Predictive model building

Bao Doquang, Dhwanit Agarwal, Akksay Singh and Shristi Singh

April 19, 2020

```
library(rsample)  # data splitting
library(glmnet)   # implementing regularized regression approaches
```

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

```
## Loaded glmnet 3.0-2
```

```
library(dplyr)    # basic data manipulation procedures
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)  # plotting
library(DAAG)
```

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

```
library(MASS)
```

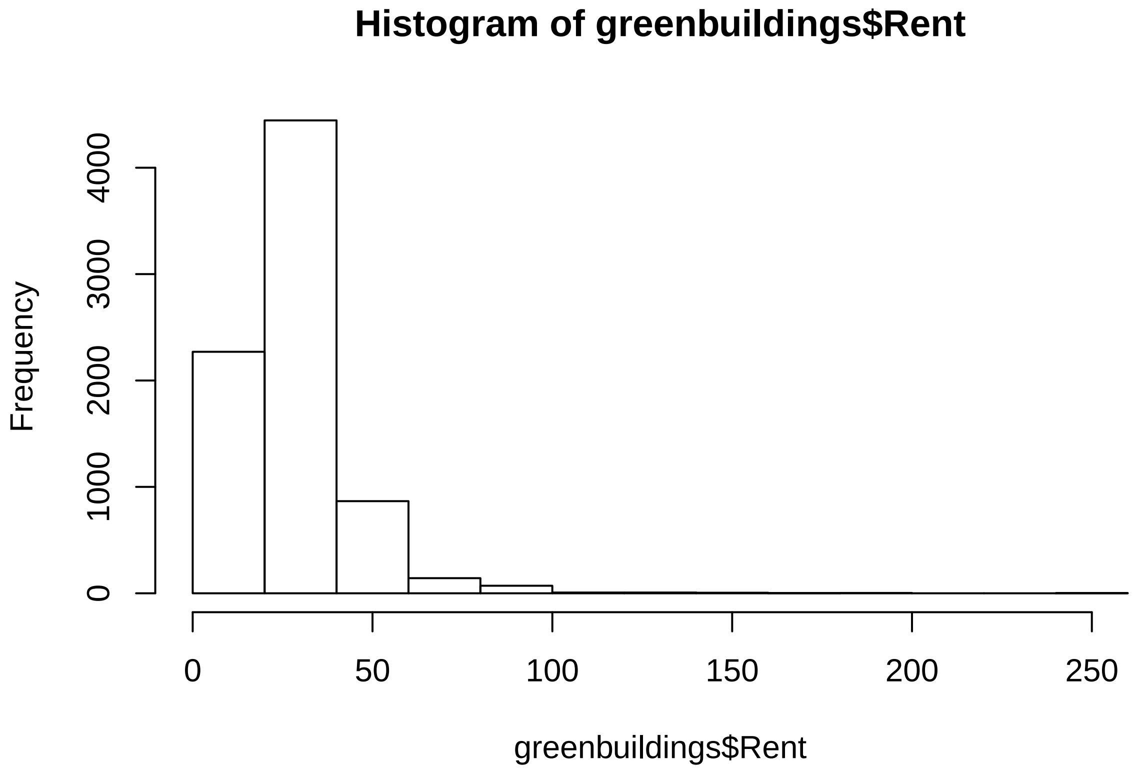
```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:DAAG':
##
##     hills
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
# import data and examine it

greenbuildings <- read.csv("greenbuildings.csv")
#View(greenbuildings)
ok <- complete.cases(greenbuildings)
greenbuildings <- greenbuildings[ok,]



# note that shares is hugely skewed
```

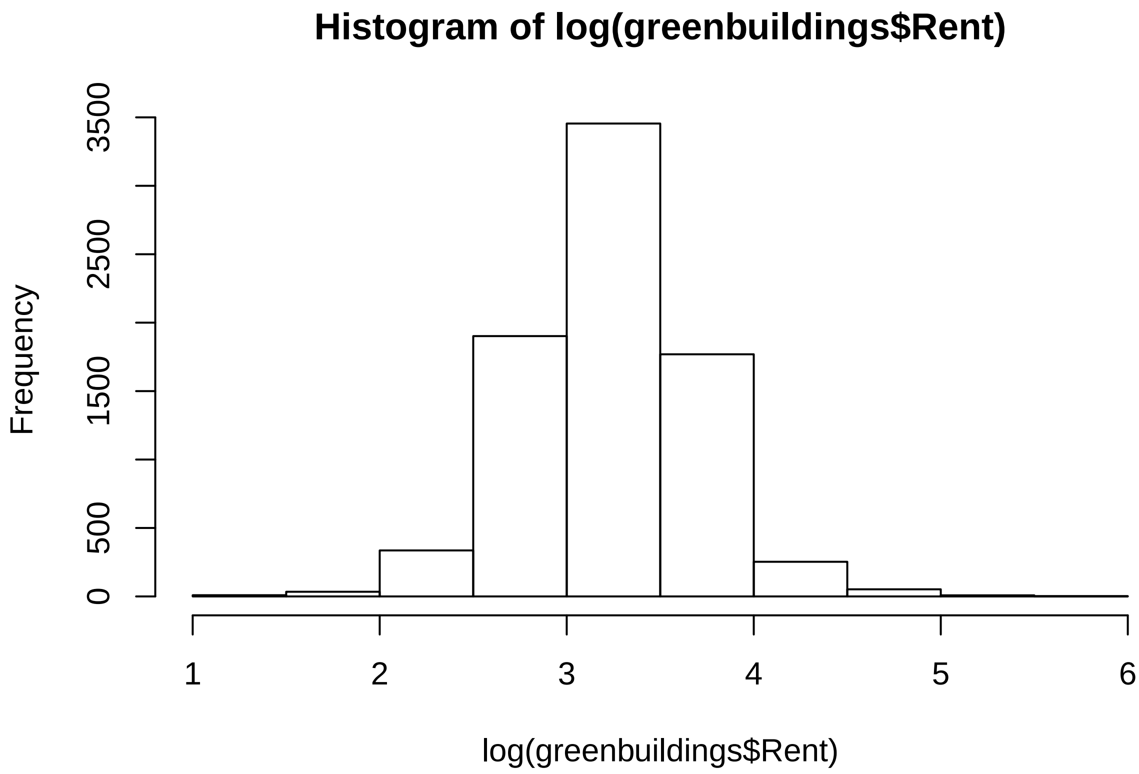
```r
# probably want a log transformation here
hist(greenbuildings$Rent)
```



```r
summary(greenbuildings$Rent)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    2.98   19.50   25.20   28.42   34.18  250.00
```

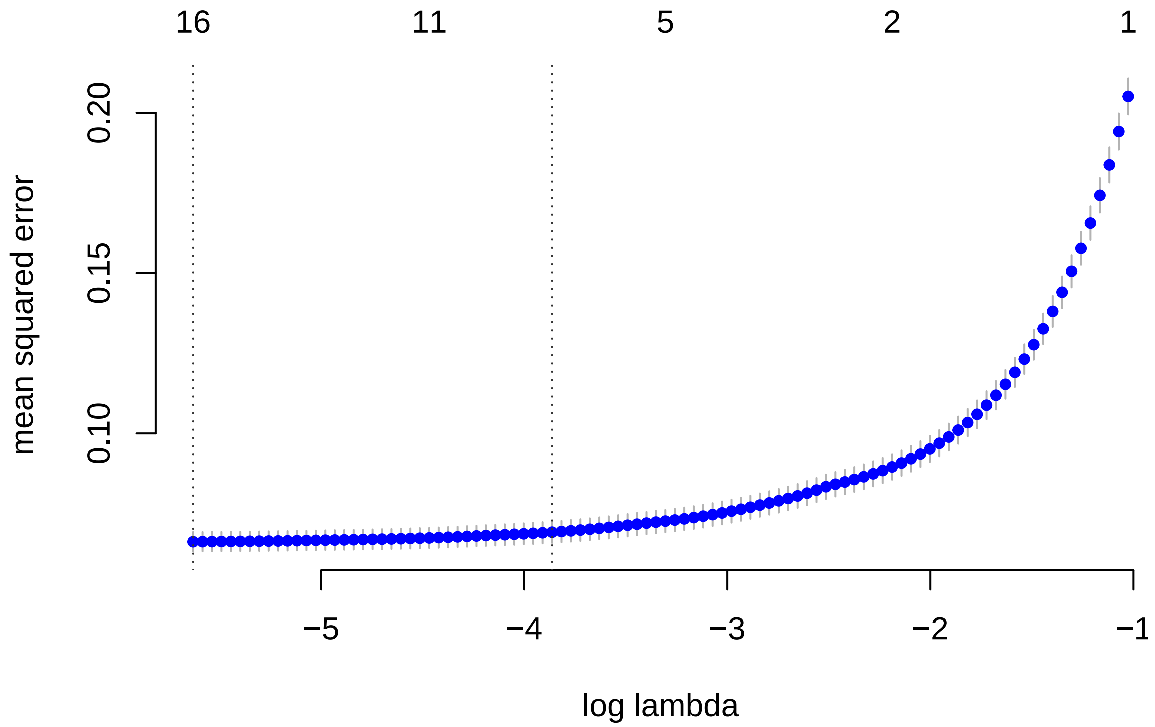
```r
# much nicer :-)
hist(log(greenbuildings$Rent))
```



```
#### lasso (glmnet does L1-L2, gamlr does L0-L1)
# I want to fit a lasso regression and do cross validation of K=10 folds
# inorder to automate finiding independent variables and training & testing my data multiple times.
# cv.gamlr command in the gamlr does it for me.
# download gamlr library
library(gamlr)

# i create a matrix of all my independent varaibles except for url from online_news data to make it eas
# the sparse.model.matrix function.
x = sparse.model.matrix( log(Rent) ~   . - CS_PropertyID - LEED -Energystar  , data=greenbuildings, stand

y = log(greenbuildings$Rent) # pull out `y' too just for convenience and do log(shares)- dependent vari



# Here I fit my lasso regression to the data and do my cross validation of k=10 n folds
# the cv.gamlr command does both things at once.
#(verb just prints progress)
cvl = cv.gamlr(x, y, nfold=10, verb=TRUE)
```

```
## fold 1,2,3,4,5,6,7,8,9,10,done.
```

```
# plot the out-of-sample deviance as a function of log lambda
plot(cvl, bty="n")
```



```
min(cvl$cvm)       # minimum MSE
```

```
## [1] 0.06619652
```

```
## [1] 0.06615445
cvl$lambda.min     # lambda for this min MSE
```

```
## [1] 0.003585894
```

```
## [1] 0.003585894


cvl$cvm[cvl$lambda == cvl$lambda.1se]  # 1 st.error of min MSE
```

```
## numeric(0)
```

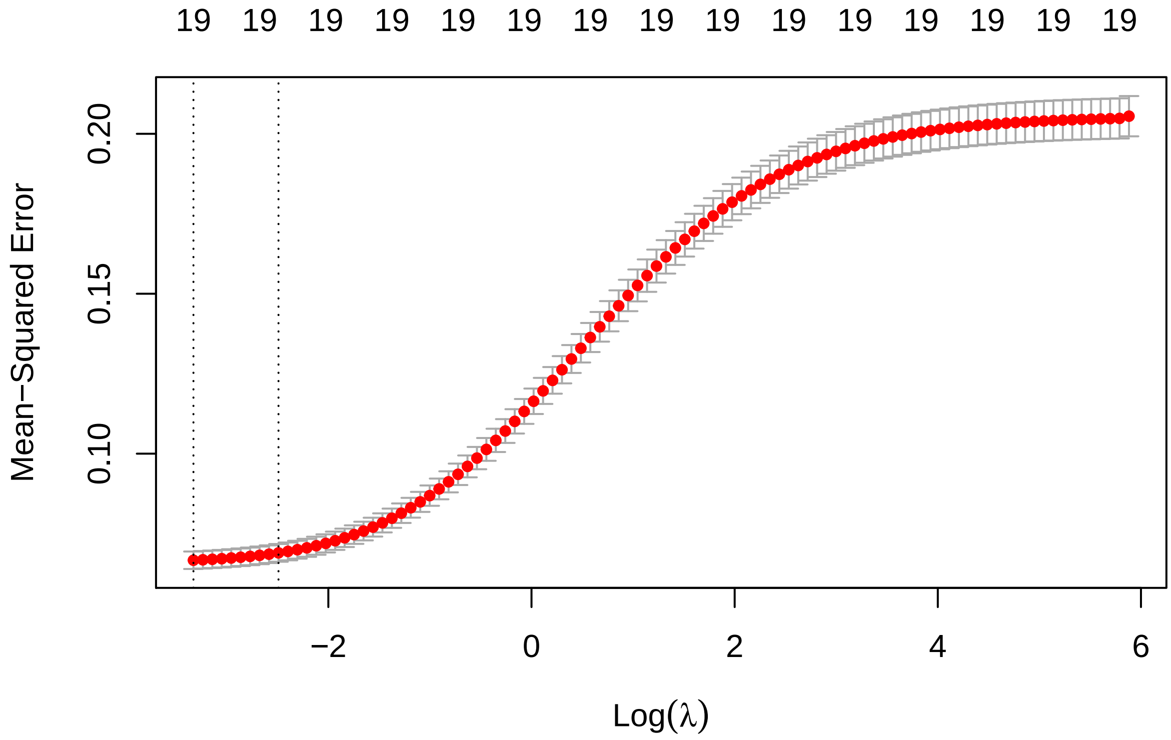
```
## [1] 0.06908108
cvl$lambda.1se  # lambda for this MSE
```

```
## [1] 0.02100266
```

```
## [1] 0.01516562

#fitted coefficients at minimum MSE
coef(cvl, select="min")
```

```
## 20 x 1 sparse Matrix of class "dgCMatrix"
##                           seg100
## intercept           2.405860e+00
## cluster             2.888478e-05
## size                7.390862e-08
## empl_gr             2.507732e-03
## leasing_rate        4.649535e-04
## stories             2.305963e-04
## age                  -9.899540e-04
## renovated             .
## class_a               9.182501e-02
## class_b               2.579294e-02
## green_rating          1.910632e-02
## net                  -3.024764e-02
## amenities             3.902586e-02
## cd_total_07          -3.056906e-05
## hd_total07            .
## total_dd_07          -1.821279e-05
## Precipitation         4.260804e-05
## Gas_Costs             .
## Electricity_Costs     .
## cluster_rent          3.087431e-02
```

```
# Apply CV Ridge regression to data
cvr <- cv.glmnet(
  x ,
  y ,
  alpha = 0
)

# plot MSE as a function of log(lambda)
plot(cvr)
```



```
min(cvr$cvm)       # minimum MSE
```

```
## [1] 0.06668643
```

```
## [1] 0.06679016  #value observed
cvr$lambda.min     # lambda for this min MSE
```

```
## [1] 0.03585894
```

```
## [1] 0.03585894

cvr$cvm[cvr$lambda == cvr$lambda.1se]  # 1 st.error of min MSE
```

```
## [1] 0.06898768
```

```
## [1] 0.06908108
cvr$lambda.1se  # lambda for this MSE
```

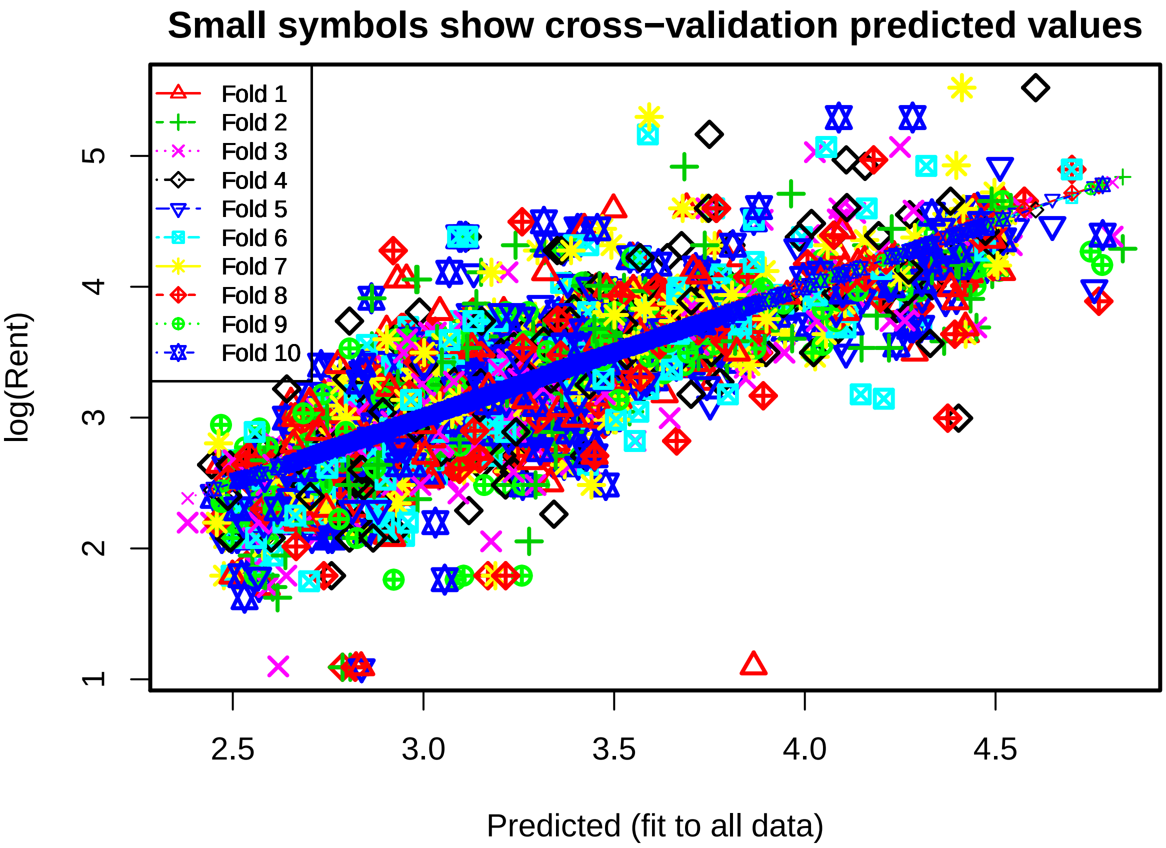
```
## [1] 0.0828388
```

```
## [1] 0.0828388

#fitted coefficients at minimum MSE
coef(cvr, select="min")
```

```
## 20 x 1 sparse Matrix of class "dgCMatrix"
##                               1
## (Intercept)        2.441321e+00
## cluster            4.932116e-05
## size               5.929577e-08
## empl_gr            3.695284e-03
## leasing_rate       9.056597e-04
## stories            7.218613e-04
## age               -9.316485e-04
## renovated         -9.610125e-03
## class_a            8.716012e-02
## class_b            2.226612e-02
## green_rating       2.177658e-02
## net               -5.986778e-02
## amenities          3.835981e-02
## cd_total_07       -4.131382e-05
## hd_total07        -4.554181e-06
## total_dd_07       -1.738643e-05
## Precipitation      2.215165e-03
## Gas_Costs         -4.588401e+00
## Electricity_Costs  3.232539e+00
## cluster_rent       2.449920e-02
```

```
#Apply OLS to data
linear_fit = lm(log(Rent) ~ . - CS_PropertyID - LEED -Energystar , data = greenbuildings) #no scaling
cvlm = cv.lm(data = greenbuildings, linear_fit, m=10, plotit = TRUE, printit = FALSE)
```

```
## Warning in predict.lm(subs.lm, newdata = data[rows.out, ]): prediction from a
## rank-deficient fit may be misleading

## Warning in predict.lm(subs.lm, newdata = data[rows.out, ]): prediction from a
## rank-deficient fit may be misleading

## Warning in predict.lm(subs.lm, newdata = data[rows.out, ]): prediction from a
## rank-deficient fit may be misleading

## Warning in predict.lm(subs.lm, newdata = data[rows.out, ]): prediction from a
## rank-deficient fit may be misleading
## Warning in predict.lm(subs.lm, newdata = data[rows.out, ]): prediction from a
## rank-deficient fit may be misleading

## Warning in predict.lm(subs.lm, newdata = data[rows.out, ]): prediction from a
## rank-deficient fit may be misleading

## Warning in predict.lm(subs.lm, newdata = data[rows.out, ]): prediction from a
## rank-deficient fit may be misleading

## Warning in predict.lm(subs.lm, newdata = data[rows.out, ]): prediction from a
## rank-deficient fit may be misleading

## Warning in predict.lm(subs.lm, newdata = data[rows.out, ]): prediction from a
## rank-deficient fit may be misleading

## Warning in predict.lm(subs.lm, newdata = data[rows.out, ]): prediction from a
## rank-deficient fit may be misleading

## Warning in cv.lm(data = greenbuildings, linear_fit, m = 10, plotit = TRUE, :
##
##  As there is >1 explanatory variable, cross-validation
##  predicted values for a fold are not a linear function
##  of corresponding overall predicted values.  Lines that
##  are shown for the different folds are approximate
```



```
print(linear_fit)
```

```
##
```

```
## Call:
## lm(formula = log(Rent) ~ . - CS_PropertyID - LEED - Energystar, 
##     data = greenbuildings)
## 
## Coefficients:
##       (Intercept)            cluster               size            empl_gr  
##         2.406e+00          3.940e-05          7.892e-08          3.504e-03  
##      leasing_rate            stories                age          renovated  
##         4.850e-04          4.224e-04         -1.049e-03          5.566e-03  
##           class_a            class_b       green_rating                net  
##         1.126e-01          5.285e-02          2.978e-02         -4.694e-02  
##         amenities        cd_total_07         hd_total07        total_dd_07  
##         3.966e-02         -6.156e-05         -2.514e-05                 NA  
##     Precipitation          Gas_Costs  Electricity_Costs       cluster_rent  
##         6.703e-04          2.219e+00         -1.569e+00          3.114e-02
```

```
#MSE for OLS = 0.0659
```