Homework 2

Matthew Perrotta

March 19, 2019

Load Packages

```r
library(tidyverse)
library(ISLR)
library(glmnet)
library(caret)
library(corrplot)
library(plotmo)
library(boot)
library(pls)
library(mgcv)
```

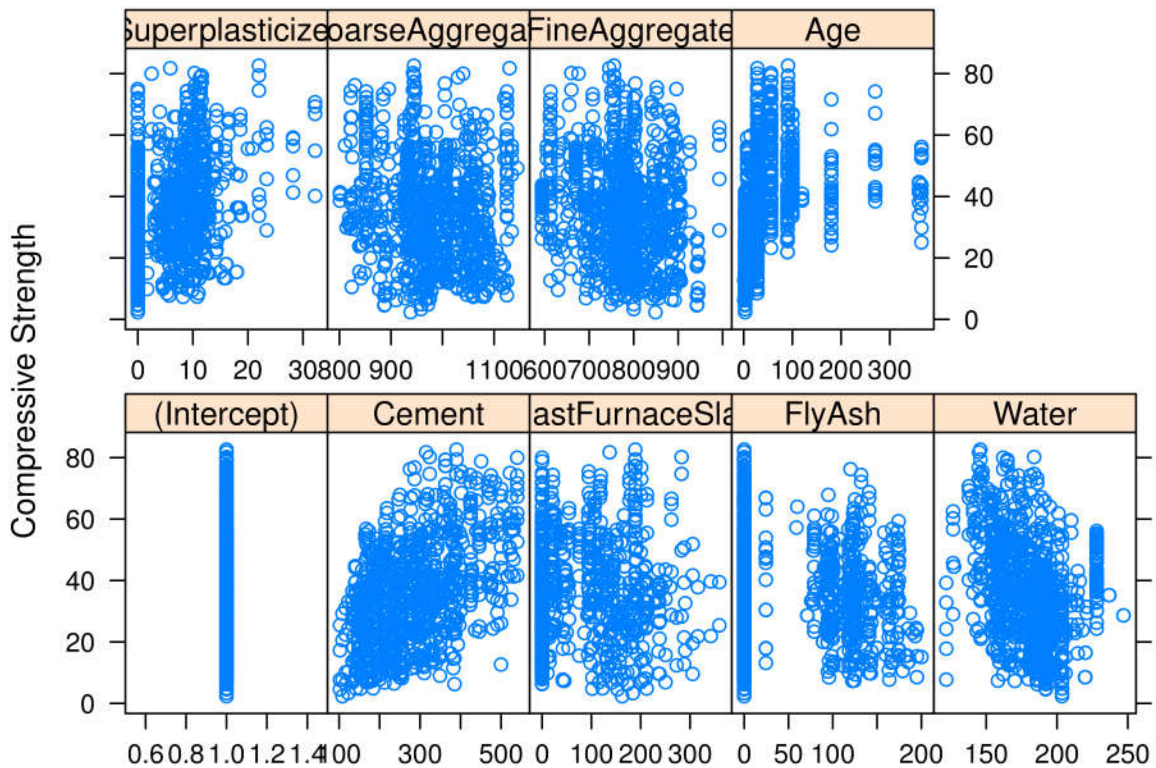
Load Data

```r
concrete = read.csv('./data/concrete.csv')
```

```r
conX = model.matrix(CompressiveStrength~., concrete)

conY = concrete$CompressiveStrength
```

Q1 Plot

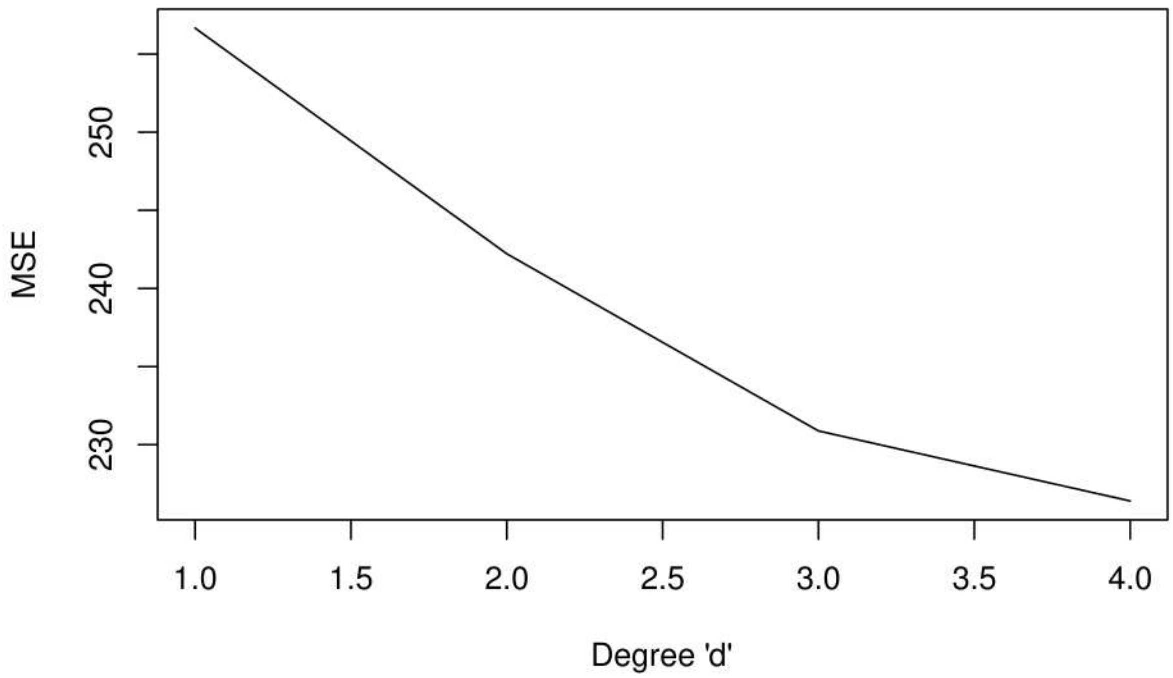
```r
featurePlot(conX, conY, plot = 'scatter', labels = c('','Compressive Strength'), type = c('p'))
```



Q2 Polynomial Regression

Using Cross-Validation to Determine Degree 'd'

```r
set.seed(1)

d = rep(NA, 4)
for (i in 1:4) {
    fit = glm(CompressiveStrength ~ poly(Water, i), data = concrete)
    d[i] = cv.glm(concrete, fit, K = 10)$d[1]
}

plot(1:4, d, xlab = "Degree 'd'", ylab = "MSE", type = "l")
```



By using cross-validation, a degree of 4 is shown to be the best from values ranging from 1 to 4.

ANOVA

```
fit1 = lm(CompressiveStrength ~ Water, data = concrete)
fit2 = lm(CompressiveStrength ~ poly(Water, 2), data = concrete)
fit3 = lm(CompressiveStrength ~ poly(Water, 3), data = concrete)
fit4 = lm(CompressiveStrength ~ poly(Water, 4), data = concrete)

anova(fit1, fit2, fit3, fit4)
```

```
## Analysis of Variance Table
##
## Model 1: CompressiveStrength ~ Water
## Model 2: CompressiveStrength ~ poly(Water, 2)
## Model 3: CompressiveStrength ~ poly(Water, 3)
## Model 4: CompressiveStrength ~ poly(Water, 4)
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1   1028 263085
## 2   1027 247712  1   15372.8 68.140 4.652e-16 ***
## 3   1026 235538  1   12174.0 53.962 4.166e-13 ***
## 4   1025 231246  1    4291.5 19.022 1.423e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

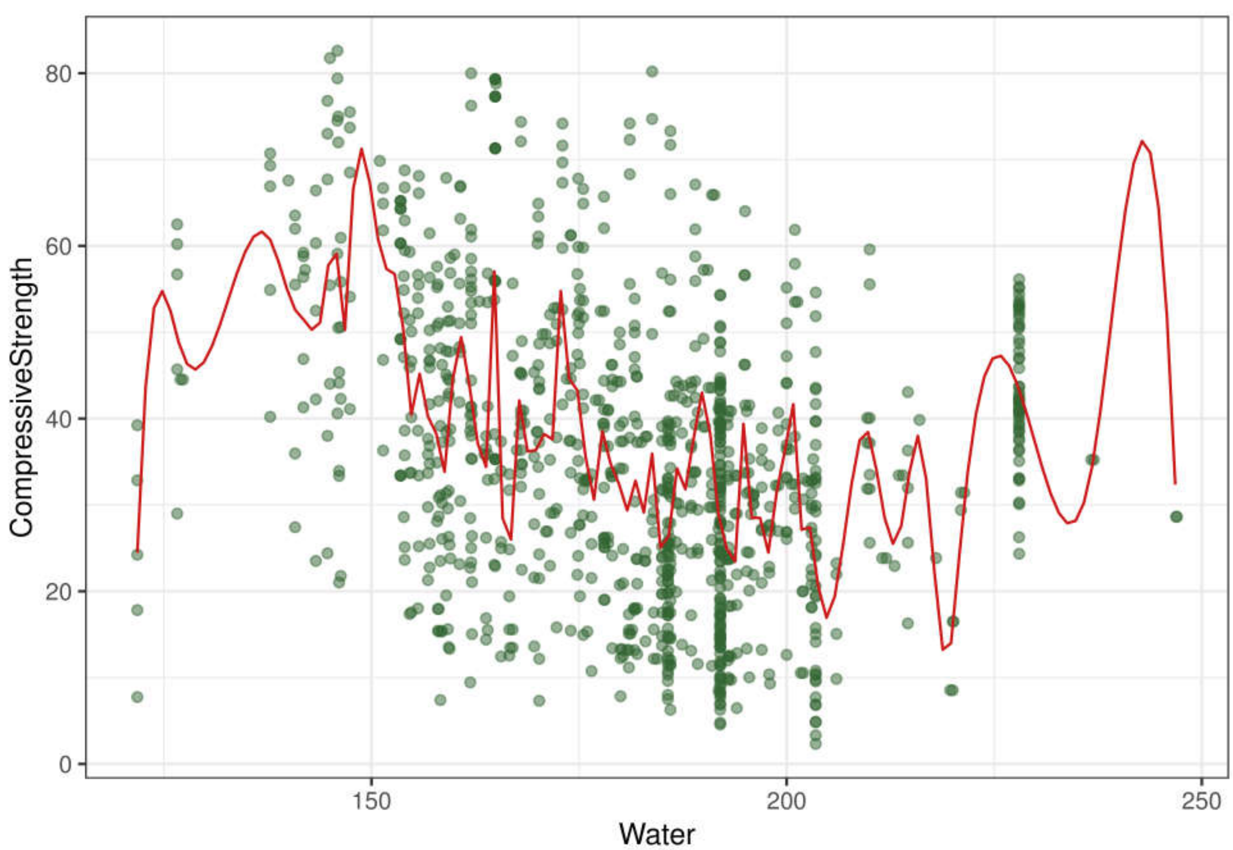
Running an ANOVA on 4 models, models 2 through 4 all have very low p values, but model 2 has the lowest p value.

Q3 Smoothing Spline

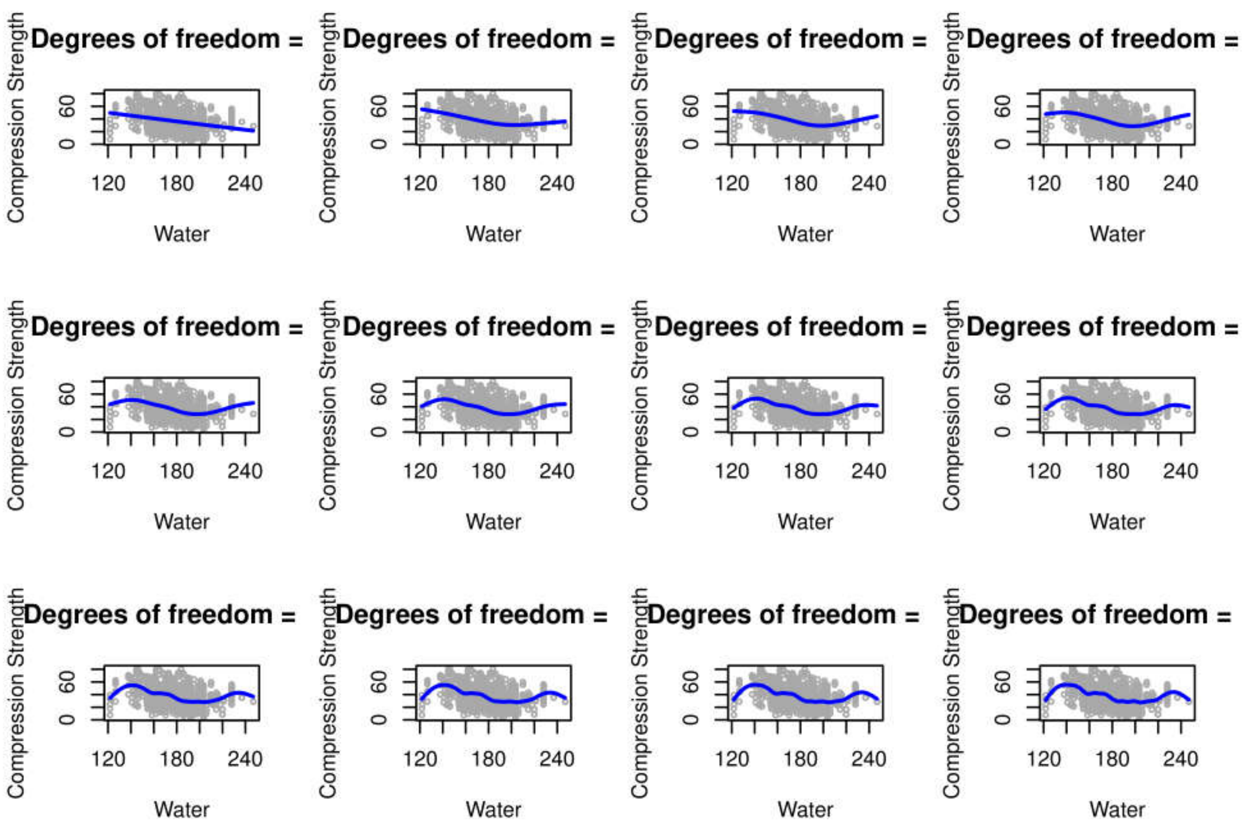
```
fit.ss = smooth.spline(concrete$Water, concrete$CompressiveStrength)
fit.ss$df
```

```
## [1] 68.88205
```

```
waterlims = range(concrete$Water)
water.grid <- seq(from = waterlims[1], to = waterlims[2])

pred.ss = predict(fit.ss,
                  x = water.grid)

pred.ss.df = data.frame(pred = pred.ss$y,
                        water = water.grid)

p = ggplot(data = concrete, aes(x = Water, y = CompressiveStrength)) +
     geom_point(color = rgb(.2, .4, .2, .5))
p +
geom_line(aes(x = water, y = pred), data = pred.ss.df,
          color = rgb(.8, .1, .1, 1)) + theme_bw()
```



```r
par(mfrow = c(3,4))
all.dfs = rep(NA, 12)
for (i in 2:13) {
  fit.ss = smooth.spline(concrete$Water, concrete$CompressiveStrength, df = i)

  pred.ss = predict(fit.ss, x = water.grid)

  plot(concrete$Water, concrete$CompressiveStrength, cex = .5, col = "darkgrey",
       xlab = "Water",
       ylab = "Compression Strength")
  title(paste("Degrees of freedom = ", round(fit.ss$df)),  outer = F)
  lines(water.grid, pred.ss$y, lwd = 2, col = "blue")
}
```



Increasing degrees of freedom increases the flexibility of the curve.

GAM

```r
gam.m1 <- gam(CompressiveStrength ~ Cement + BlastFurnaceSlag + FlyAsh + Water + Superplasticizer + Coa
gam.m2 <- gam(CompressiveStrength ~ Cement + BlastFurnaceSlag + FlyAsh + s(Water) + Superplasticizer +

anova(gam.m1, gam.m2, test = "F")
```

```
## Analysis of Deviance Table
##
## Model 1: CompressiveStrength ~ Cement + BlastFurnaceSlag + FlyAsh + Water +
##     Superplasticizer + CoarseAggregate + FineAggregate + Age
## Model 2: CompressiveStrength ~ Cement + BlastFurnaceSlag + FlyAsh + s(Water) +
##     Superplasticizer + CoarseAggregate + FineAggregate + Age
##   Resid. Df Resid. Dev     Df Deviance      F   Pr(>F)
## 1    1021.0     110413
## 2    1013.4     106140 7.5562   4272.8 5.4038 2.01e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
plot(gam.m2)
```