ECE 625 - Final Project: Predict assessed value of Edmonton's houses

Gonzalo Rubio and Breno Bahia

Preprocessing

Load some libraries first

```
library(ISLR)
library(MASS)
library(randomForest)
```

```
## randomForest 4.6-14
```

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

```
library(tree)
library(e1071)
library(gbm)
```

```
## Loading required package: survival
```

```
## Loading required package: lattice
```

```
## Loading required package: splines
```

```
## Loading required package: parallel
```

```
## Loaded gbm 2.1.3
```

```
library(neuralnet)
library(glmnet)
```

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

```
## Loading required package: foreach
```

```
## Loaded glmnet 2.0-13
```

```
library(leaps)
```

First, we preprocessed the data as to remove some predictors. Our of 35 initial features provided in the data, we deleted 15 of them. Not only we interpret some of these predictors as not useful (e.g., house suit and house stuff), we notice that some predictors were just combinations of others (e.g., full address combines house number, street name, postal code, and city). The deleted variable and associated reason for such action is given below

- taxroll number - used as ID only, not as data
- landuse description - could not find use yet
- building name - I wouldn't judge as important
- market building class - could not find use yet
- house suit - empty
- house number - relative position on street, we hope to capture that in the location info
- street name - relative position on city, we hope to capture that in the location info

- postal code - relative position on city, we hope to capture that in the location info
- city - all Edmonton
- full address - relative position on province, we hope to capture that in the location info
- build year mbc - year is being used already, mbc not yet
- site coverage - might be useful, just don't know how to use it
- geometry - could calculate area from it but this is already given in gross area
- result code - all same
- result message - all same
- result description - empty

Our understanding is that we can combine some of these features as to provide us some additional features to use in our models. As one example, even though we don't directly use the neighborhood and the address we create the variable categorical variable LOCATION, assuming values NORTH, SOUTH, EAST, WEST.

```
# Load the data
EdRSData = read.csv("EdmontonRealEstateData.csv/EdESDataProcFull.csv")

# Check missing data
apply(EdRSData,2,function(x) sum(is.na(x)))
```

```
##   PROP_TYPE  BUILD_YEAR   BUILD_AGE    NET_AREA    BASEMENT      GARAGE
##           0           4           0           0           0           0
##   FIREPLACE     ASS_VAL    LOT_SIZE  BASEMENTWO          AC VALUE_GROUP
##           0           0         819           0           0           0
##   DISP_TYPE    SITE_COV  GROSS_AREA         LON         LAT          NS
##           0           0           0           0           0           0
##          WE         NGR         GLR
##           0           0         819
```

```
# I'll get rid of them for now (bc Idk what todo)
EdRSData = EdRSData[!is.na(EdRSData$LOT_SIZE[]),];

# Check missing data
apply(EdRSData,2,function(x) sum(is.na(x)))
```

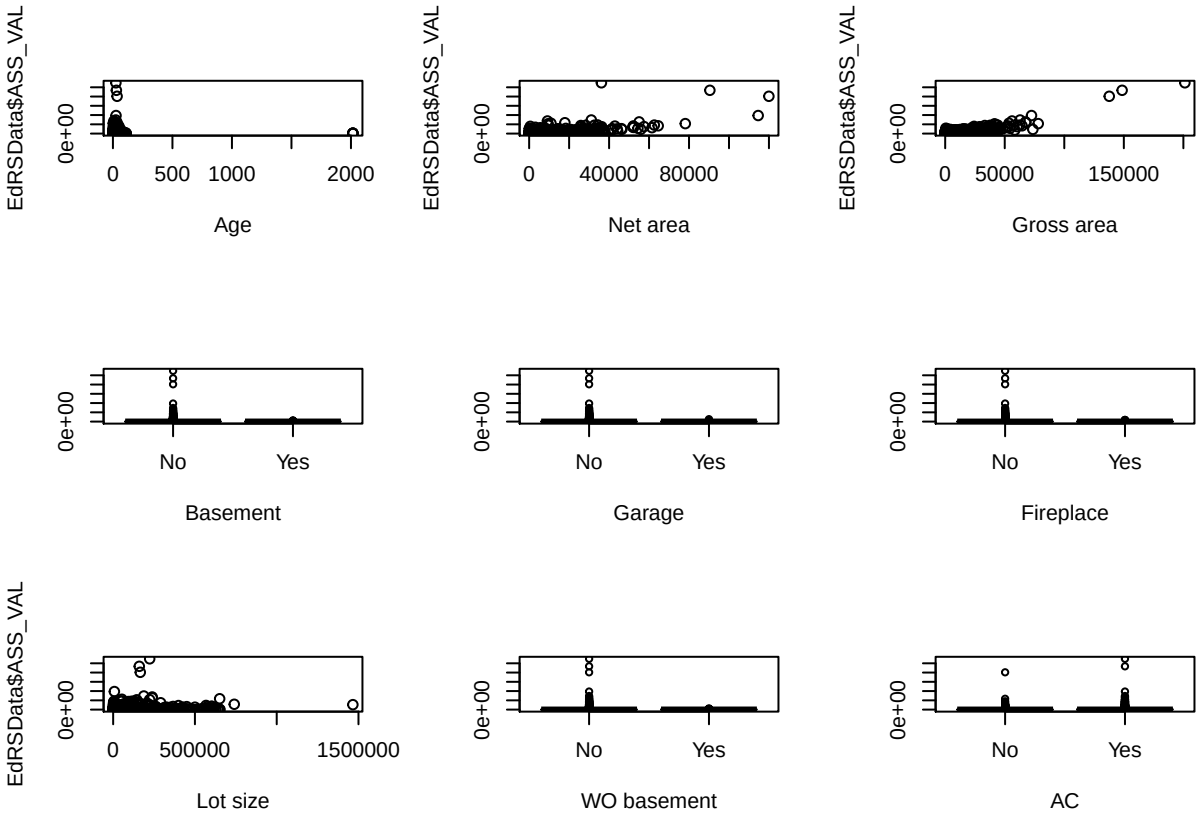
```
##   PROP_TYPE  BUILD_YEAR   BUILD_AGE    NET_AREA    BASEMENT      GARAGE
##           0           4           0           0           0           0
##   FIREPLACE     ASS_VAL    LOT_SIZE  BASEMENTWO          AC VALUE_GROUP
##           0           0           0           0           0           0
##   DISP_TYPE    SITE_COV  GROSS_AREA         LON         LAT          NS
##           0           0           0           0           0           0
##          WE         NGR         GLR
##           0           0           0
```

Other categorical variables exist, but we'll handle them later. We follow with splitting the data in training and testing datasets

We divide our dataset into two since we could see a more straigthforward relationship betweeon our numeric target and some of the predictors. We plot the data against these predictors:

```
par(mfrow = c(3,3))

plot(EdRSData$BUILD_AGE,EdRSData$ASS_VAL, xlab = "Age")
```

```
plot(EdRSData$NET_AREA,EdRSData$ASS_VAL, xlab = "Net area")
plot(EdRSData$GROSS_AREA,EdRSData$ASS_VAL, xlab = "Gross area")
plot(EdRSData$BASEMENT,EdRSData$ASS_VAL, xlab = "Basement")
plot(EdRSData$GARAGE,EdRSData$ASS_VAL, xlab = "Garage")
plot(EdRSData$FIREPLACE,EdRSData$ASS_VAL, xlab = "Fireplace")
plot(EdRSData$LOT_SIZE,EdRSData$ASS_VAL, xlab = "Lot size")
plot(EdRSData$BASEMENTWO,EdRSData$ASS_VAL, xlab = "WO basement")
plot(EdRSData$AC,EdRSData$ASS_VAL, xlab = "AC")
```



Luckly, we can already see some possible patterns with these data. For instance, the assessed value seems to increase linearly with gross area, except by some outliers.

Data splitting

We then select our data. We use $round(0.9 * (nrow(EdRSData)))$ to automatically extract 90% of our samples as training dataset.

```
# First we fix too large values for year by giving it age 0 (not sure why Im doing this)
EdRSData$BUILD_AGE[EdRSData$BUILD_AGE > 100] <- 100

# Set seed for random sampling - seed is my student ID, as I've been doing with the assignments
set.seed(1449808)

# Percentage of data to keep
Keep = 90;
```

```r
# Sampling - 70% as training dataset
Train = sample(1:nrow(EdRSData), round(Keep/100*nrow(EdRSData)))

# Extract training data
TrainEd = EdRSData[Train,]

# Extract testing data
TestEd = EdRSData[-Train,]
```

Assessing building feature information

We then check on possible relationships between our response (Assessed Value) and some selected variables. We selected some "building features"

- Age
- Gross Area
- Lot Size
- Basement
- Fireplace
- Net area

to be included in our regression analysis. This first analysis does not consider any spatial information, as our intuition says that spatial information is supposed to be more influential, in this analysis, than what we are naming "building feature". We also want to add variables as property type in this stage, but we are not sure how yet.

To chekc the relationship between the assessed value and year, instead of using the year as is, we use the building age related to 2016.

```r
# Get unique years - set Year as a matrix so you can use nrow/ncol to get their dimensions
Year =as.matrix(sort(unique(TrainEd$BUILD_AGE)))
n = nrow(Year)

# Allocate memory
AvgVal = rep(0,n)

# Get mean value for assessed value on each year
for (i in 1:n){
  AvgVal[i] = mean(TrainEd$ASS_VAL[TrainEd$BUILD_AGE == Year[i]])
}
```

We see a linear trend with year and average assessed value. We trying fitting an linear model:

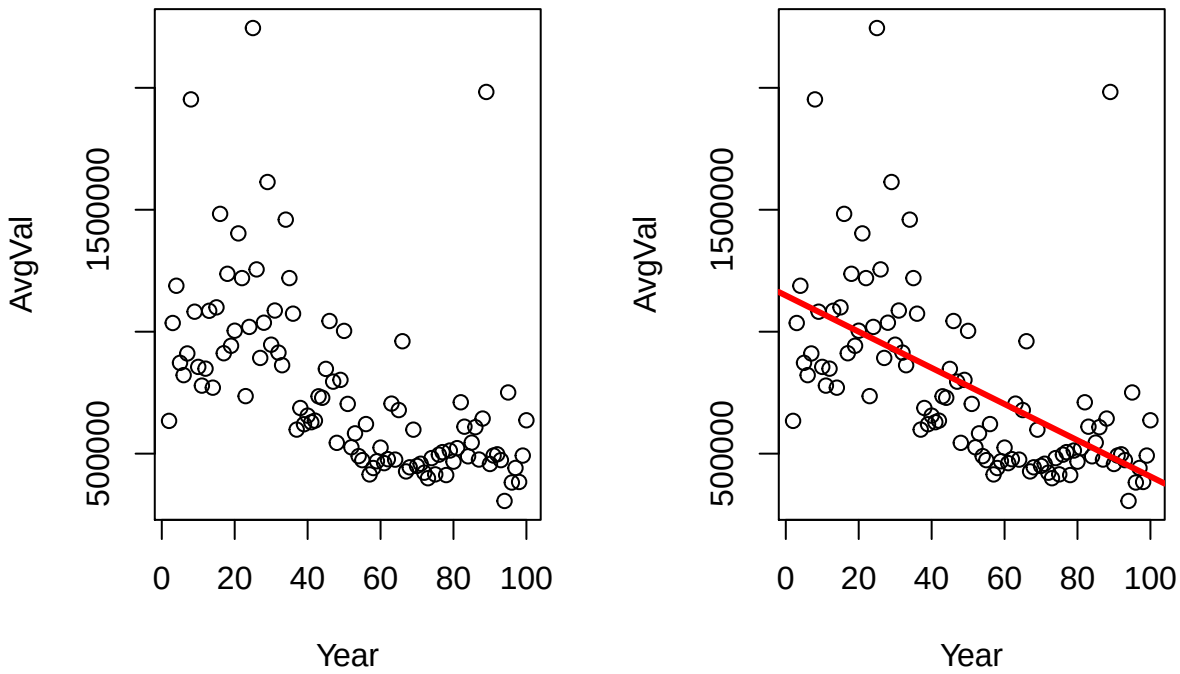
```r
# Fit linear model using lasso to avoid biasing by high leaverage points
FitLM = glm(AvgVal~Year)
summary(FitLM)
```

```
##
## Call:
## glm(formula = AvgVal ~ Year)
##
## Deviance Residuals:
```

```
##     Min       1Q   Median       3Q      Max
## -499321  -196834   -49470    66152  1495202
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1148482      61634   18.63  < 2e-16 ***
## Year           -7423       1054   -7.04 2.75e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 89864613032)
##
##     Null deviance: 1.3171e+13  on 98  degrees of freedom
## Residual deviance: 8.7169e+12  on 97  degrees of freedom
## AIC: 2781.9
##
## Number of Fisher Scoring iterations: 2
```

We expected that newer buildings would have, in average, larger assessed values than older buildings. The linear model points to such conclusion. We check its p-value to confirm a possible relationship between average assessed value and year. We agreed on the p-value as to show that the building age is a factor on its average assessed value.

```
# Divide area for plotting
par(mfrow = c(1,2))

# Plot fitted line using abline
plot(Year,AvgVal)

plot(Year,AvgVal)
abline(FitLM,lwd=3,col="red")
```



We then check on a possible relationship between gross area and the assessed value. Our intuition says that the larger the gross area, the larger the assessed value. We expect a quasi-linear relationship. Fit a linear

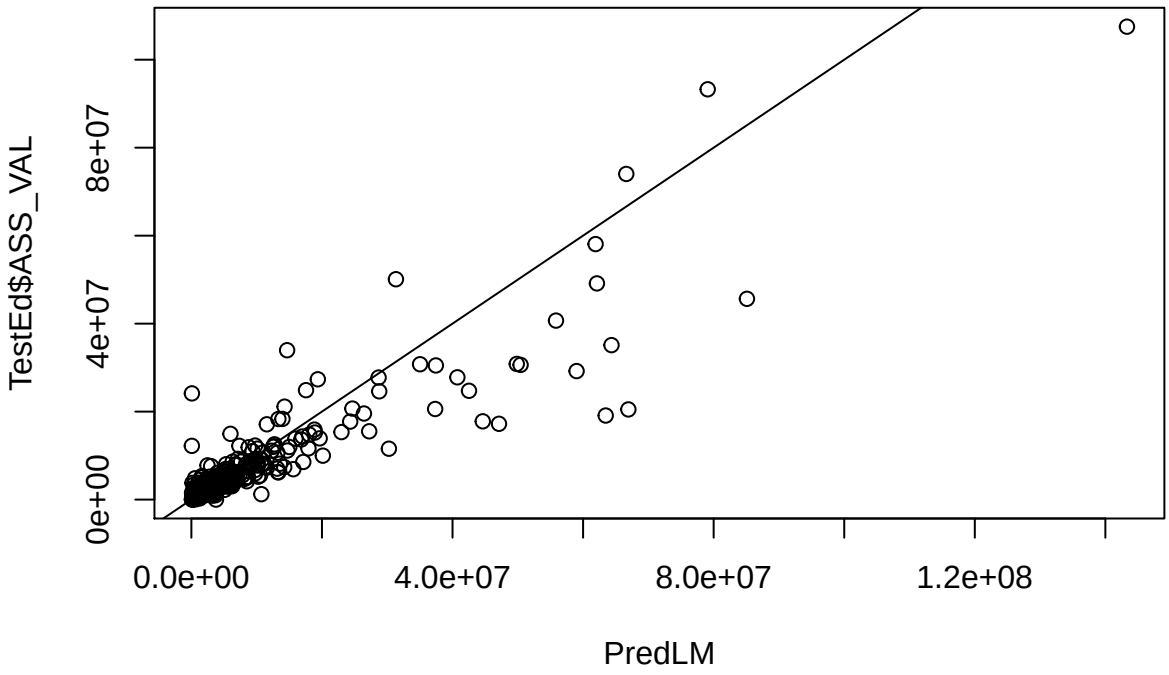
model to this two as well

```
# Fit linear model
FitLM = glm(ASS_VAL~GROSS_AREA, data = TrainEd)
summary(FitLM)
```

```
##
## Call:
## glm(formula = ASS_VAL ~ GROSS_AREA, data = TrainEd)
##
## Deviance Residuals:
##       Min         1Q     Median         3Q        Max
## -89642322     -11454      28373      84880  195014910
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 47320.151   8733.739   5.418 6.05e-08 ***
## GROSS_AREA   1833.423      3.762 487.343  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 4.160786e+12)
##
##     Null deviance: 1.2240e+18  on 56677  degrees of freedom
## Residual deviance: 2.3582e+17  on 56676  degrees of freedom
## AIC: 1807726
##
## Number of Fisher Scoring iterations: 2
```

Get some predictions on training data

```
# Prediction on test data
PredLM = predict(FitLM, TestEd)
# plot
plot(PredLM,TestEd$ASS_VAL)
abline(0,1)
```



The training error is

```
mean((PredLM - TestEd$ASS_VAL)^2)
```

```
## [1] 2.636525e+12
```

and its squared root gives us the variance

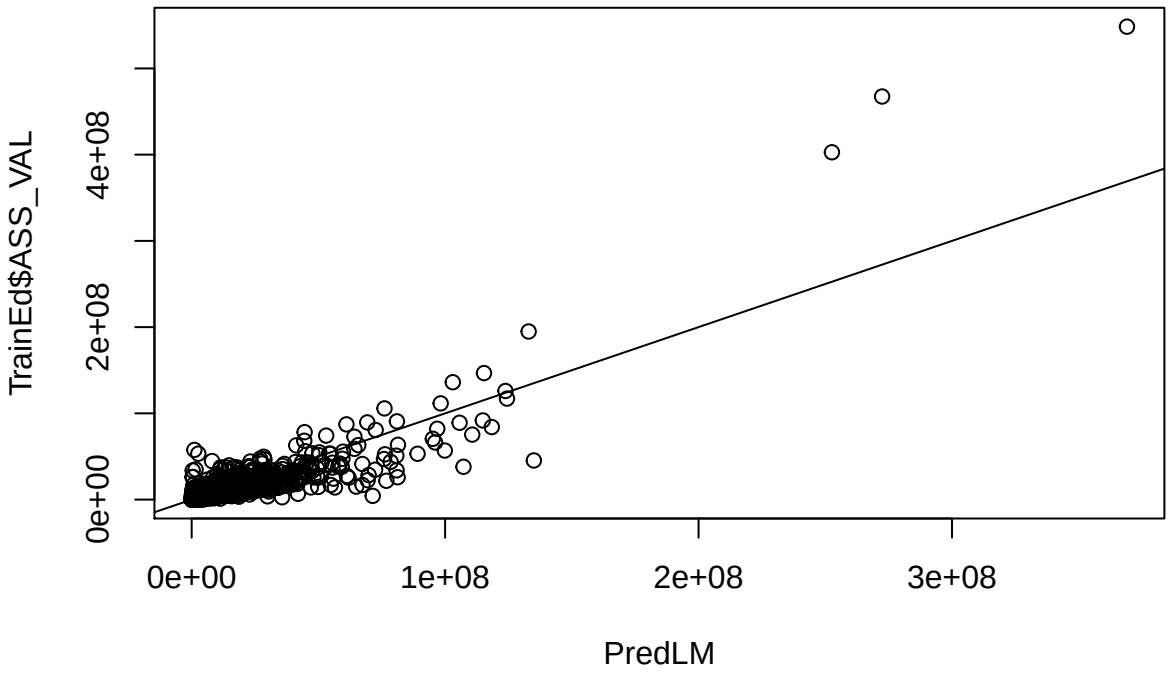
```
sqrt(mean((PredLM - TestEd$ASS_VAL)^2))
```

```
## [1] 1623738
```

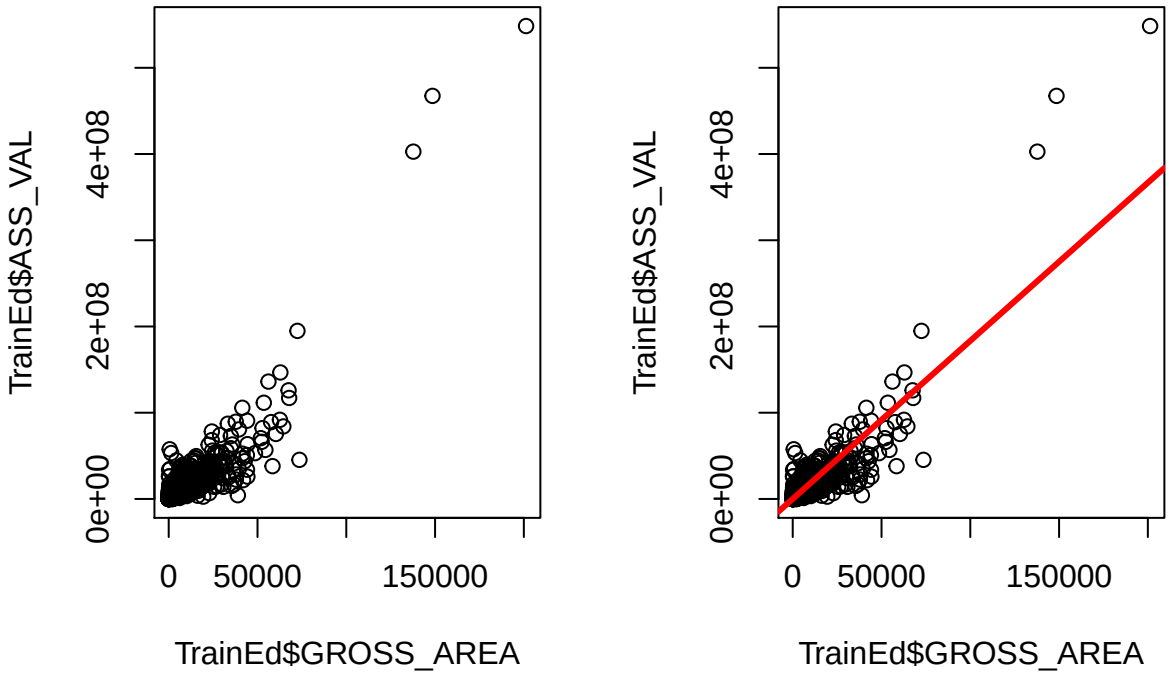
So, this model predicts the assessed values of the houses with an standard error of $ 471,732.5. Not good!

Get some predictions on test data

```
# Prediction on test data
PredLM = predict(FitLM, TrainEd)
# plot
plot(PredLM,TrainEd$ASS_VAL)
abline(0,1)
```




```
par(mfrow = c(1,2))
plot(TrainEd$GROSS_AREA,TrainEd$ASS_VAL)

plot(TrainEd$GROSS_AREA,TrainEd$ASS_VAL)
# Plot fitted line using abline
abline(FitLM,lwd=3,col="red")
```



The p-values shows statistical significance in this fit for both coefficients. We decided to stick to it gross area as a predictor in our approach.

Likewise, we expect some proportionality between assessed value and lot size. We acknowledge that there might be a relationship between lot size and gross area. We fit a linear model between assessed value and lot

size:

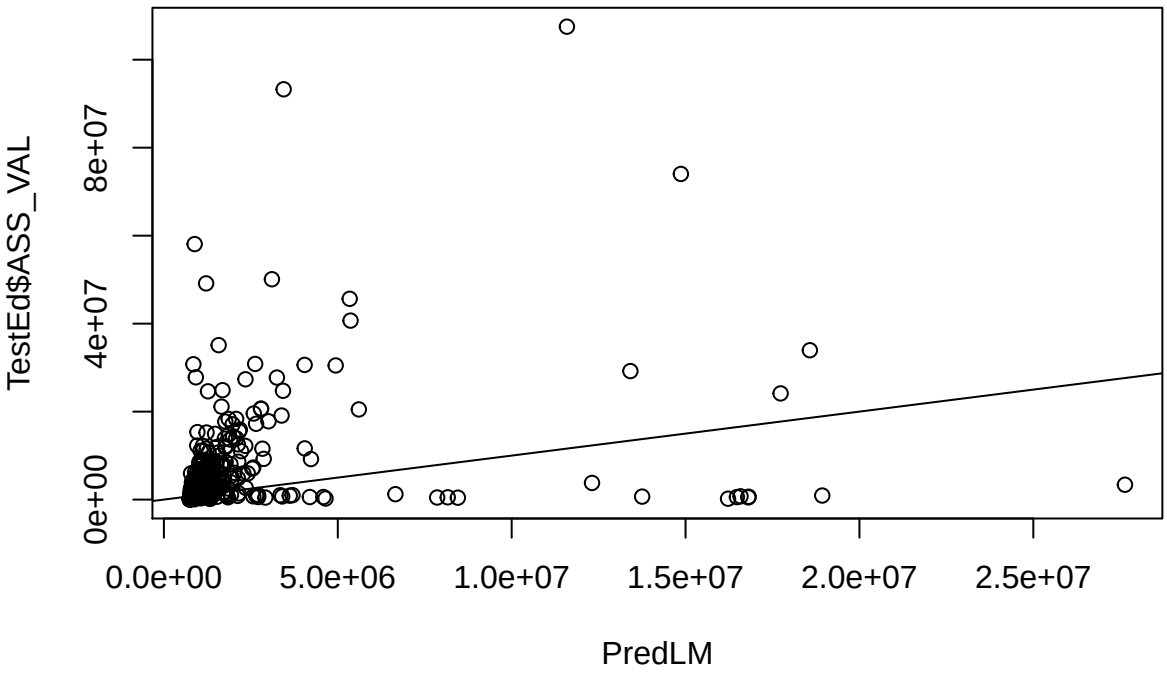
```
# Fit linear model
FitLM = glm(ASS_VAL~LOT_SIZE, data = TrainEd)
summary(FitLM)
```

```
##
## Call:
## glm(formula = ASS_VAL ~ LOT_SIZE, data = TrainEd)
##
## Deviance Residuals:
##       Min         1Q     Median         3Q        Max
## -31881011    -432897    -350238    -218054  536803536
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 7.456e+05  1.908e+04   39.08   <2e-16 ***
## LOT_SIZE    4.864e+01  8.312e-01   58.52   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 2.036629e+13)
##
##     Null deviance: 1.2240e+18  on 56677  degrees of freedom
## Residual deviance: 1.1543e+18  on 56676  degrees of freedom
## AIC: 1897741
##
## Number of Fisher Scoring iterations: 2
```

where the p-value indicates relationship between assessed value and lot size. However, although intuitively related, it is arguably hard to observe a linear relationship between lot size and assessed value (see plots).

Get some predictions on training data

```
# Prediction on test data
PredLM = predict(FitLM, TestEd)
# plot
plot(PredLM,TestEd$ASS_VAL)
abline(0,1)
```



The training error is

```
mean((PredLM - TestEd$ASS_VAL)^2)
```

```
## [1] 8.346468e+12
```

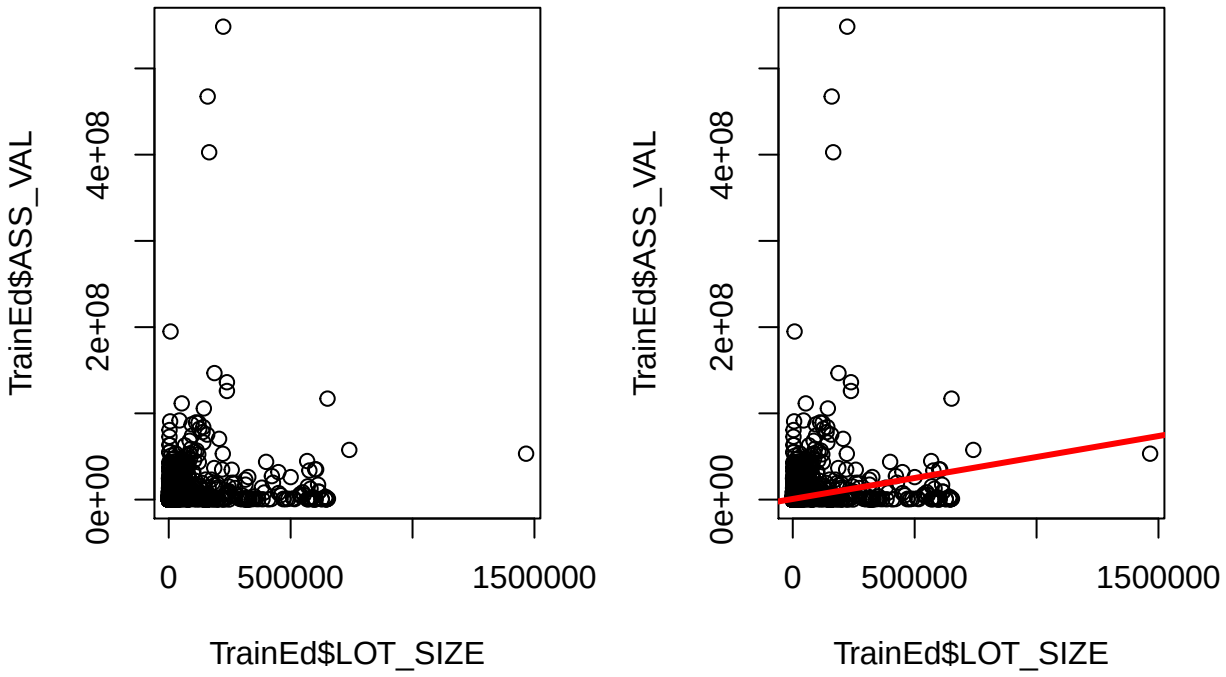
and its squared root gives us the variance

```
sqrt(mean((PredLM - TestEd$ASS_VAL)^2))
```

```
## [1] 2889025
```

So, this model predicts the assessed values of the houses with an standard error of $ NAN (probably overflow). Not good!

```
# Divide area
par(mfrow = c(1,2))
plot(TrainEd$LOT_SIZE,TrainEd$ASS_VAL)

# Plot fitted line using abline
plot(TrainEd$LOT_SIZE,TrainEd$ASS_VAL)
abline(FitLM,lwd=3,col="red")
```



The plots above suggest a non-linear relationship between assessed value and lot size. This also intuitively makes sense, as the assessed value does not increase equaly my m2, for instance.

We fit a non-linear model between assessed value and log of lot size:

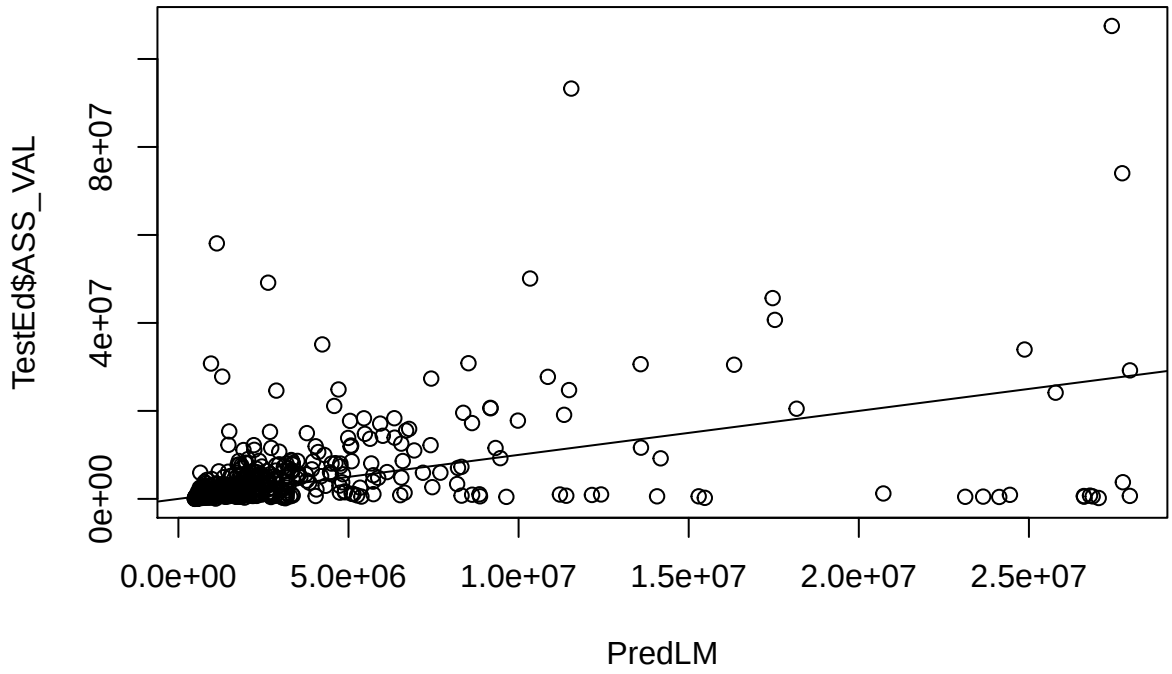
```
# Fit linear model
FitLM = lm(ASS_VAL~poly(LOT_SIZE, 3), data = TrainEd)
summary(FitLM)
```

```
##
## Call:
## lm(formula = ASS_VAL ~ poly(LOT_SIZE, 3), data = TrainEd)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -27791710   -273484   -184040    -74796 520967436
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)             872537      18064   48.30   <2e-16 ***
## poly(LOT_SIZE, 3)1  264078920    4300648   61.40   <2e-16 ***
## poly(LOT_SIZE, 3)2 -185863490    4300648  -43.22   <2e-16 ***
## poly(LOT_SIZE, 3)3  267425970    4300648   62.18   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4301000 on 56674 degrees of freedom
## Multiple R-squared:  0.1436, Adjusted R-squared:  0.1436
## F-statistic:  3168 on 3 and 56674 DF,  p-value: < 2.2e-16
```

where the p-value indicates relationship between assessed value and lot size. However, although intuitively related, it is arguably hard to observe a linear relationship between lot size and assessed value (see plots).

Get some predictions on training data

```r
# Prediction on test data
PredLM = predict(FitLM, TestEd)
# plot
plot(PredLM,TestEd$ASS_VAL)
abline(0,1)
```



The training error is

```r
mean((PredLM - TestEd$ASS_VAL)^2)
```

```
## [1] 6.852424e+12
```

and its squared root gives us the variance

```r
sqrt(mean((PredLM - TestEd$ASS_VAL)^2))
```

```
## [1] 2617714
```

It gets better than for the linear model but still bad.

As we are not sure about the lot size in our relationship here, we move to fitting a linear model to explain the assessed value using these three variables first: Year, Gross Area and Lot Size. We can then assess if lot size is actually significant or not when taking other features into our approach (also true for year and gross area).

```r
# Fit linear model
FitLM = lm(ASS_VAL~GROSS_AREA+poly(LOT_SIZE,3)+BUILD_AGE, data = TrainEd)
summary(FitLM)
```

```
##
## Call:
## lm(formula = ASS_VAL ~ GROSS_AREA + poly(LOT_SIZE, 3) + BUILD_AGE,
##     data = TrainEd)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -89634835    -31205     28184    100421 194408574
##
```

```
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          1.317e+05  1.610e+04   8.180 2.91e-16 ***
## GROSS_AREA           1.843e+03  4.123e+00 447.086  < 2e-16 ***
## poly(LOT_SIZE, 3)1   4.178e+07  2.081e+06  20.077  < 2e-16 ***
## poly(LOT_SIZE, 3)2   5.237e+07  2.089e+06  25.069  < 2e-16 ***
## poly(LOT_SIZE, 3)3 -9.257e+06  2.112e+06  -4.383 1.17e-05 ***
## BUILD_AGE          -3.046e+03  4.625e+02  -6.587 4.53e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2019000 on 56672 degrees of freedom
## Multiple R-squared:  0.8112, Adjusted R-squared:  0.8112
## F-statistic: 4.87e+04 on 5 and 56672 DF,  p-value: < 2.2e-16
```

Going back to the points of the building age, we see here that its p-value is small enough to be taken in consideration. We shall therefore include age from our model as a numerical value.

```
par(mfrow = c(1,3))
plot(TrainEd$BUILD_AGE,TrainEd$ASS_VAL)
# Plot fitted line using abline
abline(FitLM,lwd=3,col="red")
```

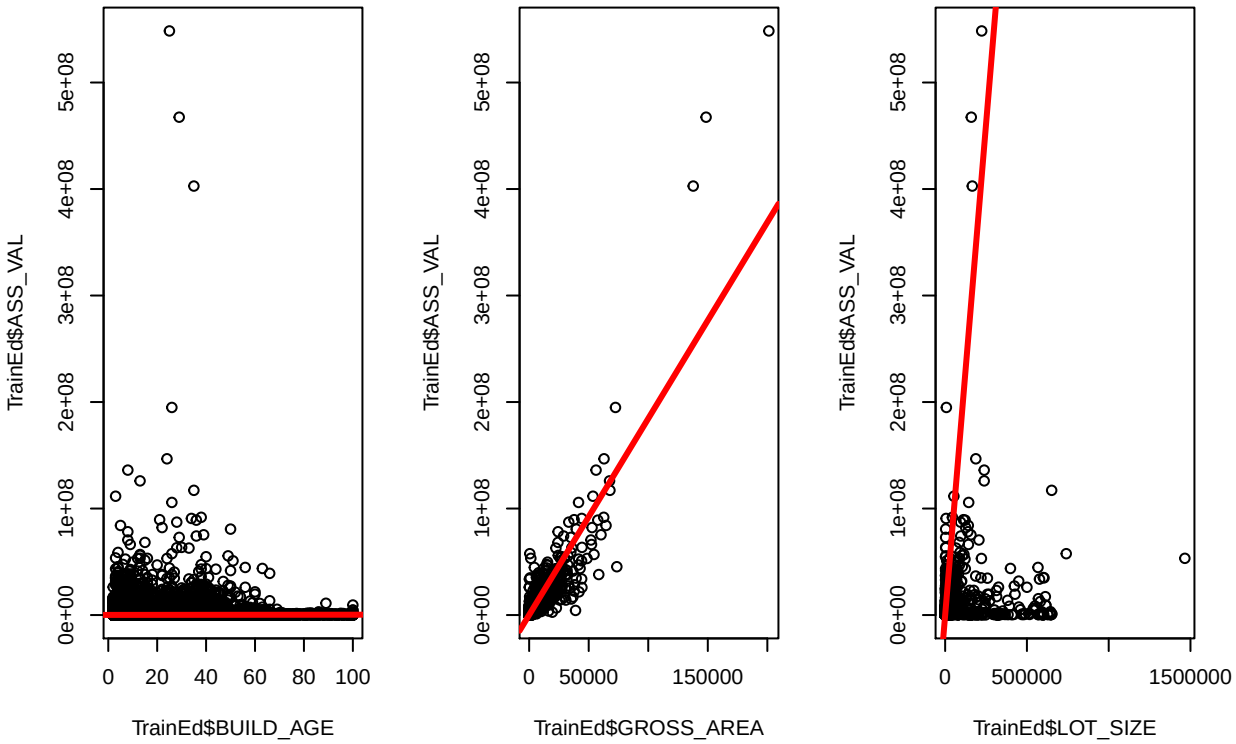
```
## Warning in abline(FitLM, lwd = 3, col = "red"): only using the first two of
## 6 regression coefficients
```

```
plot(TrainEd$GROSS_AREA,TrainEd$ASS_VAL)
# Plot fitted line using abline
abline(FitLM,lwd=3,col="red")
```

```
## Warning in abline(FitLM, lwd = 3, col = "red"): only using the first two of
## 6 regression coefficients
```

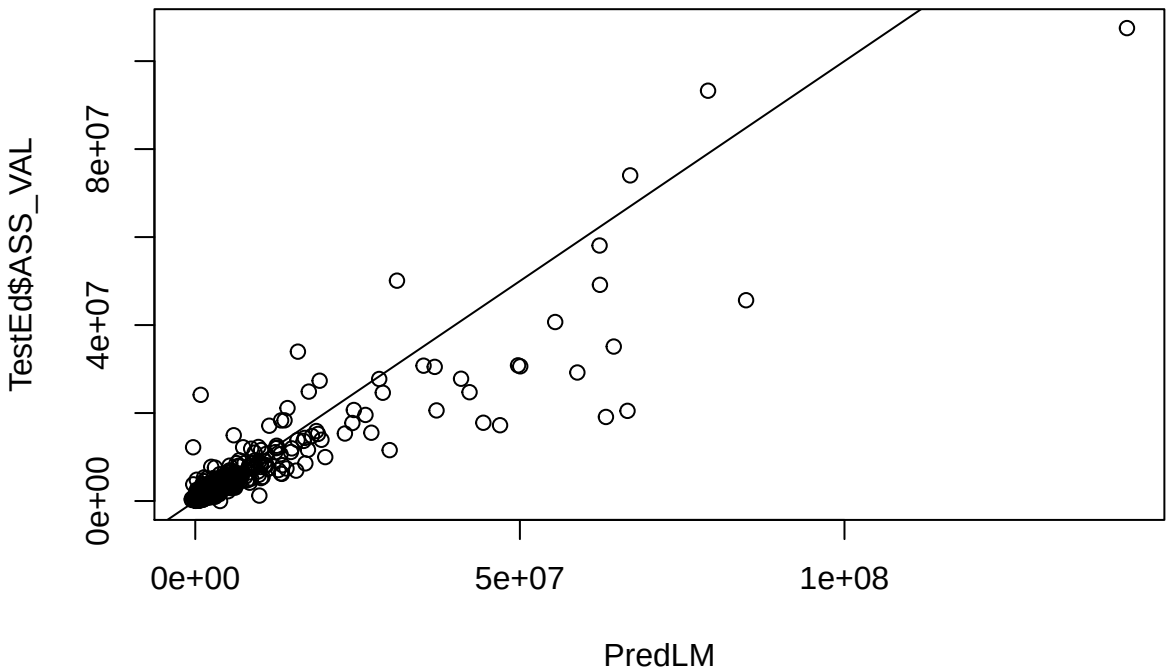
```
plot(TrainEd$LOT_SIZE,TrainEd$ASS_VAL)
# Plot fitted line using abline
abline(FitLM,lwd=3,col="red")
```

```
## Warning in abline(FitLM, lwd = 3, col = "red"): only using the first two of
## 6 regression coefficients
```



Get some predictions on training data

```
# Prediction on test data
PredLM = predict(FitLM, TestEd)
# plot
plot(PredLM,TestEd$ASS_VAL)
abline(0,1)
```



The training error is

```r
mean((PredLM - TestEd$ASS_VAL)^2)
```

```
## [1] 2.600478e+12
```

and its squared root gives us the variance

```r
sqrt(mean((PredLM - TestEd$ASS_VAL)^2))
```

```
## [1] 1612600
```

So, this model predicts the assessed values of the houses with an standard error of $ 501972.5 (probably overflow). Not good!

From the fit summary, all of the variables seem to be statistically significant for this model. The R^2 values are interpreted as good (almost 80%). We agreed that this model does not consider important features, and this error can be improved.

We now try to add other categorical variable to our model. We want to include the basement in the game. We expect that basement and assessed value be related: a building with a basement is probably more expensive than a building that has no basement. Note that Basement is set as a categorical variable, and the function

```r
# Fit linear model
FitLM = lm(ASS_VAL~GROSS_AREA+poly(LOT_SIZE,3)+BUILD_AGE+BASEMENT, data = TrainEd)
summary(FitLM)
```

```
##
## Call:
## lm(formula = ASS_VAL ~ GROSS_AREA + poly(LOT_SIZE, 3) + BUILD_AGE +
##     BASEMENT, data = TrainEd)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -89899879    -56291     42771    131515 193913944
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          8.083e+04  1.653e+04   4.889 1.01e-06 ***
## GROSS_AREA           1.847e+03  4.125e+00 447.711  < 2e-16 ***
## poly(LOT_SIZE, 3)1   4.280e+07  2.079e+06  20.583  < 2e-16 ***
## poly(LOT_SIZE, 3)2   5.151e+07  2.087e+06  24.686  < 2e-16 ***
## poly(LOT_SIZE, 3)3 -8.189e+06  2.110e+06  -3.880 0.000104 ***
## BUILD_AGE          -4.945e+03  4.838e+02 -10.222  < 2e-16 ***
## BASEMENTYes          2.365e+05  1.797e+04  13.162  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2016000 on 56671 degrees of freedom
## Multiple R-squared:  0.8118, Adjusted R-squared:  0.8117
## F-statistic: 4.073e+04 on 6 and 56671 DF,  p-value: < 2.2e-16
```

We see minor some improvements. From the summary, we see that if the building has a basement, its value is increased by more than 10000 assessed value units. Its p-value indicates statistical significance, and the variance explained in the data is of 83%. Note that the predictor AGE is now relevant to the model, as infered from its statistical significance.

Honestly, I don't think that it was significant improvement. We move by try adding a garage to our model instead

```r
# Fit linear model
FitLM = lm(ASS_VAL~GROSS_AREA+poly(LOT_SIZE,3)+BUILD_AGE+GARAGE, data = TrainEd)
summary(FitLM)
```

```
##
## Call:
## lm(formula = ASS_VAL ~ GROSS_AREA + poly(LOT_SIZE, 3) + BUILD_AGE +
##     GARAGE, data = TrainEd)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -90160601    -74645    -14352    140499 193167202
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -1.461e+05  2.453e+04  -5.955 2.62e-09 ***
## GROSS_AREA           1.852e+03  4.155e+00 445.760  < 2e-16 ***
## poly(LOT_SIZE, 3)1   4.399e+07  2.082e+06  21.129  < 2e-16 ***
## poly(LOT_SIZE, 3)2   5.048e+07  2.089e+06  24.169  < 2e-16 ***
## poly(LOT_SIZE, 3)3 -6.676e+06  2.115e+06  -3.157   0.0016 **
## BUILD_AGE           -2.148e+03  4.655e+02  -4.615 3.93e-06 ***
## GARAGEYes            3.175e+05  2.118e+04  14.985  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2015000 on 56671 degrees of freedom
## Multiple R-squared:  0.8119, Adjusted R-squared:  0.8119
## F-statistic: 4.078e+04 on 6 and 56671 DF,  p-value: < 2.2e-16
```

Well, the garage variable does not better explains the dataset than the basement. Of course, using both together yields better results

```r
# Fit linear model
FitLM = lm(ASS_VAL~GROSS_AREA+poly(LOT_SIZE,3)+BUILD_AGE+BASEMENT+GARAGE, data = TrainEd)
summary(FitLM)
```

```
##
## Call:
## lm(formula = ASS_VAL ~ GROSS_AREA + poly(LOT_SIZE, 3) + BUILD_AGE +
##     BASEMENT + GARAGE, data = TrainEd)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -90247334    -74872      8880    127495 193056214
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -1.281e+05  2.459e+04  -5.209 1.90e-07 ***
## GROSS_AREA           1.853e+03  4.153e+00 446.162  < 2e-16 ***
## poly(LOT_SIZE, 3)1   4.429e+07  2.081e+06  21.283  < 2e-16 ***
## poly(LOT_SIZE, 3)2   5.024e+07  2.087e+06  24.068  < 2e-16 ***
## poly(LOT_SIZE, 3)3 -6.417e+06  2.114e+06  -3.036   0.0024 **
## BUILD_AGE           -3.682e+03  4.957e+02  -7.428 1.12e-13 ***
## BASEMENTYes          1.692e+05  1.889e+04   8.956  < 2e-16 ***
## GARAGEYes            2.554e+05  2.228e+04  11.465  < 2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2014000 on 56670 degrees of freedom
## Multiple R-squared:  0.8122, Adjusted R-squared:  0.8122
## F-statistic: 3.501e+04 on 7 and 56670 DF,  p-value: < 2.2e-16
```

but the improvement is rather small. Also, notice how the p-value for the Garage variable increased. Although still small enough to be included in the model, we suspect that the variable garage might be of small importance for predicting the assessed value. We use the table() function to see how many buildings actually have a garage or not, and observe that only about 15% of the buildings do not have garages, which might be a reason for its lower impact in our model. In other words, it seems to be commom to have a garage in Edmonton, so adding it to the house does not make it special enough for a significant incraese in assessed value.

```
table(TrainEd$GARAGE)
```

```
##
##    No   Yes
## 12457 44221
```

```
7878/(7878+44204)
```

```
## [1] 0.1512615
```

```
table(TrainEd$BASEMENT)
```

```
##
##    No   Yes
## 31586 25092
```

On the other hand, when we do the same analysis for basement, the picture changes. In fact, more buildings do not have basements (in our samples, of course).

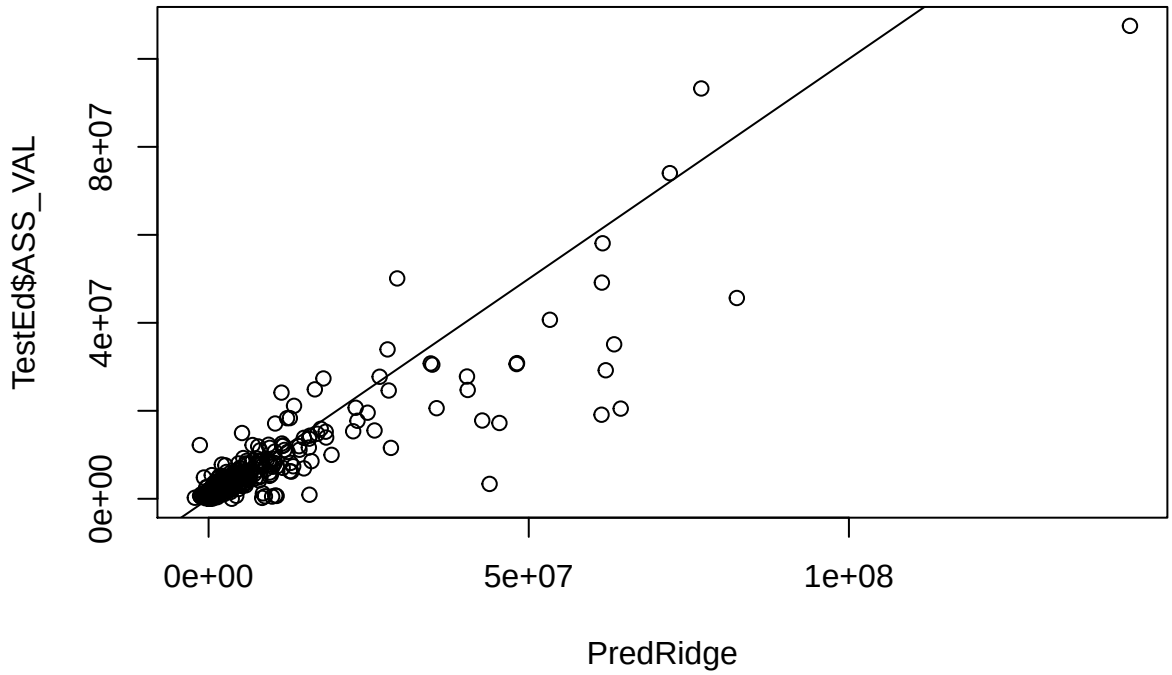
Note on model selection and the regsubset() function: We were tempted to try to train our model using regsubsets and scape the burden of training several models with different combinations of predictors and possible interactions between them. However, as this is a fairly big dataset for simple machines as ours, we decided to skip the model selection via regsubsets.

We want to follow with regularization. We wish to apply the Lasso regression to the data to obtain robust models, therefore minimizing the effect of outliers in our data. We create the model matrix by adding more features to the model above. We added net area and fireplace to our models.

```
# Create model matrix
TrainMM = model.matrix(ASS_VAL~GROSS_AREA+poly(LOT_SIZE,3)+BUILD_AGE+BASEMENT+GARAGE, data=TrainEd)
 TestMM = model.matrix(ASS_VAL~GROSS_AREA+poly(LOT_SIZE,3)+BUILD_AGE+BASEMENT+GARAGE, data=TestEd)

# CV to obtain optimal lambda
LassoCV = cv.glmnet(TrainMM, TrainEd$ASS_VAL, alpha = 1)

# OptLamb
OptLamb = LassoCV$lambda.min

# Coefficients for fit
PredRidge = predict(LassoCV, s=OptLamb, newx=TestMM)
sqrt(mean((PredRidge-TestEd$ASS_VAL)^2))
```

```
## [1] 1644966
```

```r
plot(PredRidge,TestEd$ASS_VAL)
abline(0,1)
```



We understand that linear regression, with or without regularization (Lasso and Ridge Regression), are good approaches. However, despite its aided interpretability, we do not think that this problem would be best solved through standard linear regression with or without regularization. This first step, indeed, guides us on which predictors we should pay more attention to. In this first analysis, what we can notice is that the features like basement, fireplace, garage and relative age might not be as great for predicting the assessed values of the houses. However, our intuition (which we also call p-value sometimes :P) says that such features should be included in our model. The reasoning is that, for instance, we judge reasonable a house that has a garage to have a different price (higher) than others that do not have one. However, we also acknowledge that this is not the only face to be considered. Our first analysis does not include spatial information which can be crucial for this analysis. At first we were concerned with predicting the assessed values based on houses features (sizes & areas, basement, garage, fireplace and age). Now we try to include spatial data into our model.

Including spatial information

```r
# Fit linear model
FitLM = lm(ASS_VAL~GROSS_AREA+log(LOT_SIZE+1)+BUILD_AGE+NS+WE, data = TrainEd)
summary(FitLM)
```

```
##
## Call:
## lm(formula = ASS_VAL ~ GROSS_AREA + log(LOT_SIZE + 1) + BUILD_AGE +
##     NS + WE, data = TrainEd)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -90816869    -81691     11913    105099 191834582
##
```

```
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         9.731e+05  6.676e+04  14.576  < 2e-16 ***
## GROSS_AREA          1.860e+03  4.159e+00 447.207  < 2e-16 ***
## log(LOT_SIZE + 1) -1.505e+05  1.015e+04 -14.831  < 2e-16 ***
## BUILD_AGE         -1.406e+03  4.877e+02  -2.884  0.00393 **
## NSS                2.670e+04  1.751e+04   1.525  0.12729
## WEW                1.023e+05  1.820e+04   5.622  1.9e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2035000 on 56672 degrees of freedom
## Multiple R-squared:  0.8083, Adjusted R-squared:  0.8083
## F-statistic: 4.78e+04 on 5 and 56672 DF,  p-value: < 2.2e-16
```

Going back to the points of the building age, we see here that its p-value is small enough to be taken in consideration. We shall therefore include age from our model as a numerical value.

```
par(mfrow = c(1,3))
plot(TrainEd$BUILD_AGE,TrainEd$ASS_VAL)
# Plot fitted line using abline
abline(FitLM,lwd=3,col="red")
```

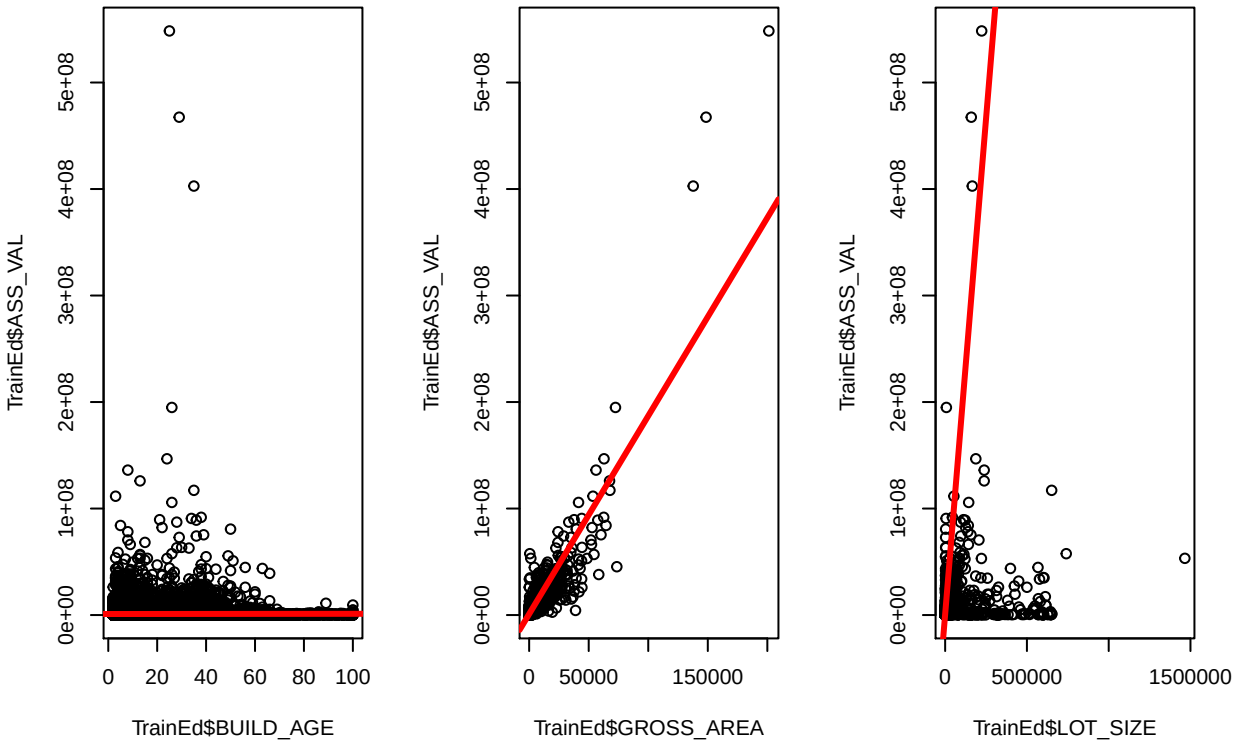
```
## Warning in abline(FitLM, lwd = 3, col = "red"): only using the first two of
## 6 regression coefficients
```

```
plot(TrainEd$GROSS_AREA,TrainEd$ASS_VAL)
# Plot fitted line using abline
abline(FitLM,lwd=3,col="red")
```

```
## Warning in abline(FitLM, lwd = 3, col = "red"): only using the first two of
## 6 regression coefficients
```

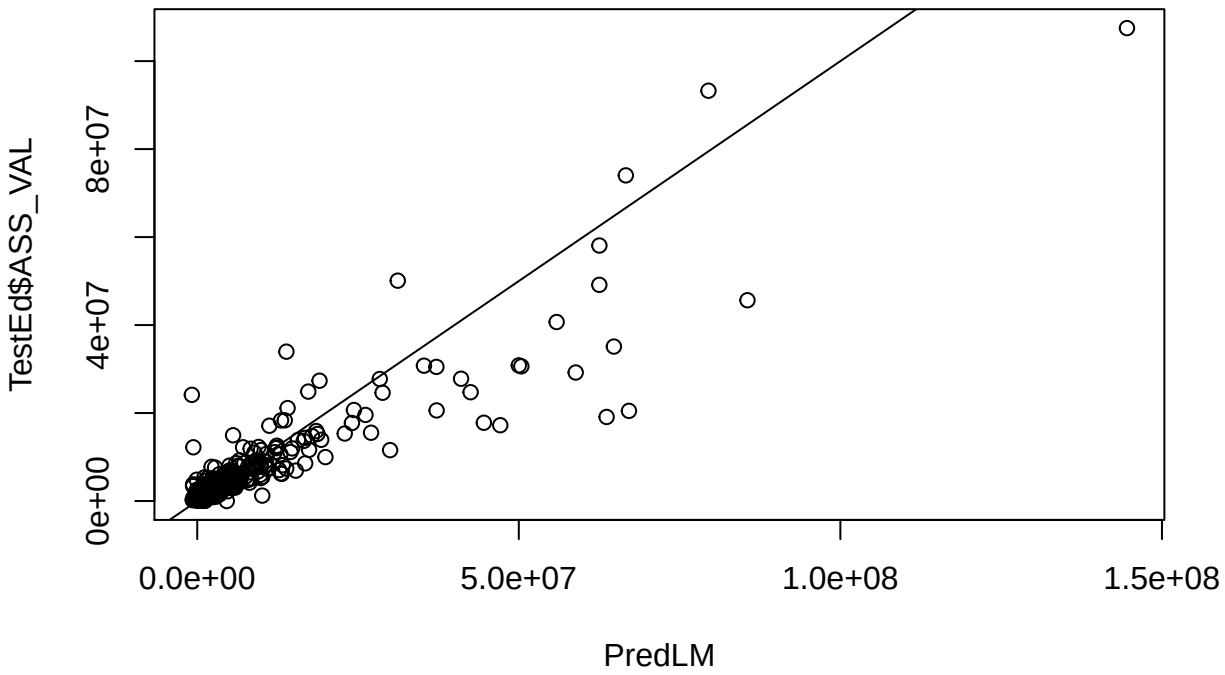
```
plot(TrainEd$LOT_SIZE,TrainEd$ASS_VAL)
# Plot fitted line using abline
abline(FitLM,lwd=3,col="red")
```

```
## Warning in abline(FitLM, lwd = 3, col = "red"): only using the first two of
## 6 regression coefficients
```



Get some predictions on training data

```
# Prediction on test data
PredLM = predict(FitLM, TestEd)
# plot
plot(PredLM,TestEd$ASS_VAL)
abline(0,1)
```



The training error is

```
mean((PredLM - TestEd$ASS_VAL)^2)
```

```
## [1] 2.688094e+12
```

and its squared root gives us the variance

```
sqrt(mean((PredLM - TestEd$ASS_VAL)^2))
```

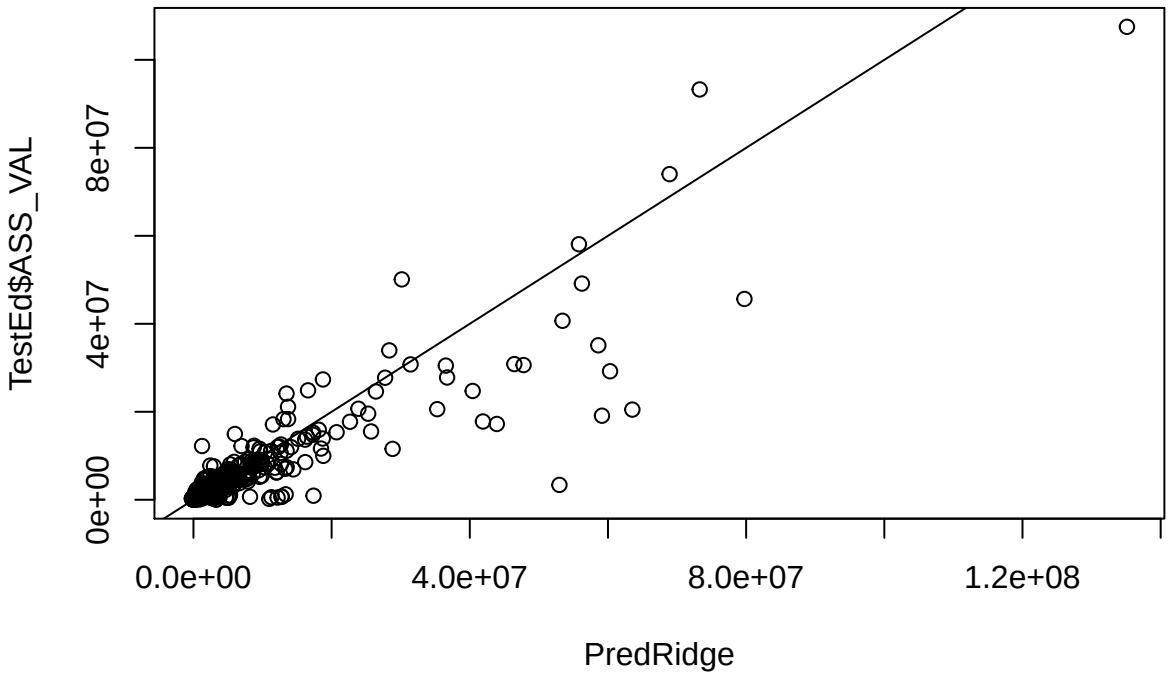
```
## [1] 1639541
```

We want to follow with regularization. We wish to apply the Lasso regression to the data to obtain robust models, therefore minimizing the effect of outliers in our data. We create the model matrix by adding more features to the model above. We added net area and fireplace to our models.

```
# Create model matrix
TrainMM = model.matrix(ASS_VAL~GROSS_AREA+poly(LOT_SIZE,3)+BUILD_AGE+BASEMENT+GARAGE+NS+WE+DISP_TYPE, da
 TestMM = model.matrix(ASS_VAL~GROSS_AREA+poly(LOT_SIZE,3)+BUILD_AGE+BASEMENT+GARAGE+NS+WE+DISP_TYPE, da

# CV to obtain optimal lambda
LassoCV = cv.glmnet(TrainMM, TrainEd$ASS_VAL, alpha = 0)

# OptLamb
OptLamb = LassoCV$lambda.min

# Coefficients for fit
PredRidge = predict(LassoCV, s=OptLamb, newx=TestMM)
sqrt(mean((PredRidge-TestEd$ASS_VAL)^2))
```

```
## [1] 1613977
```

```
plot(PredRidge,TestEd$ASS_VAL)
abline(0,1)
```



For the full dataset, WE information plays much more important role than NS. However, residential data has opposite behaviour.