پروژه ی تحلیل رگرسیون

امیرمحمد محمدقلیها

Regression

توضیحات مختصر مربوط به هر ستون داده به صورت زیر است:

Cement (kg/m3 mixture)

Blast Furnace Slag (BFS) (kg/m3 mixture)

FlyAsh (kg/m3 mixture)

Water (kg/m3 mixture)

Superplasticizer (SP) (either 0 or 1)

Coarse Aggregate (CAgg) (kg/m3 mixture)

Fine Aggregate (FAgg) (kg/m3 mixture)

Age (days)

Concrete Compressive Strength (CCS) (mega pascals)

که 7 متغیر اول غلظت ترکیبات به کار رفته در مخلوط بتن را تعیین میکنند و متغیر 8 ام عمر و متغیر 9 ام مقاومت فشاری بتن است.

حدود داده ها به صورت زیر است:

```
summary(concrete)
```

```
##      Cement           BFS             FlyAsh            Water      
##  Min.   :102.0   Min.   :  0.0   Min.   :  0.00   Min.   :121.8  
##  1st Qu.:192.4   1st Qu.:  0.0   1st Qu.:  0.00   1st Qu.:164.9  
##  Median :272.9   Median : 22.0   Median :  0.00   Median :185.0  
##  Mean   :281.2   Mean   : 73.9   Mean   : 54.19   Mean   :181.6  
##  3rd Qu.:350.0   3rd Qu.:142.9   3rd Qu.:118.30   3rd Qu.:192.0  
##  Max.   :540.0   Max.   :359.4   Max.   :200.10   Max.   :247.0  
##        SP              CAgg             FAgg            Age        
##  Min.   : 0.000   Min.   : 801.0   Min.   :594.0   Min.   :  1.00  
##  1st Qu.: 0.000   1st Qu.: 932.0   1st Qu.:731.0   1st Qu.:  7.00  
##  Median : 6.400   Median : 968.0   Median :779.5   Median : 28.00  
##  Mean   : 6.205   Mean   : 972.9   Mean   :773.6   Mean   : 45.66  
##  3rd Qu.:10.200   3rd Qu.:1029.4   3rd Qu.:824.0   3rd Qu.: 56.00  
##  Max.   :32.200   Max.   :1145.0   Max.   :992.6   Max.   :365.00  
##       CCS      
```

```
   Min.   : 2.33 ##
1st Qu.:23.71 ##
 Median :34.45 ##
 Mean   :35.82 ##
 3rd Qu.:46.13 ##
 Max.   :82.60 ##
```

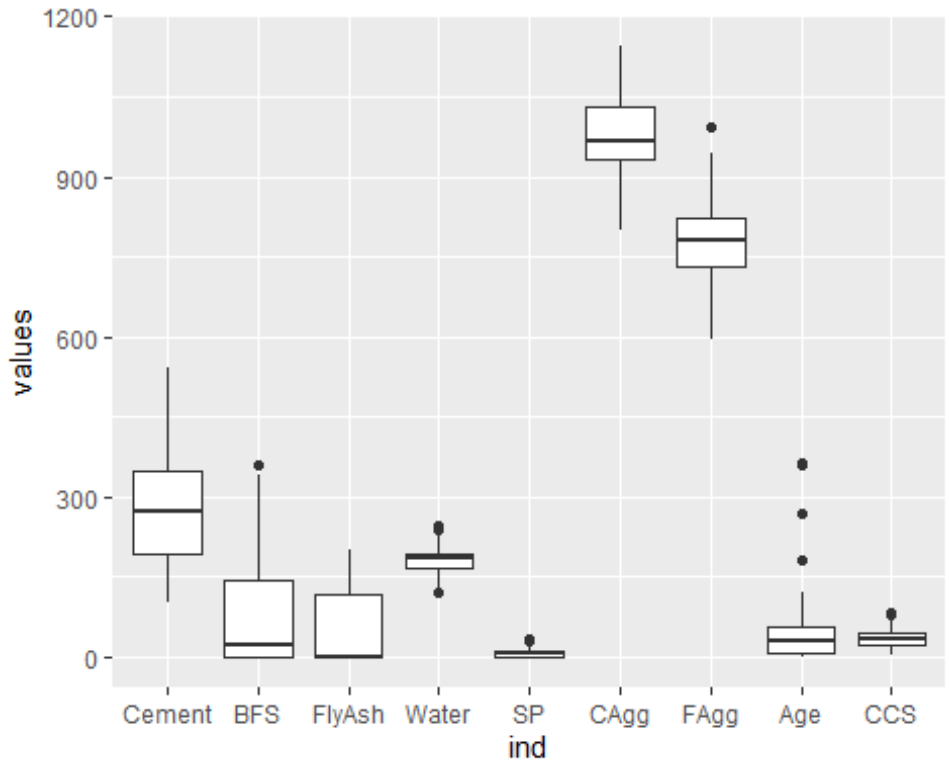
Task

در این جا قصد داریم تا با استفاده از اطلاعات مربوط به غلظت ترکیبات و عمر بتن، مقاومت فشاری آن را بدست آوریم و بتوانیم پیش بینی ای در مورد استحکام آن انجام دهیم. این تسک رگرسیونی در مقاومت مصالح و رشته ی مهندسی عمران و ساختن سازه های بتنی حائز اهمیت است.

متغیر هدف: Concrete Compressive Strength (CCS)

ابتدا نمودار جعبه ای داده ها رسم میکنیم تا پراکندگی داده ها را بررسی کنیم.

```
# before dropping the outliers
ggplot(stack(concrete),aes(x=ind,y=values))+geom_boxplot()
```



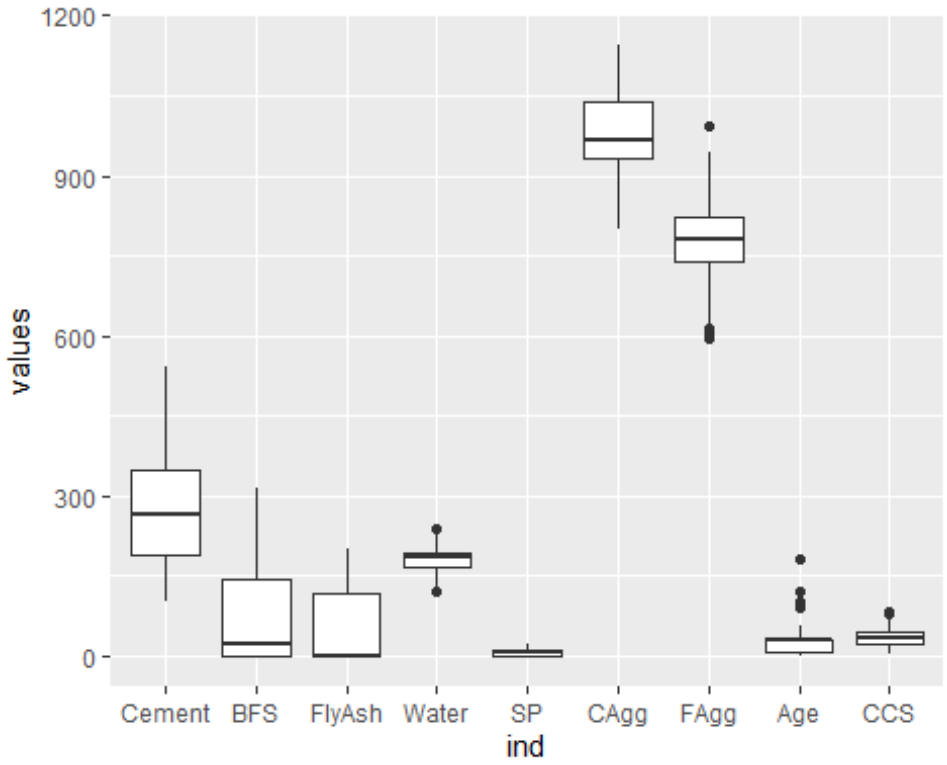
تعدادی داده ی پرت در برخی متغیر های بالا ملاحظه میکنیم که بسیار از جعبه ی خود فاصله دارند. در ابتدا داده های پرت را با استفاده از z-score حذف میکنیم. سپس داده هایی که شامل NA هستند را حذف میکنیم.

```
# remove outliers using z-score
scaled=scale(concrete, center = TRUE, scale = TRUE)
scaled=as.data.frame(scaled)
concrete[abs(scaled)>3]=NA
```

```
# remove NA from dataset
concrete=na.omit(concrete)
```

اکنون باری دیگر به نمودار جعبه ای داده ها پس از حذف داده های پرت نگاهی میندازیم.

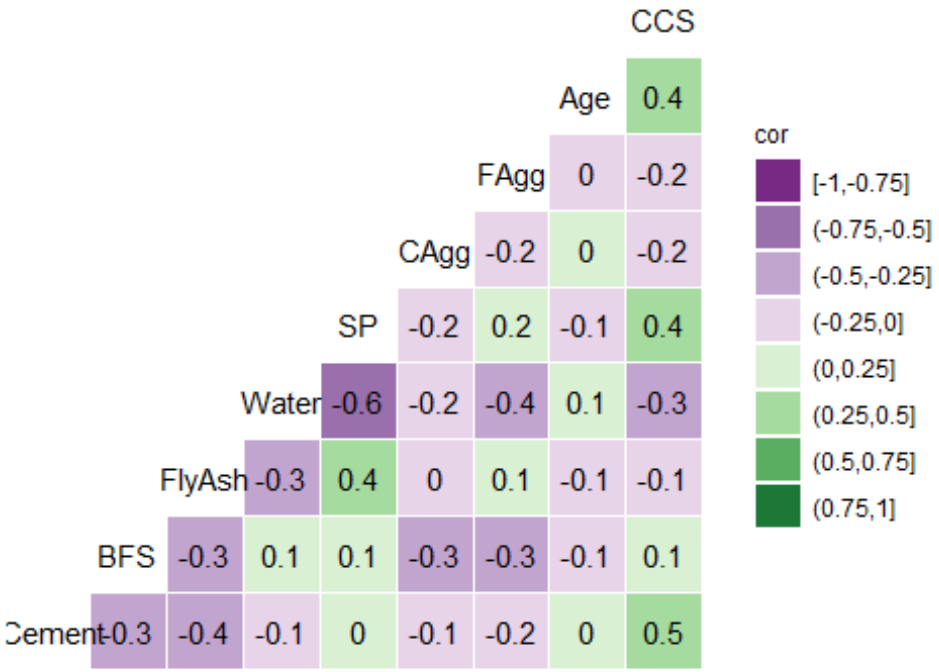
```
# after dropping the outliers
ggplot(stack(concrete),aes(x=ind,y=values))+geom_boxplot()
```



میبینیم که داده ها متمرکز تر شدند و به اندازه ی نمودار جعبه ای قبلی پراکنگی ندارند.

اکنون مشتاق هستیم تا میزان همبستگی بین متغیر ها را با بهره گیری از نمودار و شکل بررسی کنیم.

```
ggcorr(concrete, palette = "PRGn", name="cor",label=TRUE,nbreaks = 8)
```



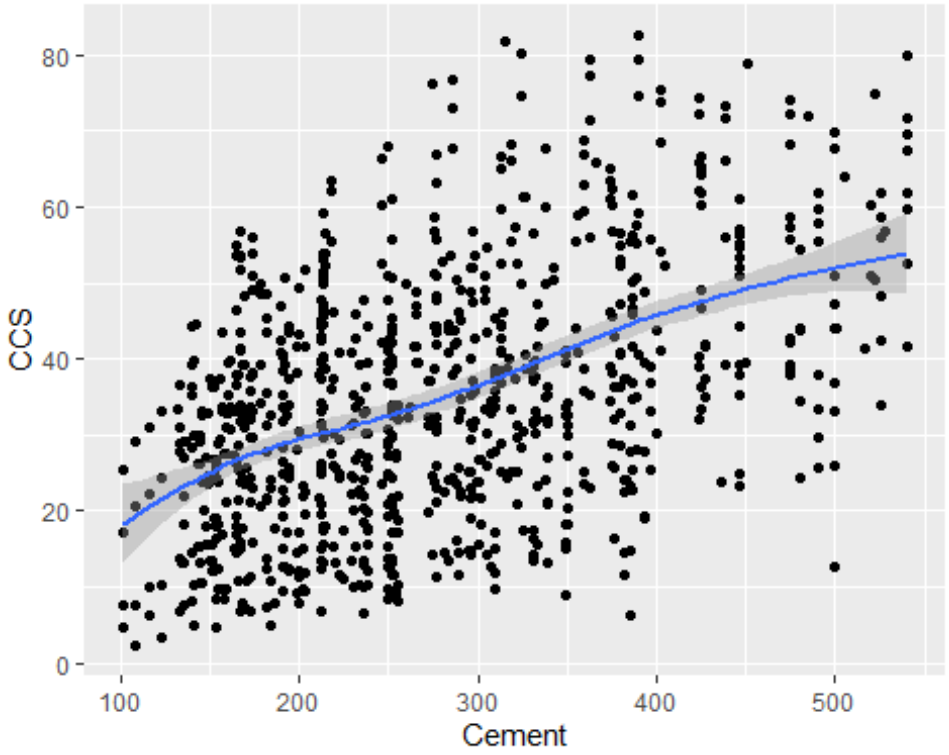
میبینیم که مقادیر

همبستگی ها آنچنان بالا نیستند که تصمیم به حذف آنها بگیریم.

اکنون برای آنکه شهودی برای فهمیدن رابطه ی بین متغیر هدف و متغیر های پیشگو بیابیم نمودار های آنها را رسم میکنیم.

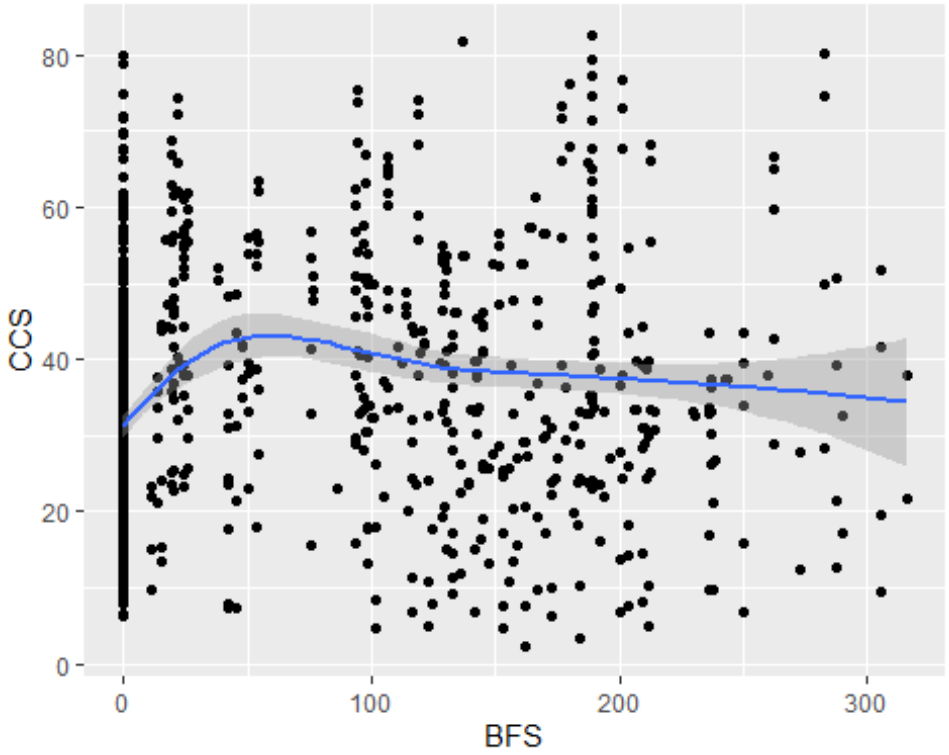
```
ggplot(concrete,aes(x=Cement , y=CCS ))+ geom_point() +geom_smooth()
```

```
'geom_smooth()` using method = 'loess' and formula 'y ~ x` ##
```



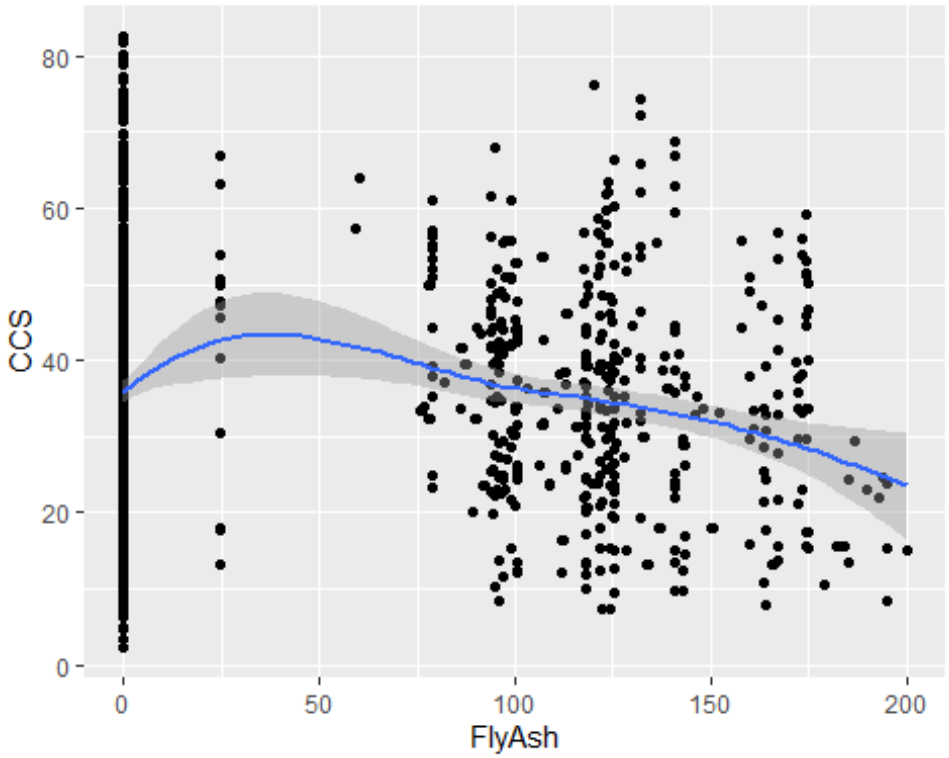

```
ggplot(concrete,aes(x=BFS , y=CCS ))+ geom_point() +geom_smooth()
```

'geom_smooth()` using method = 'loess' and formula 'y ~ x` ##



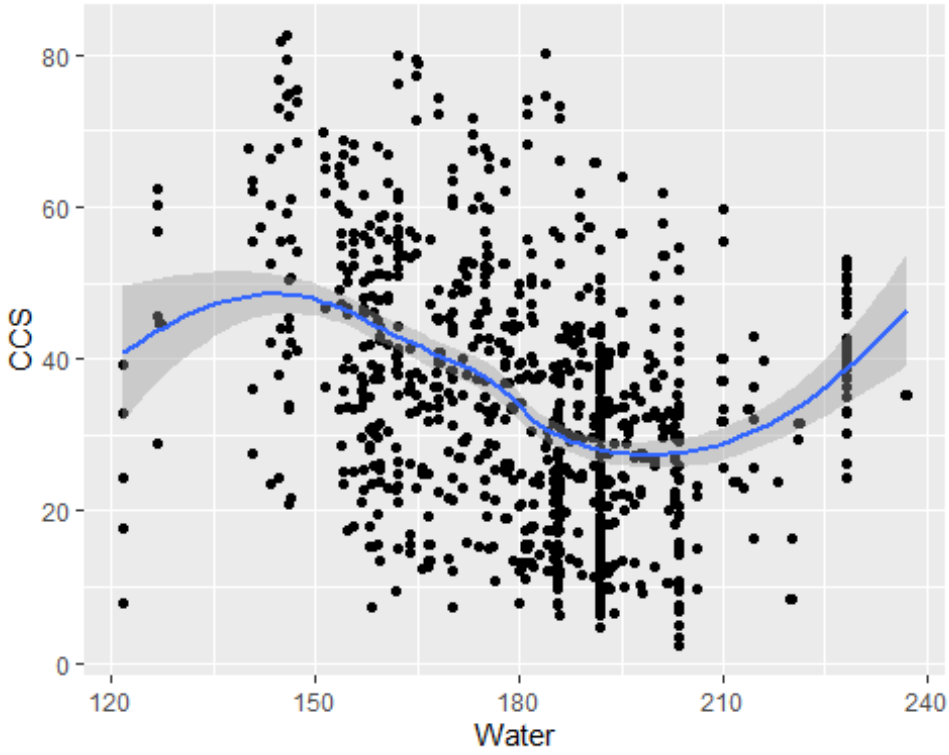
```
ggplot(concrete,aes(x=FlyAsh , y=CCS ))+ geom_point() +geom_smooth()
```

```
'geom_smooth()` using method = 'loess' and formula 'y ~ x` ##
```



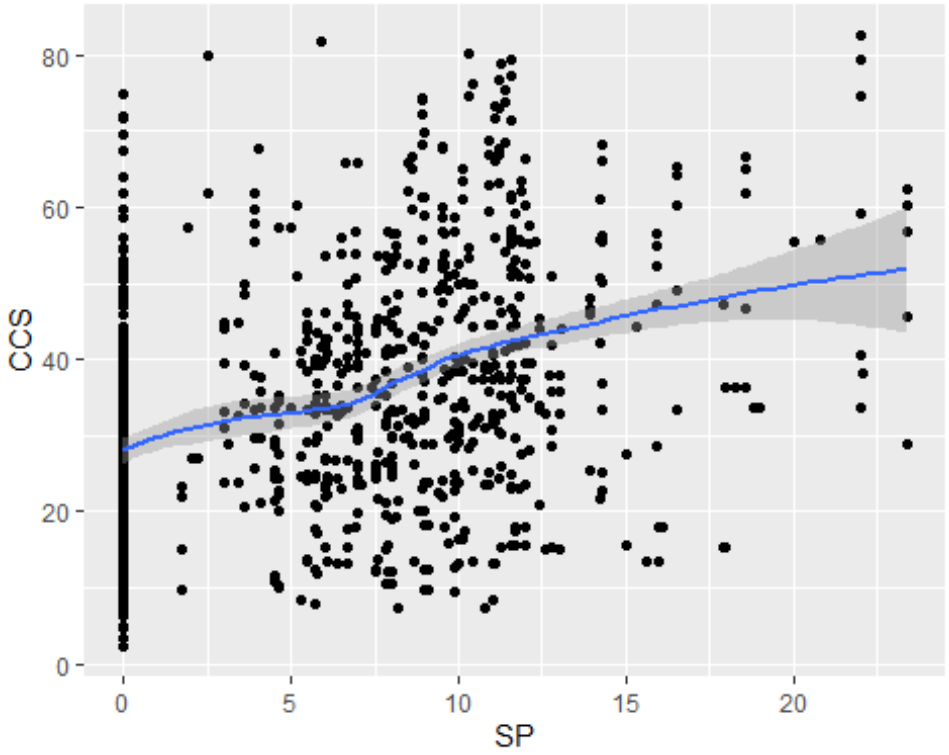

```
ggplot(concrete,aes(x=Water , y=CCS ))+ geom_point() +geom_smooth()
```

```
'geom_smooth()` using method = 'loess' and formula 'y ~ x` ##
```



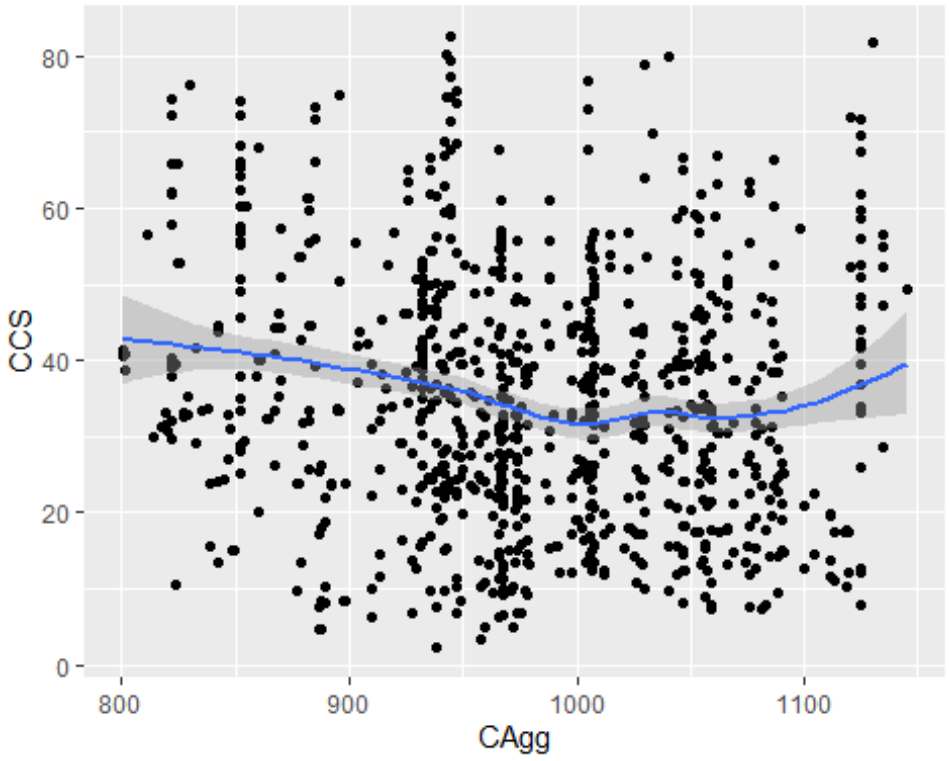

```
ggplot(concrete,aes(x=SP , y=CCS ))+ geom_point() +geom_smooth()
```

'geom_smooth()` using method = 'loess' and formula 'y ~ x` ##



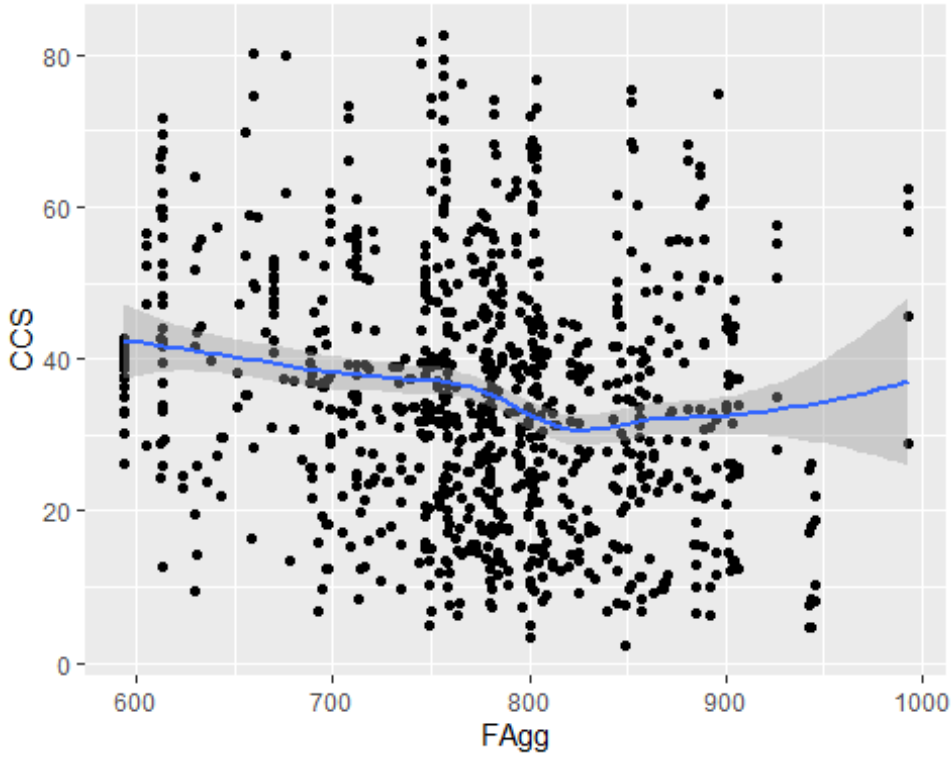
```
ggplot(concrete,aes(x=CAgg , y=CCS ))+ geom_point() +geom_smooth()
```

```
'geom_smooth()` using method = 'loess' and formula 'y ~ x` ##
```



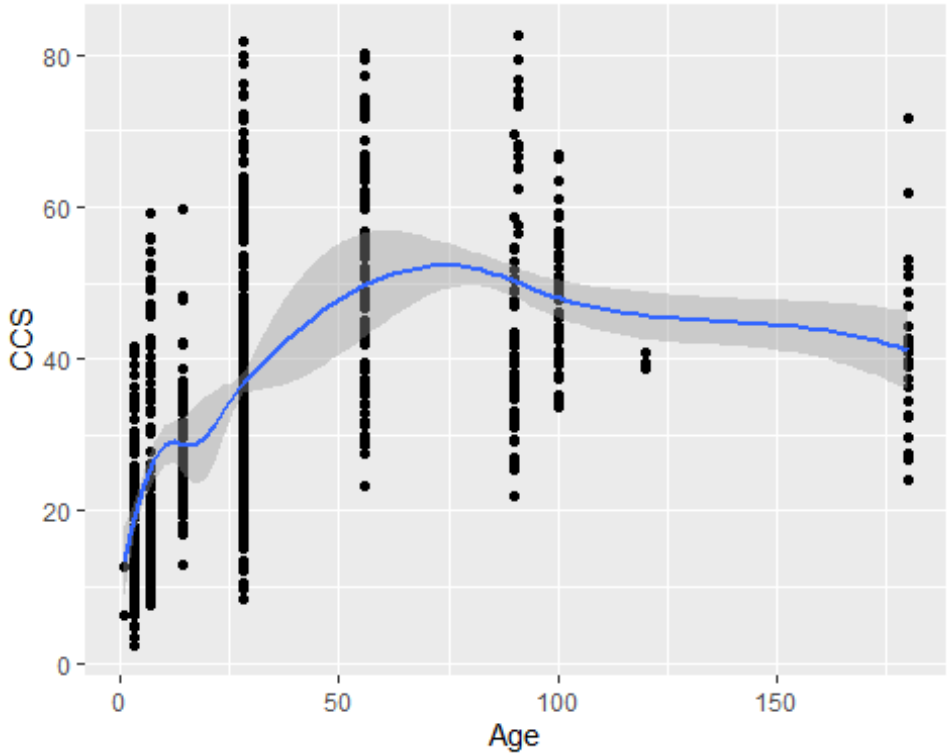

```
ggplot(concrete,aes(x=FAgg , y=CCS ))+ geom_point() +geom_smooth()
```

```
'geom_smooth()` using method = 'loess' and formula 'y ~ x` ##
```




```
ggplot(concrete,aes(x=Age , y=CCS ))+ geom_point() +geom_smooth()
```

'geom_smooth()` using method = 'loess' and formula 'y ~ x` ##



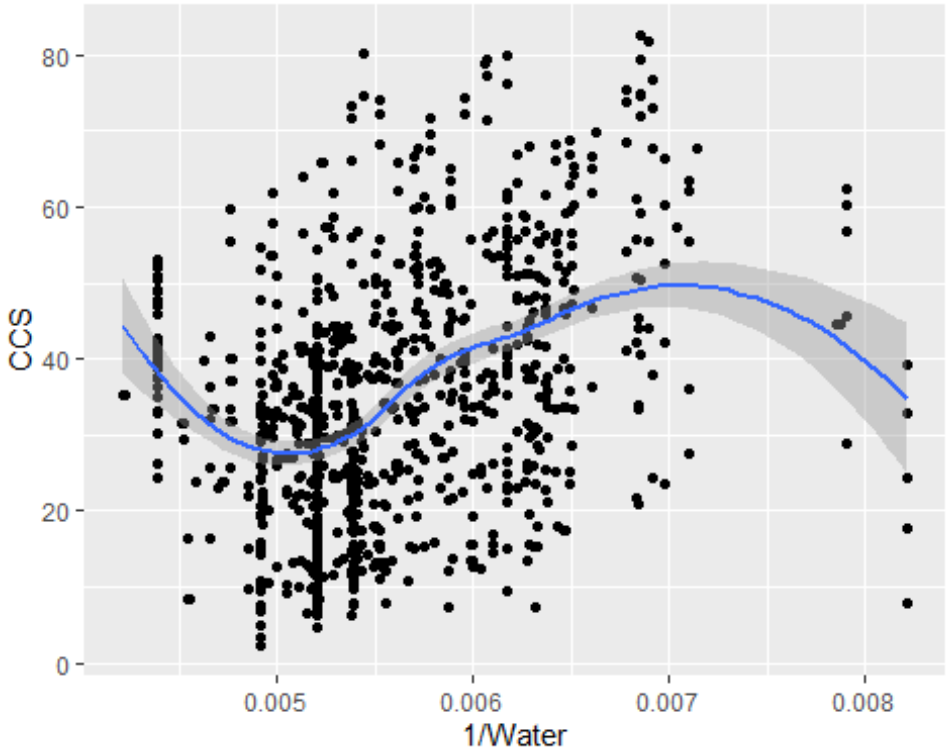
با توجه به نمودار ها،

رابطه ی تقریبا خطی بین متغیر هدف با برخی متغیر ها مانند Cement، BFS، FlyAsh، SP، CAgg، FAgg وجود دارد. دو متغیر Water و Age بنظر میاید رابطه ی خطی ای ندارند. پس سعی میکنیم توابعی روی آنها اثر دهیم تا شاهد رابطه ی تقریبا خطی شویم.

برای توابعی که به این شکل انحنا دارند یک ایده این است که ضریبی از خود تابع را با ضریبی از معکوس همان تابع جمع کنیم تا انحنای یکدیگر را خنثی کنند. برای دیدن روند انجام این کار ابتدا تابع Water/1 را رسم میکنیم.

```
ggplot(concrete,aes(x=1/Water  , y=CCS ))+ geom_point() +geom_smooth()
```

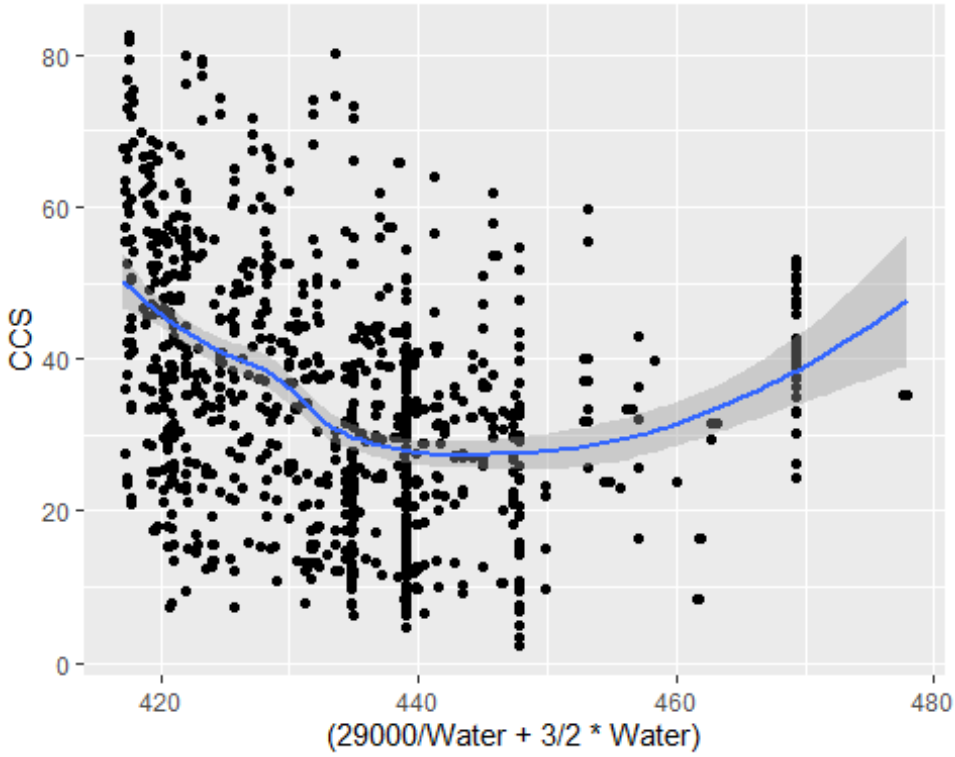
```
'geom_smooth()` using method = 'loess' and formula 'y ~ x` ##
```



قله و قعر این نمودار برعکس نمودار Water است (انگار نمودار Water را 180 درجه دوران داده ایم) تنها فرق این است که دامنه ی محور x های این دو نمودار متفاوت است. برای هم مقیاس کردن، داده های Water/1 را در 29000 و داده های Water را در 1.5 ضرب میکنیم تا مقیاس آنها متناسب شود سپس هردو را با هم جمع کنیم قله و قعر یکدیگر را خنثی میکنند یا به عبارت دیگر انحنای یکدیگر را تا حدودی میپوشانند. پس تابع را بصورت زیر رسم میکنیم.

```
ggplot(concrete,aes(x=(29000/Water+3/2*Water)  , y=CCS ))+ geom_point()
                                                          +geom_smooth()
```

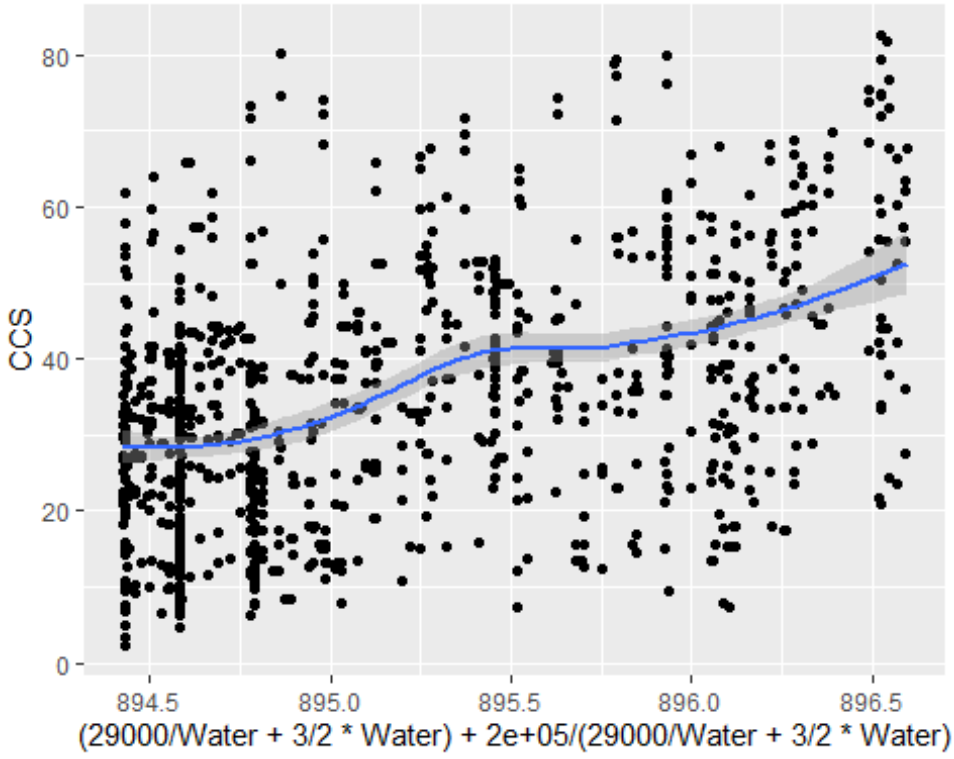
```
'geom_smooth()` using method = 'loess' and formula 'y ~ x` ##
```



میبینیم که انحنای آن ساده تر شد و بجای یک قله و یک قعر، تنها یک قعر باقی ماند. اکنون بار دیگر روند بالا را با تابع جدید انجام میدهیم (یعنی جمع کردن ضریبی از یک تابع با ضریبی از معکوس خودش) ضریب ها به گونه ای انتخاب میشوند که داده ها هم مقیاس شوند و تا قله ی نمودار اول و قعر نمودار دوم پس از جمع شدن یکدیگر را خنثی کنند.

```
ggplot(concrete,aes(x=(29000/Water+3/2*Water)+200000/(29000/Water+3/2*Water)
                                        , y=CCS ))+ geom_point() +geom_smooth()
```

```
'geom_smooth()` using method = 'loess' and formula 'y ~ x` ##
```



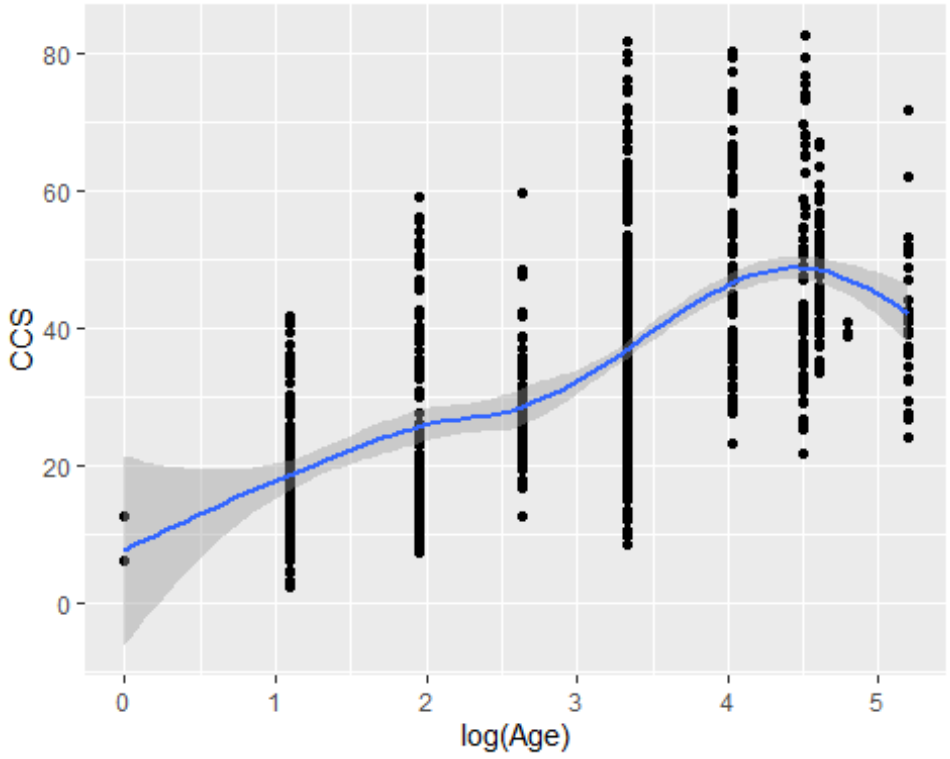
دیدیم که رابطه تا حد بسیار خوبی خطی شد.

پس میتوان گفت متغیر هدف ما با $(\frac{29000}{Water}+\frac{3}{2}Water)+\frac{200000}{(\frac{29000}{Water}+\frac{3}{2}Water)}$ رابطه ی تقریبا خطی دارد.

اکنون به دنبال تابع تبدیل یافته ای برای Age میگردیم تا رابطه ی حدودا خطی ای با متغیر هدف بیابیم.با توجه به ظاهر تابع و امتحان کردن چند تابع مقدماتی میبینیم که log انتخاب نسبتا خوبی است.

```
ggplot(concrete,aes(x=log(Age)  , y=CCS ))+ geom_point() +geom_smooth()
```

```
'geom_smooth()` using method = 'loess' and formula 'y ~ x` ##
```



پس میتوان گفت متغیر هدف ما با $log(Age)$ رابطه ی تقریبا خطی دارد.

اکنون دو ستون جدید برای دو تابع یافت شده به dataframe خود اضافه میکنیم.

```
concrete$`f(Water)`=(29000/concrete$Water+3/2*concrete$Water)+200000/(29000/c
                                                    oncrete$Water+3/2*concrete$Water)
                                                concrete$`log(Age)`=log(concrete$Age)
                                        #translocate columns to put response at the last column
                                                concrete=concrete[,c(1:8,10,11,9)]
```

که میبینیم

قبل از شروع کار و fit کردن مدل های مختلف، برای آنکه در مجموعه داده ی خود متغیر کیفی هم داشته باشیم، متغیر FlyAsh را به دو سطح کیفی تبدیل میکنیم. اگر غلظت این ماده در مخلوط صفر بود یعنی از آن استفاده ای نشده (سطح اول) و اگر غلظت آن غیر صفر بود یعنی از آن در مخلوط استفاده شده (سطح دوم) که به ترتیب به عدم استفاده شدن و استفاده شدن عدد 0 و 1 را نسبت میدهیم. با توجه به summary ای که در ابتدا از داده ها گرفته شد مشخص شد که میانه در داده های متغیر FlyAsh برابر 0 است. با بررسی کردن داده های آن مشخص شد که حدود نیمی از داده های این متغیر برابر صفر اند و در نتیجه با تبدیل این متغیر به متغیر کیفی، تعادل خوبی میان هر دو سطح باقی میماند. یعنی حدودا نیمی از آنها 0 و نیمی از آنها 1 اند. اگر treshold انتخاب شده نامناسب باشد ممکن است تاثیر آن در مدل به خوبی یافت نشود. یعنی مثلا اگر 99٪ داده ها 0 باشند، تشخیص اثر 1 ها روی مدل با داده های معدود دشوار تر خواهد شد.

```
                                                concrete[concrete$FlyAsh ==0 ,"FlyAsh"]=0
                                                concrete[concrete$FlyAsh !=0,"FlyAsh"]=1
                                                concrete$FlyAsh=as.factor(concrete$FlyAsh)
```

مدل LM

در این بخش تصمیم داریم چند مدل خطی برازش دهیم و آنها را بهبود ببخشیم و یا با هم مقایسه کنیم. ابتدا ساده ترین مدلی که شامل تمام متغیر ها است را fit میکنیم.

```
lmModel1=lm(CCS~Cement+BFS+FlyAsh+Water+SP+CAgg+FAgg+Age,data = concrete )

#summary
summary(lmModel1)
```

```
##
:Call ##
+ lm(formula = CCS ~ Cement + BFS + FlyAsh + Water + SP + CAgg ##
(FAgg + Age, data = concrete    ##
##
:Residuals ##
Min     1Q  Median     3Q    Max    ##
34.864 5.464  0.218  4.925- 28.575- ##
##
:Coefficients ##
Estimate Std. Error t value Pr(>|t|)          ##
. (Intercept) 30.083672  18.221016   1.651   0.0991 ##
*** Cement      0.107157   0.005553  19.299  < 2e-16 ##
*** BFS         0.090396   0.007046  12.829  < 2e-16 ##
*** FlyAsh1     8.958690   1.102624   8.125 1.35e-15 ##
*** Water      -0.227316   0.030599  -7.429 2.39e-13 ##
* SP           0.230078   0.092399   2.490   0.0129 ##
CAgg        -0.001654   0.006831  -0.242   0.8087 ##
FAgg        -0.002794   0.007337  -0.381   0.7034 ##
*** Age         0.222028   0.007903  28.093  < 2e-16 ##
--- ##
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 ##
##
Residual standard error: 9.019 on 972 degrees of freedom ##
Multiple R-squared:  0.7119, Adjusted R-squared:  0.7095 ##
F-statistic: 300.2 on 8 and 972 DF,  p-value: < 2.2e-16 ##
```

```
#VIF
vif(lmModel1)
```

```
Cement     BFS FlyAsh   Water      SP    CAgg    FAgg     Age ##
1.023086 3.910672 3.399898 3.096330 4.575488 3.653370 4.342342 3.909838 ##
```

به نظر میاید که دو متغیر Coarse Aggregate و Fine Aggregate معنادار نیستند و p-value بالایی دارند. پس باری دیگر مدل رگرسیون خطی را بدون این دو متغیر fit میکنیم.

علاوه بر آن با توجه به بالا بودن VIF برای برخی متغیر ها مدل را به صورت زیر کاهش میدهیم.

```
lmModel2=lm(CCS~Cement+BFS+FlyAsh+Water+SP+Age,data = concrete )

#summary
summary(lmModel2)

##
:Call ##
,lm(formula = CCS ~ Cement + BFS + FlyAsh + Water + SP + Age ##
(data = concrete    ##
##
:Residuals ##
     Min      1Q  Median      3Q     Max    ##
34.902 5.480  0.194  5.004- 28.554- ##
##
:Coefficients ##
             Estimate Std. Error t value Pr(>|t|)          ##
*** (Intercept) 24.216618   3.830657   6.322 3.93e-10 ##
*** Cement       0.108687   0.003593  30.247  < 2e-16 ##
*** BFS          0.092311   0.004542  20.324  < 2e-16 ##
*** FlyAsh1      9.197647   0.891871  10.313  < 2e-16 ##
*** Water       -0.219666   0.019356 -11.349  < 2e-16 ##
** SP            0.234487   0.084661   2.770  0.00572 ##
*** Age          0.222016   0.007890  28.138  < 2e-16 ##
---                                                    ##
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 ##
##
Residual standard error: 9.01 on 974 degrees of freedom ##
Multiple R-squared:  0.7118, Adjusted R-squared:  0.7101 ##
F-statistic:   401 on 6 and 974 DF,  p-value: < 2.2e-16 ##

#VIF
vif(lmModel2)

      Cement      BFS    FlyAsh    Water       SP      Age  ##
1.021682 2.604348 1.834408 2.394802 1.807625 1.640544 ##
```

برای آنکه ببینیم مدل ما با حذف دو متغیر بهبود یافته یا نه باید از anova استفاده کنیم. میدانیم anova برای مقایسه ی دو مدل است که فرض صفر این است که هر دو مدل به خوبی به داده ها fit میشوند و فرض مقابل این است که full model عملکرد بهتری دارد. (یعنی مدلی که متغیر های بیشتری در خود دارد)

```
anova(lmModel1,lmModel2)

Analysis of Variance Table ##
##
Model 1: CCS ~ Cement + BFS + FlyAsh + Water + SP + CAgg + FAgg + Age ##
Model 2: CCS ~ Cement + BFS + FlyAsh + Water + SP + Age ##
  Res.Df   RSS Df Sum of Sq      F Pr(>F)  ##
```

```
79062 972   1 ##
0.9284 0.0743 12.089-  2- 79074 974   2 ##
```

چون آماره ی F کوچک است درمیابیم که حذف دو متغیر CAgg و FAgg کار معقولی بوده و لطمه ای به مدل وارد نکرده.

اکنون مدلی میسازیم که در آن بجای استفاده از متغیر های Age و Water از توسیع یافته ی آنها استفاده شده باشد. دو ستون ساخته شده f(Water) و log(Age) رابطه ی تقریبا خطی ای با متغیر هدف دارند پس احتمالا مدل بهتری بسازند. این را در ادامه تست میکنیم.

```
lmModel3=lm(CCS~Cement+BFS+FlyAsh+`f(Water)`+SP+CAgg+FAgg+`log(Age)`,data =
concrete )

#summary
summary(lmModel3)
```

```
##
:Call ##
+ lm(formula = CCS ~ Cement + BFS + FlyAsh + `f(Water)` + SP ##
(CAgg + FAgg + `log(Age)`, data = concrete    ##
##
:Residuals ##
Min      1Q   Median     3Q     Max    ##
28.6153 3.9128  0.0474  3.9864- 23.0052- ##
##
:Coefficients ##
Estimate Std. Error t value Pr(>|t|)            ##
*** (Intercept) -4.836e+03  4.323e+02 -11.187  < 2e-16 ##
*** Cement       1.288e-01  3.634e-03  35.438  < 2e-16 ##
*** BFS          1.129e-01  4.651e-03  24.271  < 2e-16 ##
*** FlyAsh1      1.091e+01  7.685e-01  14.199  < 2e-16 ##
*** f(Water)`   5.297e+00  4.865e-01  10.887  < 2e-16` ##
SP          3.568e-02  6.573e-02   0.543    0.587 ##
*** CAgg        2.659e-02  3.832e-03   6.940 7.17e-12 ##
*** FAgg        3.446e-02  4.049e-03   8.512  < 2e-16 ##
*** log(Age)`   8.924e+00  1.941e-01  45.971  < 2e-16` ##
--- ##
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 ##
##
Residual standard error: 6.663 on 972 degrees of freedom ##
Multiple R-squared:  0.8427, Adjusted R-squared:  0.8414 ##
F-statistic: 651.1 on 8 and 972 DF,  p-value: < 2.2e-16 ##
```

```
#VIF
vif(lmModel3)
```

```
Cement      BFS     FlyAsh `f(Water)`      SP      CAgg      FAgg     ##
2.182051  1.959879  2.870387  2.347188  3.251085  3.465380  3.067645  ##
```

```
`log(Age)` ##
1.025278  ##
```

میبینیم متغیر SP در این مدل معنا دار نیست. پس آن را حذف میکنیم.

```
lmModel4=lm(CCS~Cement+BFS+FlyAsh+`f(Water)`+CAgg+FAgg+`log(Age)`,data =
                                                                concrete )

                                                              #summary
                                                            summary(lmModel4)

                                                                        ##
                                                                   :Call ##
+ lm(formula = CCS ~ Cement + BFS + FlyAsh + `f(Water)` + CAgg ##
                                                  (FAgg + `log(Age)`, data = concrete    ##
                                                                        ##
                                                              :Residuals ##
                                                     Min      1Q  Median      3Q     Max    ##
                                                28.8115 3.9308  0.0128  3.8952- 22.9392- ##
                                                                        ##
                                                           :Coefficients ##
                                                        Estimate Std. Error t value Pr(>|t|)          ##
                                              *** (Intercept) -4.967e+03  3.586e+02 -13.851  < 2e-16 ##
                                               *** Cement       1.293e-01  3.521e-03  36.707  < 2e-16 ##
                                                 *** BFS        1.137e-01  4.389e-03  25.909  < 2e-16 ##
                                               *** FlyAsh1      1.111e+01  6.723e-01  16.530  < 2e-16 ##
                                               *** f(Water)`    5.443e+00  4.050e-01  13.438  < 2e-16` ##
                                                 *** CAgg       2.629e-02  3.789e-03   6.937 7.28e-12 ##
                                                  *** FAgg      3.485e-02  3.983e-03   8.750  < 2e-16 ##
                                               *** log(Age)`    8.927e+00  1.940e-01  46.027  < 2e-16` ##
                                                                                                  --- ##
                                                  Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 ##
                                                                        ##
                                              Residual standard error: 6.661 on 973 degrees of freedom ##
                                              Multiple R-squared:  0.8427, Adjusted R-squared:  0.8416 ##
                                              F-statistic: 744.6 on 7 and 973 DF,  p-value: < 2.2e-16 ##

                                                                  #VIF
                                                                vif(lmModel4)

          `Cement      BFS    FlyAsh `f(Water)`     CAgg     FAgg `log(Age)    ##
1.024279  2.113372  1.917955  1.628130  2.490101  3.088383  2.882777  ##

                                                      anova(lmModel3,lmModel4)

                                                              Analysis of Variance Table ##
                                                                        ##
+ Model 1: CCS ~ Cement + BFS + FlyAsh + `f(Water)` + SP + CAgg + FAgg ##
                                                                  `log(Age)`    ##
```

```
`Model 2: CCS ~ Cement + BFS + FlyAsh + `f(Water)` + CAgg + FAgg + `log(Age) ##
  Res.Df   RSS Df Sum of Sq      F Pr(>F)  ##
1  43153 972                               ##
2  43166 973  -1- 13.083 0.2947 0.5874     ##
```

پایین بودن آماره ی F نشان دهنده ی آن است که هردو مدل به خوبی به داده ها fit میشوند. پس مدلی که شامل SP بود توضیح بهتری از داده ها به ما نمیداد و حذف آن کار معقولی به حساب میامد.

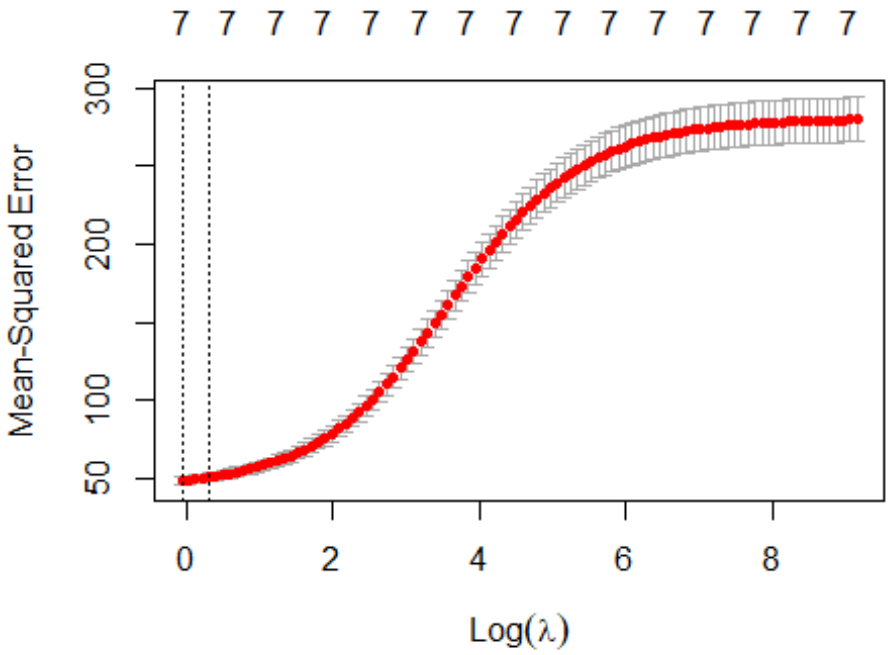
تا کنون 4 مدل مختلف امتحان کرده ایم. در مدل اول و دوم از Water و age استفاده شد و در مدل سوم و چهارم از توسیع یافته ی آنها استفاده شد. میبینیم که آماره ی R^2 در چهار مدل به صورت زیر است.

```
summary(lmModel1)$r.sq

0.7118717 [1] ##

summary(lmModel2)$r.sq

0.7118277 [1] ##

summary(lmModel3)$r.sq

0.842736 [1] ##

summary(lmModel4)$r.sq

0.8426883 [1] ##
```

یعنی این آماره تشخیص داده که مدل سوم و چهارم بهتر بوده اند. البته بالا بوددن آماره ی R^2 لزوما به معنی بهتر بودن نیست. مثلا در مقایسه ی مدل سوم و چهارم میبینیم که این آماره برای مدل سوم بهتر است. زیرا مدل سوم یک متغیر پیشگو بیشتر دارد. فرمول آماره ی R^2 به گونه ای است که هر چه تعداد متغیر های پیشگو بیشتر باشد، این آماره بزرگتر است (بین 0 و 1) ولی لزوما خطا را کم نمیکند. جلوتر پس از معرفی چند مدل دیگر نهایتا همه ی آنها را با روش cross validation میسنجیم. ولی تا به اینجا مدل چهارم را مدل بهتری میدانیم و با آن کار را ادامه میدهیم.

اکنون باراستفاده از فرمول مدل چهارم، ridge را تست میکنیم.

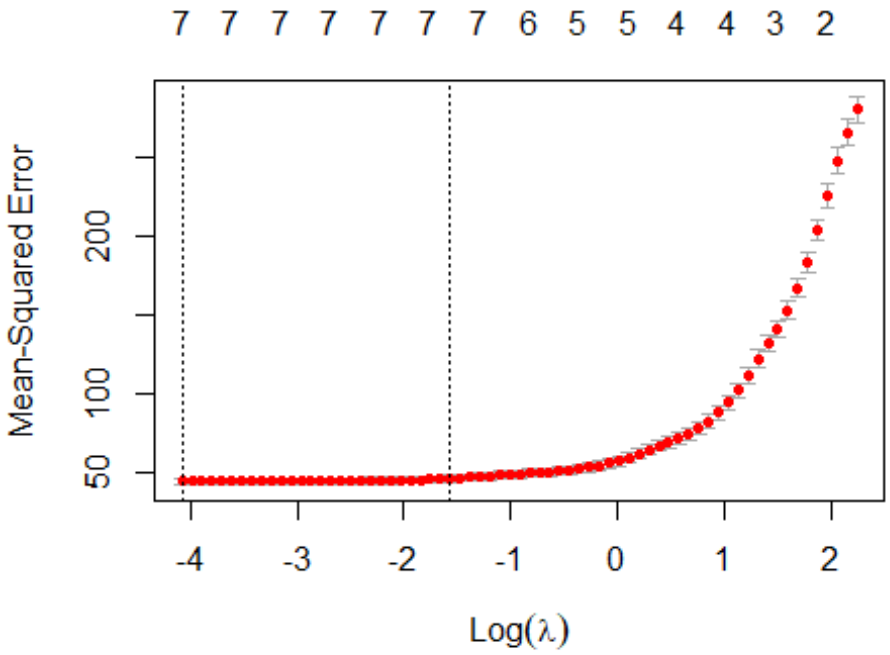
```
ridgeX=model.matrix(CCS~Cement+BFS+FlyAsh+`f(Water)`+CAgg+FAgg+`log(Age)`,con
crete)[,-1]
ridgeY=concrete$CCS

ridgeModel=glmnet(ridgeX,ridgeY,alpha=0)

cvOut=cv.glmnet (ridgeX,ridgeY,alpha =0)
plot(cvOut)
```




```
bestLambda =cvOut$lambda.min
bestLambda

0.9456815 [1] ##
```

اکنون باراستفاده از فرمول مدل چهارم، lasso را تست میکنیم.

```
lassoX=model.matrix(CCS~Cement+BFS+FlyAsh+`f(Water)`+CAgg+FAgg+`log(Age)`,con
crete)[,-1]
lassoY=concrete$CCS

lassoModel=glmnet(ridgeX,ridgeY,alpha=1)

cvOut=cv.glmnet (ridgeX,ridgeY,alpha =1)
plot(cvOut)
```




```
bestLambda =cvOut$lambda.min
                    bestLambda

0.01691493 [1] ##
```

حال مدل پنجم را به گونه ای میسازیم که تمام Interaction های ممکن بین متغیر های مدل چهارم نیز در جدول داده ها موجود باشند و بعد مدل خطی به آن برازش میکنیم و و متغیر های معنادار را نگه داشته و بی معنا ها را دور میریزیم و مدل خطی جدیدی برازش میکنیم. این کار را آنقدر انجام میدهیم تا زمانی که هیچ متغیری بی منا نباشد. در نهایت مدل ما ساخته میشود که در آخر آنها را مقایسه میکنیم.

```
concreteInter=concrete[,c(11,1,2,3,5,6,7,9,10)]
                                        for(i in 2:8){
                                    for(j in (i+1):9 ){
                    if(i!=which(colnames(concreteInter)=="FlyAsh") & j!=
                            which(colnames(concreteInter)=="FlyAsh")){
                  concreteInter[,paste0(names(concreteInter[i]),"*",names(
                concreteInter[j]))]= concreteInter[,i]* concreteInter[,j]
                                                                    }
                                                                    }
                                                                    }


                                    lmModel5=lm(CCS~.,data = concreteInter)
                                                        while(TRUE){
                                k=dim(summary(lmModel5)$coefficients)[1]
                                                                j=0
                                                            vect=c()
                                                    for (i in 2:k) {
```

```
            #checking if is there any non-significant feature to be removed
                        if(summary(lmModel5)$coefficients[i,4]>0.05){
                                                                    j=j+1
                                                                vect[j]=i
                                                                        }
                                                                        }
                                                                if(length(vect) != 0){
                                                            concreteInter=concreteInter[,-vect]
                                                                      }else{
                                                                    break()
                                                                        }
                                                                #modeling with new features
                                                            lmModel5=lm(CCS~. , data = concreteInter)
                                                                        }
                                                                summary(lmModel5)

                                                                              ##
                                                                          :Call ##
                                                            lm(formula = CCS ~ ., data = concreteInter) ##
                                                                              ##
                                                                     :Residuals ##
                                                                  Min      1Q  Median      3Q     Max    ##
                                                            25.6802 3.6755  0.0021  3.6086- 23.6985-    ##
                                                                              ##
                                                                   :Coefficients ##
                                                                    Estimate Std. Error t value Pr(>|t|)  ##
                                                        (Intercept)  4.248e+00  9.175e+00   0.463 0.643438 ##
                                                    *** Cement       1.376e+01  3.320e+00   4.145 3.70e-05 ##
                                                     *** BFS         1.905e+01  4.664e+00   4.085 4.78e-05 ##
                                                    *** FlyAsh1      9.982e+00  7.802e-01  12.794  < 2e-16 ##
                                                     *** SP          3.320e+02  6.759e+01   4.912 1.06e-06 ##
                                                    *** CAgg        -1.261e+01  1.249e+00 -10.099  < 2e-16 ##
                                                *** Cement*BFS`      1.138e-04  3.329e-05   3.417 0.000659` ##
                                                 *** Cement*SP`      2.152e-03  5.532e-04   3.890 0.000107` ##
                                               *** Cement*CAgg`      9.085e-05  2.551e-05   3.561 0.000387` ##
                                                *** Cement*FAgg`     8.979e-05  1.364e-05   6.583 7.57e-11` ##
                                            *** Cement*f(Water)` -1.542e-02  3.717e-03  -4.149 3.64e-05` ##
                                                   *** BFS*SP`       2.932e-03  6.229e-04   4.708 2.88e-06` ##
                                                  *** BFS*FAgg`      1.830e-04  2.825e-05   6.478 1.48e-10` ##
                                                *** BFS*f(Water)`   -2.141e-02  5.218e-03  -4.104 4.41e-05` ##
                                                *** BFS*log(Age)`    1.366e-02  2.051e-03   6.663 4.52e-11` ##
                                                   *** SP*CAgg`      2.294e-03  5.450e-04   4.209 2.81e-05` ##
                                                 *** SP*f(Water)`   -3.750e-01  7.553e-02  -4.965 8.12e-07` ##
                                                 *** SP*log(Age)`    2.293e-01  3.052e-02   7.515 1.31e-13` ##
                                                *** CAgg*f(Water)`   1.405e-02  1.394e-03  10.084  < 2e-16` ##
                                                *** CAgg*log(Age)`   6.854e-03  2.648e-04  25.879  < 2e-16` ##
                                                                             --- ##
                                                Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 ##
```

```
##
Residual standard error: 5.982 on 961 degrees of freedom ##
Multiple R-squared:  0.8747, Adjusted R-squared:  0.8722 ##
F-statistic:   353 on 19 and 961 DF,  p-value: < 2.2e-16 ##
```

از آنجایی که تعداد متغیر های پیشگو زیاد نیست نیازی به پیاده سازی PCA نداریم و از آن صرف نظر میکنیم.

در آخر تمام مدل ها ساخته شده را با روش Cross Validation k-fold مقایسه میکنیم. ابتدا دو تابع تعریف میکنیم. یکی برای کراس ولیدیشن مدل های خطی و دیگری برای ridge و lasso.

```
lmCrossValidation=function(df,lmModel,k){
                                        set.seed(100)
                                indexes=sample(nrow(df) ,replace = FALSE)
                    foldsIndexes = cut(indexes , breaks=k , labels=FALSE)
                                                MSE=c()
            myFormula=formula(paste(format(terms(lmModel)),collapse = ""))
            myResponse=(as.character(attr(terms(myFormula),"variables"))[-
                                    1])[attr(terms(myFormula),"response")]
                        myResponse=str_replace_all(myResponse ,"`","")
                                                for (i in 1:k) {
                    indexesOfTest = which(foldsIndexes==i , arr.ind=TRUE)
                                        trainData=df[-indexesOfTest,]
                                          testData=df[indexesOfTest,]
                       myModel=lm(formula = myFormula ,data = trainData )
                                        myPred=predict(myModel,testData)

                            MSE[i]=mean((testData[[myResponse]]-myPred)^2)
                                                }
                                                    MSE
                                                CV=mean(MSE)
                                                  return(CV)
                                                }

                                glmCrossValidation=function(df,X,Y,k,alpha){
                                        set.seed(100)
                                  indexes=sample(nrow(df) ,replace = FALSE)
                    foldsIndexes = cut(indexes , breaks=k , labels=FALSE)
                                                MSE=c()
                                              for (i in 1:k) {
                    indexesOfTest = which(foldsIndexes==i , arr.ind=TRUE)
      myModel=glmnet(X[-indexesOfTest,],Y[-indexesOfTest],alpha = alpha)
    cvOut =cv.glmnet (X[-indexesOfTest,],Y[-indexesOfTest],alpha =alpha)
                                        bestLambda =cvOut$lambda.min
            myPred=predict(myModel,s=bestLambda,newx=X[indexesOfTest,])

                                MSE[i]=mean((Y[indexesOfTest]-myPred)^2)
                                                }
                                                    MSE
                                                CV=mean(MSE)
```

```
  return(CV)
}
```

حال مدل ها را با استفاده از cv(50) مقایسه میکنیم.

```
set.seed(100)
lmCrossValidation(concrete,lmModel1,50)
```

```
## [1] 82.10653
```

```
lmCrossValidation(concrete,lmModel2,50)
```

```
## [1] 81.78077
```

```
lmCrossValidation(concrete,lmModel3,50)
```

```
## [1] 44.85757
```

```
lmCrossValidation(concrete,lmModel4,50)
```

```
## [1] 44.74055
```

```
glmCrossValidation(concrete,ridgeX,ridgeY,50,alpha =  0)
```

```
## [1] 48.11449
```

```
glmCrossValidation(concrete,lassoX,lassoY,50,alpha = 1)
```

```
## [1] 44.75104
```

```
lmCrossValidation(concreteInter,lmModel5,50)
```

```
## [1] 36.86822
```

میبینیم که مدل شامل تمام Interaction های ممکن هست بهترین عملکرد را داشته. همچنین ridge و lasso نتوانستند کمکی به بهبود مدل ما کنند.