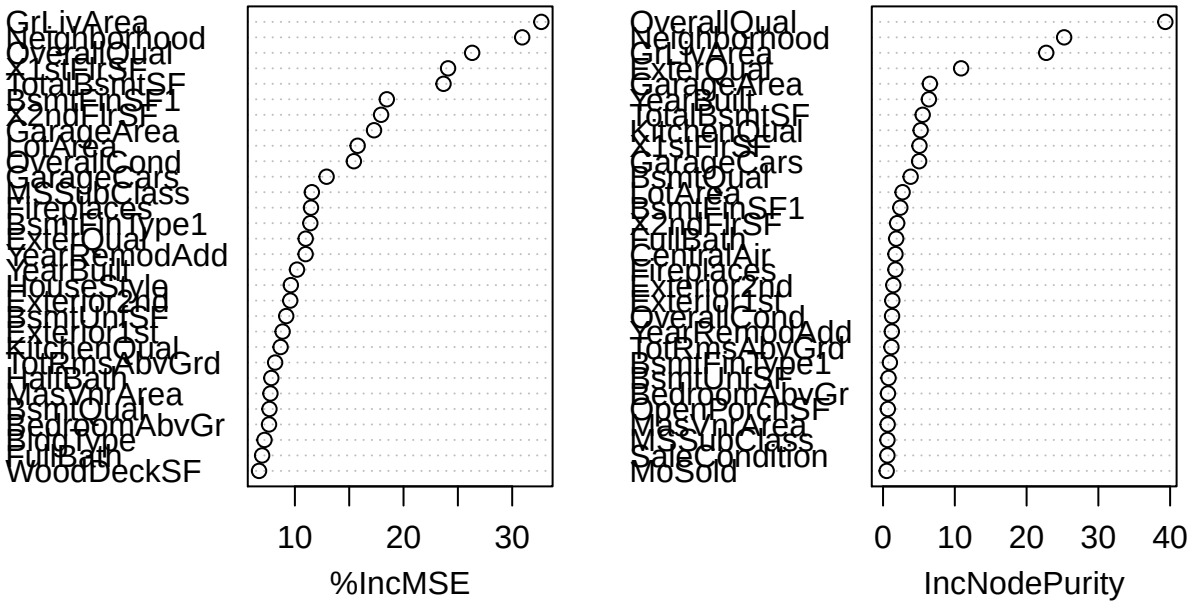
```
##  randomForest(formula = as.formula(formula), data = train_data,      importance = TRUE, ntree = 500)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 22
##
##           Mean of squared residuals: 0.02060875
##                     % Var explained: 87.59
```

```
plot(rf)
```




```
# about 100 trees, error become relatively constant
```

```
# have a look at importance of features
varImpPlot(x = rf)
```




```
# from graph below, we find these important features are same with ones used in rpart
```

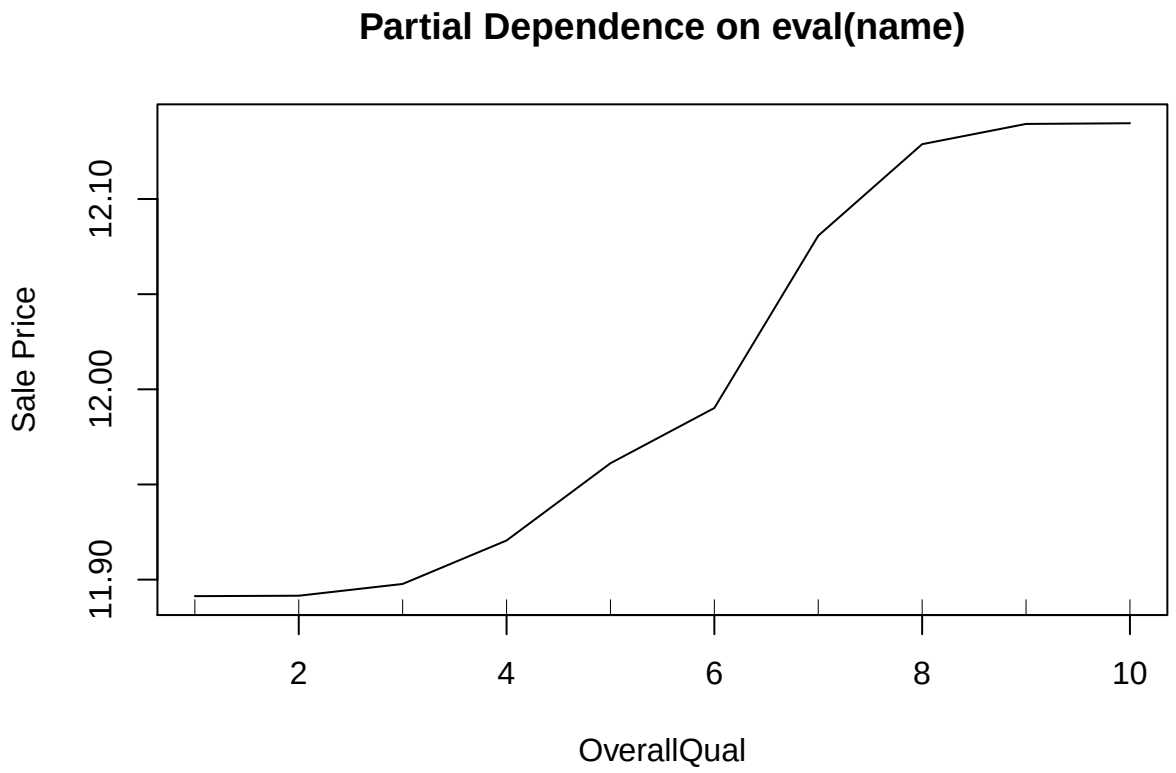
```
# importacne
rf$importance[order(-rf$importance[,"%IncMSE"]),]
```

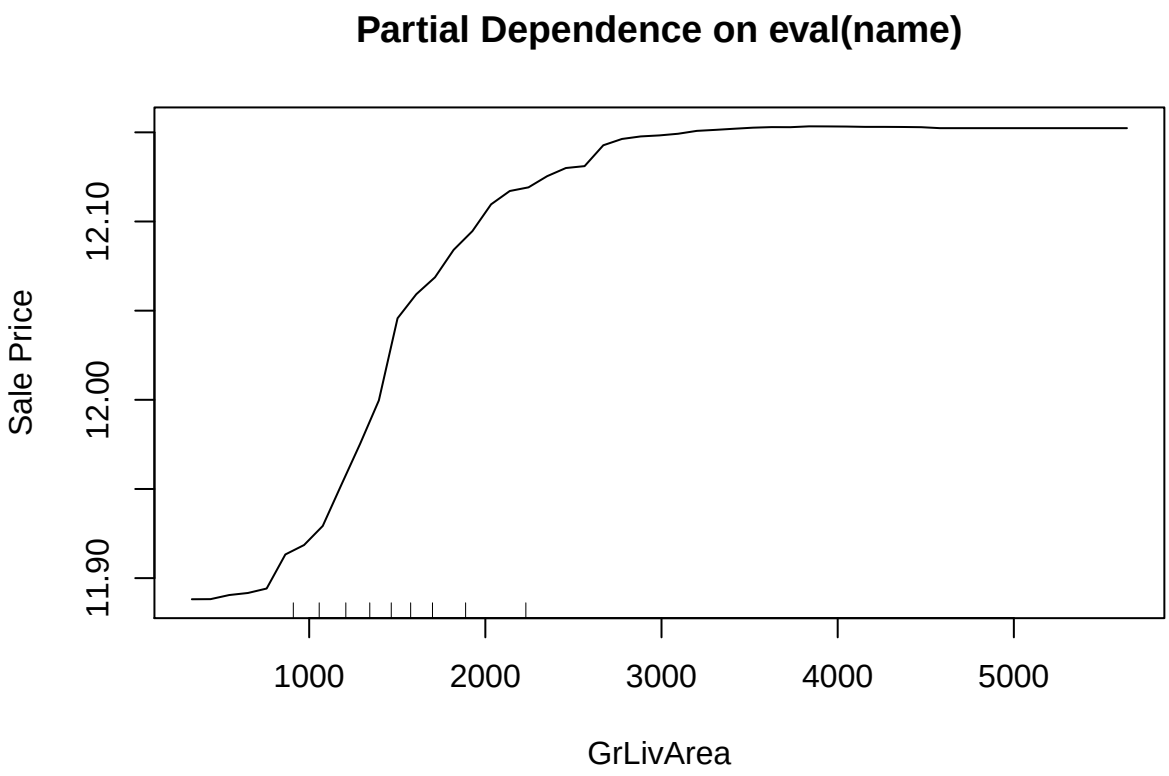
```
##                     %IncMSE IncNodePurity
## OverallQual    3.148580e-02  3.932946e+01
## GrLivArea      3.075375e-02  2.273469e+01
## Neighborhood   1.961309e-02  2.523492e+01
## TotalBsmtSF    7.440001e-03  5.521559e+00
## ExterQual      6.492283e-03  1.088007e+01
## X1stFlrSF      6.347031e-03  5.076544e+00
## YearBuilt      5.368412e-03  6.385362e+00
## GarageArea     4.591304e-03  6.523691e+00
## GarageCars     3.842325e-03  5.025631e+00
## KitchenQual    3.416878e-03  5.238965e+00
## X2ndFlrSF      2.946010e-03  1.963633e+00
## BsmtQual       2.677913e-03  3.840314e+00
## BsmtFinSF1     2.440351e-03  2.402971e+00
## LotArea        2.306143e-03  2.723933e+00
## Fireplaces     1.649463e-03  1.723764e+00
## FullBath       1.639123e-03  1.832118e+00
## YearRemodAdd   1.420180e-03  1.220581e+00
## OverallCond    1.309358e-03  1.255733e+00
## Exterior2nd    1.185529e-03  1.426146e+00
## MSSubClass     1.042792e-03  6.111629e-01
## BsmtFinType1   1.004818e-03  9.522926e-01
```

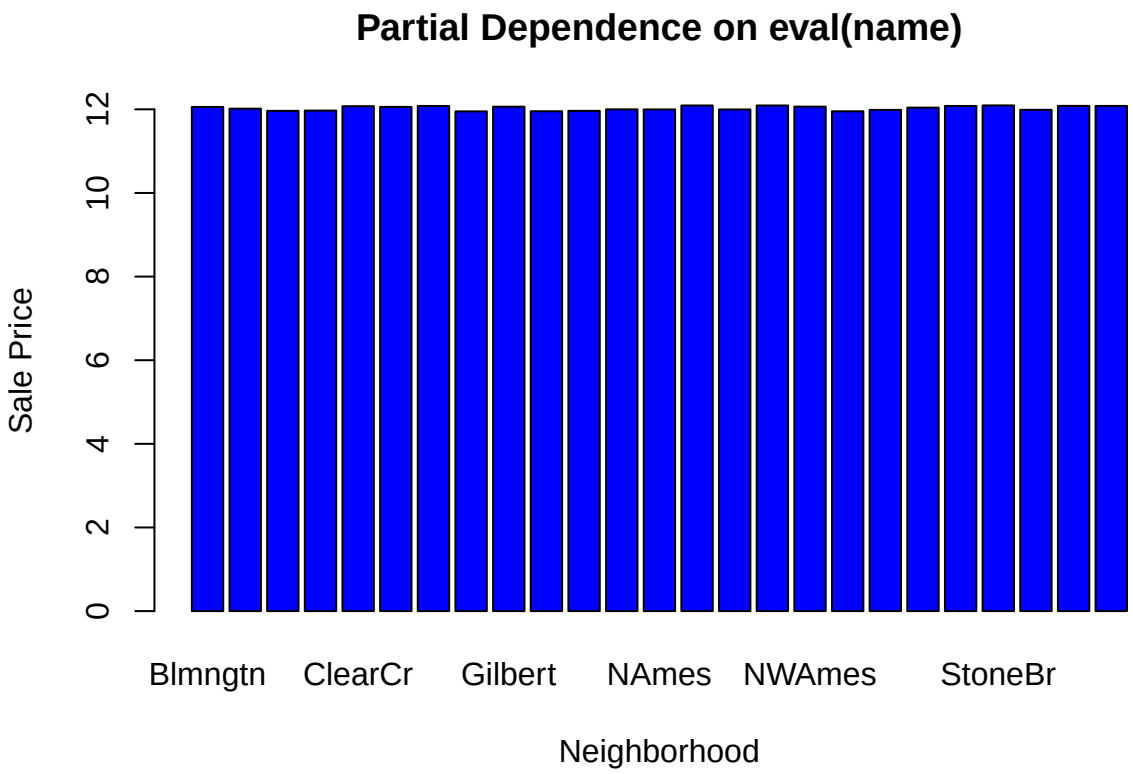
```
## Exterior1st    9.787077e-04  1.289702e+00
## TotRmsAbvGrd   7.697351e-04  1.165264e+00
## BsmtUnfSF      7.353305e-04  7.640006e-01
## CentralAir     7.171455e-04  1.728788e+00
## BedroomAbvGr   6.850154e-04  6.820703e-01
## HouseStyle     6.133715e-04  4.485347e-01
## MasVnrArea     4.981144e-04  6.377981e-01
## Foundation     4.790644e-04  4.492603e-01
## HalfBath       4.101488e-04  2.355647e-01
## MSZoning       3.867244e-04  4.928116e-01
## OpenPorchSF    3.859591e-04  6.643484e-01
## WoodDeckSF     3.188573e-04  4.529419e-01
## BldgType       2.918409e-04  2.609189e-01
## BsmtExposure   2.760152e-04  4.596846e-01
## BsmtFullBath   2.355266e-04  1.882296e-01
## BsmtCond       2.308739e-04  3.236194e-01
## Condition1     2.068819e-04  2.465797e-01
## PavedDrive     1.951225e-04  3.672062e-01
## HeatingQC      1.553498e-04  4.119391e-01
## KitchenAbvGr   1.535946e-04  1.223757e-01
## LandContour    1.426823e-04  2.487359e-01
## MasVnrType     1.393082e-04  1.501642e-01
## RoofStyle      1.311378e-04  2.494713e-01
## LotShape       1.183845e-04  2.774531e-01
## SaleCondition  8.584335e-05  6.050683e-01
## ExterCond      7.790232e-05  2.865271e-01
## ScreenPorch    6.685243e-05  1.326963e-01
## Functional     6.022592e-05  2.869502e-01
## SaleType       3.481335e-05  1.806848e-01
## BsmtFinType2   2.904259e-05  1.432463e-01
## BsmtFinSF2     1.797578e-05  9.021762e-02
## EnclosedPorch  3.383111e-06  1.617186e-01
## X3SsnPorch     5.424738e-07  1.746502e-02
## Utilities      0.000000e+00  5.338089e-04
## Street        -1.401575e-07  2.576494e-03
## RoofMatl      -3.995560e-06  3.971318e-02
## LandSlope     -4.075290e-06  1.297566e-01
## LowQualFinSF  -4.086030e-06  5.870178e-02
## Condition2    -4.115543e-06  5.531781e-02
## MoSold        -4.593197e-06  5.032809e-01
## MiscVal       -1.129638e-05  3.041375e-02
## BsmtHalfBath  -1.860837e-05  3.162967e-02
## YrSold        -3.037966e-05  2.359986e-01
## PoolArea      -3.140690e-05  1.368290e-02
## LotConfig     -3.746331e-05  1.927004e-01
## Electrical    -4.164204e-05  2.071717e-01
## Heating       -7.210249e-05  1.273305e-01
```

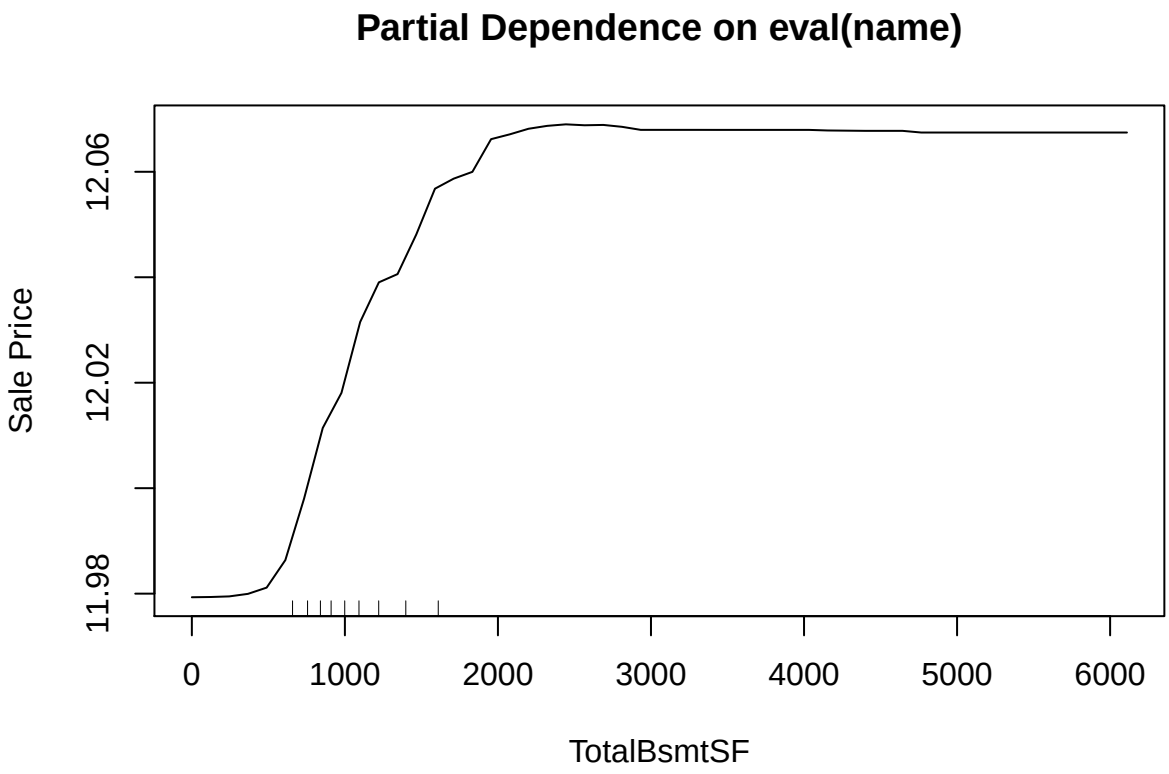
```r
# use partialPlot to check each features against saleprice
name_order <- order(-rf$importance[,"%IncMSE"])

fea_name <- rownames(rf$importance)[name_order]

for(name in fea_name[1:10]){
```

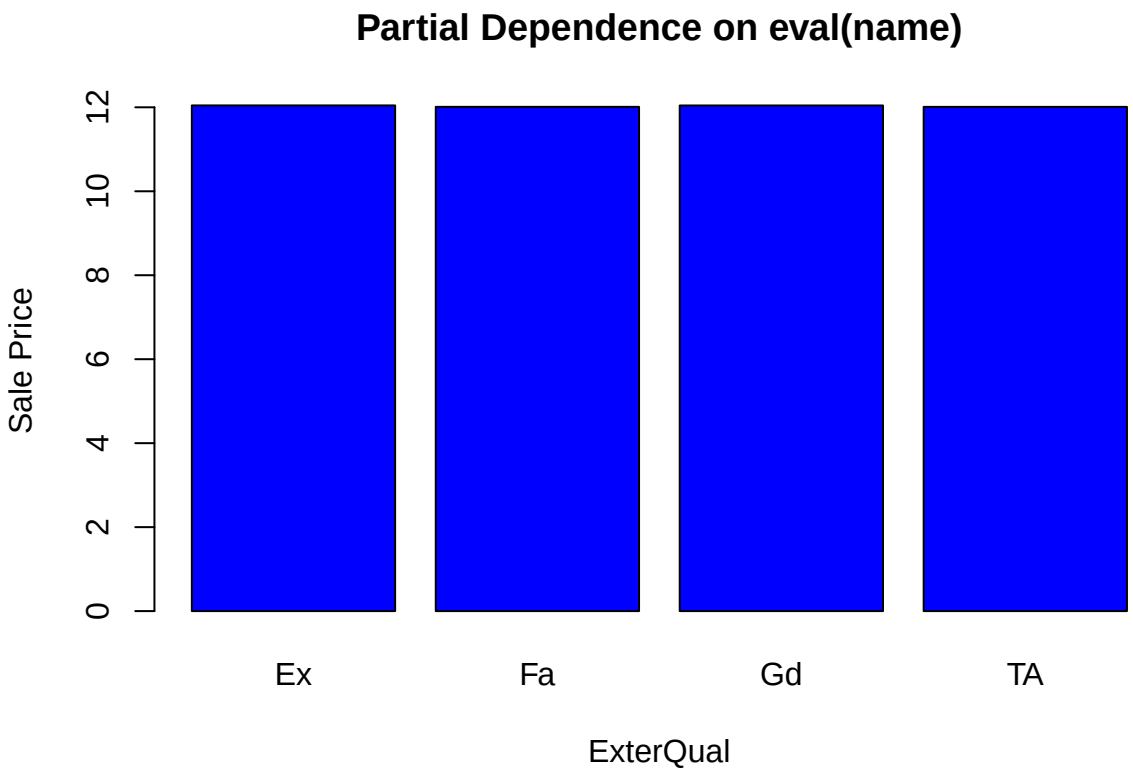
```
  partialPlot(x = rf, pred.data = train_data,
              x.var = eval(name), which.class = T,
              xlab = name, ylab = "Sale Price")
}
```

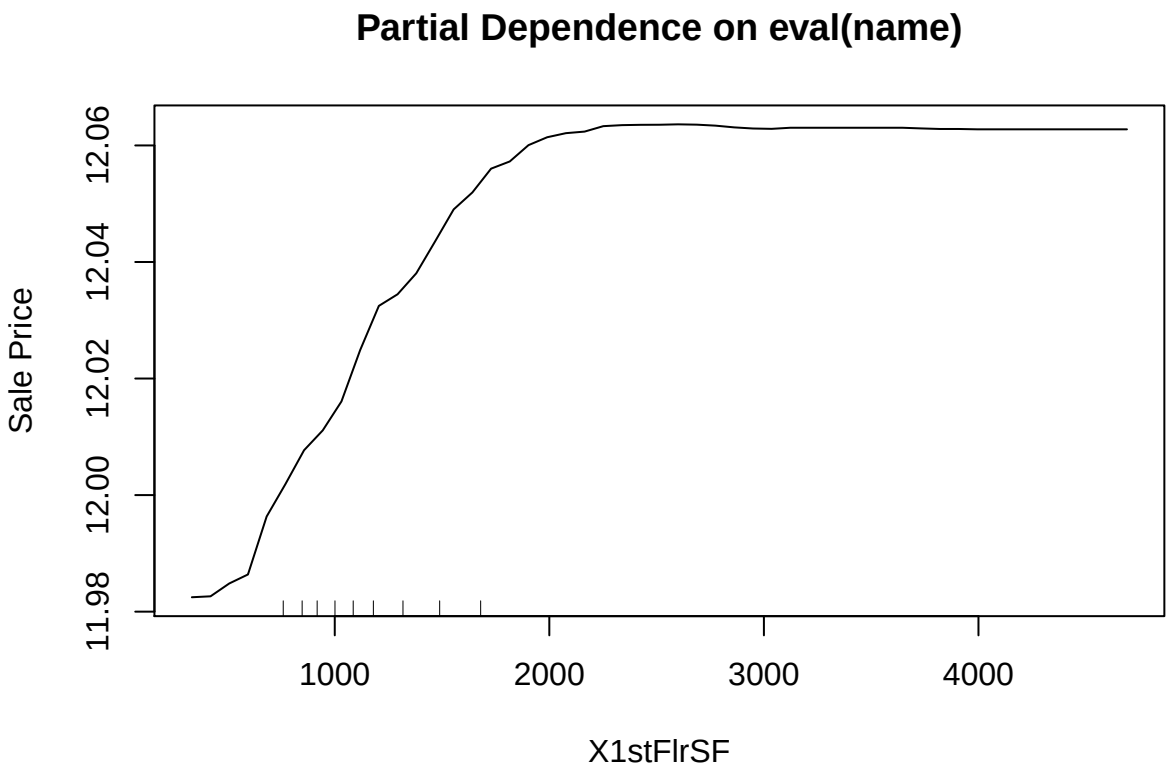


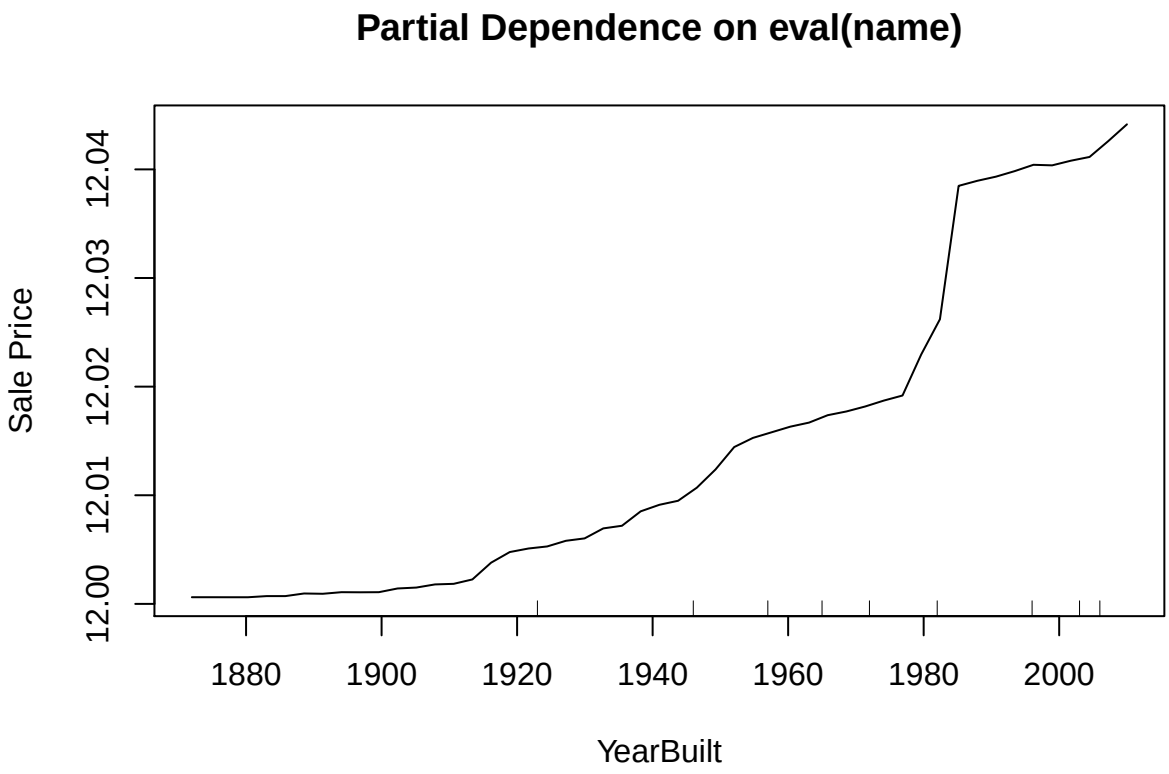

Partial Dependence on eval(name)
Sale Price
11.90
12.00
12.10
1000
2000
3000
4000
5000
GrLivArea

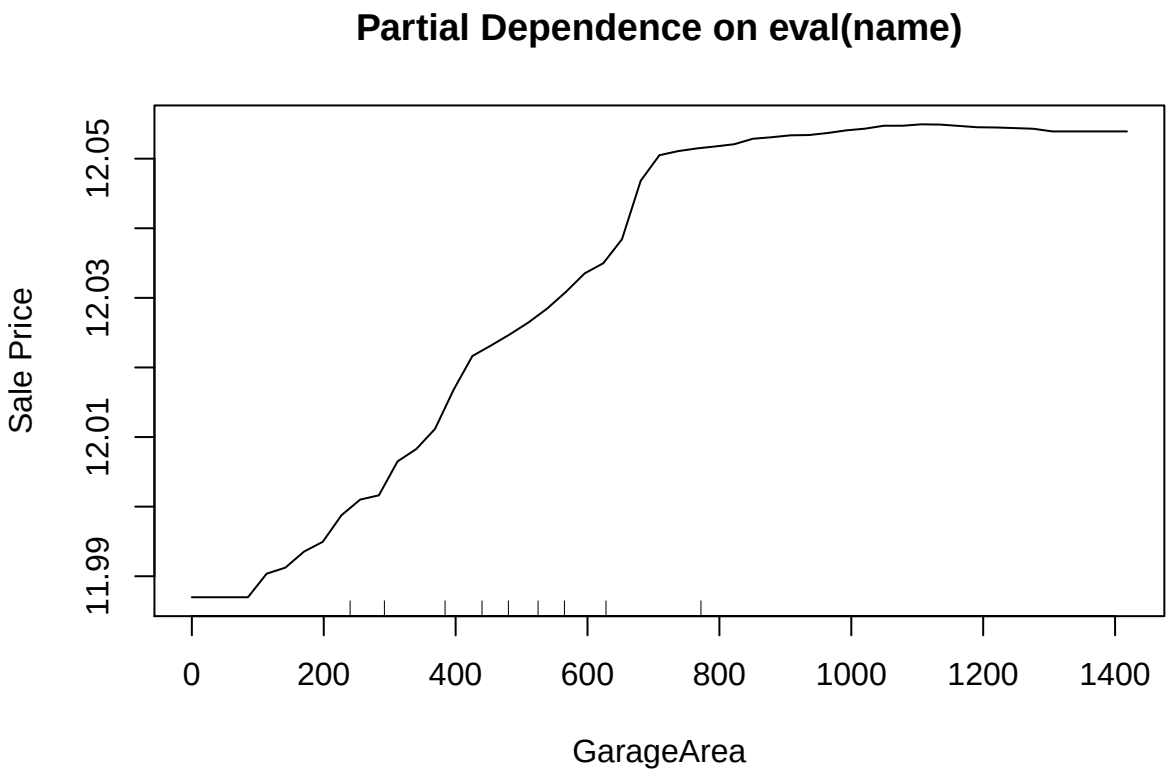

Partial Dependence on eval(name)
Sale Price
0
2
4
6
8
10
12
Blmngtn
ClearCr
Gilbert
NAmes
NWAmes
StoneBr
Neighborhood

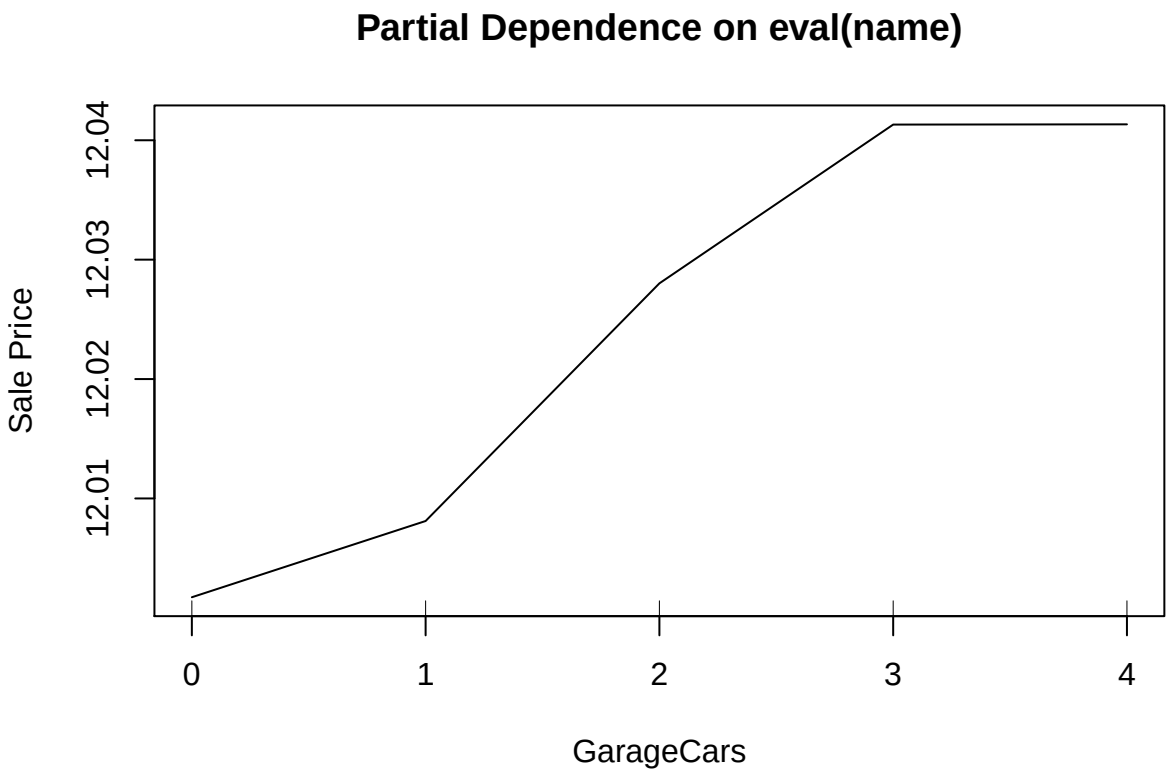

Partial Dependence on eval(name)
Sale Price
11.98
12.02
12.06
0
1000
2000
3000
4000
5000
6000
TotalBsmtSF

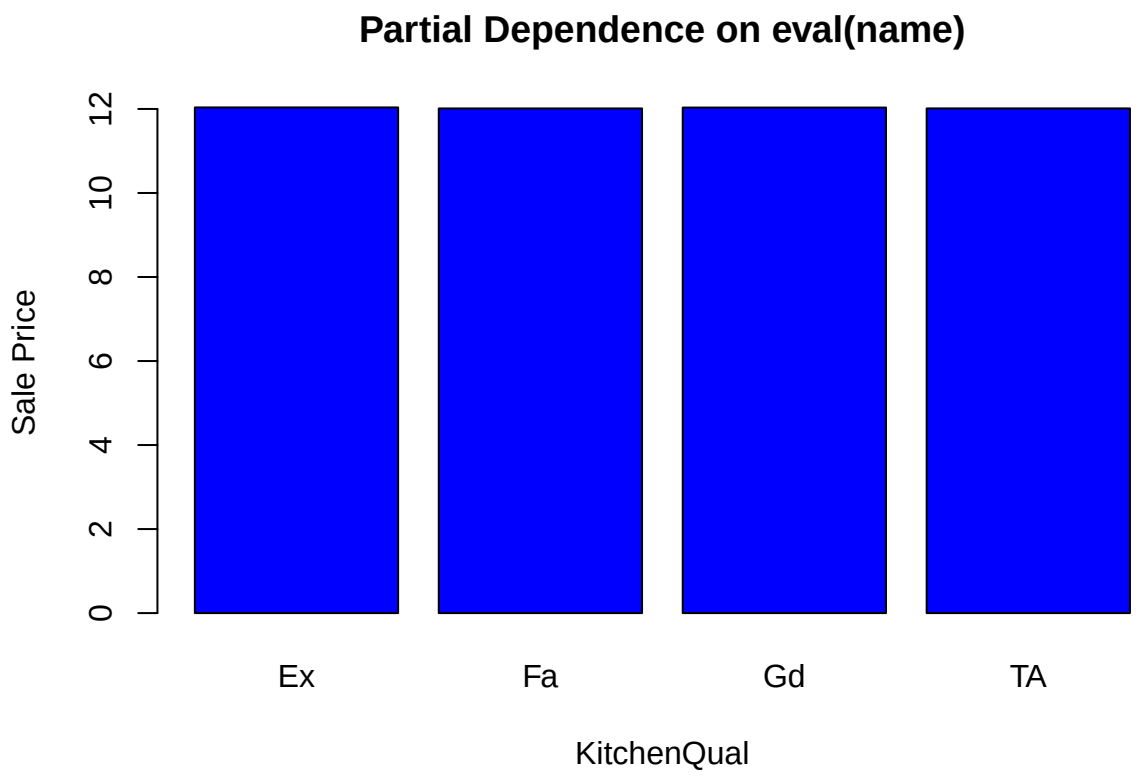

Partial Dependence on eval(name)
Sale Price
0
2
4
6
8
10
12
Ex
Fa
Gd
TA
ExterQual


Partial Dependence on eval(name)
Sale Price
11.98
12.00
12.02
12.04
12.06
1000
2000
3000
4000
X1stFlrSF


Partial Dependence on eval(name)
Sale Price
12.00
12.01
12.02
12.03
12.04
1880
1900
1920
1940
1960
1980
2000
YearBuilt


Partial Dependence on eval(name)
Sale Price
11.99
12.01
12.03
12.05
0
200
400
600
800
1000
1200
1400
GarageArea


Partial Dependence on eval(name)
Sale Price
12.01
12.02
12.03
12.04
0
1
2
3
4
GarageCars




```
# calculate SSE on test data
est_test_sp <- predict(object = rf, newdata = test_data)
sum((est_test_sp - test_data[,"SalePrice"]) ^ 2)
```

```
## [1] 6.870318
```

```
# much lower than linear regression and simple tree
```

```
# make prediction
pre_sale_price <- predict(object = rf, newdata = pre_x)
result <- data.frame(Id = c(1461:2919), SalePrice = exp(pre_sale_price))
write.csv(x = result, file = "H:/kaggle/houseprice/data/submission_6.csv",
          row.names = FALSE)
# Your submission scored 0.14541, not good.
```