Homework 2

Xinyi Lin

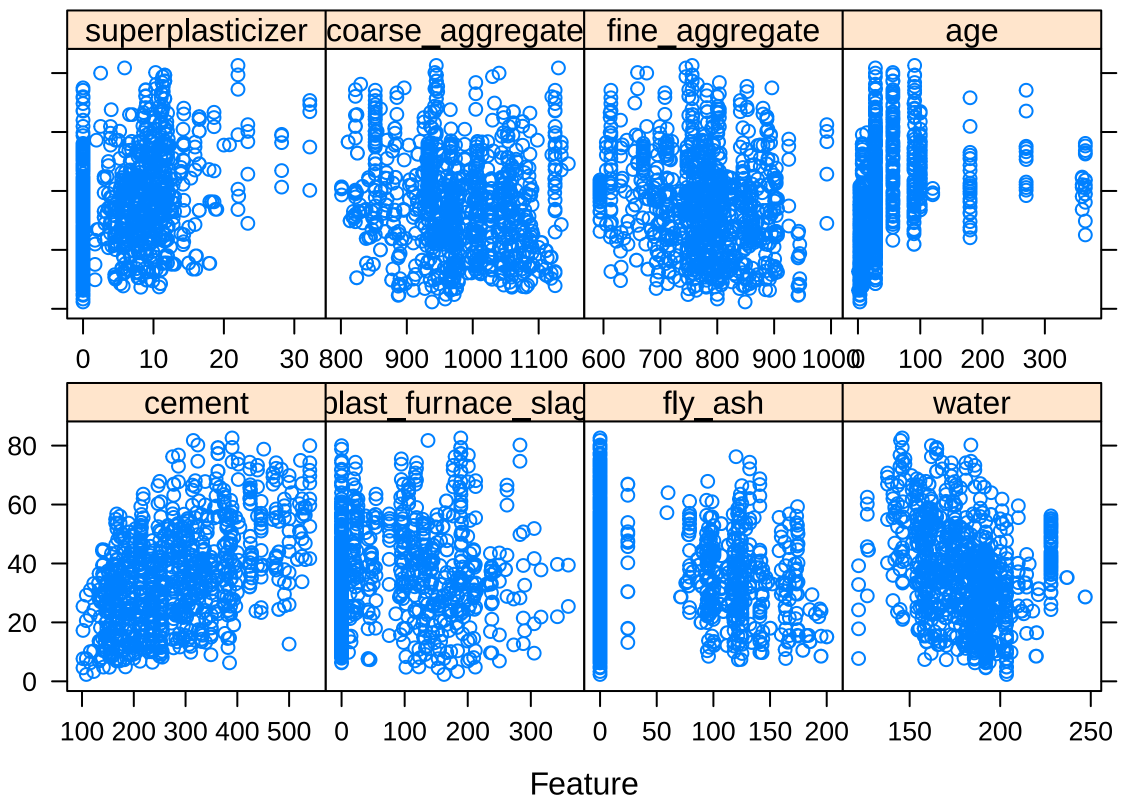
3/19/2019

```r
library(tidyverse)
library(caret)
library(boot)  # for smooth spline
library(ggplot2)
library(mgcv)   # for gam
```

```r
concrete_data = read_csv("./concrete.csv") %>%
  janitor::clean_names()
```

Question 1

```r
x = concrete_data[,1:8]
y = y = as.numeric(unlist(concrete_data[,9]))
featurePlot(x, y, "scatter")
```



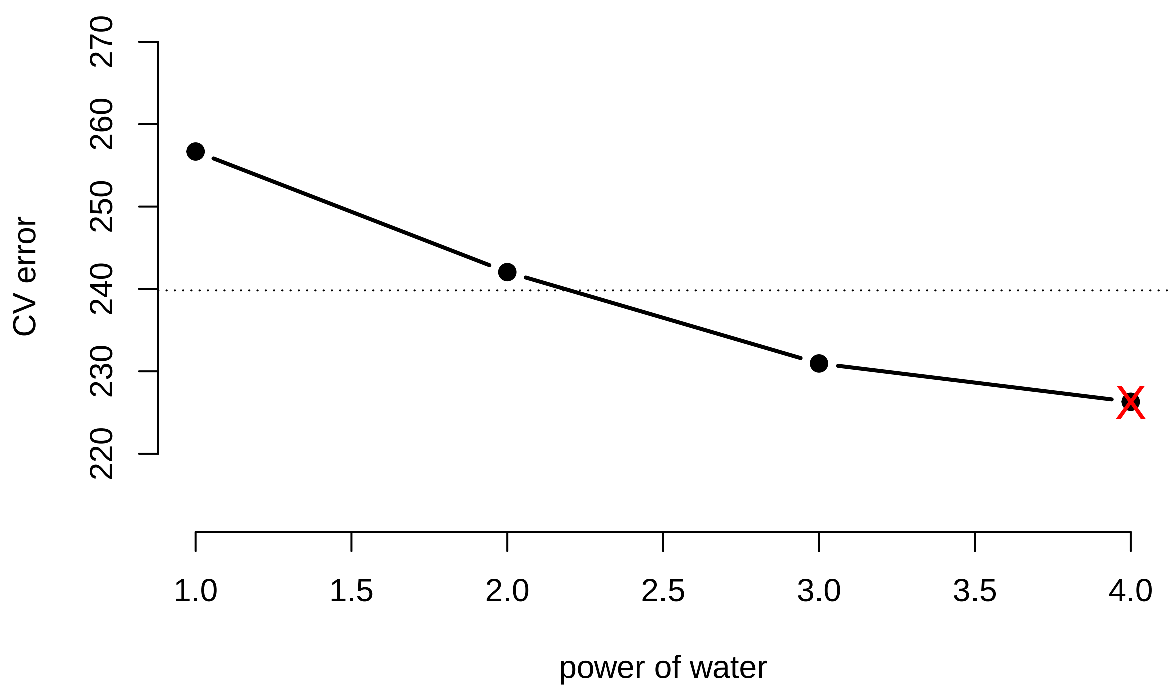
Question 2

Cross validation

```r
set.seed(123)

# container of test errors
cv.MSE <- NA

# loop over powers of water
for (i in 1:4) {
  glm.fit <-  glm(compressive_strength ~ poly(water, i), data = concrete_data)
  # we use cv.glm's cross-validation and keep the vanilla cv test error
  cv.MSE[i] <-  cv.glm(concrete_data, glm.fit, K = 10)$delta[1]
}
# inspect results object
cv.MSE
```

```
## [1] 256.6841 242.0471 230.9552 226.3080
```

```r
# illustrate results with a line plot connecting the cv.error dots
plot( x = 1:4, y = cv.MSE, xlab = "power of water", ylab = "CV error",
      type = "b", pch = 19, lwd = 2, bty = "n",
      ylim = c( min(cv.MSE) - sd(cv.MSE), max(cv.MSE) + sd(cv.MSE) ) )

# horizontal line for 1se to less complexity
abline(h = min(cv.MSE) + sd(cv.MSE) , lty = "dotted")

# where is the minimum
points( x = which.min(cv.MSE), y = min(cv.MSE), col = "red", pch = "X", cex = 1.5 )
```



According to the result, we should choose degree of freedom equals to 4.

ANOVA

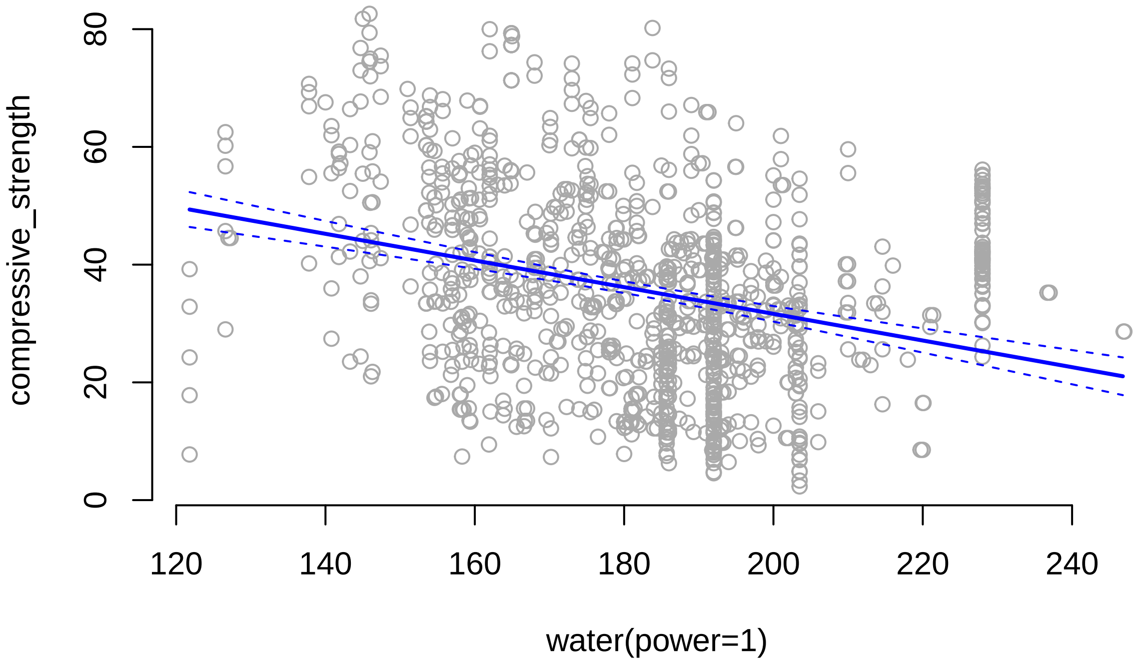
```r
# container for the models we will fit
models <- vector("list", length(cv.MSE))
# fit all 15 models
for( a in 1:length(cv.MSE)){
  models[[a]] <- glm(compressive_strength ~ poly(water, a), data = concrete_data)
}
# f-test
anova(models[[1]], models[[2]], models[[3]], models[[4]], test = "F")
```

```
## Analysis of Deviance Table
##
## Model 1: compressive_strength ~ poly(water, a)
## Model 2: compressive_strength ~ poly(water, a)
## Model 3: compressive_strength ~ poly(water, a)
## Model 4: compressive_strength ~ poly(water, a)
##   Resid. Df Resid. Dev Df Deviance      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
```

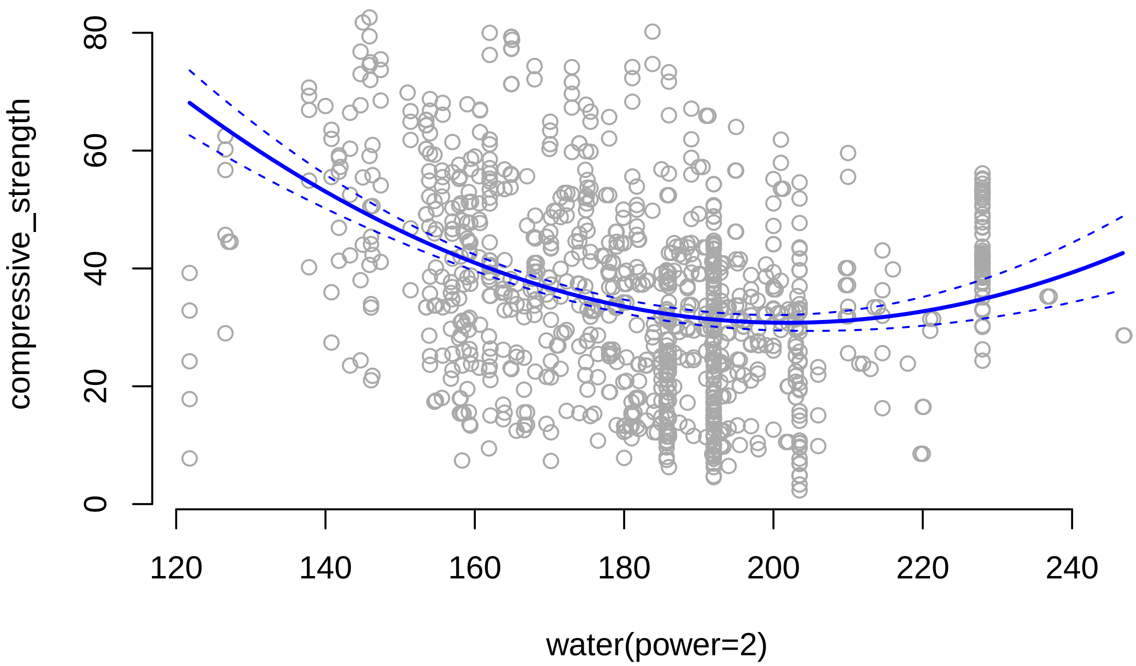
According to the result from F-test, comparing to the model with 3 degrees of freedom, the model with 4 degree of freedom is significant, so we should choose degree equals to 4.

Plots of different polynomial fits

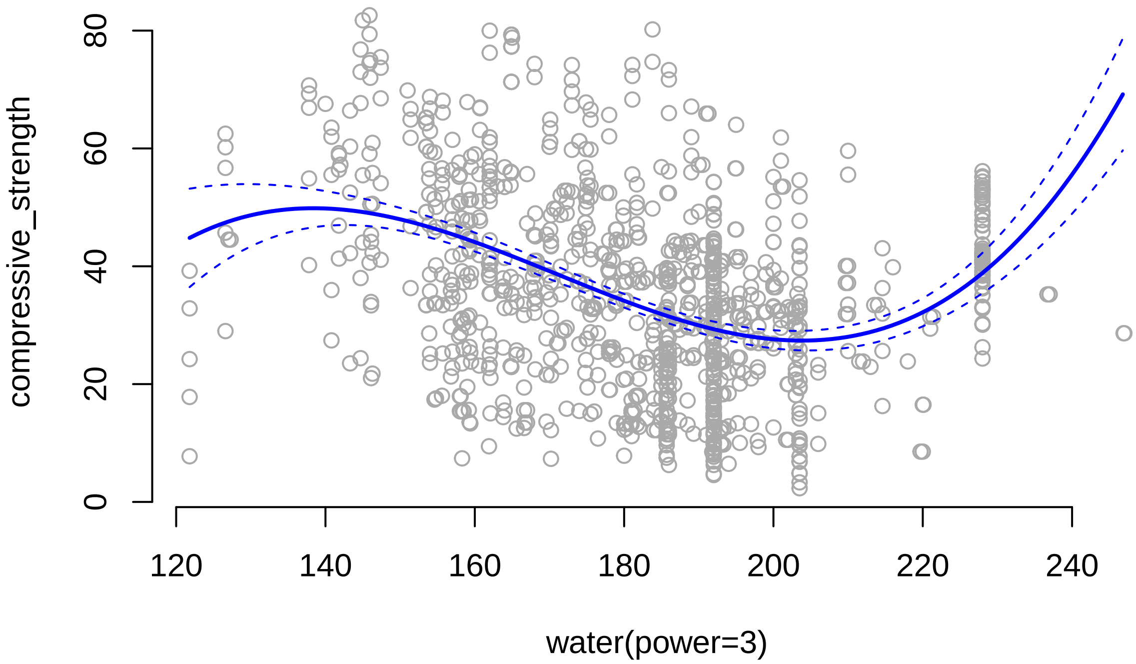
```r
plot(compressive_strength ~ water, data = concrete_data, col = "darkgrey",  bty = "n", xlab="water(powe
waterlims <-  range(concrete_data$water)
water.grid <-  seq(from = waterlims[1], to = waterlims[2])
lm.fit <-  lm(compressive_strength ~ poly(water, 1), data = concrete_data)
lm.pred <-  predict(lm.fit, data.frame(water = water.grid), se = TRUE)
# mean prediction
lines(x = water.grid , y = lm.pred$fit, col = "blue", lwd = 2)
# uncertainty bands
matlines( x = water.grid, y = cbind( lm.pred$fit + 2*lm.pred$se.fit, lm.pred$fit - 2*lm.pred$se.fit),
          lty = "dashed", col = "blue")
```



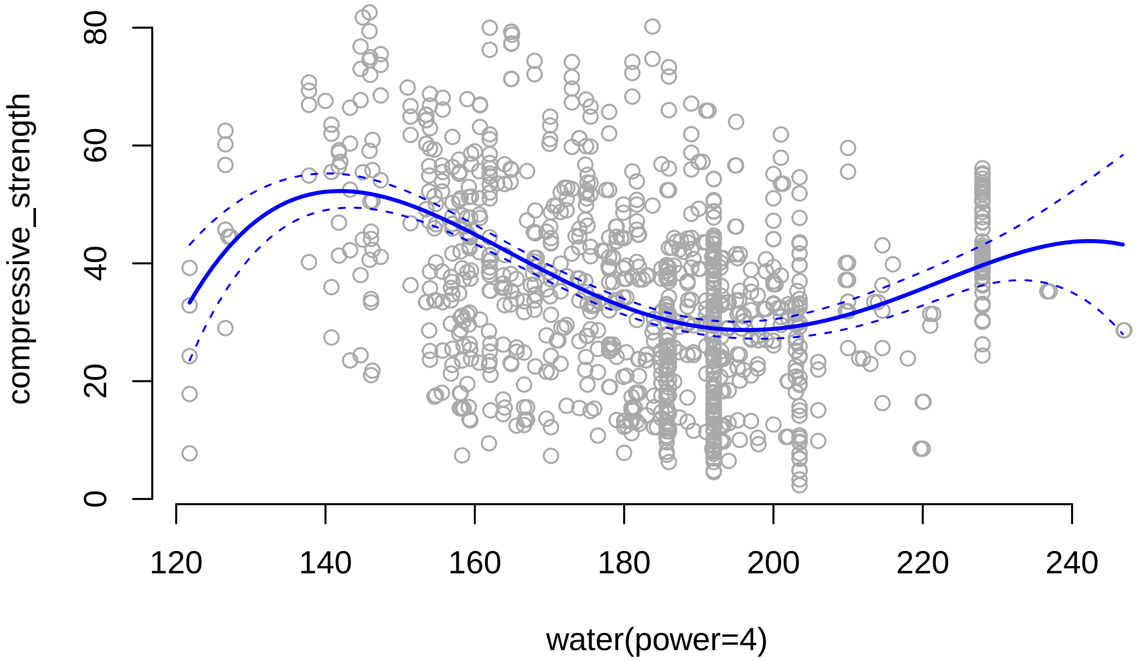

```
plot(compressive_strength ~ water, data = concrete_data, col = "darkgrey",  bty = "n", xlab="water(powe
waterlims <-  range(concrete_data$water)
water.grid <-  seq(from = waterlims[1], to = waterlims[2])
lm.fit <-  lm(compressive_strength ~ poly(water, 2), data = concrete_data)
lm.pred <-  predict(lm.fit, data.frame(water = water.grid), se = TRUE)
# mean prediction
lines(x = water.grid , y = lm.pred$fit, col = "blue", lwd = 2)
# uncertainty bands
matlines( x = water.grid, y = cbind( lm.pred$fit + 2*lm.pred$se.fit, lm.pred$fit - 2*lm.pred$se.fit),
          lty = "dashed", col = "blue")
```



```r
plot(compressive_strength ~ water, data = concrete_data, col = "darkgrey",  bty = "n", xlab="water(powe
waterlims <-  range(concrete_data$water)
water.grid <-  seq(from = waterlims[1], to = waterlims[2])
lm.fit <-  lm(compressive_strength ~ poly(water, 3), data = concrete_data)
lm.pred <-  predict(lm.fit, data.frame(water = water.grid), se = TRUE)
# mean prediction
lines(x = water.grid , y = lm.pred$fit, col = "blue", lwd = 2)
# uncertainty bands
matlines( x = water.grid, y = cbind( lm.pred$fit + 2*lm.pred$se.fit, lm.pred$fit - 2*lm.pred$se.fit),
          lty = "dashed", col = "blue")
```




```r
plot(compressive_strength ~ water, data = concrete_data, col = "darkgrey",  bty = "n", xlab="water(powe
waterlims <-  range(concrete_data$water)
water.grid <-  seq(from = waterlims[1], to = waterlims[2])
lm.fit <-  lm(compressive_strength ~ poly(water, 4), data = concrete_data)
lm.pred <-  predict(lm.fit, data.frame(water = water.grid), se = TRUE)
# mean prediction
lines(x = water.grid , y = lm.pred$fit, col = "blue", lwd = 2)
# uncertainty bands
matlines( x = water.grid, y = cbind( lm.pred$fit + 2*lm.pred$se.fit, lm.pred$fit - 2*lm.pred$se.fit),
          lty = "dashed", col = "blue")
```



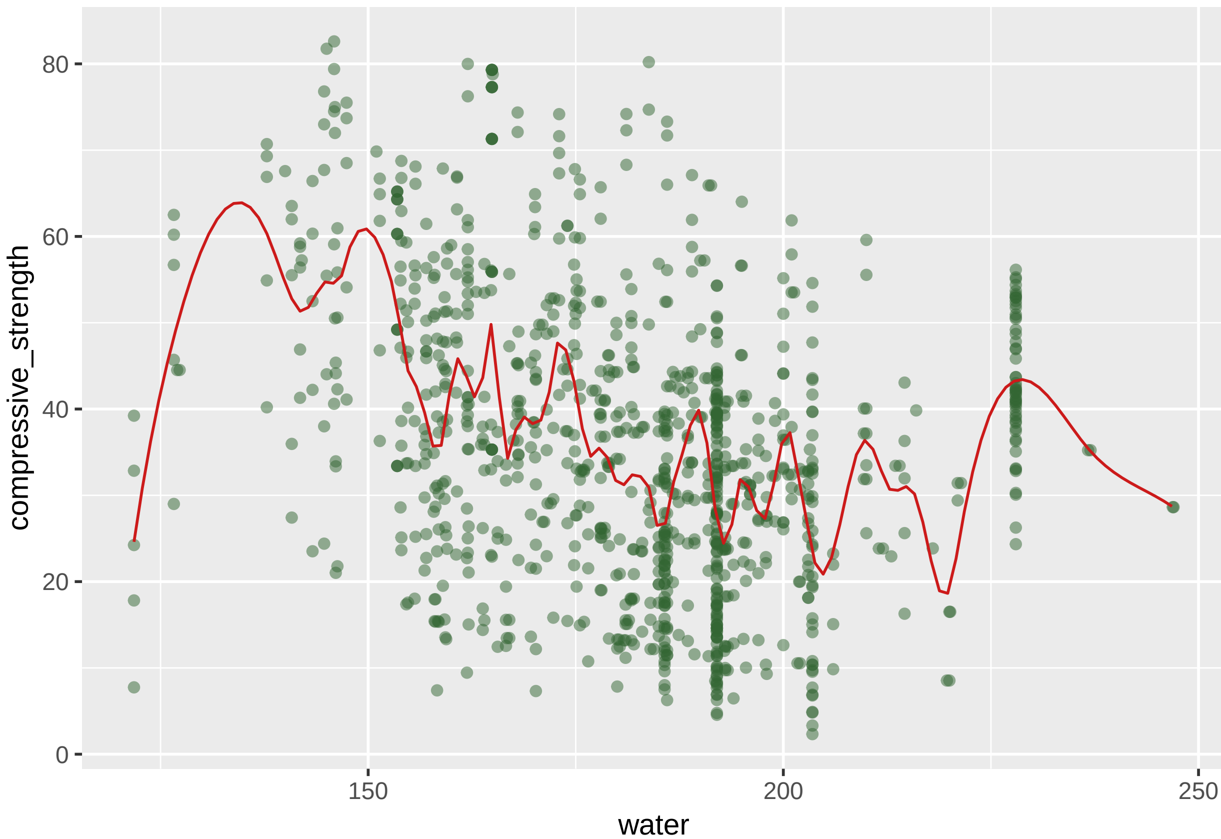
Question 3

A range of df

```
p <- ggplot(data = concrete_data, aes(x = water, y = compressive_strength)) +
     geom_point(color = rgb(.2, .4, .2, .5))
```

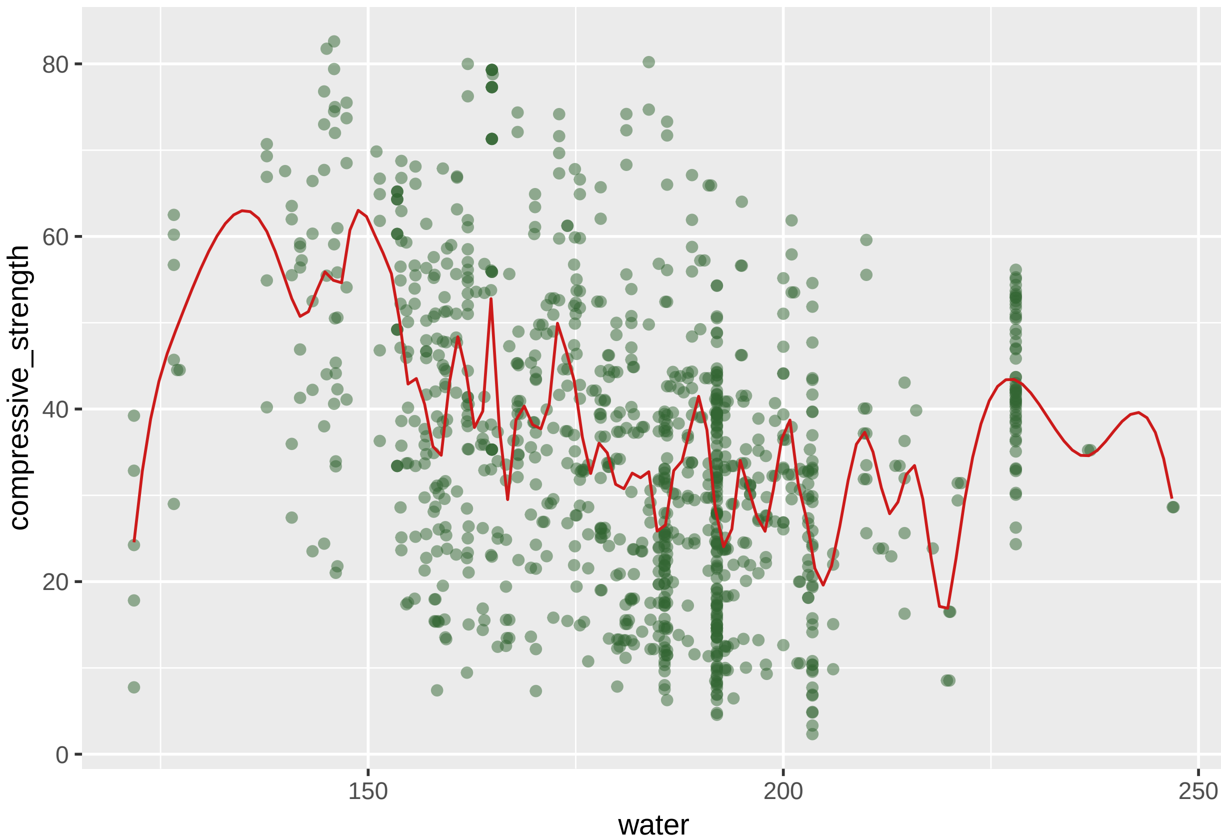
Degrees of freedom = 40

```
fit.ss <- smooth.spline(concrete_data$water, concrete_data$compressive_strength, df = 40)

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

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

p +
geom_line(aes(x = water, y = pred), data = pred.ss.df,
          color = rgb(.8, .1, .1, 1))
```



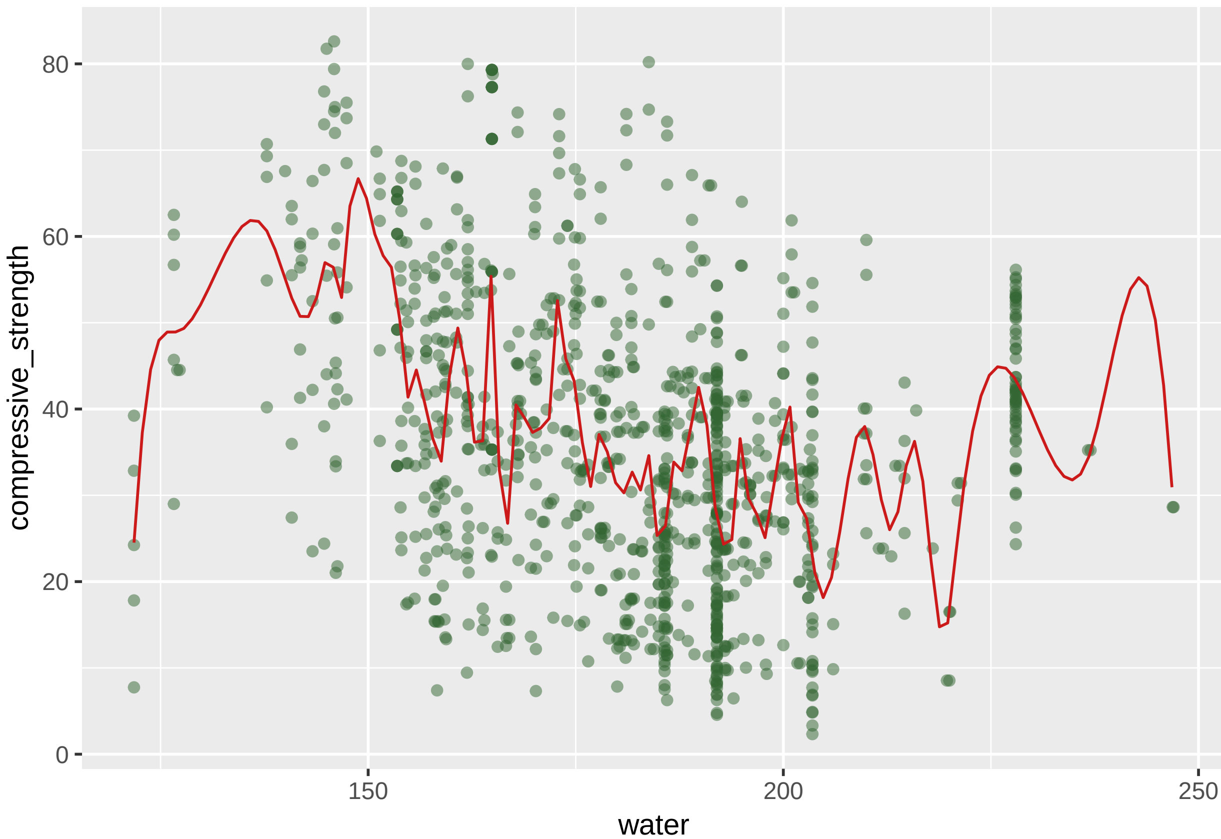
Degrees of freedom = 50

```
fit.ss <- smooth.spline(concrete_data$water, concrete_data$compressive_strength, df = 50)

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

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

p +
geom_line(aes(x = water, y = pred), data = pred.ss.df,
          color = rgb(.8, .1, .1, 1))
```



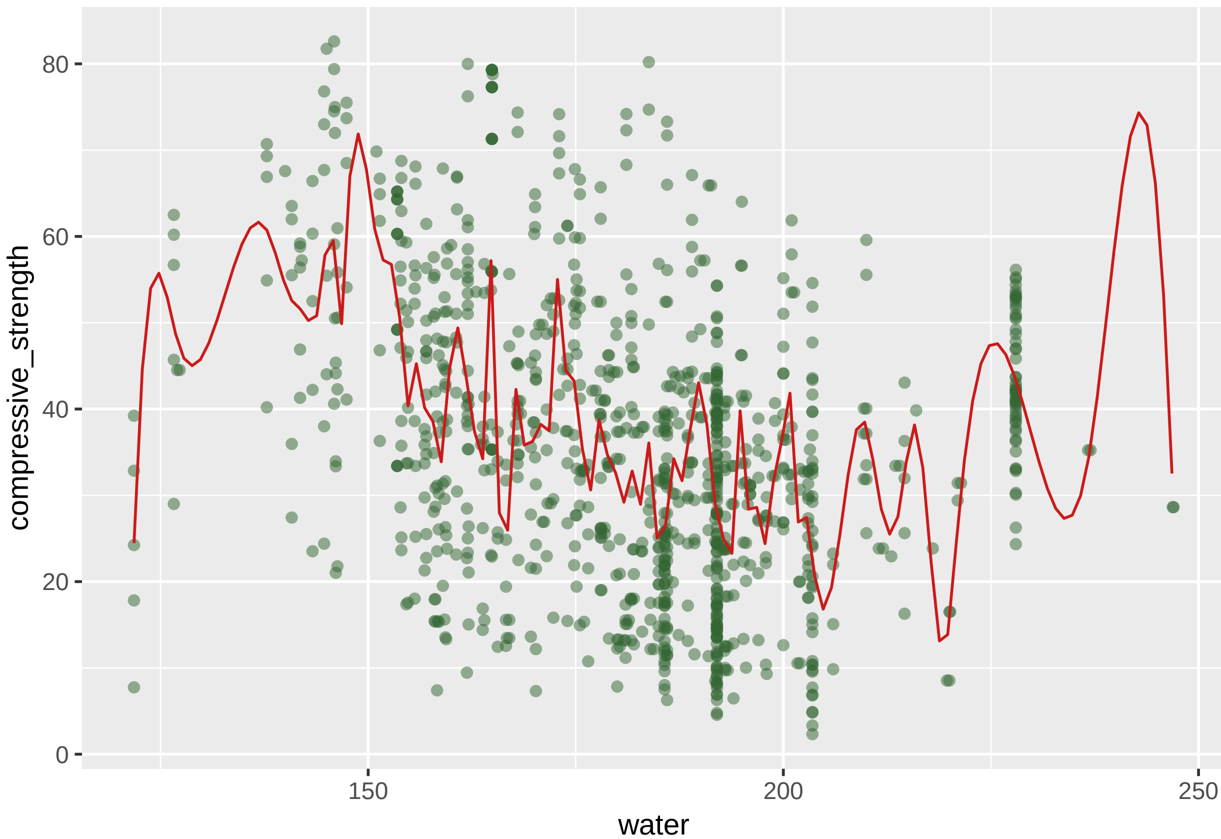
Degrees of freedom = 60

```
fit.ss <- smooth.spline(concrete_data$water, concrete_data$compressive_strength, df = 60)

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

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

p +
geom_line(aes(x = water, y = pred), data = pred.ss.df,
          color = rgb(.8, .1, .1, 1))
```



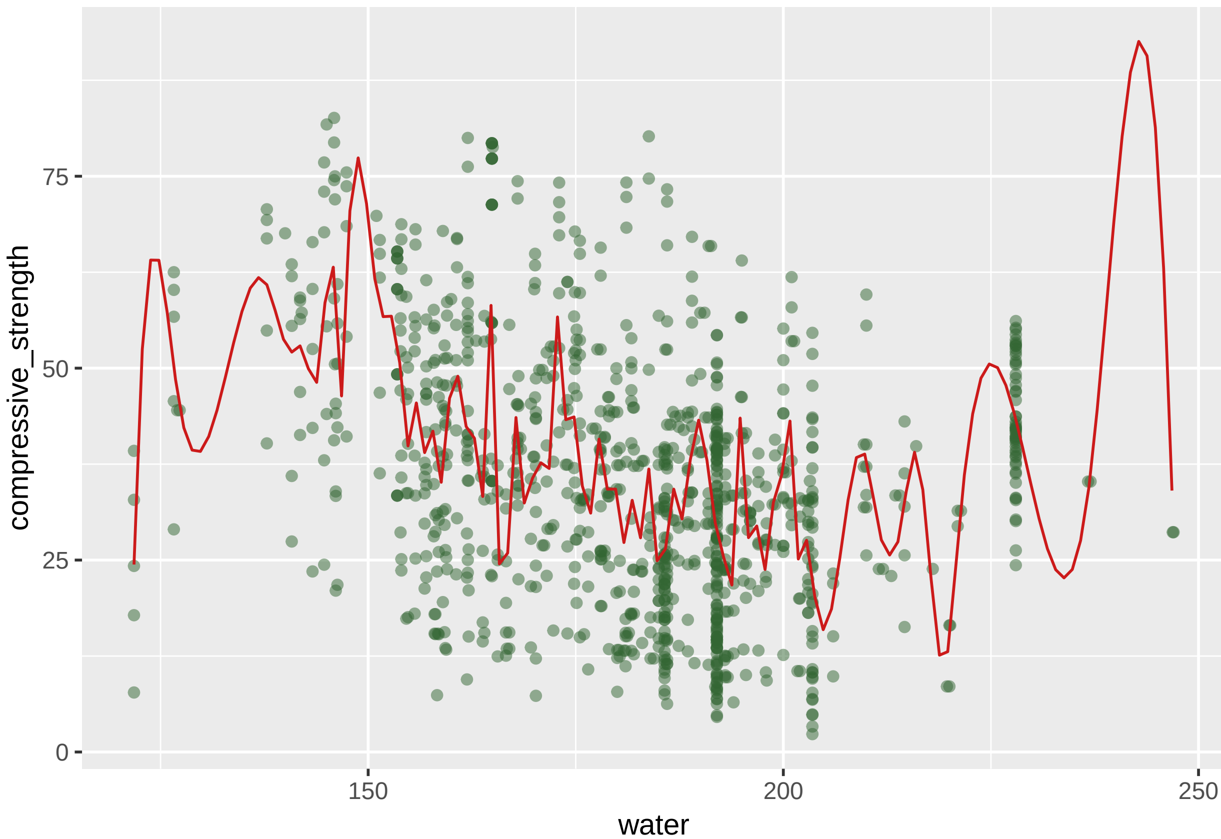
Degrees of freedom = 70

```
fit.ss <- smooth.spline(concrete_data$water, concrete_data$compressive_strength, df = 70)

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

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

p +
geom_line(aes(x = water, y = pred), data = pred.ss.df,
          color = rgb(.8, .1, .1, 1))
```



Degrees of freedom = 80

```
fit.ss <- smooth.spline(concrete_data$water, concrete_data$compressive_strength, df = 80)

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

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

p +
geom_line(aes(x = water, y = pred), data = pred.ss.df,
          color = rgb(.8, .1, .1, 1))
```



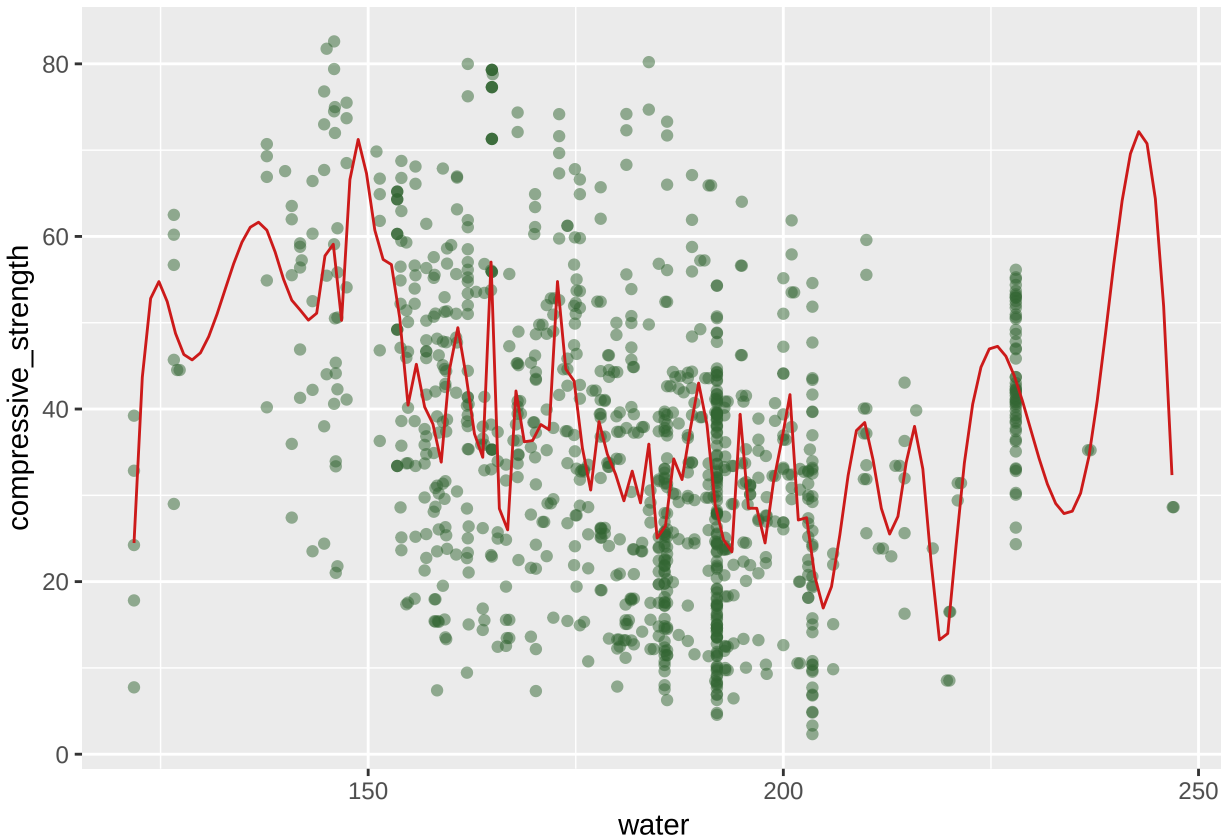
With degrees of freedom increase, the fitted model become more flexible.

Generalized cross-validation

```
fit.ss <- smooth.spline(concrete_data$water, concrete_data$compressive_strength)
fit.ss$df
```

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

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

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

p <- ggplot(data = concrete_data, aes(x = water, y = compressive_strength)) +
     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))
```



The degree of freedom obtained by generalized cross-validation is 68.88 and the fitted model is very flexible.

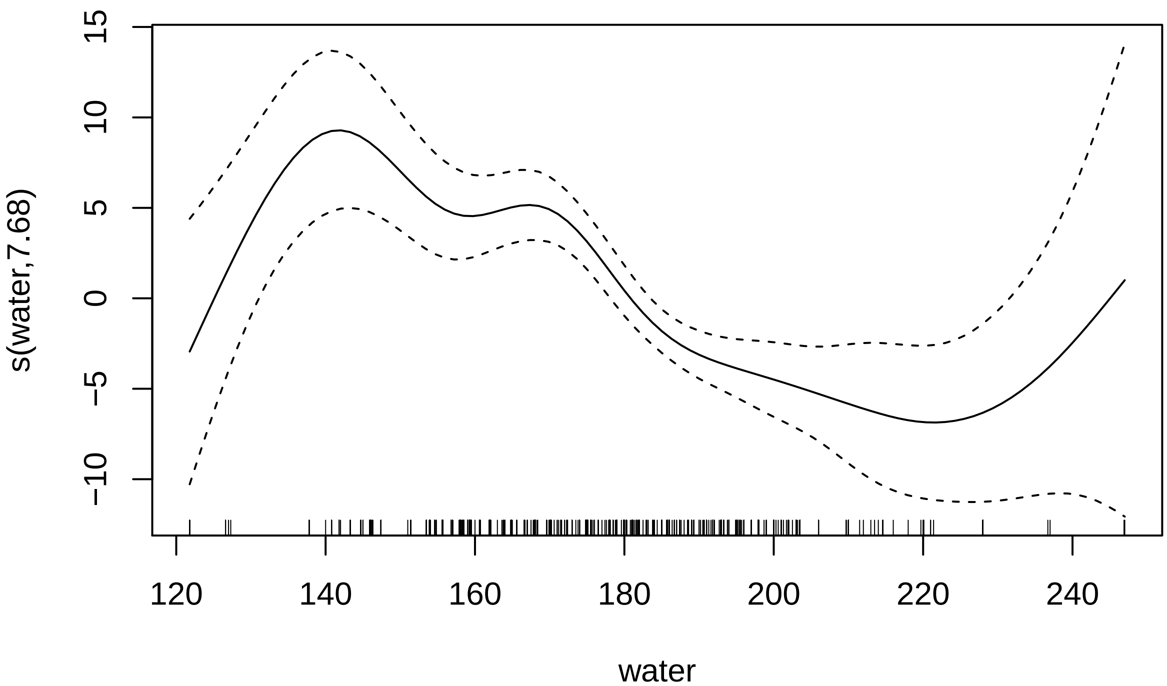
Question 4

```
gam.m1 <- gam(compressive_strength ~ cement+blast_furnace_slag+fly_ash+s(water)+superplasticizer+coarse_
summary(gam.m1)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## compressive_strength ~ cement + blast_furnace_slag + fly_ash +
##     s(water) + superplasticizer + coarse_aggregate + fine_aggregate +
##     age
##
## Parametric coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -40.147761  21.017401  -1.910   0.0564 .
## cement               0.115093   0.008750  13.153  < 2e-16 ***
## blast_furnace_slag   0.098998   0.010409   9.510  < 2e-16 ***
## fly_ash              0.080112   0.013046   6.141 1.18e-09 ***
## superplasticizer     0.142826   0.097775   1.461   0.1444
## coarse_aggregate     0.011652   0.009734   1.197   0.2316
## fine_aggregate       0.018961   0.011294   1.679   0.0935 .
## age                  0.110772   0.005740  19.300  < 2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##            edf Ref.df F  p-value
## s(water) 7.682  8.556 6 6.72e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.625   Deviance explained =   63%
## GCV = 106.26  Scale est. = 104.64    n = 1030
```

```r
plot(gam.m1)
```



According to the result, we can find that when water equals to around 145, the strength is the highest and when water equals to around 225, the strength is the lowest.