|  |
| --- |
| STAT 420- David M. Dapiaz  4-24-2015 |

|  |
| --- |
| Outlier Detection and Mitigation in Linear Regression  Shairoz Sohail (ssohai3), Kevin Nellessen (nellssn2) |
|  |
|  |

Introduction

In this paper we outline a method for increasing robustness of a simple linear regression by identifying and systematically adjusting outliers. We seek mainly to answer the question:

*Does a systematic reduction of outlier residual distance for a linear regression fitted to a training set help increase predictive power of the model for a testing set?*

For the purposes of this paper we define ‘outliers’ to mean points with a high Cook’s Distance (for a description of Cook’s Distance, please see section “Materials and Methods”).

The data that will be utilized is the “Diamonds” dataset, found within the GGplot2[1]  package. The description of the dataset as given by the source is as follows:

**Description**

A dataset containing the prices and other attributes of almost 54,000 diamonds. The variables are as follows:

**Usage**

data(diamonds)

**Format**

A data frame with 53940 rows and 10 variables

**Details**

* price. price in US dollars (\$326–\$18,823)
* carat. weight of the diamond (0.2–5.01)
* cut. quality of the cut (Fair, Good, Very Good, Premium, Ideal)
* colour. diamond colour, from J (worst) to D (best)
* clarity. a measurement of how clear the diamond is (I1 (worst), SI1, SI2, VS1, VS2, VVS1, VVS2, IF (best))
* x. length in mm (0–10.74)
* y. width in mm (0–58.9)
* z. depth in mm (0–31.8)
* depth. total depth percentage = z / mean(x, y) = 2 \* z / (x + y) (43–79)
* table. width of top of diamond relative to widest point (43–95)

\*For the purposes of this paper, only the “carat” and “price” variables will be used.

# Materials and Methods:

**The dataset used is outlined in the previous section.**

**Simple Linear Regression** will be utilized for the model in this paper. Informally, simple linear regression is a way of measuring the type of relationship between two variables, usually denoted a predictor and a response. Collecting observations of these variables, determining a coordinate system and using a parameter to minimize a specific metric between the data is usually the procedure behind many types of regression. In simple linear regression using the **Least-Squares** approach, the metric being minimized is the distance of the data points to a line that generalizes their relationship. Thus, the simple linear regression equation is as follows:

 y = \alpha + \beta x,

This is the equation of a straight line with y-intercept https://mixtt.files.wordpress.com/2008/06/alpha.jpg and slope http://www.backupassist.com/images/beta-300x300.png. Usually, the “y” denotes the response variable and “x” denotes the predictor variable. For an estimated (sample) simple linear regression we get an error termand sample parameters and this becomes:

where εi’s are independent Normal( 0 , σ2 ) (iid Normal( 0 , σ2 ))

As mentioned, the Least-Squares approach minimizes the squared distances between this line and the data points, and these distances are known as ‘residuals’. Formally:

Simple Linear regression makes several key assumptions:

* + There is a linear relationship between the predictor and response variable
  + The predictor variable is normally distributed
  + The errors are normally distributed with mean 0 and variance

A key problem with minimizing the squared distances is that observations with extreme values get their residuals vastly over-represented. In a predictive framework this is usually not desired as these observations ‘pull’ the regression line away from what we expect to observe and thus reduce our predictive power. Some regression designs like Quantile Regression avoid this by minimizing absolute distances and are known as more robust (resistant to outliers) methods, but are computationally difficult.

One of the central concepts of this paper is Cook’s Distance. It is defined as:

D_i = \frac{ \sum_{j=1}^n (\hat Y_j\ - \hat Y_{j(i)})^2 }{p \ \mathrm{MSE}},

Where:

\hat Y_j \, is the prediction from the full regression model for observation *j*;

\hat Y_{j(i)}\, is the prediction for observation *j* from a refitted regression model in which observation *i* has been omitted;

p is the number of fitted parameters in the model;

 \mathrm{MSE} \, is the [mean square error](http://en.wikipedia.org/wiki/Mean_square_error) of the regression model.

Informally, this is a measure of a point’s influence on the regression line when performing a least squares regression analysis. In a predictive setting, unless there is a cluster of outliers we wish to reduce the effect these outliers have on our predictions as they can be due to noise or measurement error. In certain situations, we may simply delete the outliers, but the goal of this paper is to provide a method that still provides these observations with some influence, but proportional to our assumptions as to the nature of the outliers.

We define a **K-Adjusted Residual** as:

Where 0 < k< 1 is a constant chosen before performing the regression.

The steps to utilizing the Adjusted Residual are as follows:

1. Decide on “k” term and a cutoff for Cook’s Distance to classify outliers
2. Perform Simple Linear Regression
3. Classify points that fall above Cook’s Distance cutoff and use the Adjusted Residual equation to come up with an adjusted value
4. Delete the classified points and replace them with the adjusted values

# Results:

We use a training set of 1,000 random observations of diamond carats from the diamonds data set. We check our SLR assumptions and train a linear model on these observations with diamond carat as the predictor variable and diamond price as the response variable. We then utilized the K-Adjusted Residual method and trained a second linear model on the adjusted data with the same predictor and response variables. We then use these models on the full data set (53,940 observations) to predict diamond prices given their carats, the differences in error of these predictions is the benchmark used for model performance.

Standard Error for non-adjusted model: 133,711.7

Standard Error for adjusted model: 132, 529.1

This accounts for a ~.8% reduction in error. Considering the training set used was ~1.853% of the full data this is a noticeable improvement in prediction power.

We provide model parameters below for model1 (non-adjusted) and model2 (adjusted)

Non Adjusted Model

## Residuals:  
## Min 1Q Median 3Q Max   
## -14115.9 -674.9 -156.3 267.7 8743.9   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -1549.88 97.09 -15.96 <2e-16 \*\*\*  
## carat 6723.31 99.64 67.47 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1689 on 1003 degrees of freedom  
## Multiple R-squared: 0.8195, Adjusted R-squared: 0.8193   
## F-statistic: 4553 on 1 and 1003 DF, p-value: < 2.2e-16

Adjusted Model

## Residuals:  
## Min 1Q Median 3Q Max   
## -10561.0 -681.2 -137.2 288.9 8666.2   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -1626.85 94.47 -17.22 <2e-16 \*\*\*  
## carat 6826.41 96.95 70.41 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1643 on 1003 degrees of freedom  
## Multiple R-squared: 0.8317, Adjusted R-squared: 0.8316   
## F-statistic: 4958 on 1 and 1003 DF, p-value: < 2.2e-16

These analyses were done using k=0.5. The choice of k can be dependent on quite a few things, most of them out of the scope of this paper. A few things we can use to choose k include:

-Probability of experimental error

-Function of Cook’s Distance

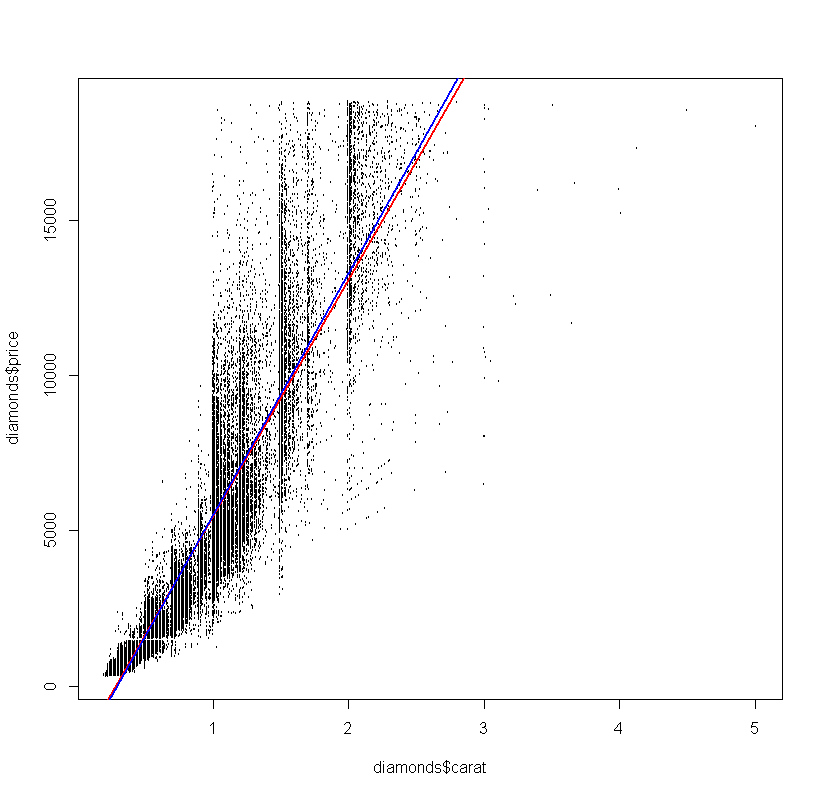
-Prior belief about observations

-Clustering tendencies of observations

-Size of data/ training set

# Conclusion:

We observe that shifting outlier points in a systematic manner improves the predictive power of a simple linear regression in terms of standard error. Over larger datasets this should help offset the problem of noise disrupting predictive power of the model. In theory, as long as all assumptions are satisfied, this method should be generalizable to multiple linear regression. Finally, we provide the plot of the full data with lines imposed for both the regular (red) and adjusted model (blue) to demonstrate the increased robustness of the adjusted model.



\*Forced 0 intercept was considered, but ultimately results in worse model and dropped

References:

-David M. Dalpiaz, Visiting Professor of Statistics,

University of Illinois at Urbana-Champaign

-Practical Regression and ANOVA Using R  
 Julian J. Faraway

[1]

-H. Wickham. ggplot2: elegant graphics for data analysis. Springer New York,

2009.

Appendix:

Training Set: A training set is a set of [data](http://en.wikipedia.org/wiki/Data) used in various areas of [information science](http://en.wikipedia.org/wiki/Information_science) to discover potentially predictive relationships

Testing Set: A test set is a set of [data](http://en.wikipedia.org/wiki/Data) used in various areas of [information science](http://en.wikipedia.org/wiki/Information_science) to assess the strength and utility of a predictive relationship.

\*Forced 0 intercept increases standard error significantly (~ 25,000) and so was abandoned. Models must be used carefully for diamonds with carat<1.

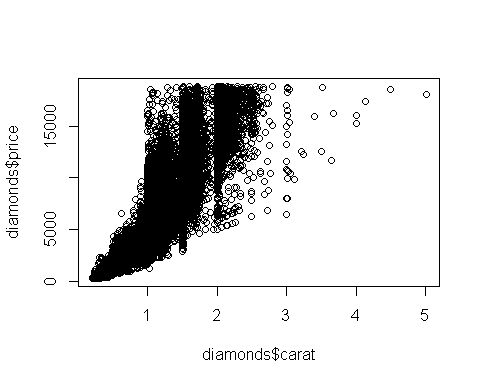
Full code and output used for paper is attached on next page

#install.packages('dplyr')  
#install.packages('ggplot2')  
#install.packages('broom')  
library(dplyr)

library(ggplot2)

library(broom)

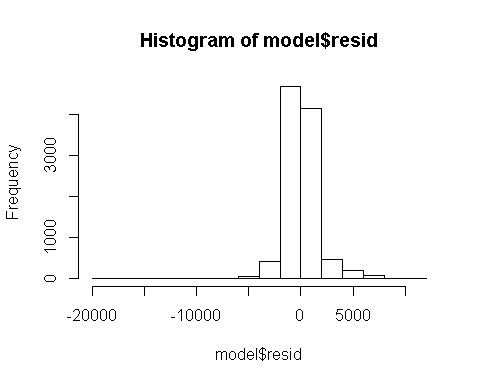
data(diamonds)  
  
plot(diamonds$carat, diamonds$price)



#Creating Training Set called "trainout"  
set.seed(10)  
train= diamonds[sample(nrow(diamonds), 10000), ]  
outliers= diamonds[diamonds$carat>4, ,]  
trainout= rbind(train, outliers)  
  
#Fitting Linear Model to Training Data  
model= lm(price ~ carat, data=trainout)  
summary(model)

##   
## Call:  
## lm(formula = price ~ carat, data = trainout)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -18018.3 -767.5 -32.8 498.2 10697.5   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -2164.66 29.75 -72.77 <2e-16 \*\*\*  
## carat 7624.95 32.04 237.97 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1530 on 10003 degrees of freedom  
## Multiple R-squared: 0.8499, Adjusted R-squared: 0.8499   
## F-statistic: 5.663e+04 on 1 and 10003 DF, p-value: < 2.2e-16

hist(model$resid)



#Creating Test Data to Pass  
Test= data.frame(diamonds$carat)  
Test2= diamonds[, 1, drop=FALSE]  
head(Test2)

## carat  
## 1 0.23  
## 2 0.21  
## 3 0.23  
## 4 0.29  
## 5 0.31  
## 6 0.24

#Passing Test Data to Trained Model and getting MSE  
ModelTest=predict(model, Test2)  
residuals= (ModelTest - diamonds$price)  
SqrResiduals= (residuals)^2  
MSE= sum(SqrResiduals)/(length(diamonds)-2)  
SE= sqrt(MSE)  
SE

## [1] 127261.8

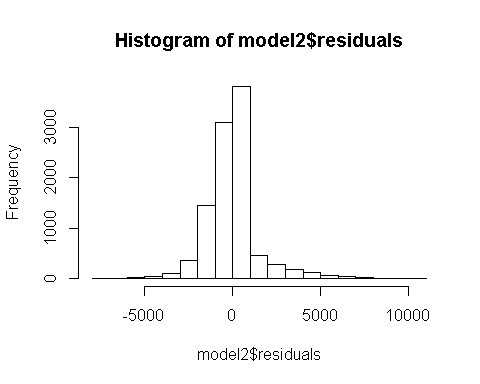
#Creating Adjusted Residuals and Using   
HighCook <- augment(model)$.cooksd > 100/nrow(trainout)  
HighCookAdj <- augment(model)$price - 0.9\*resid(model)  
  
  
trainout$new\_price <- trainout$price  
trainout$new\_price[HighCook] <- HighCookAdj[HighCook]  
  
  
model2 <- lm(new\_price ~ carat, data = trainout)  
summary(model)

##   
## Call:  
## lm(formula = price ~ carat, data = trainout)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -18018.3 -767.5 -32.8 498.2 10697.5   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -2164.66 29.75 -72.77 <2e-16 \*\*\*  
## carat 7624.95 32.04 237.97 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1530 on 10003 degrees of freedom  
## Multiple R-squared: 0.8499, Adjusted R-squared: 0.8499   
## F-statistic: 5.663e+04 on 1 and 10003 DF, p-value: < 2.2e-16

summary(model2)

##   
## Call:  
## lm(formula = new\_price ~ carat, data = trainout)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -7506.3 -797.8 -11.2 554.3 10642.2   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -2294.97 28.40 -80.8 <2e-16 \*\*\*  
## carat 7808.69 30.59 255.2 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1461 on 10003 degrees of freedom  
## Multiple R-squared: 0.8669, Adjusted R-squared: 0.8669   
## F-statistic: 6.515e+04 on 1 and 10003 DF, p-value: < 2.2e-16

hist(model2$residuals)



#Passing Test Data to Trained Model and getting MSE  
ModelTest2=predict(model2, Test2)  
residuals2= (ModelTest2 - diamonds$price)  
SqrResiduals2= (residuals2)^2  
MSE2= sum(SqrResiduals2)/(length(diamonds)-2)  
SE2= sqrt(MSE2)  
  
#Standard Error of Original Model vs. Model with Adjusted Residual method  
SE

## [1] 127261.8

SE2

## [1] 127170.8

plot(diamonds$carat, diamonds$price, cex=0.3, ylim=c(0,20000))  
abline(model, col='red', lwd=2)  
abline(model2, col='blue', lwd=2)

