Introducing our Dataset

```r
library(MASS)
library(ISLR)
library(caret)
names(Boston)
```

```
 [1] "crim"    "zn"      "indus"   "chas"
"nox"     "rm"      "age"
 [8] "dis"     "rad"     "tax"     "ptratio"
"black"   "lstat"   "medv"
```

Background to Our Tool: Linear Regression

```
y = xb + c
```

Exploring the data

```
?Boston
```

Exploring the data

```r
str(Boston)
```

```
'data.frame':   506 obs. of  14 variables:
 $ crim   : num  0.00632 0.02731 0.02729
0.03237 0.06905 ...
 $ zn     : num  18 0 0 0 0 0 12.5 12.5 12.5
12.5 ...
 $ indus  : num  2.31 7.07 7.07 2.18 2.18
2.18 7.87 7.87 7.87 7.87 ...
 $ chas   : int  0 0 0 0 0 0 0 0 0 0 ...
 $ nox    : num  0.538 0.469 0.469 0.458
0.458 0.458 0.524 0.524 0.524 0.524 ...
 $ rm     : num  6.58 6.42 7.18 7 7.15 ...
 $ age    : num  65.2 78.9 61.1 45.8 54.2
58.7 66.6 96.1 100 85.9 ...
 $ dis    : num  4.09 4.97 4.97 6.06 6.06 ...
 $ rad    : int  1 2 2 3 3 3 5 5 5 5 ...
 $ tax    : num  296 242 242 222 222 222 311
311 311 311 ...
 $ ptratio: num  15.3 17.8 17.8 18.7 18.7
18.7 15.2 15.2 15.2 15.2 ...
 $ black  : num  397 397 393 395 397 ...
 $ lstat  : num  4.98 9.14 4.03 2.94 5.33 ...
 $ medv   : num  24 21.6 34.7 33.4 36.2 28.7
22.9 27.1 16.5 18.9 ...
```

Exploring the data

```r
head(Boston)
```

```
     crim zn indus chas   nox    rm  age
dis rad tax ptratio  black
1 0.00632 18  2.31    0 0.538 6.575 65.2
4.0900   1 296    15.3 396.90
2 0.02731  0  7.07    0 0.469 6.421 78.9
4.9671   2 242    17.8 396.90
3 0.02729  0  7.07    0 0.469 7.185 61.1
4.9671   2 242    17.8 392.83
4 0.03237  0  2.18    0 0.458 6.998 45.8
6.0622   3 222    18.7 394.63
5 0.06905  0  2.18    0 0.458 7.147 54.2
6.0622   3 222    18.7 396.90
6 0.02985  0  2.18    0 0.458 6.430 58.7
6.0622   3 222    18.7 394.12
  lstat medv
1  4.98 24.0
2  9.14 21.6
3  4.03 34.7
4  2.94 33.4
5  5.33 36.2
6  5.21 28.7
```

Perform the train/test split

- createDataPartition can preserve the relative frequencies in our dependent variable

```
set.seed(7)
train_Index <-
createDataPartition(Boston$medv, p = 0.8,
list=FALSE)
train_Boston <- Boston[train_Index,]
test_Boston <- Boston[-train_Index,]
```

Model 1 Single Variable Linear Regression

```
model1 <- train(medv ~ lstat, method = 'lm',
data = train_Boston)
# view coefficients
coef(model1$finalModel)
```

```
(Intercept)       lstat
 34.8025753  -0.9506744
```

```
model1
```

```
Linear Regression

407 samples
  1 predictor

No pre-processing
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 407, 407, 407, 407,
407, 407, ...
Resampling results:

  RMSE      Rsquared
  6.491572  0.5451794

Tuning parameter 'intercept' was held
constant at a value of TRUE
```

Model 2 Two Variable Linear Regression

```
model2 <- train(medv ~ lstat + age, method =
'lm', data = train_Boston)
coef(model2$finalModel)
```

```
(Intercept)       lstat         age
 33.1842068  -1.0460706   0.0412899
```

```
model2
```

```
Linear Regression

407 samples
  2 predictor

No pre-processing
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 407, 407, 407, 407,
407, 407, ...
Resampling results:

  RMSE      Rsquared
  6.266065  0.5511262

Tuning parameter 'intercept' was held
constant at a value of TRUE
```

```
varImp(model2)
```

```
lm variable importance

      Overall
lstat     100
age         0
```

Model 3 Multiple Variable Linear Regression

```
model3 <- train(medv ~ ., method = "lm", data
= train_Boston)
coef(model3$finalModel)
```

```
  (Intercept)          crim             zn
indus          chas
 37.443258137  -0.114804777   0.041559970
-0.026526572   2.644829514
          nox            rm            age
dis           rad
-19.104962411   3.919062842   0.012302002
-1.468870235   0.278393830
          tax       ptratio          black
lstat
 -0.010477531  -0.978105565   0.006832878
-0.532421893
```

```
model3
```

```
Linear Regression

407 samples
 13 predictor

No pre-processing
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 407, 407, 407, 407,
407, 407, ...
Resampling results:

  RMSE      Rsquared
  5.267044  0.6945903

Tuning parameter 'intercept' was held
constant at a value of TRUE
```

```r
varImp(model3)
```

```
lm variable importance

        Overall
lstat   100.000
rm       90.086
ptratio  69.081
dis      66.392
nox      46.709
rad      34.644
crim     28.419
chas     27.366
zn       25.067
tax      21.715
black    20.238
age       4.898
indus     0.000
```

Exercise: Try a few variable combinations yourself

```r
## your code goes here ##
```

What might be affecting our model?

-
-

Correlation

- assess correlation by creating a correlation matrix

```
cor(Boston)
```

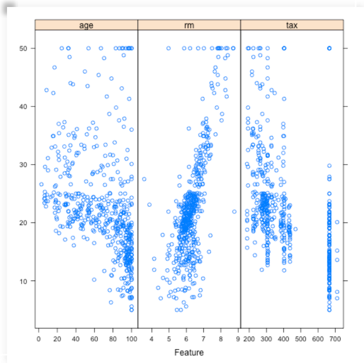
```
                crim          zn       indus
chas         nox
crim     1.00000000 -0.20046922  0.40658341
-0.055891582  0.42097171
zn      -0.20046922  1.00000000 -0.53382819
-0.042696719 -0.51660371
indus    0.40658341 -0.53382819  1.00000000
0.062938027  0.76365145
chas    -0.05589158 -0.04269672  0.06293803
1.000000000  0.09120281
nox      0.42097171 -0.51660371  0.76365145
0.091202807  1.00000000
rm      -0.21924670  0.31199059 -0.39167585
0.091251225 -0.30218819
age      0.35273425 -0.56953734  0.64477851
0.086517774  0.73147010
dis     -0.37967009  0.66440822 -0.70802699
-0.099175780 -0.76923011
rad      0.62550515 -0.31194783  0.59512927
-0.007368241  0.61144056
tax      0.58276431 -0.31456332  0.72076018
-0.035586518  0.66802320
ptratio  0.28994558 -0.39167855  0.38324756
-0.121515174  0.18893268
black   -0.38506394  0.17552032 -0.35697654
0.048788485 -0.38005064
lstat    0.45562148 -0.41299457  0.60379972
-0.053929298  0.59087892
medv    -0.38830461  0.36044534 -0.48372516
```

```
0.175260177 -0.42732077
                rm          age          dis
rad         tax
crim    -0.21924670  0.35273425 -0.37967009
0.625505145  0.58276431
zn       0.31199059 -0.56953734  0.66440822
-0.311947826 -0.31456332
indus   -0.39167585  0.64477851 -0.70802699
0.595129275  0.72076018
chas     0.09125123  0.08651777 -0.09917578
-0.007368241 -0.03558652
nox     -0.30218819  0.73147010 -0.76923011
0.611440563  0.66802320
rm       1.00000000 -0.24026493  0.20524621
-0.209846668 -0.29204783
age     -0.24026493  1.00000000 -0.74788054
0.456022452  0.50645559
dis      0.20524621 -0.74788054  1.00000000
-0.494587930 -0.53443158
rad     -0.20984667  0.45602245 -0.49458793
1.000000000  0.91022819
tax     -0.29204783  0.50645559 -0.53443158
0.910228189  1.00000000
ptratio -0.35550149  0.26151501 -0.23247054
0.464741179  0.46085304
black    0.12806864 -0.27353398  0.29151167
-0.444412816 -0.44180801
lstat   -0.61380827  0.60233853 -0.49699583
0.488676335  0.54399341
medv     0.69535995 -0.37695457  0.24992873
-0.381626231 -0.46853593
           ptratio       black      lstat
medv
crim     0.2899456 -0.38506394  0.4556215
-0.3883046
zn      -0.3916785  0.17552032 -0.4129946
0.3604453
indus    0.3832476 -0.35697654  0.6037997
-0.4837252
chas    -0.1215152  0.04878848 -0.0539293
0.1752602
nox      0.1889327 -0.38005064  0.5908789
-0.4273208
rm      -0.3555015  0.12806864 -0.6138083
0.6953599
age      0.2615150 -0.27353398  0.6023385
-0.3769546
dis     -0.2324705  0.29151167 -0.4969958
0.2499287
rad      0.4647412 -0.44441282  0.4886763
-0.3816262
tax      0.4608530 -0.44180801  0.5439934
-0.4685359
ptratio  1.0000000 -0.17738330  0.3740443
-0.5077867
black   -0.1773833  1.00000000 -0.3660869
0.3334608
lstat    0.3740443 -0.36608690  1.0000000
-0.7376627
medv    -0.5077867  0.33346082 -0.7376627
1.0000000
```

Looking for outliers

- Scatterplots are useful

```
# small example
var <- c("rm", "age", "tax")
featurePlot(x = Boston[, var],
            y = Boston$medv,
            plot = "scatter",
            layout = c(3,1))
```



Improving our model