Predictive Modeling of Mechanical Behavior from Process Parameters in Additve Manufacturing

Austin Ngo

May 3, 2019

Contents

1 Introduction

Additive Manufacturing (AM) or 3-D Printing of metallic alloys and materials is a relatively recent category of processes for fabricating mechanical components, generally from a powder or wire feedstock. The benefits of using AM are a decrease in material waste (as opposed to conventional cutting and machining processes), the ability to create geometrically complex components with Computer Automated Design (CAD) software, and the flexibility of fabricating virtually any design on a single AM machine.

For AM processes to be implemented in the manufacture of specialized, critical components used in aerospace or naval applications, manufacturers and engineers must have a comprehensive understanding of how to optimize process parameters for the desired applications. A common AM process is Selective Laser Melting (SLM), which involves melting a bed of metallic powder into the desired shape with a laser, building a component layer by layer. The thermodynamic complexities between hundreds (or thousands) of rapid laser passes and the metallic powder feed create ample opportunity for the formation of defects, any of which could initiate mechanical failure in a critical application.

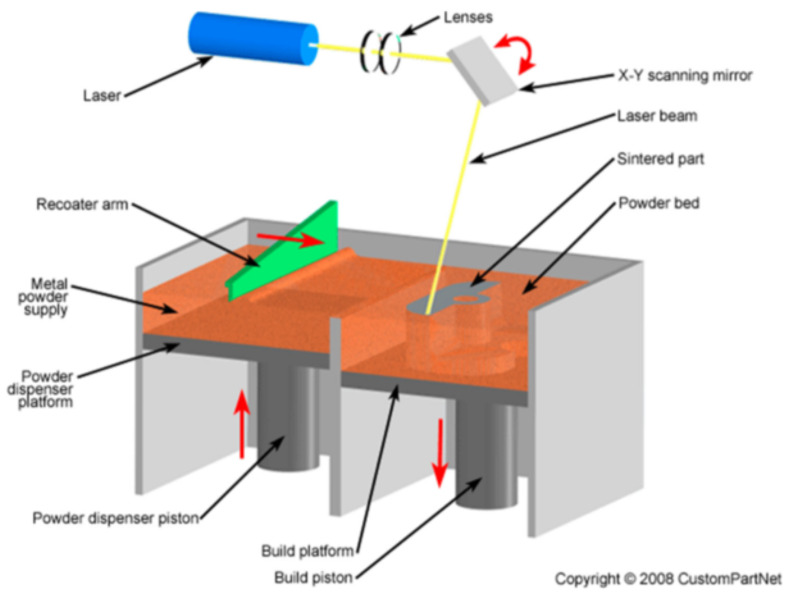


Figure 1: A typical metallic powder-bed Additive Manufacturing (AM) System. A laser energy source is directed to melt powder into the desired geometry, and the next layer of powder is added and subsequently melted until the build is complete. (1)

1.1 Purpose of Semester Project Assignment

My graduate research is focused on studying the mechanical behavior of SLM-processed alloys. I would like to be able to use data science and data analysis techniques to predict the mechanical properties of a component build based on its processing input parameters. However, as I do not yet have adequate experimental data, this semester project will be a proof-of-concept exploration using a dataset related to polymer 3-D Printing.

2 Semester Project Procedure

2.1 Part a) Define Question

Ultimately, my semester project question is, *Given a dataset of AM process parameters and the resulting mechanical performance, how well can I predict the end result from inputting process parameters?*

My methods to answer this project question will involve exploratory data analysis (EDA) of the variable behavior in these tests, as well as building a variety of predictive models and evaluating their accuracy.

My ideal dataset would include several process input parameters, systematically varied. For example, some of these parameters could include energy source power (or temperature), travel speed across the print bed, material type, and print bed heating temperature. In terms of resulting output mechanical data, some typical and relatively easy-to-acquire data could come from tensile testing, such as Ultimate Tensile Strength (UTS) and strain/elongation.

For this project, the data set I am able to access comes from Kaggle:

"3D Printer Dataset for Mechanical Engineers" (2) Link: (https://www.kaggle.com/afumetto/3dprinter)

This is a dataset of 50 polymer 3-D Printing builds, systematically varying certain process parameters and materials (ABS or PLA plastic), as well as some mechanical data on the results of the build. The printer used was an Ultimaker S5 3-D printer. Mechanical testing was done using a Sincotec GMBH tester capable of force of 20 kN.

There were 50 different builds run, or 50 observations n and 12 variables. 9 variables are input process parameters and 3 are output mechanical data results. 2 of the input variables (infill pattern and material) are discrete categorical variables.

2.2 SemProj Part b) Cleaning and EDA

As the data comes from Kaggle, it was relatively simple to insert the .csv file into R.

```
MyData <- read.csv(file = "H:/Git/19s-dsci353-353m-453-aqn5/1-assignments/SemProj-453/3dprinter/data.cs
head(MyData, n = 3) # Show what the dataset contains
```

```
##   layer_height wall_thickness infill_density infill_pattern
## 1         0.02              8             90           grid
## 2         0.02              7             90      honeycomb
## 3         0.02              1             80           grid
##   nozzle_temperature bed_temperature print_speed material fan_speed
## 1                220              60          40      abs         0
## 2                225              65          40      abs        25
## 3                230              70          40      abs        50
##   roughness tension_strenght elongation
## 1        25               18        1.2
## 2        32               16        1.4
## 3        40                8        0.8
```

These are the library packages I will be calling for this project:

```
library(tidyverse)
```

```
## -- Attaching packages ----------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.1.0       v purrr   0.3.0
## v tibble  2.0.1       v dplyr   0.8.0.1
## v tidyr   0.8.2       v stringr 1.4.0
## v readr   1.3.1       v forcats 0.4.0

## -- Conflicts -------------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
library(leaps)
library(modelr)
library(broom)
```

```
##
## Attaching package: 'broom'

## The following object is masked from 'package:modelr':
##
##     bootstrap
```

```
library(tree)
library(randomForest)
```

```
## randomForest 4.6-14

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

##
## Attaching package: 'randomForest'
```

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

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

2.2.1 Databook

Here I define each of the variables:

Input Parameters:

Layer height - height/thickness of each layer of the build (mm)

Wall thickness - thickness of the outer "contour" edge, as opposed to the bulk structure (mm)

Infill density - percentage of filling the internal structure (%)

Infill pattern - pattern the material is deposited in, grid or honeycomb

Nozzle temperature - nozzle temperature controls properties of melting, extrusion and cooling (C)

Bed temperature - temperature of build platform (C)

Print speed - rate of nozzle depositing material (mm/s)

Material - plastic type, acrylonitrile butadiene styrene (ABS) or polylactic acid (PLA)

Fan Speed - affects material cooling rate (%)

Output Parameters:

Roughness - sample surface roughness ($micron$)

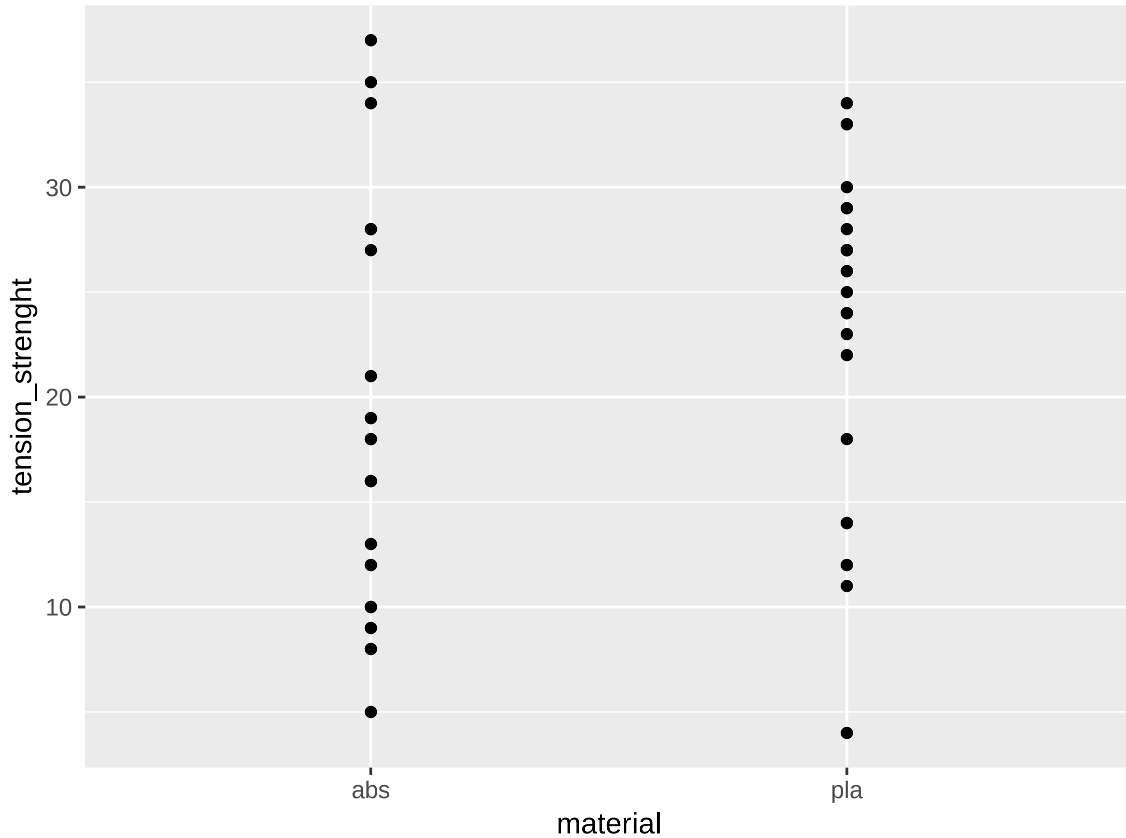
Tension strength - ultimate tensile strength (MPa)

Elongation - sample strain (%)

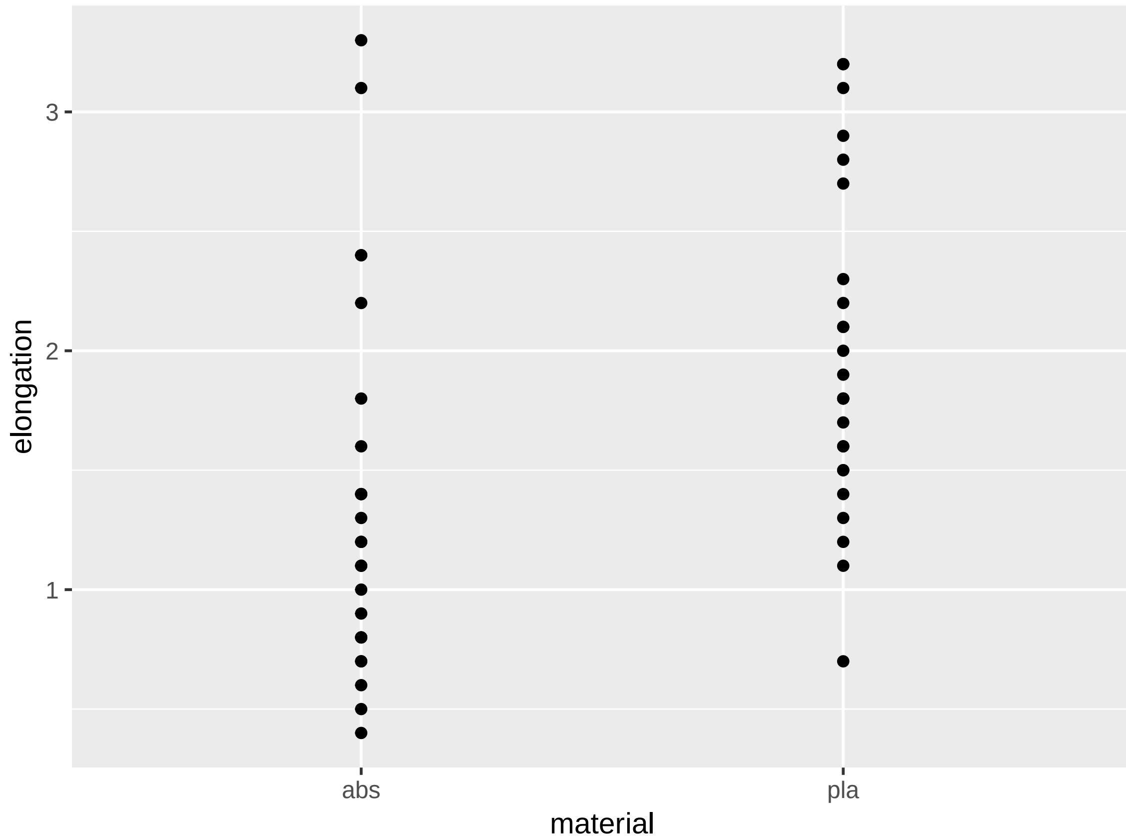
2.2.2 Exploratory Data Analysis

See what the data looks like, paying special attention to the difference of the two materials.

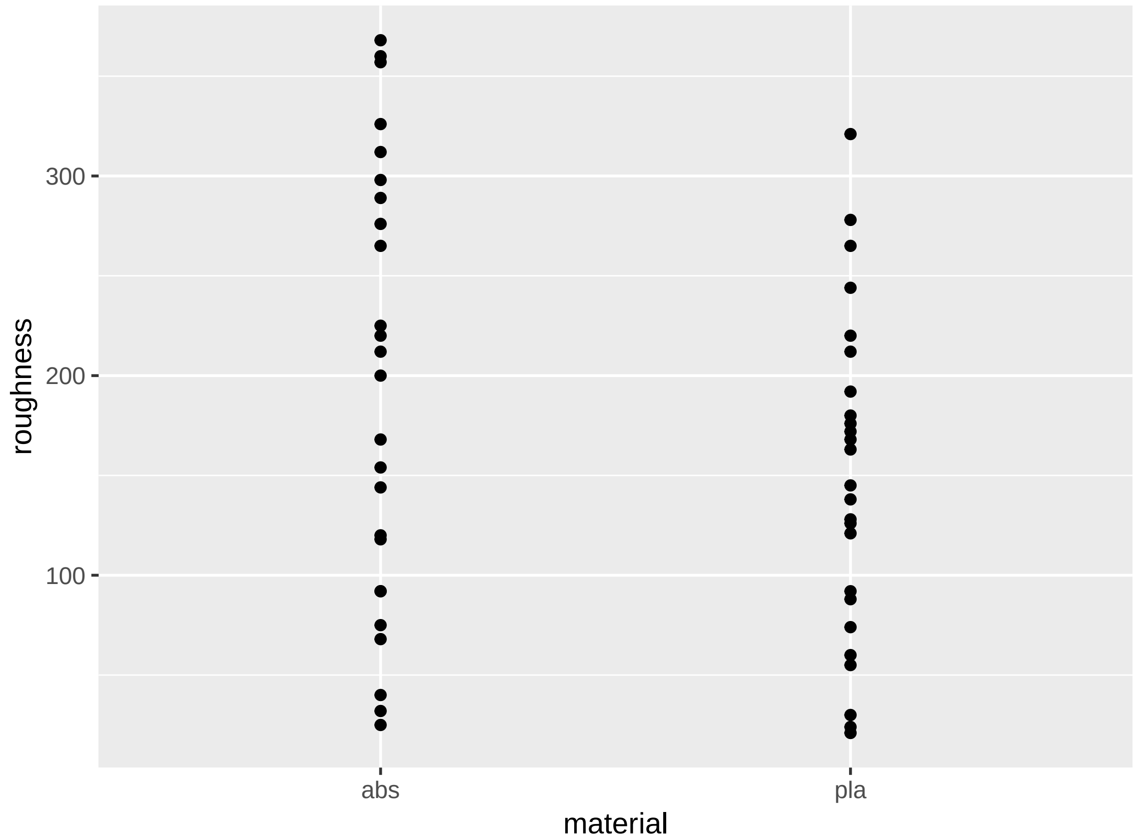
```
attach(MyData)
ggplot(data = MyData) + geom_point(mapping = aes(x = material, y = tension_strenght))
```



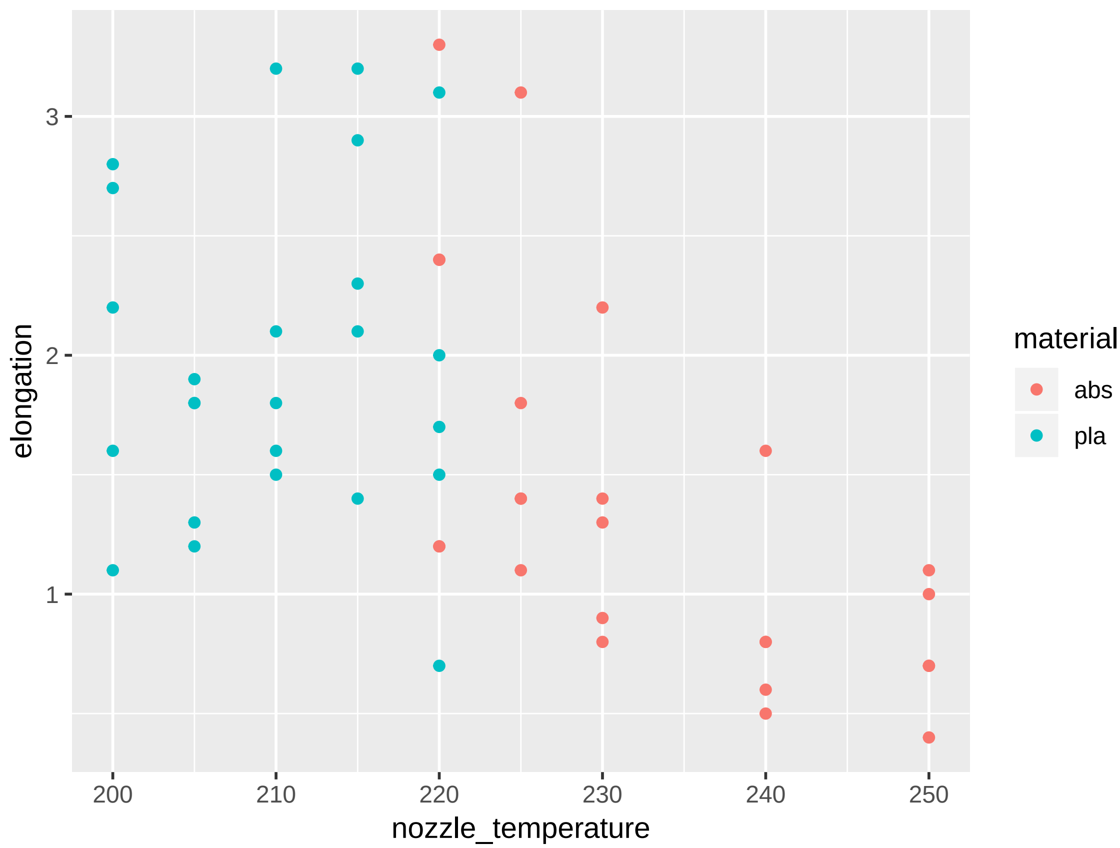

```r
ggplot(data = MyData) + geom_point(mapping = aes(x = material, y = elongation))
```



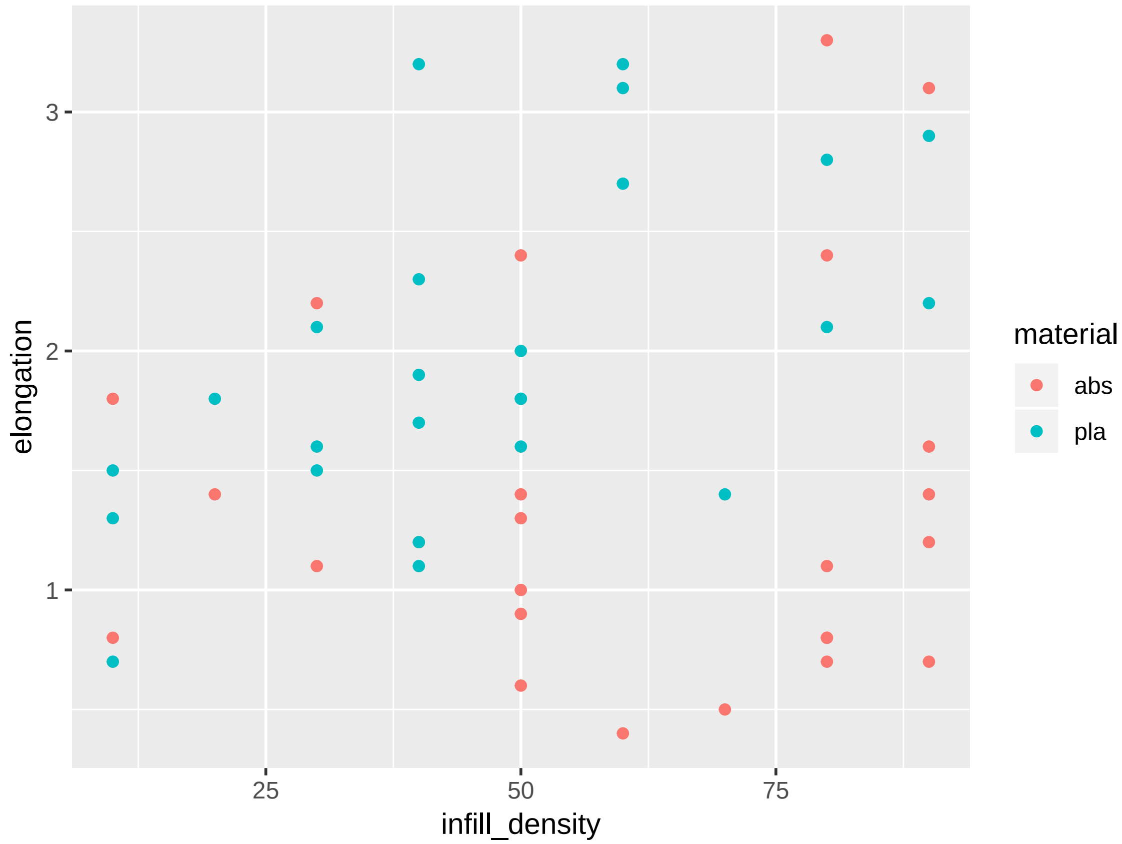

```
ggplot(data = MyData) + geom_point(mapping = aes(x = material, y = roughness))
```



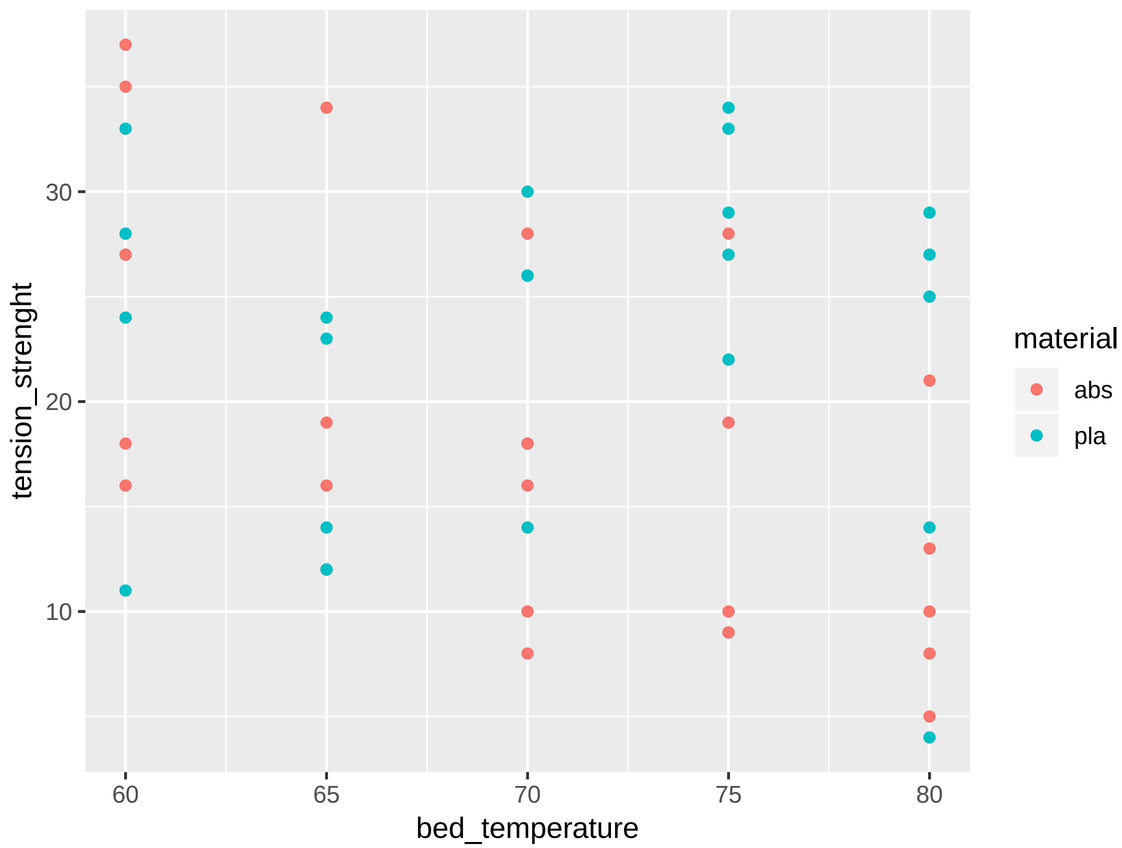

```
ggplot(data = MyData) + geom_point(mapping = aes(x = nozzle_temperature, y = elongation, color = materi
```



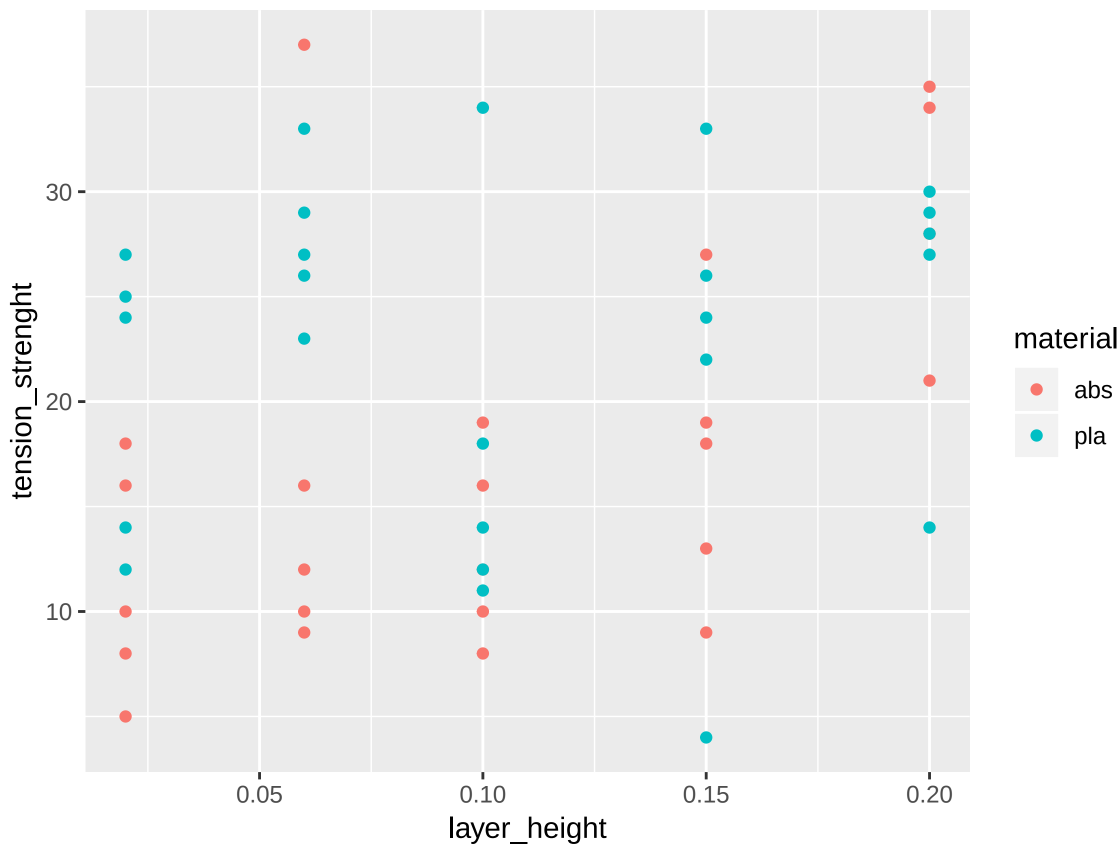

```
ggplot(data = MyData) + geom_point(mapping = aes(x = infill_density, y = elongation, color = material))
```




```
ggplot(data = MyData) + geom_point(mapping = aes(x = bed_temperature, y = tension_strenght, color = mate
```




```
ggplot(data = MyData) + geom_point(mapping = aes(x = layer_height, y = tension_strenght, color = materia
```



The most significant difference between the two materials is the nozzle operating temperature. ABS has a higher glass transition temperature and therefore must be melted and extruded at a higher nozzle temperature than PLA. Comparing the materials and output data side-by-side, it seems that these materials have similar mechanical properties, however, so I will not focus too much on separating them for further data analysis. Because this project is a proof of concept test on the viability of predictive modeling with a relatively small number of observations, I will disregard material differences instead of splitting my dataset further.

Next I look for correlations to find strong predictors of each of the output parameters.

```
res <- cor(MyData[,-c(4,8)], y = NULL, use = "complete.obs", method = "pearson")
round(res, 2)
```

```
##                    layer_height wall_thickness infill_density
## layer_height               1.00          -0.19           0.00
## wall_thickness            -0.19           1.00           0.10
## infill_density             0.00           0.10           1.00
## nozzle_temperature         0.00          -0.12           0.24
## bed_temperature            0.00          -0.03           0.00
## print_speed               -0.06          -0.42          -0.09
## fan_speed                  0.00          -0.03           0.00
## roughness                  0.80          -0.23           0.12
## tension_strenght           0.34           0.40           0.36
## elongation                 0.51           0.18           0.16
##                    nozzle_temperature bed_temperature print_speed
## layer_height                     0.00            0.00       -0.06
## wall_thickness                  -0.12           -0.03       -0.42
## infill_density                   0.24            0.00       -0.09
```

```
## nozzle_temperature              1.00          0.60        0.00
## bed_temperature                 0.60          1.00        0.00
## print_speed                     0.00          0.00        1.00
## fan_speed                       0.60          1.00        0.00
## roughness                       0.35          0.19        0.12
## tension_strenght               -0.41         -0.25       -0.26
## elongation                     -0.53         -0.30       -0.23
##                    fan_speed roughness tension_strenght elongation
## layer_height            0.00      0.80             0.34       0.51
## wall_thickness         -0.03     -0.23             0.40       0.18
## infill_density          0.00      0.12             0.36       0.16
## nozzle_temperature      0.60      0.35            -0.41      -0.53
## bed_temperature         1.00      0.19            -0.25      -0.30
## print_speed             0.00      0.12            -0.26      -0.23
## fan_speed               1.00      0.19            -0.25      -0.30
## roughness               0.19      1.00             0.05       0.10
## tension_strenght       -0.25      0.05             1.00       0.84
## elongation             -0.30      0.10             0.84       1.00
```

It seems there are only a few strong correlations of the input parameters to the output parameters, such as roughness vs layer height and elongation vs nozzle temperature or layer height. Of course tensile strength and elongation correlate strongly but not with the third output, roughness.

Part of this difficulty of finding good correlative relationships is certainly due to the small size of the dataset, and another factor may be the large spread that can occur in mechanical defects due to the quality and consistency of the AM printing process.

As far as assessing the validity of these correlations, it would makes sense to see a strong correlation between printing layer height and roughness, as thicker layering could be expected to result in a rougher surface finish and less geometrically precise build. Looking at the general negative trend between nozzle temperature and elongation, it is possible that the higher temperatures damage the structural integrity of the polymer feed, allowing the resulting build to stretch further.

2.3 SemProj Part c) Modeling and Statistical Learning

2.3.1 Modeling Approaches

The main challenge is that this is a fairly small dataset where n = 50 and p = 12, while an ideal dataset would have n>>p. Because of this issue, I will first approach modeling with *subset and stepwise selection methods*. I will also look at *regression trees and random forest* methods, as they are easily interpretable and will appear more informative than linear models, given the low number of observation points. Finally, from domain knowledge I am interested in examining the viability of a *multivariate regression* model. Most studies of additive manufacturing process parameters will examine at least two or three process input variables, and I would like to do the same applying data science methods.

2.3.2 Best Subset Selection

First I look at elongation output, and want to eliminate the other two output variables so they are not mixed with the input variables.

```
invisible(attach(MyData))
```

```
## The following objects are masked from MyData (pos = 3):
##
##     bed_temperature, elongation, fan_speed, infill_density,
##     infill_pattern, layer_height, material, nozzle_temperature,
##     print_speed, roughness, tension_strenght, wall_thickness
```

```r
set.seed(5) # for reproducible results
# Eliminate other output variables for now
MyData$tension_strenght <- NULL
MyData$roughness <- NULL
train <-  sample(1:nrow(MyData), nrow(MyData)/2) #create training dataset, splitting data in half for t
```

```r
regfit.full <- regsubsets(elongation ~ ., data = MyData, nvmax = 9)
```

```
## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax,
## force.in = force.in, : 1 linear dependencies found

## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax,
## force.in = force.in, : nvmax reduced to 8
```

```r
reg.summary <- summary(regfit.full)
reg.summary
```

```
## Subset selection object
## Call: regsubsets.formula(elongation ~ ., data = MyData, nvmax = 9)
## 9 Variables  (and intercept)
##                         Forced in Forced out
## layer_height                FALSE      FALSE
## wall_thickness              FALSE      FALSE
## infill_density              FALSE      FALSE
## infill_patternhoneycomb     FALSE      FALSE
## nozzle_temperature          FALSE      FALSE
## bed_temperature             FALSE      FALSE
## print_speed                 FALSE      FALSE
## materialpla                 FALSE      FALSE
## fan_speed                   FALSE      FALSE
## 1 subsets of each size up to 8
## Selection Algorithm: exhaustive
##          layer_height wall_thickness infill_density
## 1  ( 1 ) " "          " "            " "
## 2  ( 1 ) "*"          " "            " "
## 3  ( 1 ) "*"          " "            "*"
## 4  ( 1 ) "*"          " "            "*"
## 5  ( 1 ) "*"          " "            "*"
## 6  ( 1 ) "*"          " "            "*"
## 7  ( 1 ) "*"          "*"            "*"
## 8  ( 1 ) "*"          "*"            "*"
##          infill_patternhoneycomb nozzle_temperature bed_temperature
## 1  ( 1 ) " "                     "*"                " "
## 2  ( 1 ) " "                     "*"                " "
## 3  ( 1 ) " "                     "*"                " "
## 4  ( 1 ) " "                     "*"                " "
## 5  ( 1 ) " "                     "*"                "*"
## 6  ( 1 ) " "                     "*"                "*"
## 7  ( 1 ) " "                     "*"                " "
## 8  ( 1 ) "*"                     "*"                "*"
##          print_speed materialpla fan_speed
## 1  ( 1 ) " "         " "         " "
## 2  ( 1 ) " "         " "         " "
## 3  ( 1 ) " "         " "         " "
## 4  ( 1 ) "*"         " "         " "
```

```
## 5  ( 1 ) " "         "*"         " "
## 6  ( 1 ) "*"         "*"         " "
## 7  ( 1 ) "*"         "*"         "*"
## 8  ( 1 ) "*"         "*"         " "
```

```r
coef(regfit.full, 8)
```

```
##             (Intercept)            layer_height          wall_thickness
##            18.435640218             6.382036581             0.030827163
##          infill_density infill_patternhoneycomb      nozzle_temperature
##             0.009890413            -0.062281806            -0.109642480
##         bed_temperature             print_speed             materialpla
##             0.104223026            -0.003374979            -1.783729697
```

```r
reg.summary$rsq
```

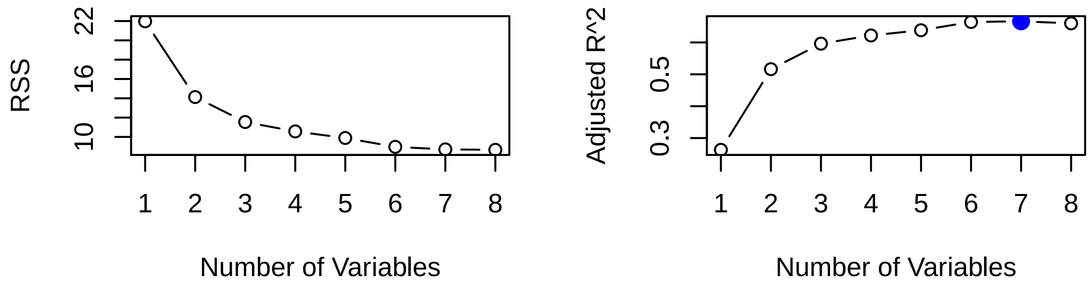
```
## [1] 0.2781999 0.5358405 0.6208192 0.6527703 0.6751489 0.7052882 0.7139090
## [8] 0.7153737
```

Here we see two important pieces of information. First, the most important input parameters are identified. Nozzle temperature, layer height, and infill density seem to be most important, respectively. Second, we see how the adjusted R^2 value increases as we add more variables. The addition from a 1-variable model (nozzle temperature) to 2-variable modeling (nozzle temperature and layer height) gives the larges jump in R^2 value. To find the best number of variables we should look at statistics such as Cp, adjusted R^2, and BIC.

```r
par(mfrow = c(2,2))
plot(reg.summary$rss, xlab = "Number of Variables", ylab = "RSS", type = "b")
plot(reg.summary$adjr2, xlab = "Number of Variables", ylab = "Adjusted R^2", type = "b")
which.max(reg.summary$adjr2)
```

```
## [1] 7
```

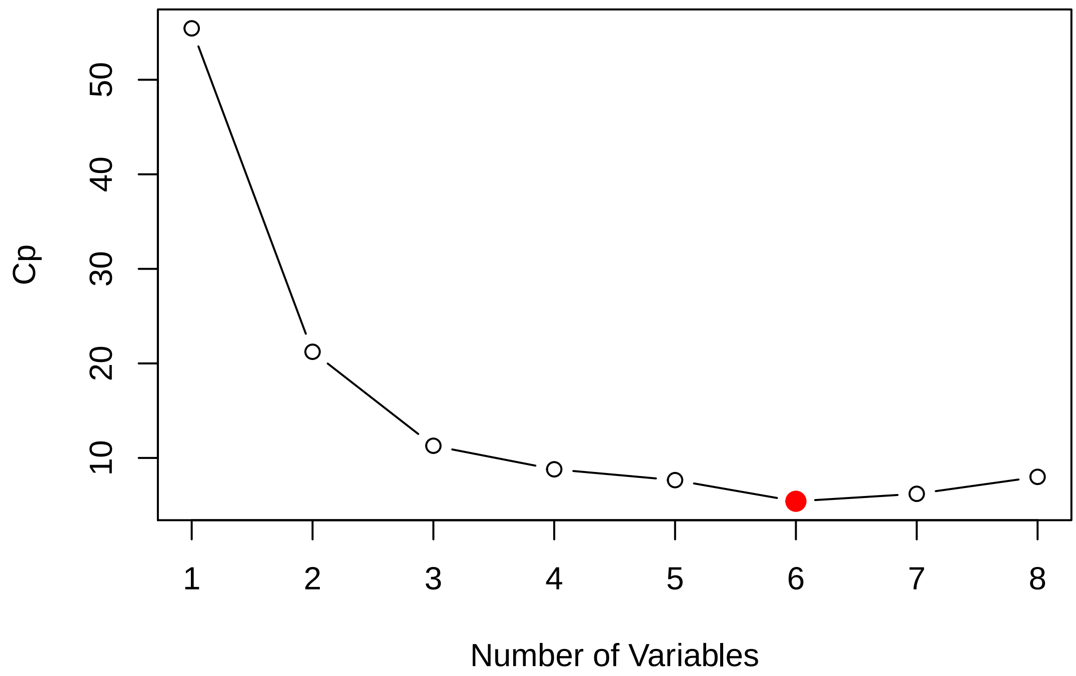
```r
points(7, reg.summary$adjr2[7], col = "blue", cex = 2, pch = 20)
```




```
plot(reg.summary$cp, xlab = "Number of Variables", ylab = "Cp", type = "b")
which.min(reg.summary$cp)
```

```
## [1] 6
```

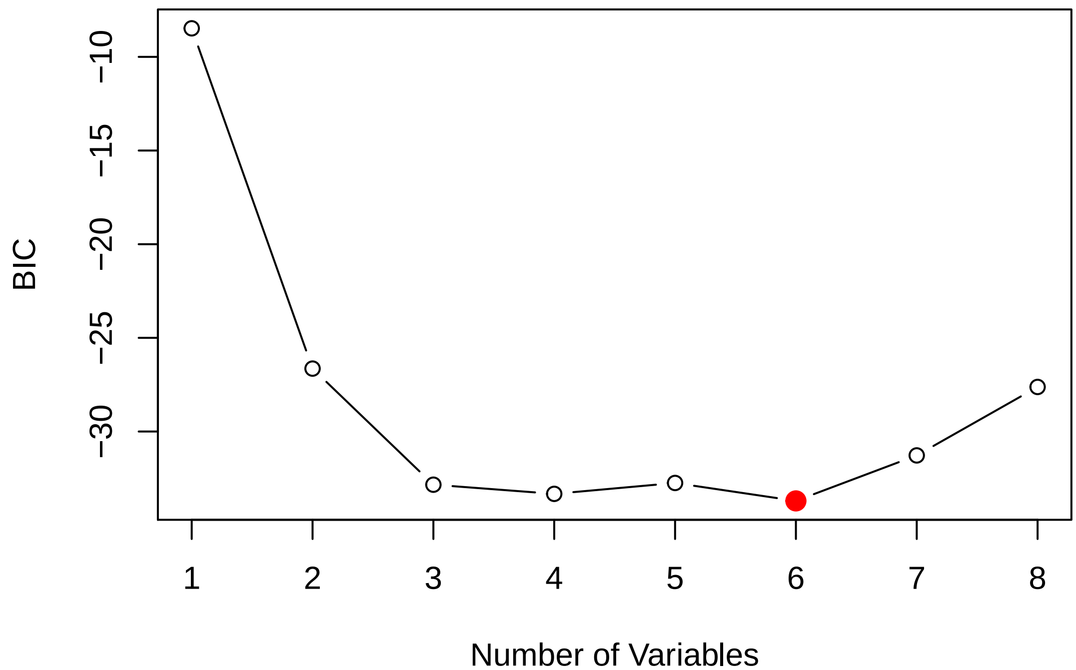
```
points(6, reg.summary$cp[6], col = "red", cex = 2, pch = 20)
```




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

```
## [1] 6
```

```
plot(reg.summary$bic, xlab = "Number of Variables", ylab = "BIC", type = 'b')
points(6, reg.summary$bic[6], col = "red", cex = 2, pch = 20)
```



Looking at the minima and maxima of these statistics, it seems that the best variable model will have 6 or 7 variables. This will help us in the future when we finally set parameters on our regression modeling methods.

2.3.3 Forward Stepwise Selection

Here we can compare our BSS procedure to the FSS method and find any discrepancies.

```
regfit.fwd <- regsubsets(elongation ~ ., data = MyData, nvmax = 8, method = "forward")
```

```
## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax,
## force.in = force.in, : 1 linear dependencies found
```

```
summary(regfit.fwd)
```

```
## Subset selection object
## Call: regsubsets.formula(elongation ~ ., data = MyData, nvmax = 8,
##     method = "forward")
## 9 Variables  (and intercept)
##                           Forced in Forced out
## layer_height                  FALSE      FALSE
## wall_thickness                FALSE      FALSE
## infill_density                FALSE      FALSE
## infill_patternhoneycomb       FALSE      FALSE
## nozzle_temperature            FALSE      FALSE
## bed_temperature               FALSE      FALSE
## print_speed                   FALSE      FALSE
## materialpla                   FALSE      FALSE
## fan_speed                     FALSE      FALSE
```

```
## 1 subsets of each size up to 8
## Selection Algorithm: forward
##          layer_height wall_thickness infill_density
## 1  ( 1 ) " "          " "            " "
## 2  ( 1 ) "*"          " "            " "
## 3  ( 1 ) "*"          " "            "*"
## 4  ( 1 ) "*"          " "            "*"
## 5  ( 1 ) "*"          "*"            "*"
## 6  ( 1 ) "*"          "*"            "*"
## 7  ( 1 ) "*"          "*"            "*"
## 8  ( 1 ) "*"          "*"            "*"
##          infill_patternhoneycomb nozzle_temperature bed_temperature
## 1  ( 1 ) " "                     "*"                " "
## 2  ( 1 ) " "                     "*"                " "
## 3  ( 1 ) " "                     "*"                " "
## 4  ( 1 ) " "                     "*"                " "
## 5  ( 1 ) " "                     "*"                " "
## 6  ( 1 ) " "                     "*"                "*"
## 7  ( 1 ) " "                     "*"                "*"
## 8  ( 1 ) "*"                     "*"                "*"
##          print_speed materialpla fan_speed
## 1  ( 1 ) " "         " "         " "
## 2  ( 1 ) " "         " "         " "
## 3  ( 1 ) " "         " "         " "
## 4  ( 1 ) "*"         " "         " "
## 5  ( 1 ) "*"         " "         " "
## 6  ( 1 ) "*"         " "         " "
## 7  ( 1 ) "*"         "*"         " "
## 8  ( 1 ) "*"         "*"         " "
```

```r
coef(regfit.fwd, 8)
```

```
##             (Intercept)            layer_height          wall_thickness
##            18.435640218             6.382036581             0.030827163
##          infill_density infill_patternhoneycomb      nozzle_temperature
##             0.009890413            -0.062281806            -0.109642480
##         bed_temperature             print_speed             materialpla
##             0.104223026            -0.003374979            -1.783729697
```

FSS gives the same top three input factors and assigns them the same order of importance as BSS, though the importance of the other factors is different.

2.3.3.1 Evaluating Tensile Strength

Now that I have explained my methods, I can also apply them to predicting on a second output, tensile strength.

```r
MyData <- read.csv(file = "H:/Git/19s-dsci353-353m-453-aqn5/1-assignments/SemProj-453/3dprinter/data.csv
#Discard other two output parameters
MyData$elongation <- NULL
MyData$roughness <- NULL
#BSS method
regfit.full <- regsubsets(tension_strenght ~ ., data = MyData, nvmax = 8)
```

```
## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax,
## force.in = force.in, : 1 linear dependencies found
```

```r
reg.summary <- summary(regfit.full)
reg.summary
```

```
## Subset selection object
## Call: regsubsets.formula(tension_strenght ~ ., data = MyData, nvmax = 8)
## 9 Variables  (and intercept)
##                          Forced in Forced out
## layer_height                 FALSE      FALSE
## wall_thickness               FALSE      FALSE
## infill_density               FALSE      FALSE
## infill_patternhoneycomb      FALSE      FALSE
## nozzle_temperature           FALSE      FALSE
## bed_temperature              FALSE      FALSE
## print_speed                  FALSE      FALSE
## materialpla                  FALSE      FALSE
## fan_speed                    FALSE      FALSE
## 1 subsets of each size up to 8
## Selection Algorithm: exhaustive
##          layer_height wall_thickness infill_density
## 1  ( 1 ) " "          " "            " "           
## 2  ( 1 ) " "          " "            "*"           
## 3  ( 1 ) "*"          " "            "*"           
## 4  ( 1 ) "*"          "*"            "*"           
## 5  ( 1 ) "*"          "*"            "*"           
## 6  ( 1 ) "*"          "*"            "*"           
## 7  ( 1 ) "*"          "*"            "*"           
## 8  ( 1 ) "*"          "*"            "*"           
##          infill_patternhoneycomb nozzle_temperature bed_temperature
## 1  ( 1 ) " "                     "*"                " "            
## 2  ( 1 ) " "                     "*"                " "            
## 3  ( 1 ) " "                     "*"                " "            
## 4  ( 1 ) " "                     "*"                " "            
## 5  ( 1 ) "*"                     "*"                " "            
## 6  ( 1 ) " "                     "*"                "*"            
## 7  ( 1 ) "*"                     "*"                "*"            
## 8  ( 1 ) "*"                     "*"                "*"            
##          print_speed materialpla fan_speed
## 1  ( 1 ) " "         " "         " "      
## 2  ( 1 ) " "         " "         " "      
## 3  ( 1 ) " "         " "         " "      
## 4  ( 1 ) " "         " "         " "      
## 5  ( 1 ) " "         " "         " "      
## 6  ( 1 ) " "         "*"         " "      
## 7  ( 1 ) " "         "*"         " "      
## 8  ( 1 ) "*"         "*"         " "      
```

```r
reg.summary$rsq
```

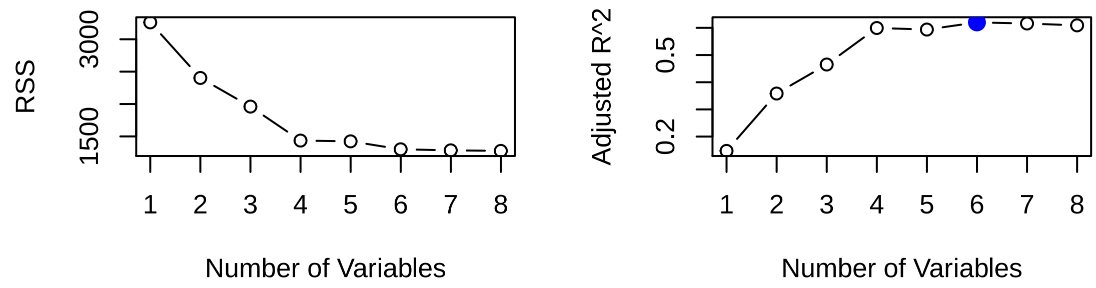
```
## [1] 0.1647610 0.3845934 0.4978543 0.6321826 0.6352339 0.6666586 0.6708548
## [8] 0.6730003
```

```r
#Plot R^2 maximum
par(mfrow = c(2,2))
plot(reg.summary$rss, xlab = "Number of Variables", ylab = "RSS", type = "b")
plot(reg.summary$adjr2, xlab = "Number of Variables", ylab = "Adjusted R^2", type = "b")
```

```r
which.max(reg.summary$adjr2)
```

```
## [1] 6
```

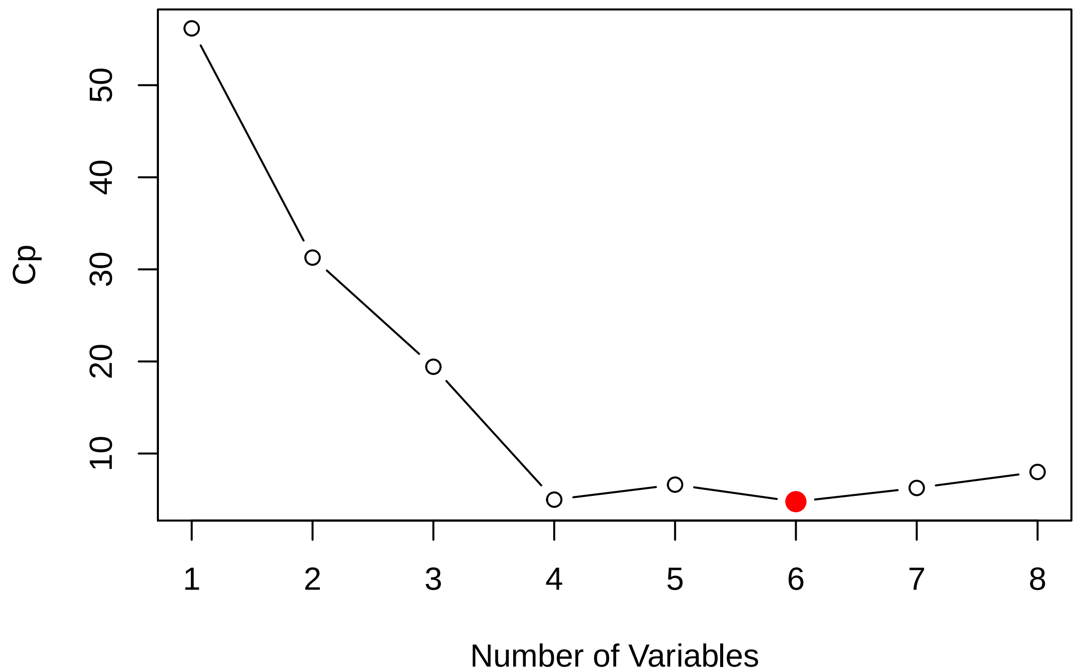
```r
points(6, reg.summary$adjr2[6], col = "blue", cex = 2, pch = 20)
```




```r
plot(reg.summary$cp, xlab = "Number of Variables", ylab = "Cp", type = "b")
which.min(reg.summary$cp)
```

```
## [1] 6
```

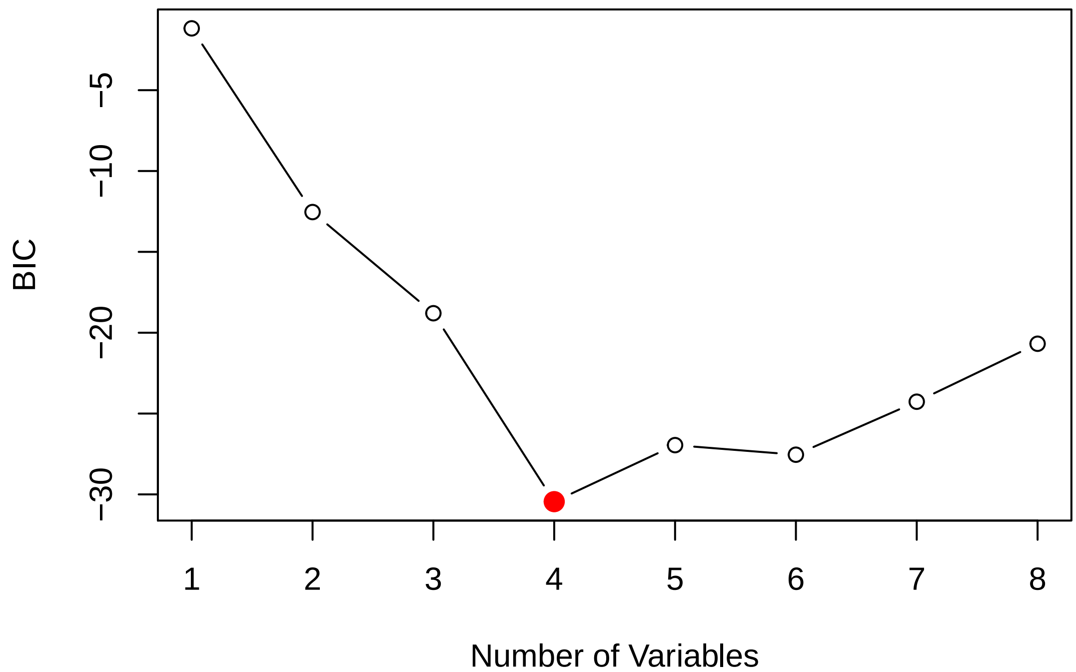
```r
points(6, reg.summary$cp[6], col = "red", cex = 2, pch = 20)
```




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

```
## [1] 4
```

```r
plot(reg.summary$bic, xlab = "Number of Variables", ylab = "BIC", type = 'b')
points(4, reg.summary$bic[4], col = "red", cex = 2, pch = 20)
```



For tension strength, we see that the optimal number of variables for modeling would also be at most 6, if not fewer.

These selection methods were also applied to the roughness variable and suggest using a model of 4 - 6 variables.

To summarize, the most important predictor variables for each response were as follows:

- Nozzle temperature, layer height, and infill density most important factors for elongation.
- Nozzle temperature and infill density for tensile strength
- Nozzle temperature and layer height for roughness

2.3.4 Tree Methods

Tree methods were examined for all three response variables, but to show the methods I will only show code for the modeling of tensile strength.

First separate training and test data.

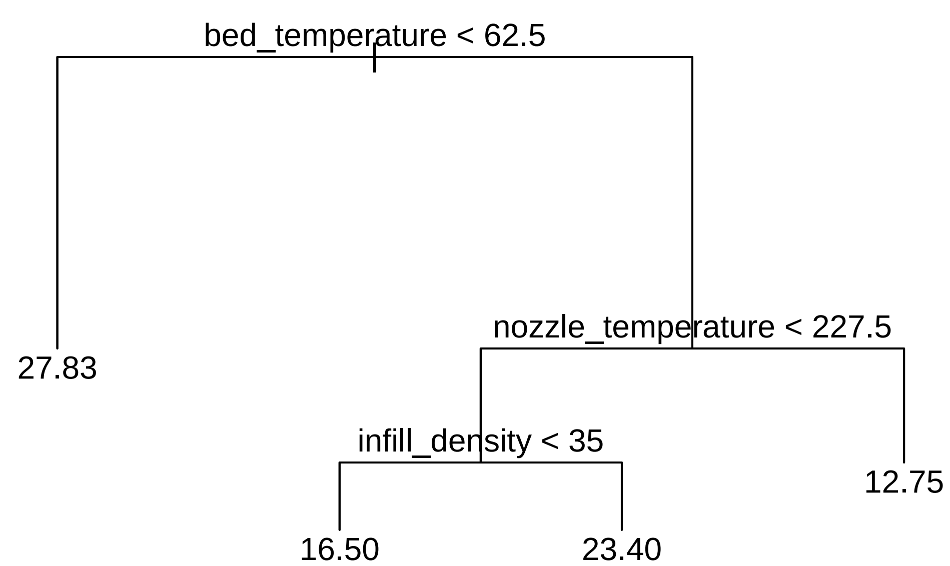
```
set.seed(5)
MyData$elongation <- NULL
MyData$roughness <- NULL
train <-  sample(1:nrow(MyData), nrow(MyData)/2) #create training dataset, splitting data in half for t
tree.MyData <- tree(tension_strenght ~ ., data = MyData, subset = train)
summary(tree.MyData)
```

```
##
## Regression tree:
```

```
## tree(formula = tension_strenght ~ ., data = MyData, subset = train)
## Variables actually used in tree construction:
## [1] "bed_temperature"    "nozzle_temperature" "infill_density"
## Number of terminal nodes:  4
## Residual mean deviance:  38.91 = 817 / 21
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -11.8300  -3.7500   0.1667   0.0000   3.6000   9.5000
```

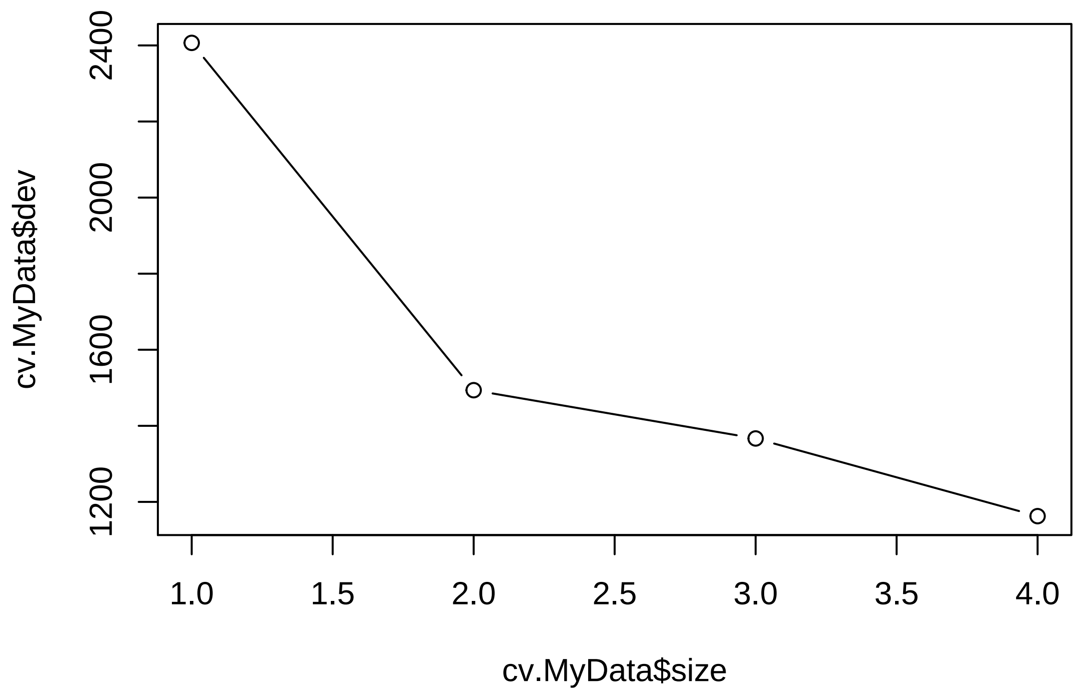
We are modeling against sample tensile strength. It seems that only three variables have been used to construct the tree. We now plot the tree.

```r
plot(tree.MyData)
text(tree.MyData, pretty = 0)
```



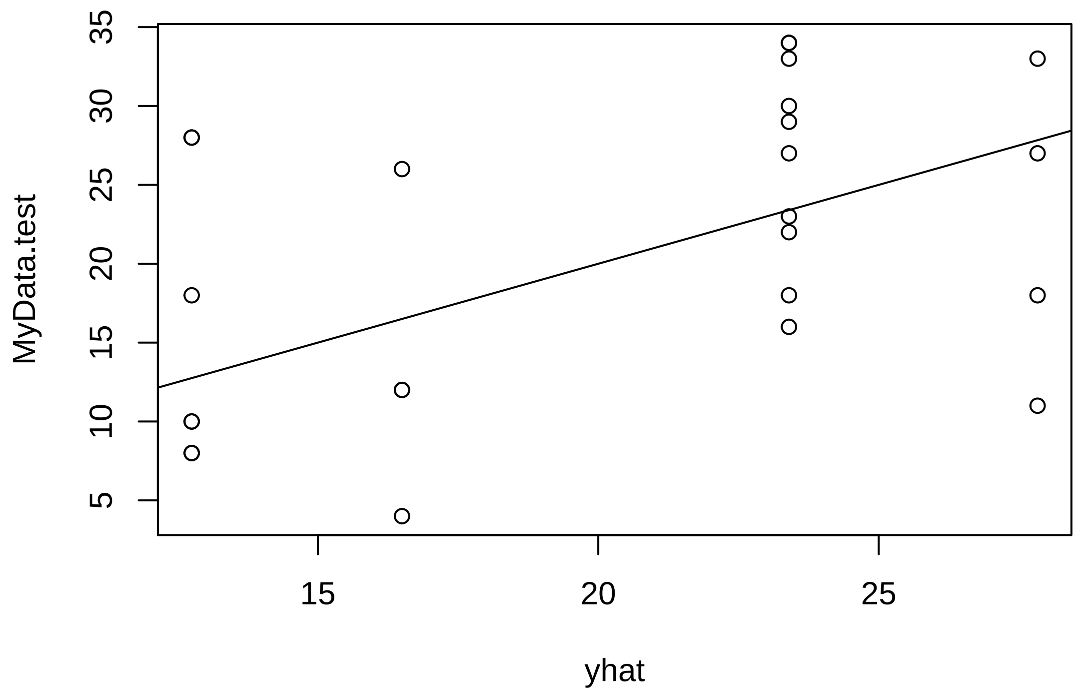
The tree indicates that samples with lower bed temperature, nozzle temperature and low infill density have a strength of approximately 16.5 MPA. Now to cross-validate and check if pruning will improve performance.

```r
cv.MyData <- cv.tree(tree.MyData) #Cross-validate our tree
plot(cv.MyData$size, cv.MyData$dev, type = 'b')
```



It seems that three variables gives the least deviation in modeling, so no pruning is necessary. Finally, I make predictions on the test set.

```
yhat <- predict(tree.MyData, newdata = MyData[-train,])
MyData.test <- MyData[-train, "tension_strenght"]
plot(yhat, MyData.test)
abline(0,1)
```




```
mean((yhat - MyData.test)^2)
```

```
## [1] 69.52728
```

The test set MSE for this regression tree is 69.5, so that the square root of this MSE is about 8.3. These test predictions are then within 8.3 MPA of the tensile strength.

For elongation, the test set MSE for this regression tree is 0.604, so that the square root of this MSE is about 0.77. These test predictions are then within 0.77% of the elongation. This is a fairly large error, close to the minimum value in the range of elongation data.

For roughness, the test set MSE for this regression tree is 3609, so that the square root of this MSE is about 60.1. These test predictions are then within 60 microns of the roughness.

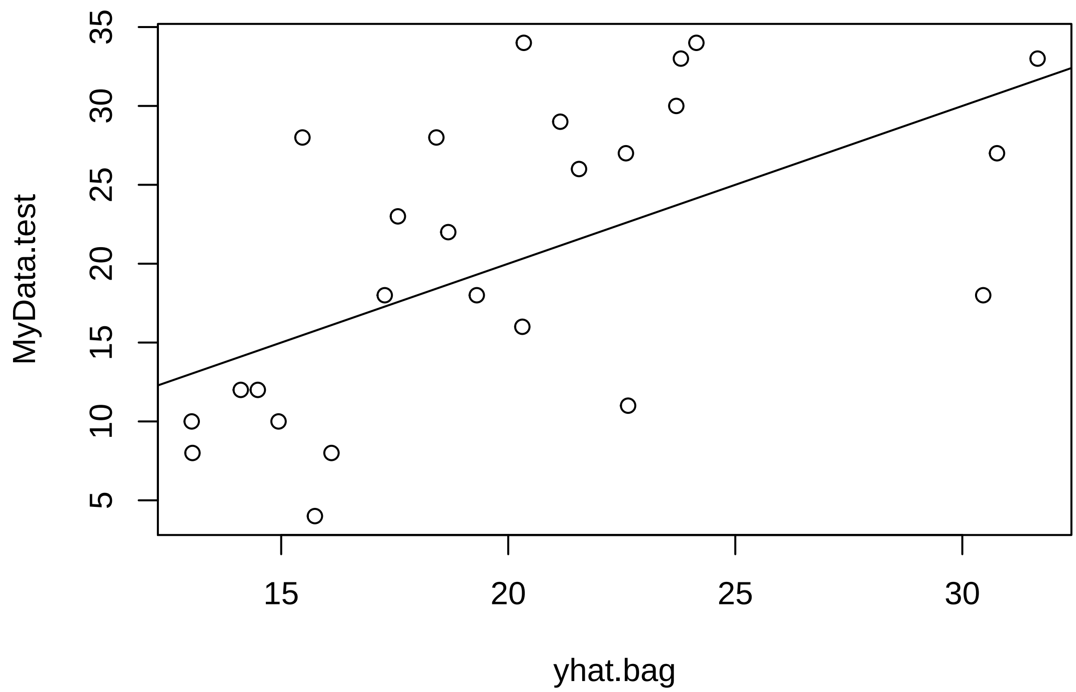
Now I use random forest methods to see if they improve on my first regression trees.

```
library(randomForest)
set.seed(1)
rf.MyData <- randomForest(tension_strenght ~ .,data = MyData, subset = train, mtry = 9, importance = TRU
rf.MyData
```

```
##
## Call:
##  randomForest(formula = tension_strenght ~ ., data = MyData, mtry = 9,      importance = TRUE, subse
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 9
##
##           Mean of squared residuals: 38.89705
```

```
##                     % Var explained: 43.73
```

```
yhat.bag <- predict(rf.MyData, newdata = MyData[-train,])
plot(yhat.bag, MyData.test)
abline(0,1)
```




```
mean((yhat.bag - MyData.test)^2)
```

```
## [1] 55.96079
```

```
importance(rf.MyData)
```

```
##                       %IncMSE IncNodePurity
## layer_height       -0.3765176     129.07264
## wall_thickness      6.0506506     178.09284
## infill_density     12.6201133     381.00587
## infill_pattern     -2.3033151      22.05321
## nozzle_temperature 14.1416762     325.68397
## bed_temperature     6.7010472     200.68297
## print_speed         4.1480449      65.74663
## material            3.0475128      29.63385
## fan_speed           6.1357331     193.01315
```

Error is about 56, a reduction to the optimal regression tree. Across all trees, the most important variables were extruding nozzle temperature and infill density. Only 43% of variance was successfully accounted for in this model

For elongation, error is about 0.38, a reduction to the optimal regression tree. Across all trees, the most important variables were extruding nozzle temperature, infill density, and layer height, respectively. 70% of variance was successfully accounted for in this model.

For roughness, error is about 2630, a reduction to the optimal regression tree. Across all trees, the most important variables were layer height and nozzle temperature. Only 55% of variance was successfully accounted for in this model.

2.3.5 Multivariate Regression

Now let us examine the relationship between the important variables identified in previous EDA and modeling methods, setting response against multiple strong predictors linearly and evaluating the model.

```r
lm.fit <- lm(tension_strenght ~ nozzle_temperature + infill_density, data = MyData, subset = train)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = tension_strenght ~ nozzle_temperature + infill_density,
##     data = MyData, subset = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.876  -2.797  -2.044   5.747  10.065
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        91.98174   20.67074   4.450 0.000201 ***
## nozzle_temperature -0.37127    0.09888  -3.755 0.001095 **
## infill_density      0.20789    0.06306   3.297 0.003286 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.682 on 22 degrees of freedom
## Multiple R-squared:  0.4315, Adjusted R-squared:  0.3799
## F-statistic:  8.35 on 2 and 22 DF,  p-value: 0.002003
```

Let's compare these values to the response (here, tensile strength) against all predictive input variables.

```r
lm.fit.all <- lm(tension_strenght ~ ., data = MyData, subset = train)
summary(lm.fit.all)
```

```
##
## Call:
## lm(formula = tension_strenght ~ ., data = MyData, subset = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.9536 -3.0738  0.6258  2.6983  8.4491
##
## Coefficients: (1 not defined because of singularities)
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             330.96650   83.61601   3.958  0.00113 **
## layer_height             38.36576   19.97467   1.921  0.07277 .
## wall_thickness            0.73470    0.47684   1.541  0.14292
## infill_density            0.22336    0.05581   4.002  0.00103 **
## infill_patternhoneycomb   1.71020    2.64663   0.646  0.52732
## nozzle_temperature       -2.00944    0.56628  -3.548  0.00267 **
## bed_temperature           1.97911    0.72998   2.711  0.01541 *
```

```
## print_speed              -0.05328    0.04214  -1.264  0.22417
## materialpla             -41.07909   13.23448  -3.104  0.00682 **
## fan_speed                      NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.371 on 16 degrees of freedom
## Multiple R-squared:  0.7329, Adjusted R-squared:  0.5994
## F-statistic: 5.489 on 8 and 16 DF,  p-value: 0.001901
```

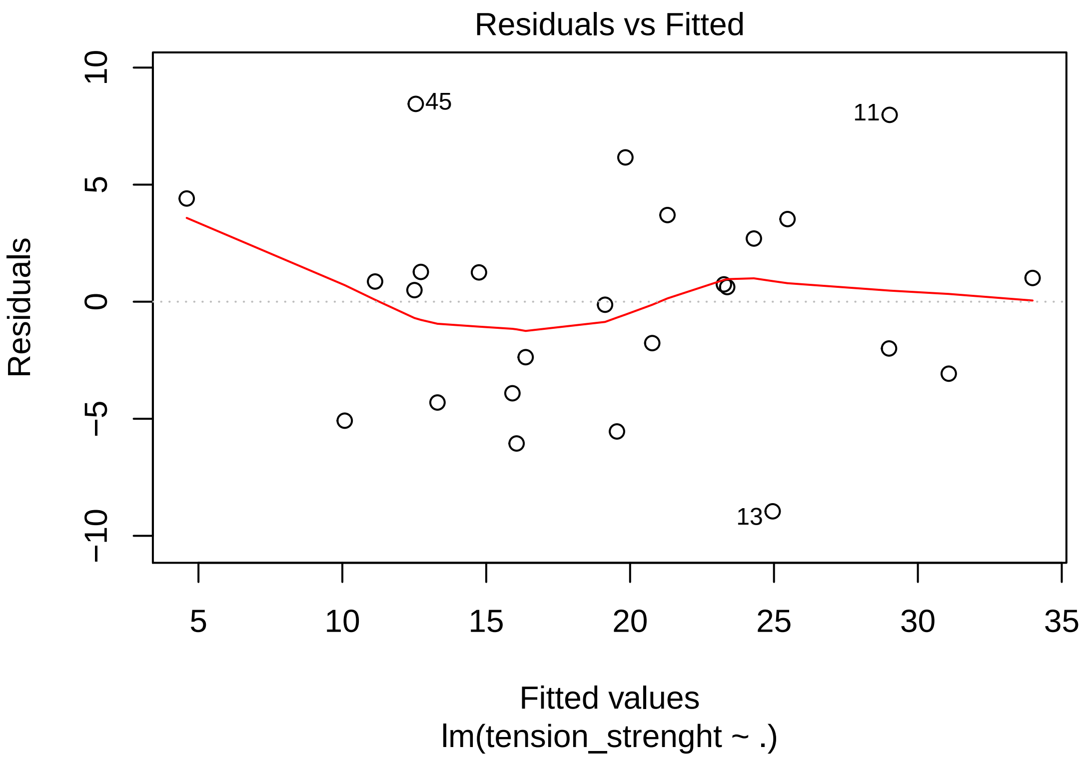
```r
# Look at interaction term for two most important predictors
summary(lm(tension_strenght ~ infill_density*nozzle_temperature, data = MyData, subset = train))
```

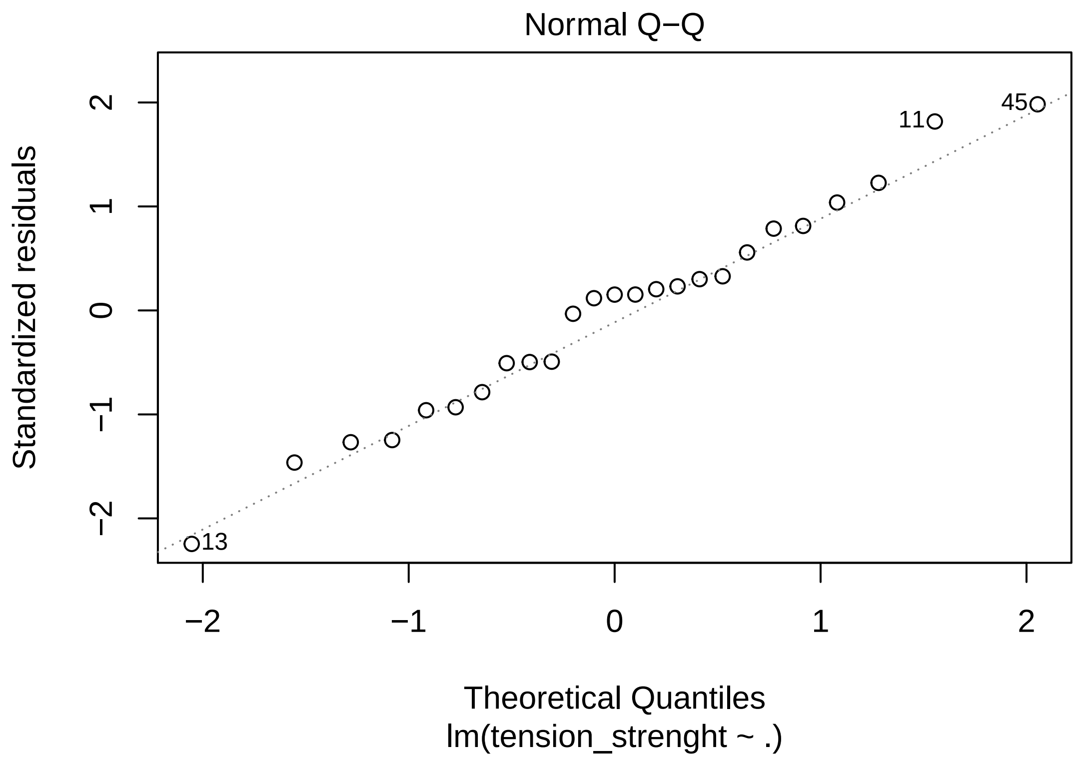
```
##
## Call:
## lm(formula = tension_strenght ~ infill_density * nozzle_temperature,
##     data = MyData, subset = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.4224  -3.4300   0.4468   4.7524   8.7533
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      -15.844862  43.084254  -0.368  0.71673
## infill_density                     2.226239   0.733711   3.034  0.00631
## nozzle_temperature                 0.114350   0.196224   0.583  0.56627
## infill_density:nozzle_temperature  -0.008930   0.003237  -2.759  0.01177
##
## (Intercept)
## infill_density                    **
## nozzle_temperature
## infill_density:nozzle_temperature *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.86 on 21 degrees of freedom
## Multiple R-squared:  0.5827, Adjusted R-squared:  0.5231
## F-statistic: 9.776 on 3 and 21 DF,  p-value: 0.0003076
```

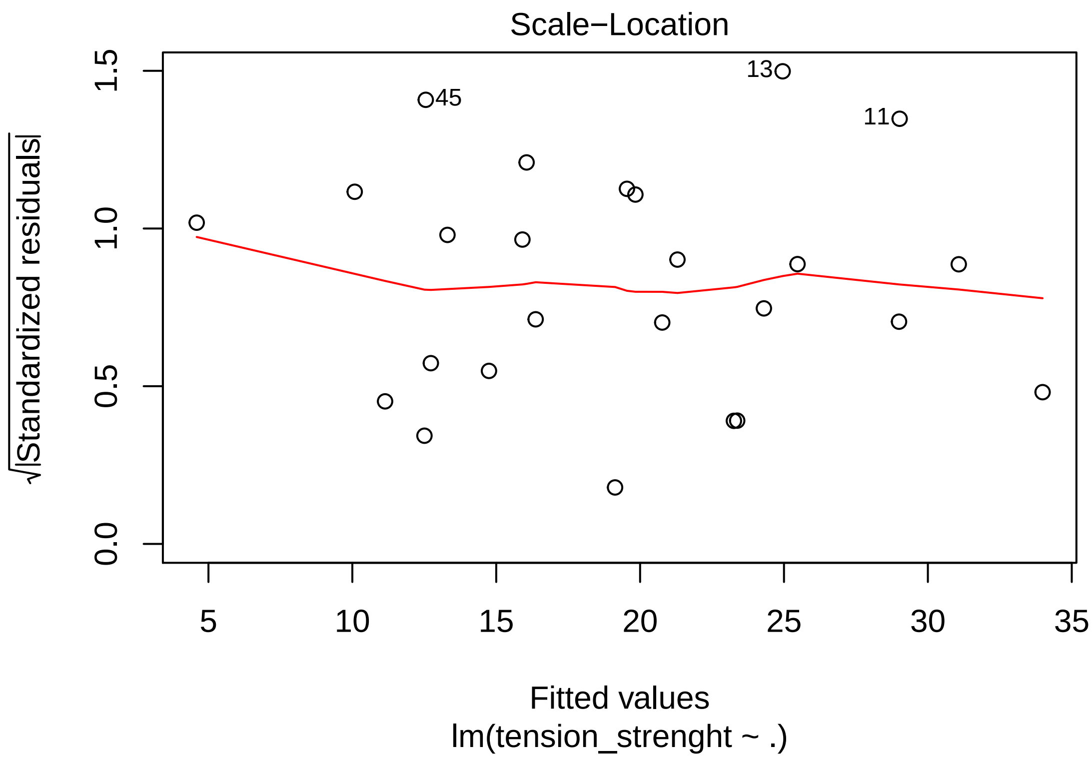
While the p-value < 0.05 indicates the results are statistically significant, the adjusted R^2 values of the first model indicates that less than 40% of the data variability is accounted for. When considered against all predictor variables, about 60% is accounted for. These linear models do not work well for tensile strength.

The models better fit for elongation, with 60 - 65% of the data explained by the model. Roughness shows the best fit with multi-linear modeling, with adjusted R^2 accounting for 85% of the model variance. We also see a reduction in error of about 15 - 20% from previous models.

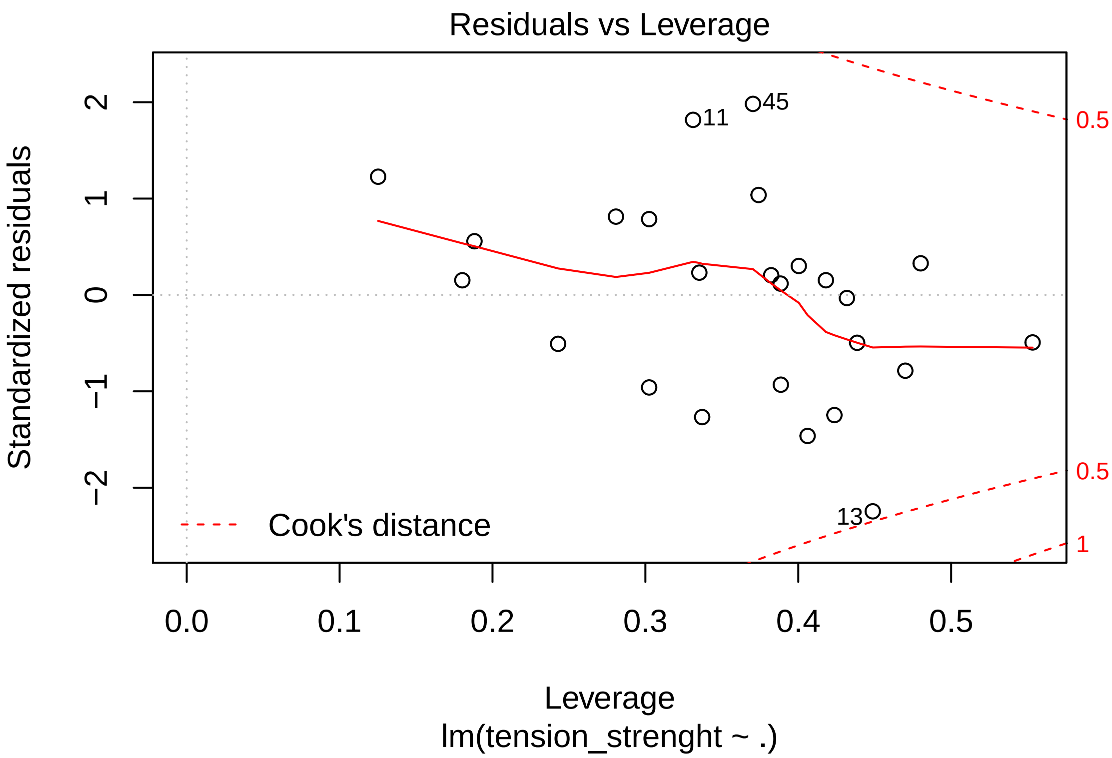
```r
#Example of lm model results for tensile strength
plot(lm.fit.all)
```


Residuals vs Fitted
Residuals
Fitted values
lm(tension_strenght ~ .)
10
5
0
−5
−10
5
10
15
20
25
30
35
45
11
13


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(tension_strenght ~ .)
13
11
45
−2
−1
0
1
2



Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
45
13
11
5
10
15
20
25
30
35
Fitted values
lm(tension_strenght ~ .)



2.3.6 Results Discussion

From knowledge concerning the 3-D printing process, the predictor variables which were chosen as most significant by these models make sense. No matter material type, the process parameter most focused on is some sort of heat energy input (3). In the case of an Ultimaker, this translates to nozzle temperature. Layer height should correspond with roughness, as the fineness of each layer in a layer-by-layer building process determines the quality of the final product finished surface. Infill density would be expected to correspond with mechanical properties and not external surface roughness, which seems to be the case in these models. The density of a testing specimen will of course determine how well it holds intact and resists tensile loading.

2.4 SemProj Part d) Present your final models and learnings - Conclusions

In beginning this project, I was unsure about its viability in terms of statistical significance, given the relatively small dataset (n = 50, p = 14). This exploration of modeling methods on processing data and performance data has shown that, while the models are not beyond reproach and improvement, several findings can be gleaned from the process.

Ultimately, I do not think any modeling methods accurately model tensile strength. Random forest methods were able to account for 70% of elongation data, and up to 85% for roughness data. Here I show the fit of the roughness multivariate model, as it is the best result of this project:

These results suggest that perhaps only a few dozen more observations would be needed to make the models more reliable for some output variables, rather than thousands of observations commonly seen in data science studies. While mechanical performance is intrinsically more variable due to destructive testing, some parameters such as roughness are easier to model. As additive manufacturing is a relatively new field for processing research and optimization, it is plausible that such an ideal dataset will be produced in the next few years.

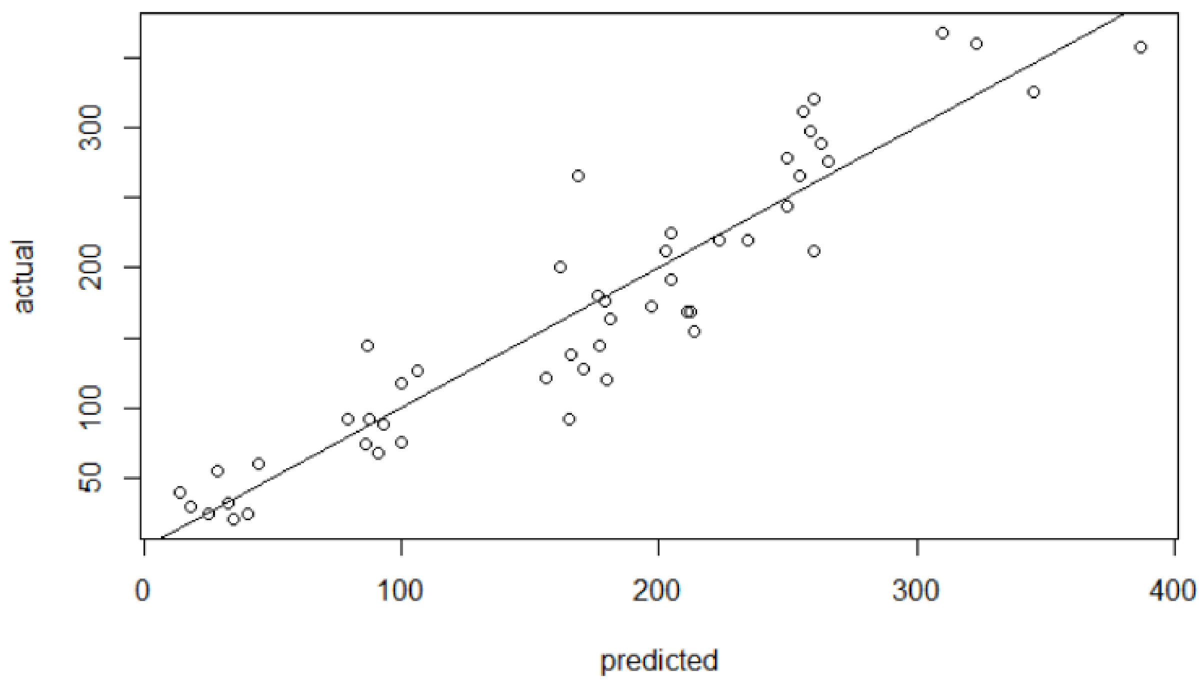


Figure 2: Roughness MultiVariate model

Beyond fit of the models, these statistical methods are also useful for identifying the most influential process input parameters and how they affect performance output parameters. While these relationships typically been determined through physical observation and qualitative inference, we can now confirm these findings with quantitative methods. I have not seen other attempts at modeling additive manufacturing processes currently in the literature, as most systematic examinations only focus on qualitative performance metrics.

3 Acknowledgements

I would like to acknowledge Professor Roger French and Menghong Wang of the Solar Durability and Lifetime Extension Center (SDLE) for providing guidance and advice throughout my first data science project.

4 References

(1) *Sames, W. J., List, F. A., Pannala, S., Dehoff, R. R. & Babu, S. S. The metallurgy and processing science of metal additive manufacturing. Int. Mater. Rev. 61, 315-360 (2016).*

(2) *Okudan, Ahmet. TR/Selcuk University Mechanical Engineering.*

(3) *Sames, W. J., List, F. A., Pannala, S., Dehoff, R. R. & Babu, S. S. The metallurgy and processing science of metal additive manufacturing. Int. Mater. Rev. 61, 315–360 (2016).*

5 Licensing

==