Housing Prices

Load required libraries for models that we are testing

Load the data sets

This loads up the data sets dfTraining and dfAnalysis so we can use them to create models

```
rm(list = ls())
df <- read.csv("dfTrain1.csv")
```

Decision Tree

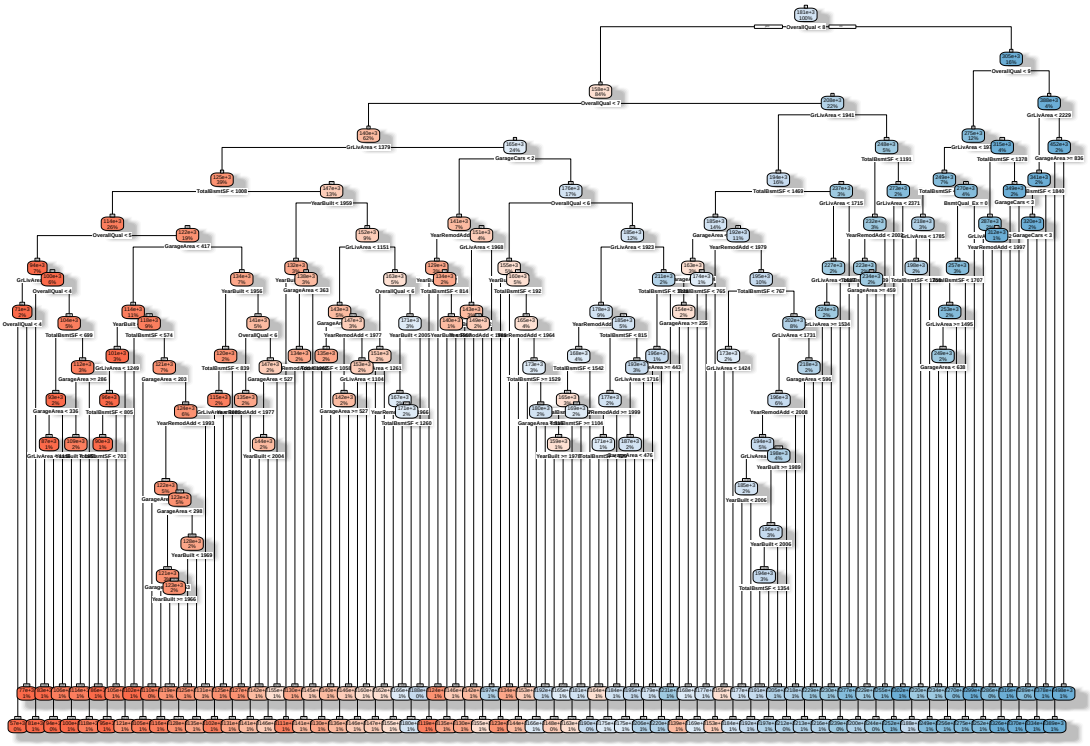
Create a decision tree model

```
# Decision Tree
tree <- rpart(formula = SalePrice ~ OverallQual + YearBuilt + YearRemodAdd + TotalBsmtSF + GrLivArea + 
```

Decision Tree plot

```
# Visualize the decision tree with rpart.plot
rpart.plot(tree, box.palette="RdBu", shadow.col="gray", nn=TRUE)
```

```
## Warning: labs do not fit even at cex 0.15, there may be some overplotting
```



Linear Regression

Regression Model

```
# Linear Regression (using variables that are high correlation)
reg <- lm(formula = SalePrice ~ OverallQual + YearBuilt + YearRemodAdd + TotalBsmtSF + GrLivArea + Garag

# Model Performance
print(reg)
```

```
##
## Call:
## lm(formula = SalePrice ~ OverallQual + YearBuilt + YearRemodAdd +
##     TotalBsmtSF + GrLivArea + GarageCars + BsmtQual_Ex + KitchenQual_Ex,
##     data = df)
##
## Coefficients:
##    (Intercept)     OverallQual       YearBuilt    YearRemodAdd     TotalBsmtSF
##     -1.004e+06       1.473e+04       2.448e+02       2.442e+02       2.231e+01
##      GrLivArea      GarageCars    BsmtQual_Ex  KitchenQual_Ex
##      4.999e+01       1.323e+04       3.708e+04       3.593e+04
```

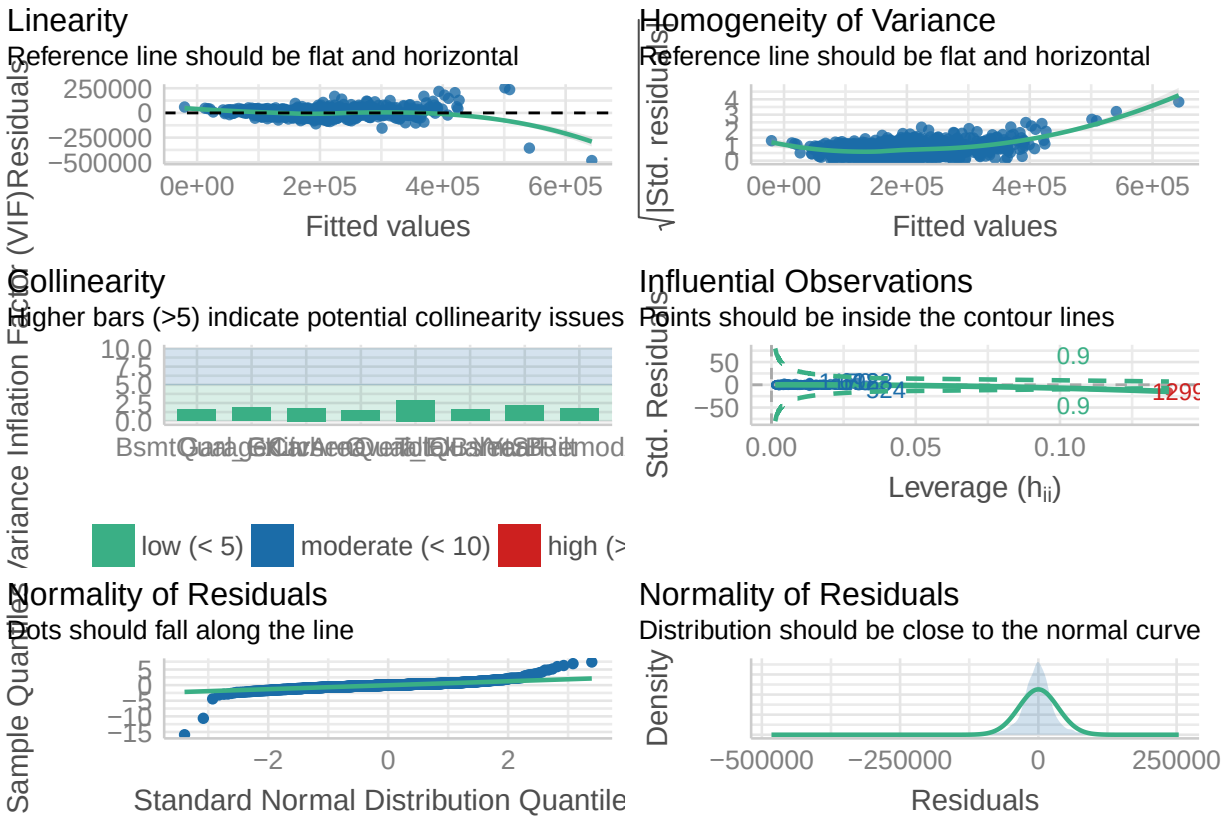
```
r2(reg)
```

```
## # R2 for Linear Regression
##        R2: 0.802
##   adj. R2: 0.801
```

```
model_performance(reg)
```

```
## # Indices of model performance
##
## AIC       |       BIC |    R2 | R2 (adj.) |      RMSE |     Sigma
## ----------------------------------------------------------------------
## 34742.881 | 34795.743 | 0.802 |     0.801 | 35328.983 | 35438.380
```

```
#Visualization of model checks
check_model(reg)
```



Neural Network

Adding a neural network model

```
# Neural Network
nn <-nnet.formula(formula = SalePrice ~ OverallQual + YearBuilt + YearRemodAdd + TotalBsmtSF + GrLivArea
```

```
## # weights:  111
```

```
## initial  value 56997365982750.078125
## final  value 9208736243659.042969
## converged
```