Kaggle Project

Bridget, Eva, and Annie

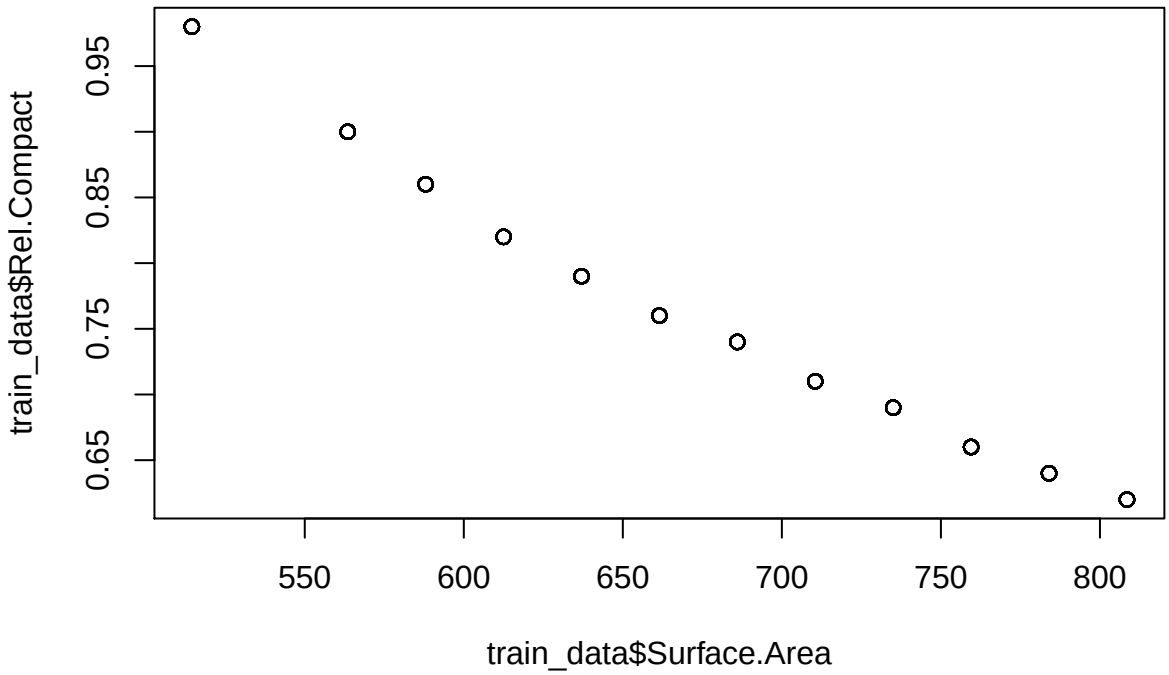
November 21, 2017

```
# read in the data
train_data=read.csv(file="/home/bridget/Dropbox/STATS202/kaggle/train.csv",header=T)
test_data=read.csv(file="/home/bridget/Dropbox/STATS202/kaggle/test.csv",header=T)

# check correlations
cor(train_data)
```

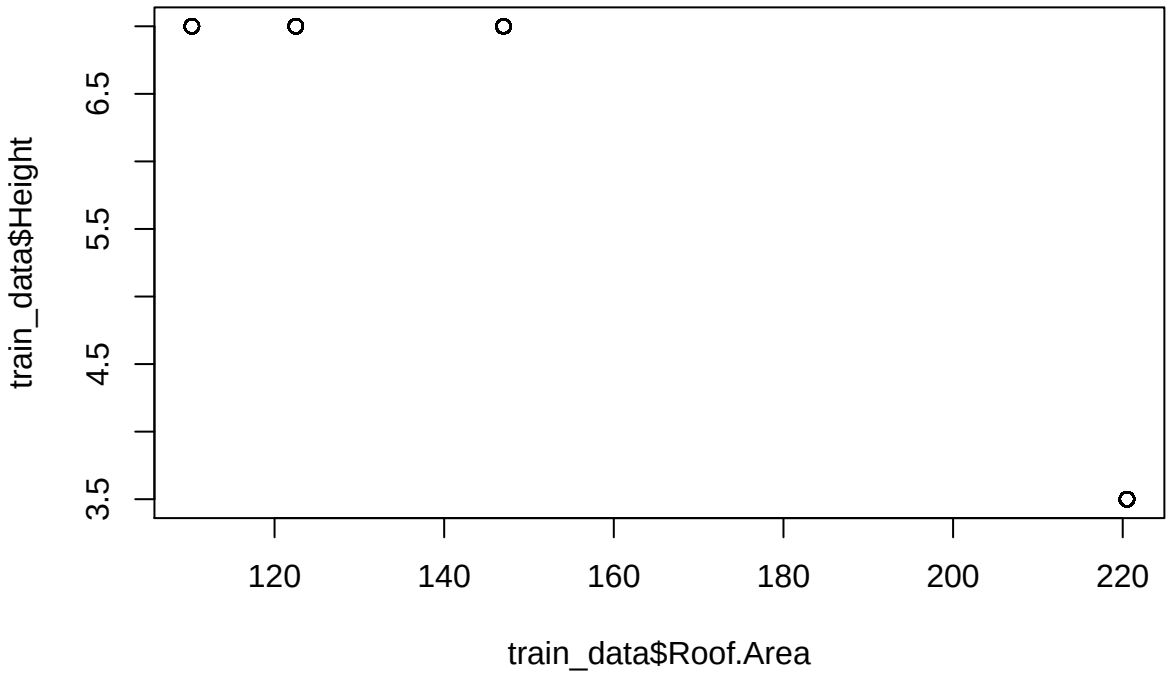
```
##                          ID  Rel.Compact Surface.Area    Wall.Area
## ID             1.000000000 -0.079452294  0.079556731  0.050111914
## Rel.Compact   -0.079452294  1.000000000 -0.991920018 -0.188572478
## Surface.Area   0.079556731 -0.991920018  1.000000000  0.177694681
## Wall.Area      0.050111914 -0.188572478  0.177694681  1.000000000
## Roof.Area      0.052340236 -0.864831757  0.877916558 -0.315192553
## Height        -0.050350469  0.824310797 -0.855434592  0.304627802
## Orientation   -0.063890529  0.024387804 -0.028384160 -0.018812858
## Glazing.Area   0.035542350 -0.005116515  0.002566042 -0.002268867
## Glazing.Distr -0.004501353 -0.006858788  0.007691487 -0.017346230
## Outcome       -0.037014518  0.613645457 -0.651401903  0.477015505
##                  Roof.Area       Height  Orientation Glazing.Area
## ID              0.05234024 -0.050350469 -0.063890529  0.035542350
## Rel.Compact    -0.86483176  0.824310797  0.024387804 -0.005116515
## Surface.Area    0.87791656 -0.855434592 -0.028384160  0.002566042
## Wall.Area      -0.31519255  0.304627802 -0.018812858 -0.002268867
## Roof.Area       1.00000000 -0.973178598 -0.018219452  0.003578560
## Height         -0.97317860  1.000000000  0.015269106 -0.001719509
## Orientation    -0.01821945  0.015269106  1.000000000  0.005121389
## Glazing.Area    0.00357856 -0.001719509  0.005121389  1.000000000
## Glazing.Distr   0.01585741 -0.017802456  0.009653998  0.218478751
## Outcome        -0.86029097  0.888969806  0.008817138  0.269249436
##                Glazing.Distr      Outcome
## ID              -0.004501353 -0.037014518
## Rel.Compact     -0.006858788  0.613645457
## Surface.Area     0.007691487 -0.651401903
## Wall.Area       -0.017346230  0.477015505
## Roof.Area        0.015857406 -0.860290971
## Height          -0.017802456  0.888969806
## Orientation      0.009653998  0.008817138
## Glazing.Area     0.218478751  0.269249436
## Glazing.Distr    1.000000000  0.071155433
## Outcome          0.071155433  1.000000000
```

```
# plot relationship between Surface.Area and Rel.Compact
plot(train_data$Surface.Area,train_data$Rel.Compact)
```




```
# should drop one of these

# plot the relationship between Roof.Area and Height
plot(train_data$Roof.Area,train_data$Height)
```




```
# change height and orientation variables to categorical variables
#train_data$Height=as.factor(train_data$Height)
train_data$Orientation=as.factor(train_data$Orientation)

# remove ID variable since it just labels the rows
# remove relative compactness because it is linearly correlated with surface area
```

```r
# remove surface area because it is equal to wall area + 2 *(roof area)
summary(lm(Surface.Area~Wall.Area+Roof.Area,data=train_data))
```

```
##
## Call:
## lm(formula = Surface.Area ~ Wall.Area + Roof.Area, data = train_data)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -1.348e-11 -3.090e-13 -9.900e-14 -1.800e-14  1.003e-10
##
## Coefficients:
##              Estimate Std. Error   t value Pr(>|t|)
## (Intercept) 6.671e-12  1.528e-12 4.366e+00 1.47e-05 ***
## Wall.Area   1.000e+00  3.736e-15 2.677e+14  < 2e-16 ***
## Roof.Area   2.000e+00  3.635e-15 5.502e+14  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.993e-12 on 647 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 1.563e+29 on 2 and 647 DF,  p-value: < 2.2e-16
```

```r
names(train_data)
```

```
##  [1] "ID"            "Rel.Compact"   "Surface.Area"  "Wall.Area"
##  [5] "Roof.Area"     "Height"        "Orientation"   "Glazing.Area"
##  [9] "Glazing.Distr" "Outcome"
```

```r
train_data=train_data[,-c(1,2,3)]

# linear regression on remaining data
lm.fit=lm(Outcome~.,data=train_data)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = Outcome ~ ., data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.7083 -1.5954  0.2048  1.5287  7.6019
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -36.286557   3.698369  -9.812  < 2e-16 ***
## Wall.Area       0.053243   0.002819  18.889  < 2e-16 ***
## Roof.Area       0.036890   0.011353   3.249  0.00122 **
## Height          5.684714   0.293408  19.375  < 2e-16 ***
## Orientation3    0.230632   0.333258   0.692  0.48916
## Orientation4   -0.123776   0.333211  -0.371  0.71041
## Orientation5    0.109721   0.337915   0.325  0.74552
## Glazing.Area   19.932986   0.902111  22.096  < 2e-16 ***
## Glazing.Distr   0.211373   0.077826   2.716  0.00679 **
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.011 on 641 degrees of freedom
## Multiple R-squared:  0.9132, Adjusted R-squared:  0.9122
## F-statistic: 843.5 on 8 and 641 DF,  p-value: < 2.2e-16
```

```
# test whether we need all orientation variables
set.seed(1)
train_sample=sample(1:nrow(train_data),0.7*nrow(train_data))
lm.fit=lm(Outcome~.,data=train_data,subset=train_sample)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = Outcome ~ ., data = train_data, subset = train_sample)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.7289 -1.4425  0.1676  1.4201  7.9489
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -31.947129   4.324375  -7.388 7.44e-13 ***
## Wall.Area       0.050759   0.003208  15.822  < 2e-16 ***
## Roof.Area       0.025642   0.013219   1.940    0.053 .
## Height          5.411518   0.345397  15.668  < 2e-16 ***
## Orientation3    0.383869   0.394551   0.973    0.331
## Orientation4   -0.374340   0.380786  -0.983    0.326
## Orientation5    0.106667   0.389601   0.274    0.784
## Glazing.Area   20.069577   1.075028  18.669  < 2e-16 ***
## Glazing.Distr   0.159292   0.090172   1.767    0.078 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.928 on 445 degrees of freedom
## Multiple R-squared:  0.9181, Adjusted R-squared:  0.9166
## F-statistic: 623.1 on 8 and 445 DF,  p-value: < 2.2e-16
```

```
mean((predict(lm.fit,newdata=train_data[-train_sample,])-train_data[-train_sample,]$Outcome)^2)
```

```
## [1] 10.34505
```

```
train_sample=sample(1:nrow(train_data),0.7*nrow(train_data))
lm.fit=lm(Outcome~.-Orientation,data=train_data,subset=train_sample)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = Outcome ~ . - Orientation, data = train_data, subset = train_sample)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.6488 -1.5130  0.1017  1.4363  7.9290
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)   -35.693657   4.271643  -8.356 8.22e-16 ***
## Wall.Area       0.054130   0.003404  15.902  < 2e-16 ***
## Roof.Area       0.034555   0.013162   2.625  0.00895 **
## Height          5.582507   0.339838  16.427  < 2e-16 ***
## Glazing.Area   19.615162   1.060424  18.497  < 2e-16 ***
## Glazing.Distr   0.249617   0.091664   2.723  0.00672 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.956 on 448 degrees of freedom
## Multiple R-squared:  0.9124, Adjusted R-squared:  0.9115
## F-statistic: 933.7 on 5 and 448 DF,  p-value: < 2.2e-16
```

```
mean((predict(lm.fit,newdata=train_data[-train_sample,])-train_data[-train_sample,]$Outcome)^2)
```

```
## [1] 9.800774
```

```
# given that the validation set error is smaller for the model that does not include Orientation,
# we drop orientation as well.

# test whether we need height/roof.area variable
lm.fit=lm(Outcome~.-Roof.Area-Orientation,data=train_data,subset=train_sample)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = Outcome ~ . - Roof.Area - Orientation, data = train_data,
##     subset = train_sample)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.3865 -1.5587  0.0759  1.3548  8.4417
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -24.860889   1.111852 -22.360  < 2e-16 ***
## Wall.Area       0.053512   0.003418  15.656  < 2e-16 ***
## Height          4.717316   0.083475  56.511  < 2e-16 ***
## Glazing.Area   19.695455   1.066916  18.460  < 2e-16 ***
## Glazing.Distr   0.244269   0.092241   2.648  0.00838 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.975 on 449 degrees of freedom
## Multiple R-squared:  0.9111, Adjusted R-squared:  0.9103
## F-statistic:  1150 on 4 and 449 DF,  p-value: < 2.2e-16
```

```
mean((predict(lm.fit,newdata=train_data[-train_sample,])-train_data[-train_sample,]$Outcome)^2)
```

```
## [1] 9.973021
```

```
lm.fit=lm(Outcome~.-Height-Orientation,data=train_data,subset=train_sample)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = Outcome ~ . - Height - Orientation, data = train_data,
```

```
##     subset = train_sample)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.7490  -2.0346  -0.4513   1.3454  11.4879
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   30.605894   1.769073   17.30   <2e-16 ***
## Wall.Area      0.054275   0.004304   12.61   <2e-16 ***
## Roof.Area     -0.175126   0.004061  -43.12   <2e-16 ***
## Glazing.Area  20.103206   1.340299   15.00   <2e-16 ***
## Glazing.Distr  0.193367   0.115822    1.67   0.0957 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.737 on 449 degrees of freedom
## Multiple R-squared:  0.8597, Adjusted R-squared:  0.8585
## F-statistic: 687.8 on 4 and 449 DF,  p-value: < 2.2e-16
```

```
mean((predict(lm.fit,newdata=train_data[-train_sample,])-train_data[-train_sample,]$Outcome)^2)
```

```
## [1] 15.10154
```

```
# looks like we can drop the roof.area variable
# try adding an interaction term
lm.fit=lm(Outcome~.-Orientation+Wall.Area:Roof.Area,data=train_data,subset=train_sample)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = Outcome ~ . - Orientation + Wall.Area:Roof.Area,
##     data = train_data, subset = train_sample)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.586 -1.318 -0.052  1.333  7.851
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -5.474e+01  5.736e+00  -9.542  < 2e-16 ***
## Wall.Area            1.108e-01  1.218e-02   9.093  < 2e-16 ***
## Roof.Area            1.424e-01  2.575e-02   5.531 5.44e-08 ***
## Height               5.662e+00  3.321e-01  17.052  < 2e-16 ***
## Glazing.Area         1.934e+01  1.036e+00  18.664  < 2e-16 ***
## Glazing.Distr        2.498e-01  8.946e-02   2.793  0.00545 **
## Wall.Area:Roof.Area -3.305e-04  6.839e-05  -4.833 1.85e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.885 on 447 degrees of freedom
## Multiple R-squared:  0.9168, Adjusted R-squared:  0.9157
## F-statistic: 820.8 on 6 and 447 DF,  p-value: < 2.2e-16
```

```
mean((predict(lm.fit,newdata=train_data[-train_sample,])-train_data[-train_sample,]$Outcome)^2)
```

```
## [1] 9.066568
```

```
lm.fit=lm(Outcome~.-Roof.Area-Orientation+Wall.Area:Roof.Area,data=train_data,subset=train_sample)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = Outcome ~ . - Roof.Area - Orientation + Wall.Area:Roof.Area,
##     data = train_data, subset = train_sample)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.4014 -1.5576  0.0842  1.3460  8.4541
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -2.474e+01  1.927e+00 -12.835  < 2e-16 ***
## Wall.Area            5.396e-02  6.763e-03   7.980 1.24e-14 ***
## Height               4.696e+00  2.915e-01  16.108  < 2e-16 ***
## Glazing.Area         1.970e+01  1.068e+00  18.440  < 2e-16 ***
## Glazing.Distr        2.441e-01  9.236e-02   2.643   0.0085 **
## Wall.Area:Roof.Area -2.738e-06  3.523e-05  -0.078   0.9381
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.978 on 448 degrees of freedom
## Multiple R-squared:  0.9111, Adjusted R-squared:  0.9101
## F-statistic: 918.2 on 5 and 448 DF,  p-value: < 2.2e-16
```

```
mean((predict(lm.fit,newdata=train_data[-train_sample,])-train_data[-train_sample,]$Outcome)^2)
```

```
## [1] 9.970661
```

```
lm.fit=lm(Outcome~.-Roof.Area-Orientation+Wall.Area:Roof.Area+Roof.Area:Height,
          data=train_data,subset=train_sample)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = Outcome ~ . - Roof.Area - Orientation + Wall.Area:Roof.Area +
##     Roof.Area:Height, data = train_data, subset = train_sample)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.586 -1.318 -0.052  1.333  7.851
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -2.334e+01  1.884e+00 -12.390  < 2e-16 ***
## Wall.Area            1.108e-01  1.218e-02   9.093  < 2e-16 ***
## Height               1.177e+00  6.960e-01   1.691  0.09147 .
## Glazing.Area         1.934e+01  1.036e+00  18.664  < 2e-16 ***
## Glazing.Distr        2.498e-01  8.946e-02   2.793  0.00545 **
## Wall.Area:Roof.Area -3.305e-04  6.839e-05  -4.833 1.85e-06 ***
## Roof.Area:Height     2.034e-02  3.678e-03   5.531 5.44e-08 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.885 on 447 degrees of freedom
## Multiple R-squared:  0.9168, Adjusted R-squared:  0.9157
## F-statistic: 820.8 on 6 and 447 DF,  p-value: < 2.2e-16
```

```
mean((predict(lm.fit,newdata=train_data[-train_sample,])-train_data[-train_sample,]$Outcome)^2)
```

```
## [1] 9.066568
```

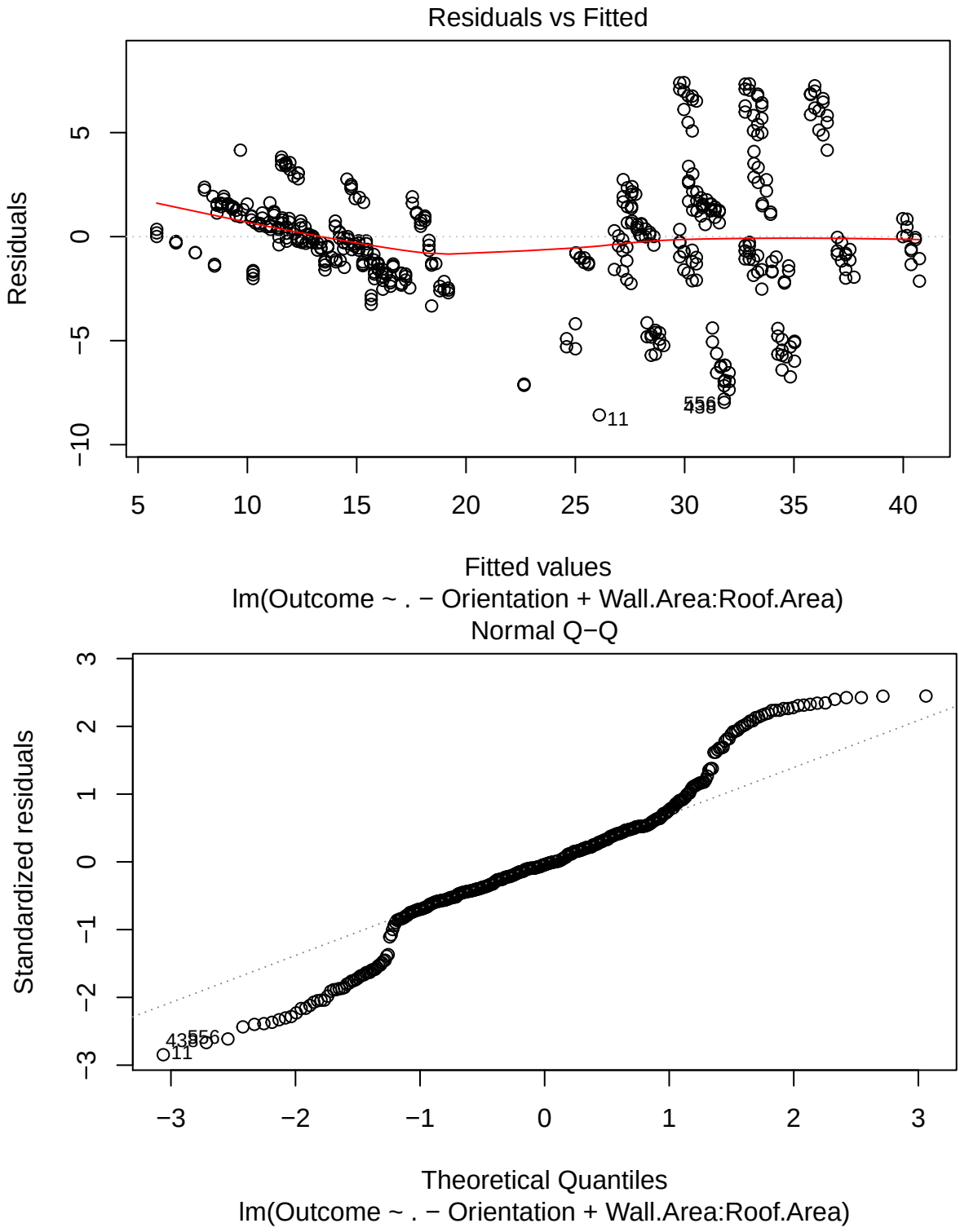
```
train_sample=sample(1:nrow(train_data),0.7*nrow(train_data))
lm.fit=lm(Outcome~.-Orientation+Wall.Area:Roof.Area,data=train_data,subset=train_sample)
summary(lm.fit)
```

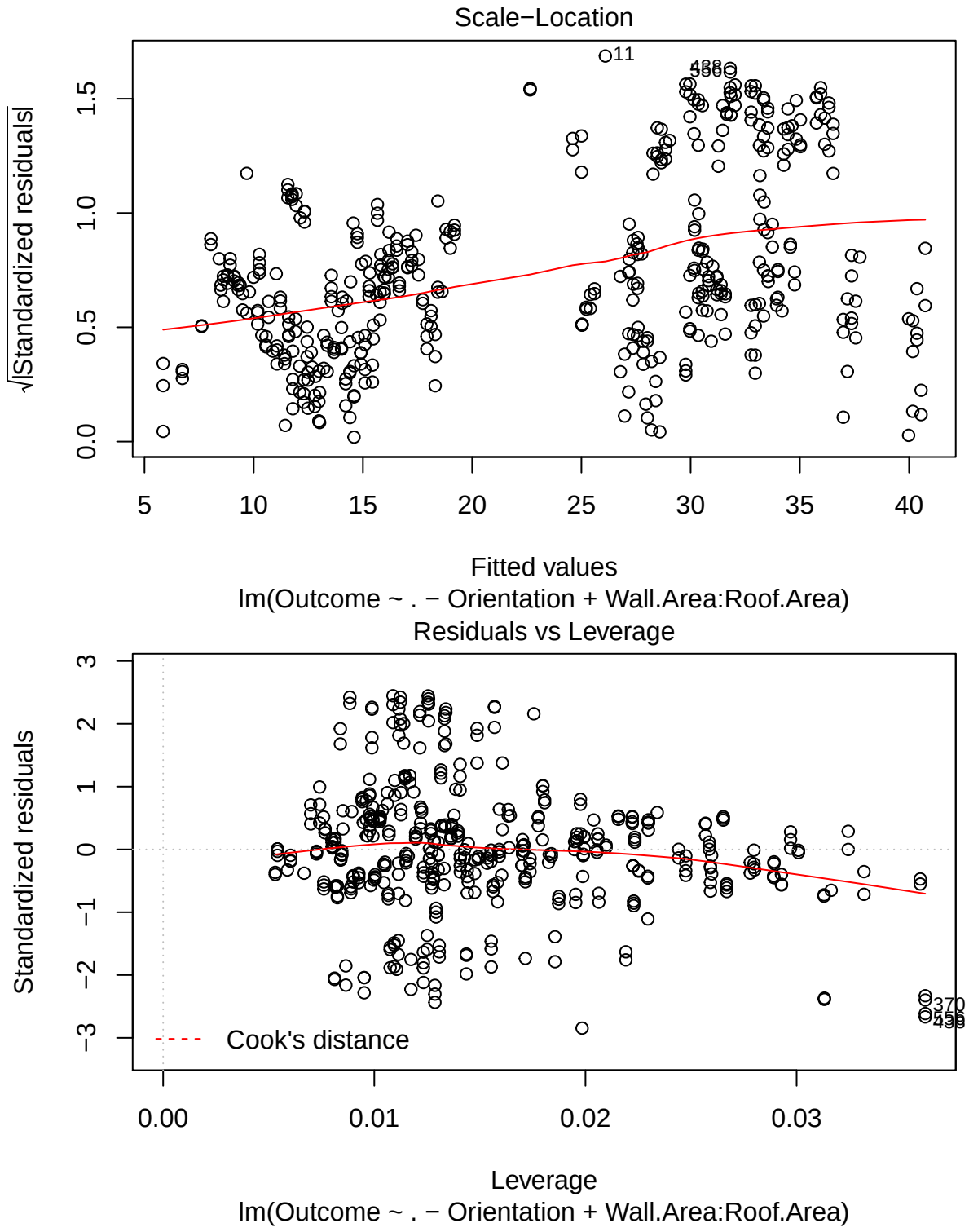
```
##
## Call:
## lm(formula = Outcome ~ . - Orientation + Wall.Area:Roof.Area,
##     data = train_data, subset = train_sample)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.5725 -1.3977 -0.1275  1.4312  7.4043
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -5.766e+01  6.029e+00  -9.564  < 2e-16 ***
## Wall.Area            1.113e-01  1.280e-02   8.699  < 2e-16 ***
## Roof.Area            1.534e-01  2.693e-02   5.697 2.22e-08 ***
## Height               5.965e+00  3.427e-01  17.404  < 2e-16 ***
## Glazing.Area         1.994e+01  1.090e+00  18.301  < 2e-16 ***
## Glazing.Distr        1.904e-01  9.486e-02   2.007   0.0454 *
## Wall.Area:Roof.Area -3.417e-04  7.134e-05  -4.789 2.28e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.042 on 447 degrees of freedom
## Multiple R-squared:  0.9131, Adjusted R-squared:  0.912
## F-statistic:   783 on 6 and 447 DF,  p-value: < 2.2e-16
```

```
mean((predict(lm.fit,newdata=train_data[-train_sample,])-train_data[-train_sample,]$Outcome)^2)
```

```
## [1] 6.952516
```

```
# this looks like a good model!
# check residuals/outliers/high leverage points
plot(lm.fit)
```


Residuals vs Fitted
Residuals
Fitted values
lm(Outcome ~ . − Orientation + Wall.Area:Roof.Area)
Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(Outcome ~ . − Orientation + Wall.Area:Roof.Area)




```
# there's weird structure in these residuals...

# try one more interaction term
train_sample=sample(1:nrow(train_data),0.7*nrow(train_data))
lm.fit=lm(Outcome~.-Orientation+Wall.Area:Roof.Area+Glazing.Distr:Glazing.Area,
          data=train_data,subset=train_sample)
```

```
summary(lm.fit)
```

```
##
## Call:
## lm(formula = Outcome ~ . - Orientation + Wall.Area:Roof.Area +
##     Glazing.Distr:Glazing.Area, data = train_data, subset = train_sample)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.2974 -1.3703 -0.1818  1.5405  7.4333
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 -6.563e+01  5.581e+00 -11.758  < 2e-16 ***
## Wall.Area                    1.218e-01  1.183e-02  10.290  < 2e-16 ***
## Roof.Area                    1.786e-01  2.501e-02   7.143 3.73e-12 ***
## Height                       6.184e+00  3.245e-01  19.056  < 2e-16 ***
## Glazing.Area                 2.835e+01  1.796e+00  15.783  < 2e-16 ***
## Glazing.Distr                8.979e-01  1.497e-01   5.996 4.17e-09 ***
## Wall.Area:Roof.Area         -3.915e-04  6.633e-05  -5.902 7.13e-09 ***
## Glazing.Area:Glazing.Distr -3.183e+00  5.839e-01  -5.452 8.26e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.833 on 446 degrees of freedom
## Multiple R-squared:  0.9233, Adjusted R-squared:  0.9221
## F-statistic: 767.5 on 7 and 446 DF,  p-value: < 2.2e-16
```

```
mean((predict(lm.fit,newdata=train_data[-train_sample,])-train_data[-train_sample,]$Outcome)^2)
```

```
## [1] 8.126965
```

```
##########################################
# looks like the best regression model so far removes ID, relative compactness, surface area,
# and Orientation, and adds Wall.Area*Roof.Area

# fit all of the data and make an output test set
names(train_data)
```

```
## [1] "Wall.Area"     "Roof.Area"     "Height"        "Orientation"
## [5] "Glazing.Area"  "Glazing.Distr" "Outcome"
```

```
lm.fit=lm(Outcome~.-Orientation+Wall.Area:Roof.Area,data=train_data)
names(test_data)
```

```
## [1] "ID"            "Rel.Compact"   "Surface.Area"  "Wall.Area"
## [5] "Roof.Area"     "Height"        "Orientation"   "Glazing.Area"
## [9] "Glazing.Distr"
```

```
ID=test_data$ID
test_data=test_data[,-c(1,2,3)]
test_data$Orientation=as.factor(test_data$Orientation)
out=data.frame(ID,predict(lm.fit,newdata=test_data))
names(out)=c("ID","Outcome")
write.csv(out, file = "test_linear_reg_bb_2017_11_21.csv",row.names=FALSE)
```

```
# leaderboard score was 2.98759.
```

```
# best subset selection
library(leaps)
n=ncol(train_sample)-1
regfit.full=regsubsets(Outcome~.,data=train_data,subset=train_sample,nvmax=n)
reg.summary=summary(regfit.full)
which.min(reg.summary$cp)
```

```
## [1] 5
```

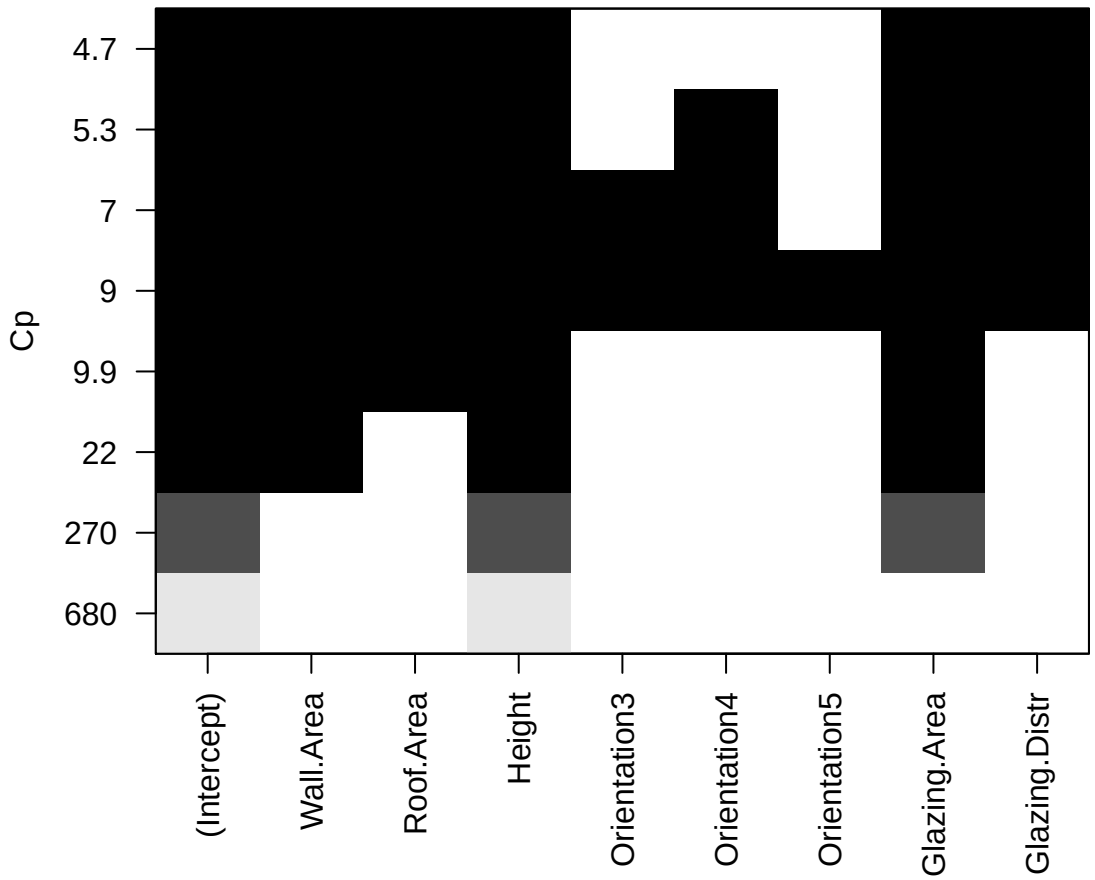
```
which.min(reg.summary$adjr2)
```

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

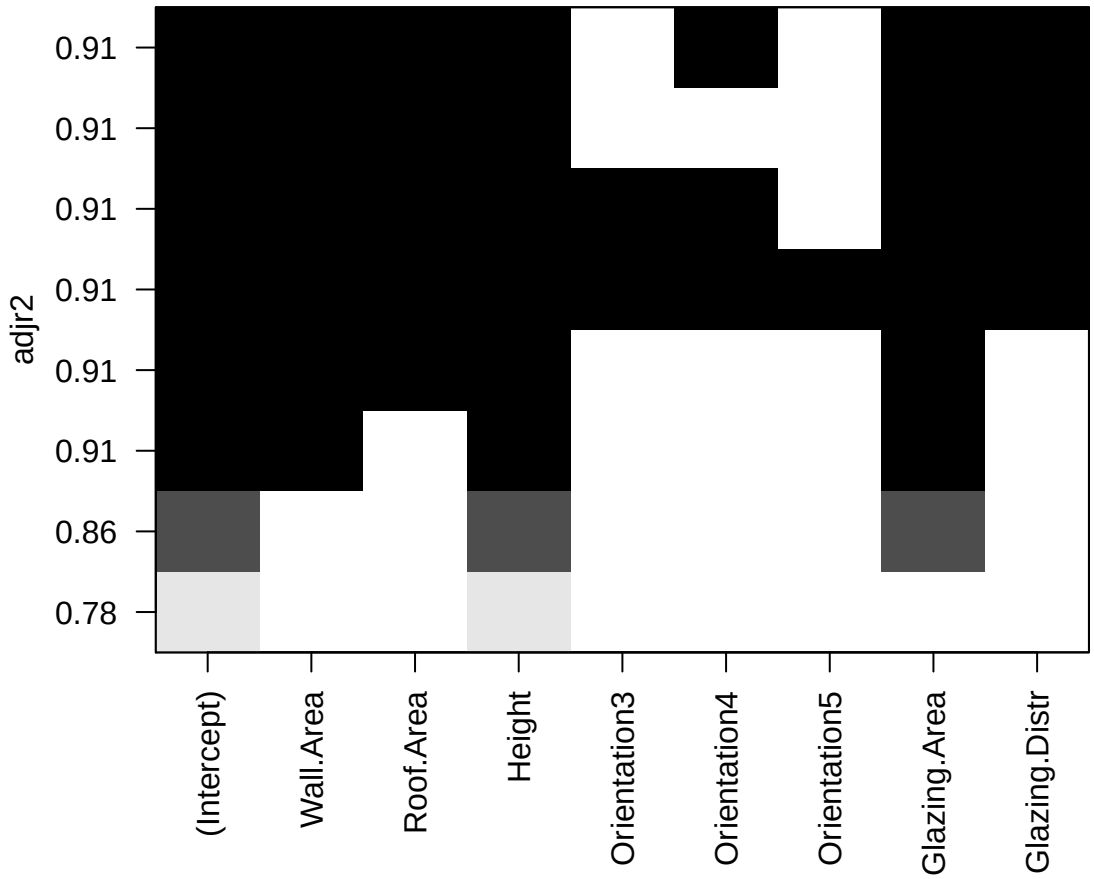
```
which.min(reg.summary$bic)
```

```
## [1] 5
```

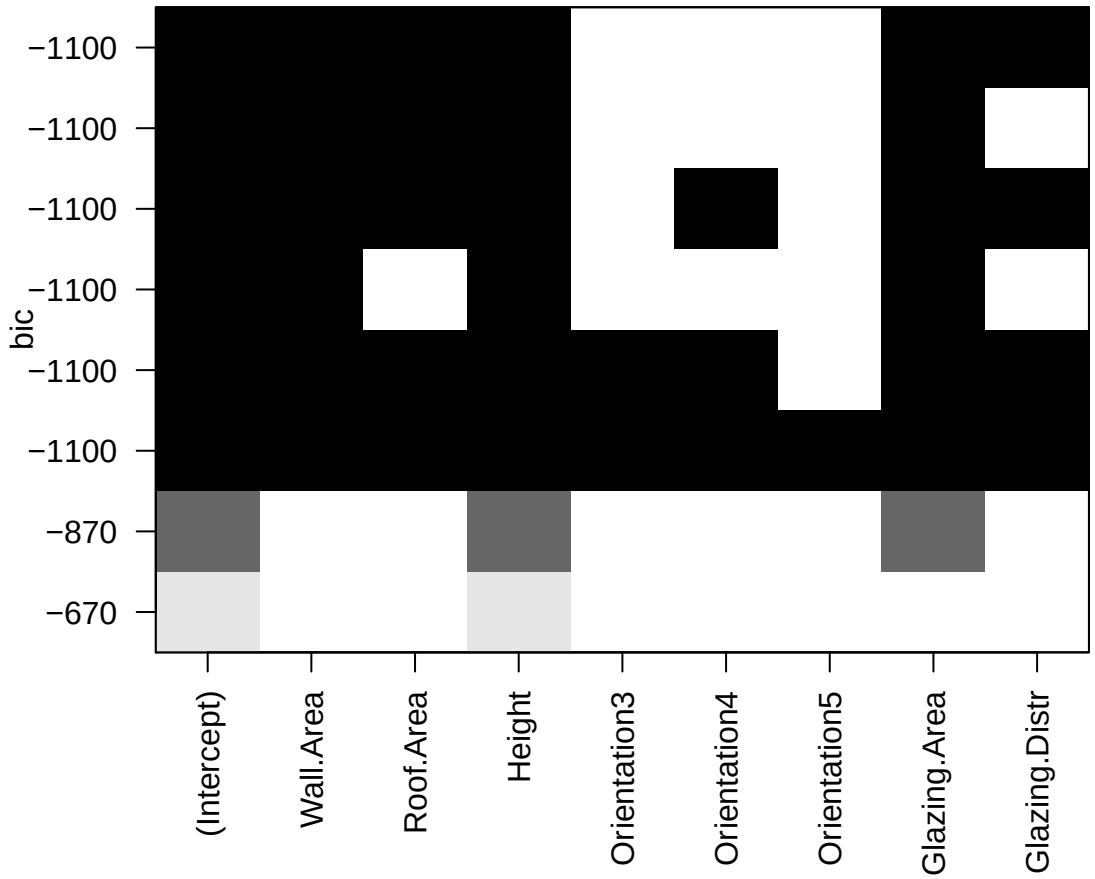
```
plot(regfit.full,scale="Cp")
```




```
plot(regfit.full,scale="adjr2")
```




```
plot(regfit.full, scale="bic")
```




```
# this roughly shows again that the orientation variable is not that useful.
```

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

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

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

```
set.seed(2)
# try on all of the data
train_data=read.csv(file="/home/bridget/Dropbox/STATS202/kaggle/train.csv",header=T)
names(train_data)
```

```
##  [1] "ID"            "Rel.Compact"   "Surface.Area"  "Wall.Area"
##  [5] "Roof.Area"     "Height"        "Orientation"   "Glazing.Area"
##  [9] "Glazing.Distr" "Outcome"
```

```
# remove ID
train_data=train_data[,-1]
n=ncol(train_data)-1
mse=rep(NA,n)
for (i in 1:n){
  rf.train=randomForest(Outcome~.,data=train_data,subset=train_sample,importance=TRUE,
                        ntree=1000,mtry=i)
  mse[i]=mean((predict(rf.train,newdata=train_data[-train_sample,])
               -train_data[-train_sample,]$Outcome)^2)
}
```

```
mse
```

```
## [1] 4.5087000 1.5498738 0.8118018 0.4734871 0.4029490 0.4072314 0.4068964
## [8] 0.4022655
```

```
importance(rf.train)
```

```
##                  %IncMSE IncNodePurity
## Rel.Compact     29.02284     3222.1756
## Surface.Area    21.69960    11763.7246
## Wall.Area       25.45587     1680.8849
## Roof.Area       23.01567    12904.8795
## Height          22.64337    12697.1506
## Orientation    -22.31035       20.3779
## Glazing.Area   152.21330     3522.2040
## Glazing.Distr   39.15866      773.3670
```

```
# try removing orientation
m=n-1
mse=rep(NA,m)
for (i in 1:m){
  rf.train=randomForest(Outcome~.-Orientation,data=train_data,subset=train_sample,
                        importance=TRUE,ntree=1000,mtry=i)
  mse[i]=mean((predict(rf.train,newdata=train_data[-train_sample,])
               -train_data[-train_sample,]$Outcome)^2)
}
mse
```

```
## [1] 4.1673809 1.4592134 0.6766390 0.4348446 0.3974275 0.4001467 0.4115470
```

```
importance(rf.train)
```

```
##                  %IncMSE IncNodePurity
## Rel.Compact     25.80072     3443.6149
## Surface.Area    22.68484    12483.7198
## Wall.Area       25.46491     1633.8242
## Roof.Area       22.96541    12885.4435
## Height          21.34900    11802.6204
## Glazing.Area   152.12798     3493.6154
## Glazing.Distr   39.15434      780.7422
```

```
# try interaction terms
new_train_data=data.frame(train_data,train_data$Wall.Area*train_data$Roof.Area)
mse=rep(NA,n)
for (i in 1:n){
rf.train=randomForest(Outcome~.-Orientation,data=new_train_data,subset=train_sample,
                      importance=TRUE,ntree=1000,mtry=i)
mse[i]=mean((predict(rf.train,newdata=new_train_data[-train_sample,])-
               new_train_data[-train_sample,]$Outcome)^2)
}
mse
```

```
## [1] 4.2494755 1.8199874 0.9567923 0.5193377 0.4075115 0.3919063 0.4034246
## [8] 0.4134357
```

```
importance(rf.train)
```

```
##                                          %IncMSE IncNodePurity
```

```
## Rel.Compact                                    27.33074      3380.244
## Surface.Area                                   15.00343      6483.043
## Wall.Area                                      23.39908      1484.987
## Roof.Area                                      22.29843     12313.095
## Height                                         22.32964     12563.159
## Glazing.Area                                  149.59396      3510.676
## Glazing.Distr                                  37.83034       777.893
## train_data.Wall.Area...train_data.Roof.Area    14.40227      6012.350
```

```
# looks like this is a good model with i = 6.

new_train_data=data.frame(train_data,train_data$Wall.Area*train_data$Glazing.Area)
mse=rep(NA,n)
for (i in 1:n){
rf.train=randomForest(Outcome~.,data=new_train_data,subset=train_sample,
                      importance=TRUE,ntree=1000,mtry=i)
mse[i]=mean((predict(rf.train,newdata=new_train_data[-train_sample,])
             -new_train_data[-train_sample,]$Outcome)^2)
}
mse
```

```
## [1] 3.3655880 0.7547503 0.4569363 0.4214773 0.4130237 0.4152779 0.4229093
## [8] 0.4162522
```

```
importance(rf.train)
```

```
##                                                %IncMSE IncNodePurity
## Rel.Compact                                    22.40218    2738.40381
## Surface.Area                                   22.83928   12305.83305
## Wall.Area                                      21.00531    1209.47254
## Roof.Area                                      22.13123   12222.52544
## Height                                         22.33787   12492.92763
## Orientation                                   -11.40701      20.05413
## Glazing.Area                                   43.49539    1166.35280
## Glazing.Distr                                  26.87688     379.87272
## train_data.Wall.Area...train_data.Glazing.Area 77.28499    3993.56891
```

```
mse=rep(NA,n)
for (i in 1:n){
rf.train=randomForest(Outcome~.,data=new_train_data,subset=train_sample,
                      importance=TRUE,ntree=1000,mtry=i,interaction_depth=4)
mse[i]=mean((predict(rf.train,newdata=new_train_data[-train_sample,])
             -new_train_data[-train_sample,]$Outcome)^2)
}
mse
```

```
## [1] 3.4097038 0.7326848 0.4518546 0.4200604 0.4160338 0.4149918 0.4183489
## [8] 0.4210316
```

```
importance(rf.train)
```

```
##                                                %IncMSE IncNodePurity
## Rel.Compact                                    22.92437     2828.4457
## Surface.Area                                   22.31641    11845.6108
## Wall.Area                                      20.66657     1188.4973
## Roof.Area                                      23.07363    12882.2500
## Height                                         22.01470    12200.8953
```

```
## Orientation                                    -16.16679       20.0177
## Glazing.Area                                    43.35169     1231.4900
## Glazing.Distr                                   26.59603      393.0627
## train_data.Wall.Area...train_data.Glazing.Area  74.03627     3866.8015
```

```
#####################################################
new_train_data=data.frame(train_data,train_data$Wall.Area*train_data$Roof.Area)
names(new_train_data)
```

```
##  [1] "Rel.Compact"
##  [2] "Surface.Area"
##  [3] "Wall.Area"
##  [4] "Roof.Area"
##  [5] "Height"
##  [6] "Orientation"
##  [7] "Glazing.Area"
##  [8] "Glazing.Distr"
##  [9] "Outcome"
## [10] "train_data.Wall.Area...train_data.Roof.Area"
```

```
rf.train=randomForest(Outcome~.-Orientation,data=new_train_data,
                      importance=TRUE,ntree=1000,mtry=6)
test_data=read.csv(file="/home/bridget/Dropbox/STATS202/kaggle/test.csv",header=T)
ID=test_data$ID
test_data=test_data[,-1]
new_test_data=data.frame(test_data,test_data$Wall.Area*test_data$Roof.Area)
names(new_test_data)
```

```
## [1] "Rel.Compact"
## [2] "Surface.Area"
## [3] "Wall.Area"
## [4] "Roof.Area"
## [5] "Height"
## [6] "Orientation"
## [7] "Glazing.Area"
## [8] "Glazing.Distr"
## [9] "test_data.Wall.Area...test_data.Roof.Area"
```

```
names(new_test_data)[9]=names(new_train_data)[10]
out=data.frame(ID,predict(rf.train,newdata=new_test_data))
names(out)=c("ID","Outcome")
write.csv(out, file = "test_rand_forest_bb_2017_11_21.csv",row.names=FALSE)

# leaderboard score was 0.50795.
```