Report

2018 年 6 月 16 日

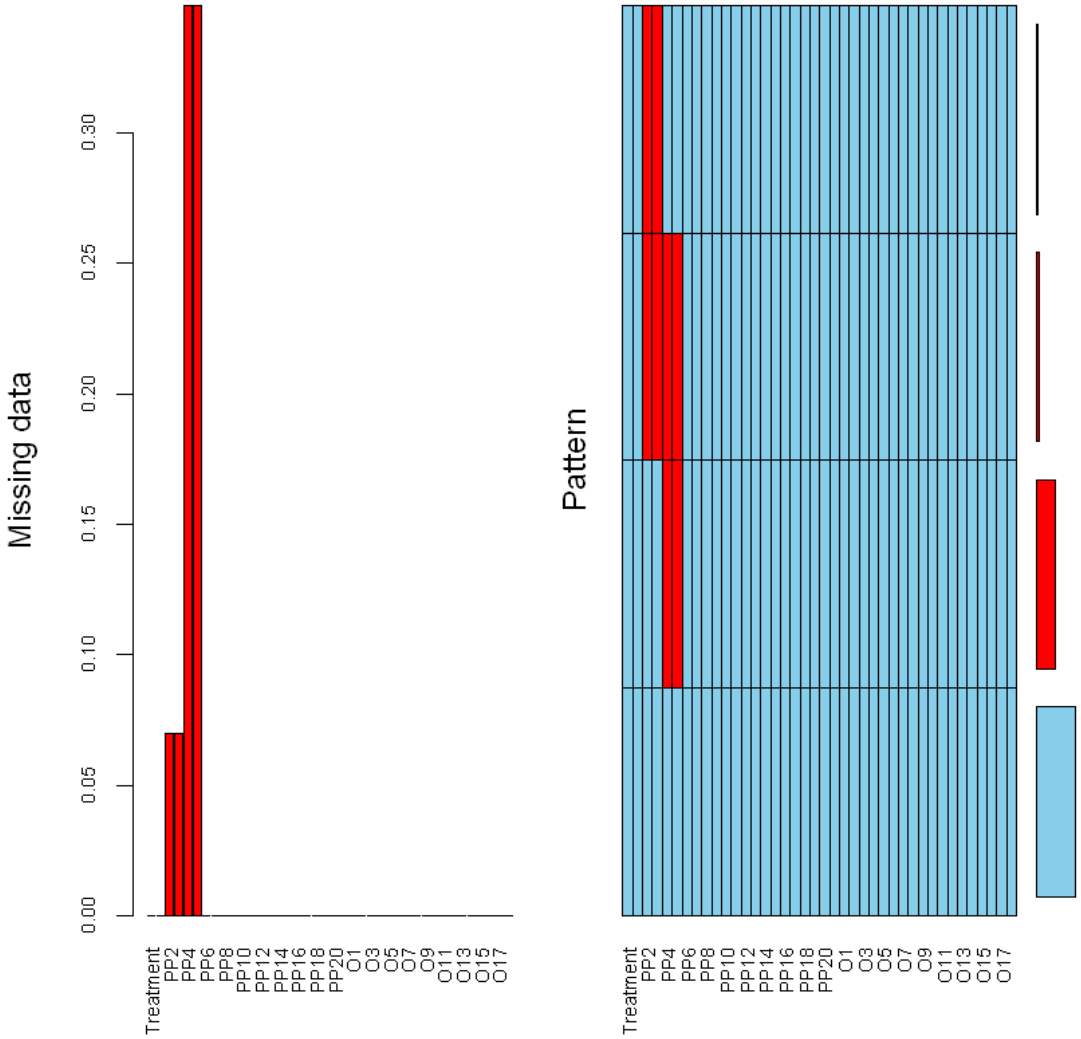
1 初步分析

一共 86 个案例，其中自变量 21 个，因变量 18 个。

```
In [1]: powder <- read.csv("powder.csv")
        rownames(powder) <- powder$Treatment
        predictors.index <- grep("PP", colnames(powder),value=T)
        responses.index <- grep("O", colnames(powder),value=T)
        powder.scaled<-powder
        powder.scaled[-1]<-scale(powder[-1]) # normalized data
        sort(apply(powder.scaled[-1], 2, function(col) e1071::skewness(col, na.rm = TRUE))) #
        mice::md.pattern(powder) # 查看数据缺失情况
        mice_plot <- VIM::aggr(powder,
                               labels=names(powder), cex.axis=.7,
                               gap=3, ylab=c("Missing data","Pattern"))
```

O2 -2.41012090617785 **O4** -1.40777397531153 **PP20** -1.33014411559028 **PP4** -1.2484765068501 **O9** -1.18241555070265 **PP21** -1.18001345177117 **PP6** -0.988223512249271 **O8** -0.966793399387697 **O5** -0.929583359617949 **PP16** -0.919975140294685 **O1** -0.919137653433223 **O3** -0.861700238487786 **O7** -0.839156421717957 **PP10** -0.804235906741372 **O14** -0.773036056606136 **PP1** -0.701160792649371 **O13** -0.662514740201373 **PP13** -0.649446892306293 **PP14** -0.630696894579872 **PP5** -0.595742012125316 **PP15** -0.585685543045854 **O6** -0.500878784694723 **PP7** -0.464330779708397 **PP19** -0.349150105510126 **PP12** -0.340014855884538 **PP3** -0.29209738512383 **O15** -0.284532644298891 **PP8** -0.267512051493445 **O17** -0.248481976049269 **O18** -0.246557216433503 **PP11** -0.207625367738082 **PP2** -0.187100414417378 **PP18** 0.114352638465019 **O12** 0.11554077017309 **O10** 0.235389595610199 **PP17** 0.253787799326091 **O11** 0.280341565729542 **PP9** 0.415379027250592 **O16** 0.689398470357599

	Treatment	PP1	PP6	PP7	PP8	PP9	PP10	PP11	PP12	PP13	...	O14	O15	O16	O17
54	1	1	1	1	1	1	1	1	1	1	...	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1	...	1	1	1	1
26	1	1	1	1	1	1	1	1	1	1	...	1	1	1	1
4	1	1	1	1	1	1	1	1	1	1	...	1	1	1	1
	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0




```
In [2]: print(na.index<-unique(which(is.na(powder),arr.ind = T)[,1]))
```

```
 [1] 51 57 58 71 83 84 53 54 56 59 60 61 62 63 64 66 67 68 69 70 72 73 74 76 77
```

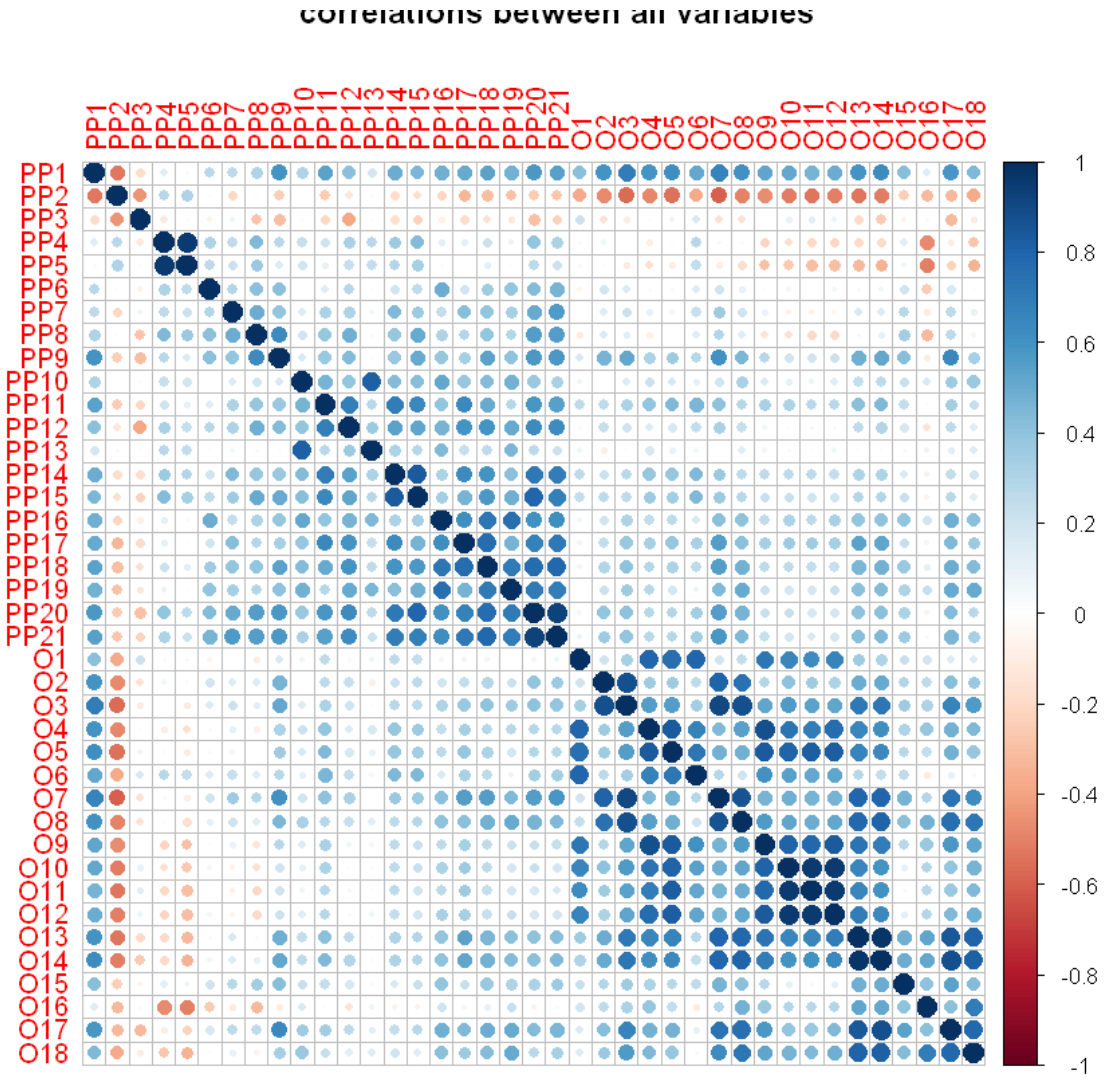
```
[26] 78 79 80 81 82 85 86
```

以上可以看到整个数据的缺失情况。在 86 个案例中，有 54 个完整，32 个有缺失。其中 PP2 和 PP3 各有 6 个缺失；PP4 和 PP5 各有 30 个缺失。

1.1 变量之间的相关性

看看所有变量（自变量加响应变量）的相关矩阵图

```
In [3]: correlations <- list("predictors" = cor(powder[predictors.index],use = "complete"),
                             "responses" = cor(powder[responses.index], use = "all.obs"),
                             "all" = cor(powder[-1], use = "complete"))
        corrplot::corrplot(correlations[["all"]], order = "original",
                           main = "correlations between all variables")
```

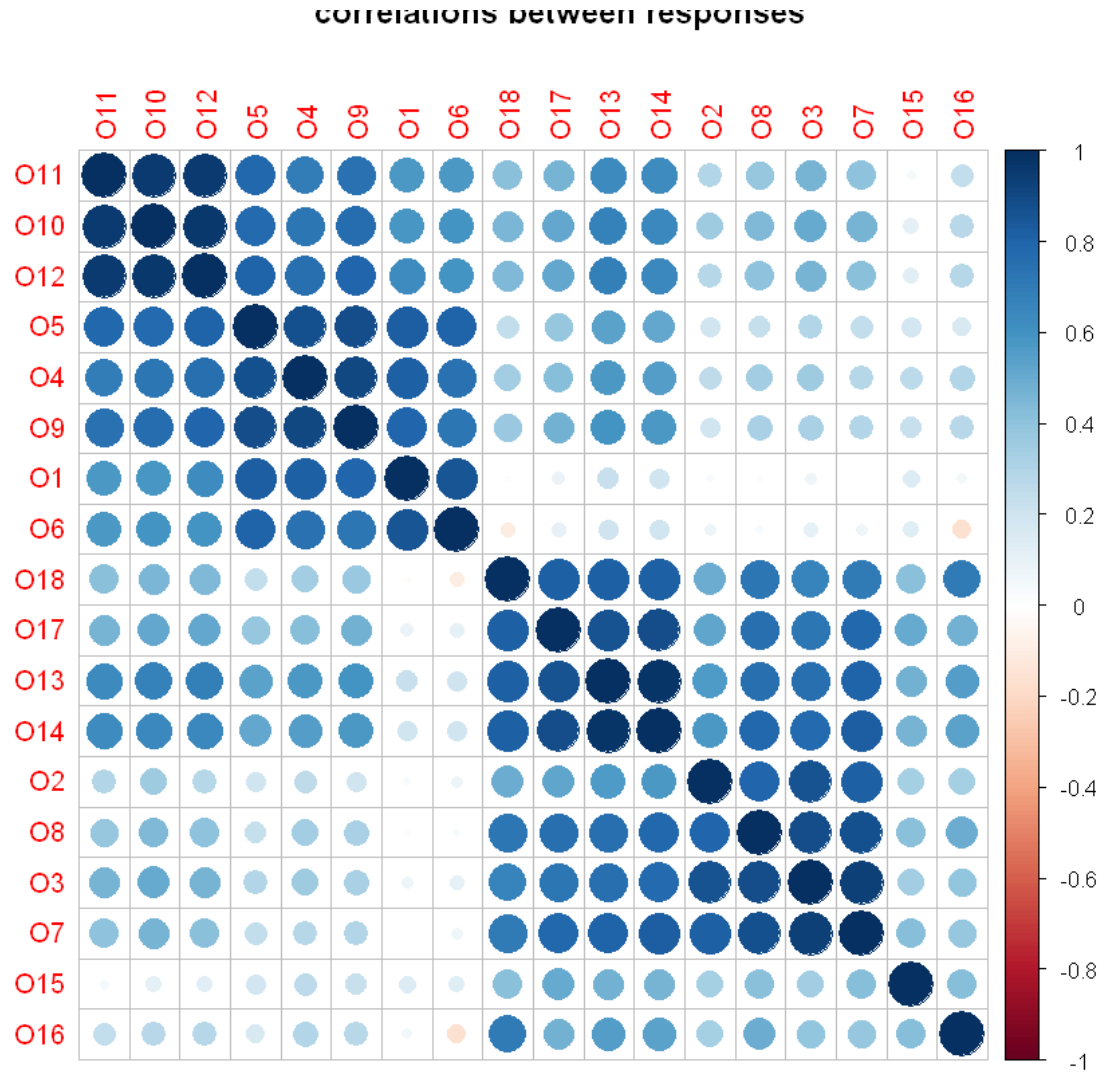


自变量和因变量很明显地分为两块，这说明许多自变量之间相关性很高（物理属性比较相近），而许多因变量之间相关性也很高（污渍类别比较接近）；但从反对角块来看，大部分自变量与因变量之间的相关性并不强。

1.1.1 响应变量之间的相关性

先看看响应变量的相关矩阵图：

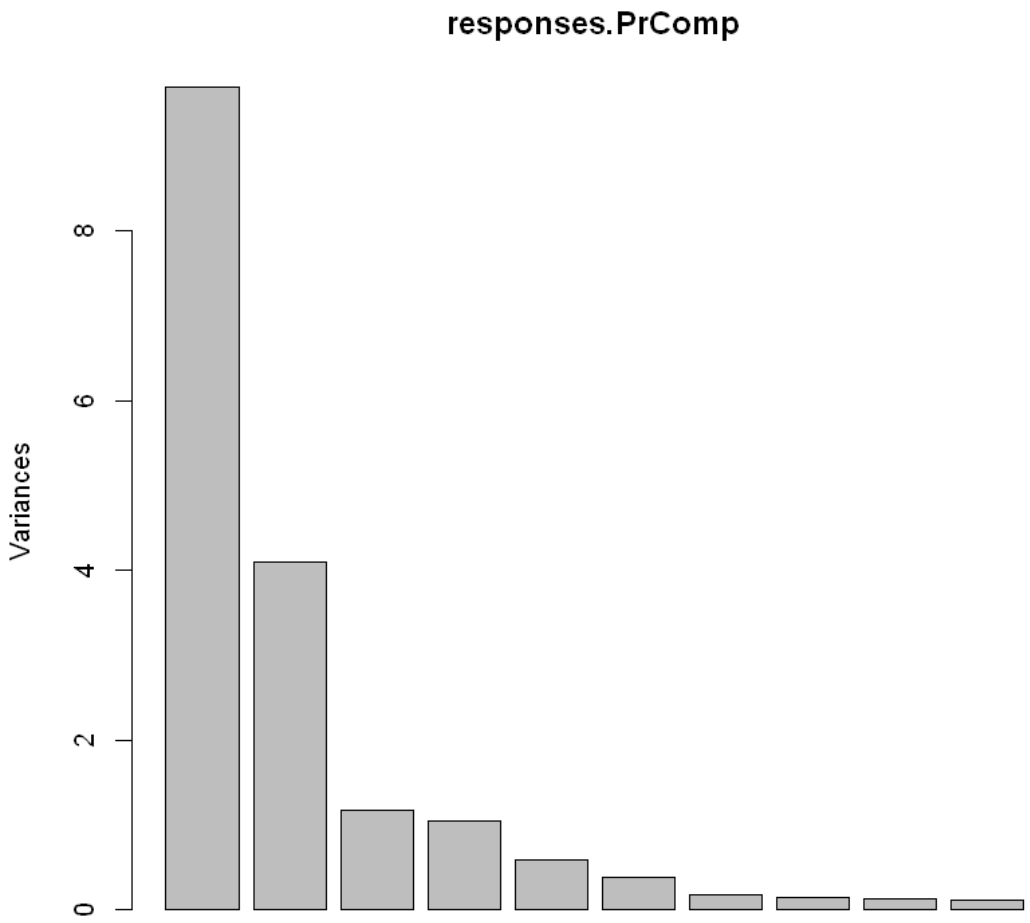
```
In [4]: corrplot::corrplot(correlations[["responses"]], order = "hclust",
                           main = "correlations between responses")
```



再用主成分分析（PCA）来看一看：（用标准化后的数据）

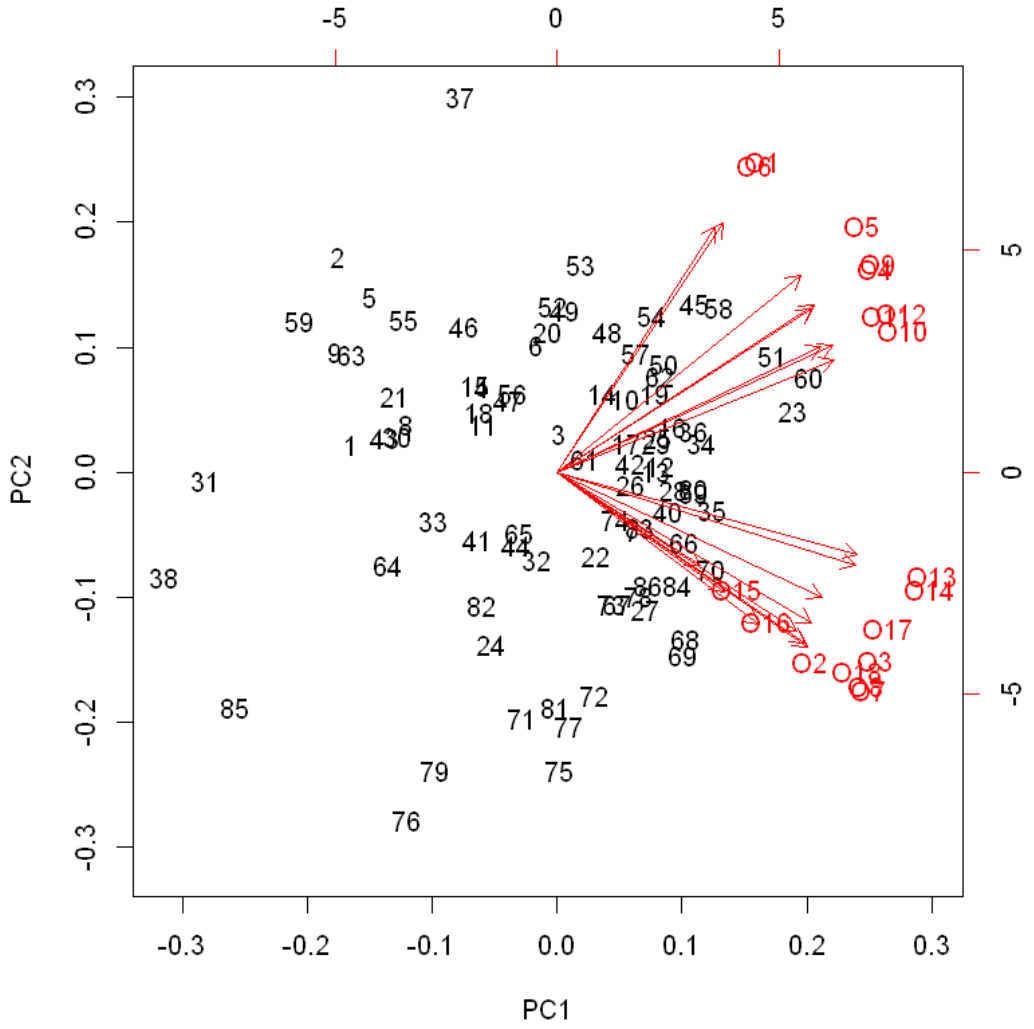
```
In [5]: ## responses 的 PCA
        responses.PrComp <- prcomp(powder[responses.index], center = TRUE, scale. = TRUE)
        plot(responses.PrComp)
        head(responses.PrComp$sd ^ 2 / sum(responses.PrComp$sd ^ 2))
```

1. 0.538860040584814 2. 0.227943997829765 3. 0.0651116577449708 4. 0.0577012473152216 5. 0.0325696519880201 6. 0.0214420177031978



前两个主成分占了变异性的大部分。

```
In [6]: biplot(responses.PrComp)
```

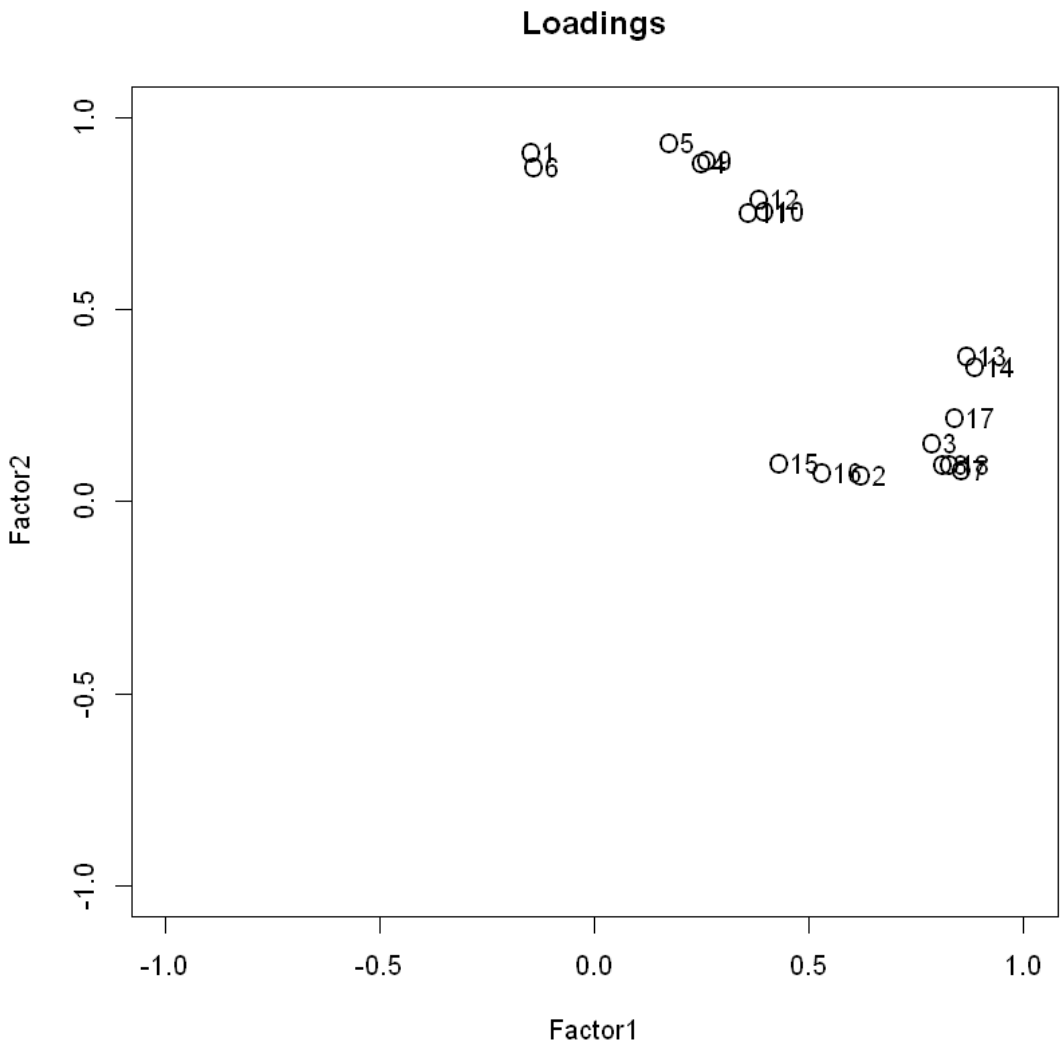


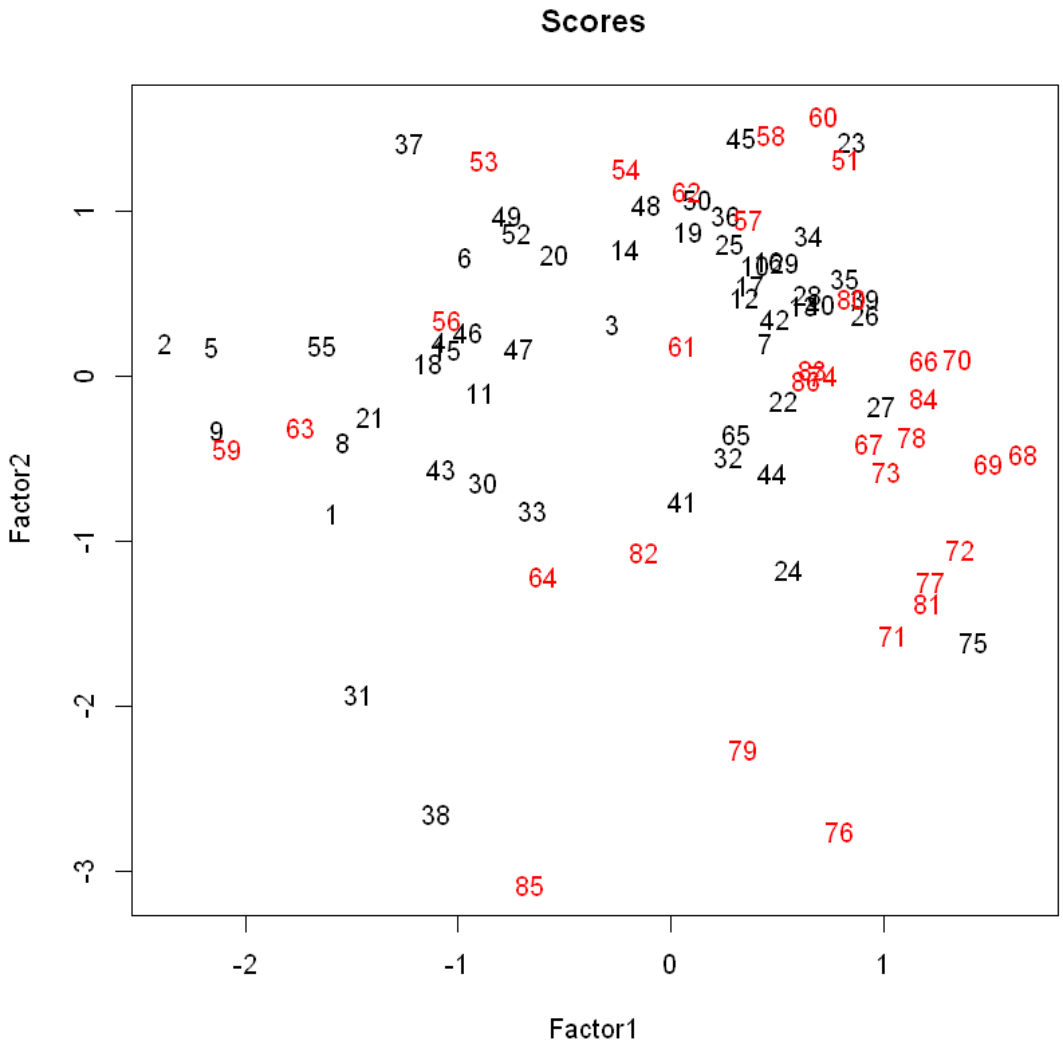
从前两个主成分的图上来看，响应变量主要分为两类。

用因子分析（factor analysis）并选取 `factors=2`：

```
In [7]: # responses 的 factor analysis
        responses.factanal <- factanal(powder.scaled[responses.index], factors = 2, scores = "
        plot(responses.factanal$loadings[, 1:2], type = "n", main = "Loadings",
             xlim = c(-1, 1), ylim = c(-1, 1))
        text(responses.factanal$loadings[, 1], responses.factanal$loadings[, 2],
             labels = row.names(responses.factanal$loadings))
        plot(responses.factanal$scores[, 1:2], type = "n", main = "Scores")
        na.col<-rep("black",nrow(powder))
```

```
na.col[na.index]<-"red"
text(responses.factanal$scores[,1],responses.factanal$scores[,2],labels=1:86,col=na.col
```





因子分析给出和主成分分析相似的建议——---都指向将响应变量分为两类，或许分别对应两类污渍，比如“溶于水”、“不溶于水”。此外，在这张图中含有缺失的案例分布得比较均匀，没有什么明显得 pattern，这说明我们应该可以用常规的 MAR 假设下得填补方法处理缺失数据。

此外，在这张图上，我们还将有缺失的案例用红色标出，我们看到缺失似乎与 factor1 体现出一定的正相关，即我们有理由怀疑这不是完全随机缺失（MCAR）

1.1.2 响应变量的分类与降维

相关矩阵、主成分分析和因子分析，都指向把响应变量分为以下两类：+ 污渍 1：[O1，O4，O5，O6，O9，O10，O11，O12] + 污渍 2：[O2，O3，O7，O8，O13，O14，O15，O16，O17，O18]

进一步查看刚刚因子分析的结果：

```
In [8]: responses.factanal
```

```
Call:
factanal(x = powder.scaled[responses.index], factors = 2, scores = "regression")

Uniquenesses:
   O1    O2    O3    O4    O5    O6    O7    O8    O9   O10   O11   O12   O13
0.152 0.588 0.330 0.147 0.088 0.224 0.231 0.301 0.126 0.242 0.274 0.204 0.039
  O14   O15   O16   O17   O18
0.029 0.774 0.676 0.190 0.251

Loadings:
    Factor1 Factor2
O1  -0.131   0.911
O2   0.638
O3   0.804   0.154
O4   0.266   0.884
O5   0.193   0.935
O6  -0.124   0.872
O7   0.873
O8   0.830
O9   0.281   0.892
O10  0.428   0.758
O11  0.392   0.757
O12  0.416   0.789
O13  0.902   0.382
O14  0.920   0.354
O15  0.464   0.103
O16  0.564
O17  0.872   0.223
O18  0.860

               Factor1 Factor2
SS loadings      6.927   6.208
Proportion Var   0.385   0.345
Cumulative Var   0.385   0.730
```

```
Test of the hypothesis that 2 factors are sufficient.
The chi square statistic is 676.08 on 118 degrees of freedom.
The p-value is 3.81e-79
```

前两个因子解释了 73% 的方差。所以我们考虑将 18 个响应变量降至两个，可以采用因子分析的 scores；或者主成分分析的前两个主成分 PC1 和 PC2。

```
In [80]: powder.scaled[c('factor1','factor2')]<-responses.factanal$scores
         powder.scaled[c('PC1','PC2')]<-responses.PrComp$x[,1:2]
         powder.scaled
```

Treatment	PP1	PP2	PP3	PP4	PP5	PP6	PP7
1	-0.839025303	1.39146301	0.07625361	1.32378599	1.87486315	-0.35645854	0.33930
2	-0.989392899	0.33386692	1.18356672	0.54105272	0.69730584	-0.71291709	0.38121
3	-0.065074440	-0.98644413	1.31443099	0.88480309	1.09035753	0.74559245	1.23234
4	-0.980547746	1.68785937	1.23389913	0.24165725	0.56548972	-0.53468781	-1.87234
5	-2.205969946	1.49924351	1.69695734	1.34857255	1.75383199	-0.30893074	-1.80786
6	0.086767348	0.15872362	-0.42203729	-1.71452365	-2.12515685	-0.52874684	-1.51125
7	1.801989880	-0.06357365	0.03598768	0.82283670	1.00647455	0.51389440	-0.05402
8	0.407035586	0.63699956	-1.15689054	1.06417946	1.25492796	0.36834049	0.37154
9	-0.750573776	1.51945235	-0.51766888	0.96438097	1.41909894	0.18417025	-0.52472
10	0.972019716	0.56963675	-2.13837261	0.76217487	1.02764502	0.19902269	-2.01742
11	-0.100086503	1.52618863	-0.90522847	-0.54498969	-0.57851441	0.77232684	0.23936
12	0.194751921	-0.34649745	0.04605416	0.38450607	0.54192259	0.58518611	-0.05079
13	0.342539681	0.19914130	-1.54445012	0.11250626	-0.03167724	1.86843686	-0.78263
14	1.223369473	-2.71766831	1.46542823	-2.37267187	-1.77804108	-0.05940976	0.50372
15	-0.199963019	-0.75741058	-0.15024226	0.70020849	0.85708295	-0.22872756	-0.63433
16	0.736148977	-1.47145635	1.49562768	0.45821345	0.10453275	1.37236539	0.23291
17	0.062074630	0.86603311	-1.41358585	0.26253014	-0.11076691	1.67535515	1.30649
18	-0.417774905	-0.11746390	1.38489637	0.08967654	0.28627921	-0.43963220	-2.42686
19	-0.149840487	-0.38691513	0.21215112	0.52735489	0.17803022	-0.11881951	0.55208
20	-1.106959720	-0.42059654	0.14671899	-0.21298032	-0.55255062	0.13664244	-2.23665
21	-1.243322492	1.82932127	0.32288244	0.73804060	0.90022277	0.76341538	0.57465
22	0.652488574	0.19914130	-0.28110653	0.54105272	0.59065461	1.09908051	1.35485
23	1.860589016	0.14525106	-0.29620626	0.37667874	-0.18106884	0.76341538	0.24258
24	-0.282149230	0.18566874	-0.26097357	0.38124468	0.52953987	0.43963220	-0.12172
25	-0.651065807	0.14525106	-0.05964391	-2.22917077	-2.04606718	-2.01399077	-0.70526
26	0.299419562	0.79193402	-1.87161081	0.34797852	0.51476048	0.62380245	0.71006
27	-0.149103391	-0.92581760	0.10645306	0.14838153	-0.39117574	0.06238024	0.63913
28	-0.005369659	-0.54184959	0.30274947	0.50778656	0.14088204	0.70400562	1.14852
29	0.272515555	-0.11746390	-1.27768833	0.19991147	-0.13832846	1.30998514	-1.57573
30	-1.586440707	0.98728616	0.46381319	-1.60689782	-1.47446456	0.15149488	-0.87613
...	...	...	...	...	...	...	...
57	0.29868247	NA	NA	0.306885022	-0.08640090	-1.96646296	-0.74395
58	-0.71482462	NA	NA	NA	NA	0.47824854	1.28715
59	-2.95117406	0.70436237	-0.50760240	NA	NA	0.90896928	0.62623
60	-0.24197749	-0.29260720	-0.02441122	NA	NA	0.65647782	-0.09915
61	-0.16200257	-0.16461786	0.75574120	NA	NA	0.05940976	1.14207
62	0.26072202	-0.66310265	0.51414561	NA	NA	0.95946758	-2.05933
63	0.47190004	1.31062764	-0.73409826	NA	NA	0.40695684	1.09371
64	-1.06678799	-0.12420018	1.53589361	NA	NA	-1.54168320	-0.41188
65	-0.72661815	0.39449345	-1.36828667	-0.001642342	-0.19504934	-4.20918129	0.34252

在后续研究中，我们重点研究 `factor1` 和 `factor2` 这两个响应变量。

1.1.3 区分训练集和测试集

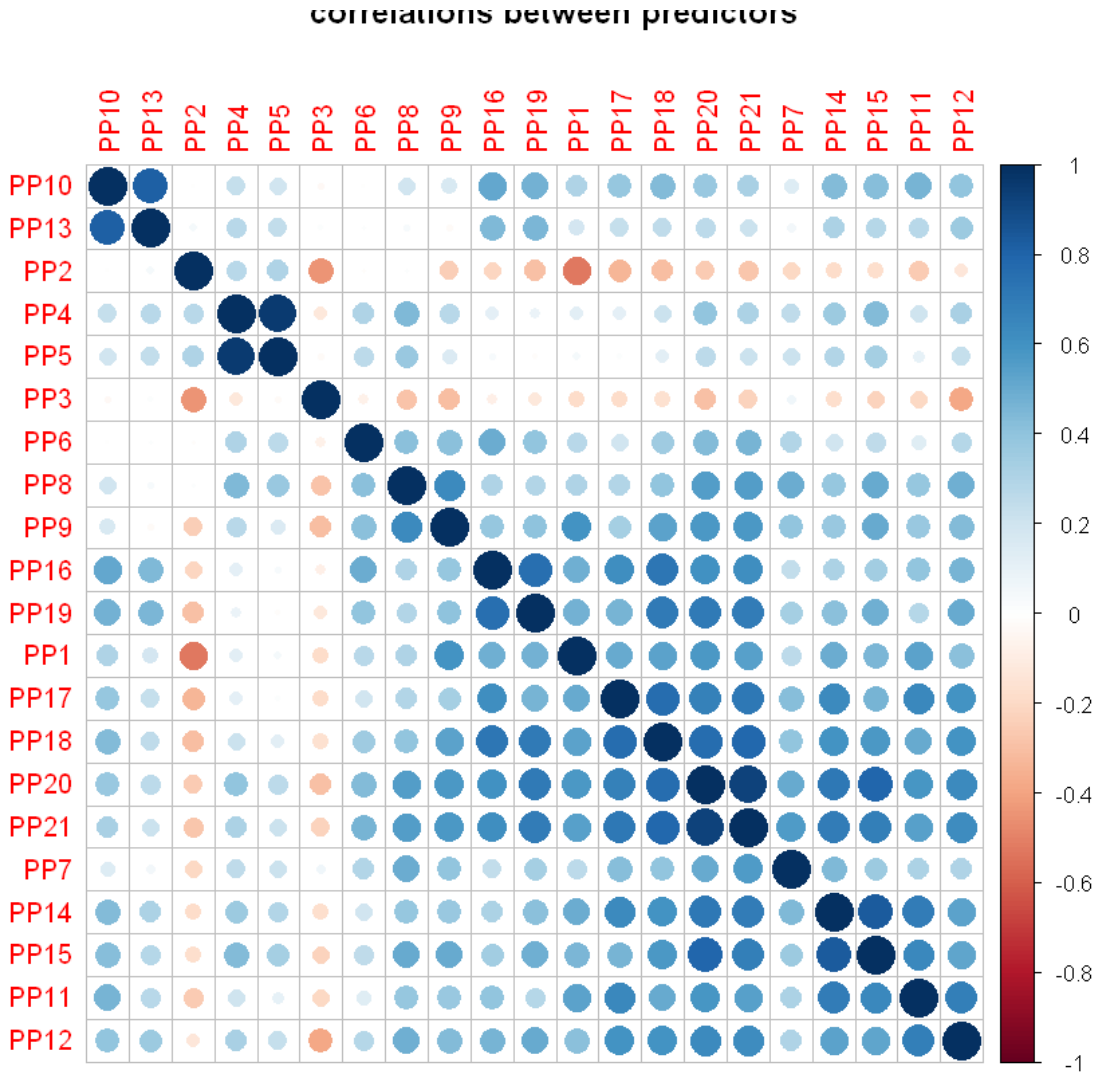
再进一步研究自变量之前，我们应该将数据分为训练集和测试集，并在最终预测前将测试集严格置于“黑盒子”中。按照作业要求，我们取前 10 个样本作为训练集。

```
In [266]: powder.scaled.test<-head(powder.scaled,10)
          powder.scaled.train<-powder.scaled[-c(1:10),]
          powder.train<-head(powder,10)
          powder.test<-powder[-c(1:10),]
```

1.1.4 自变量之间的相关性

我们再来看看自变量之间的相关矩阵图（按照 hierachical 聚类排序，由于自变量中数据有缺失，暂时先考虑完全数据）：

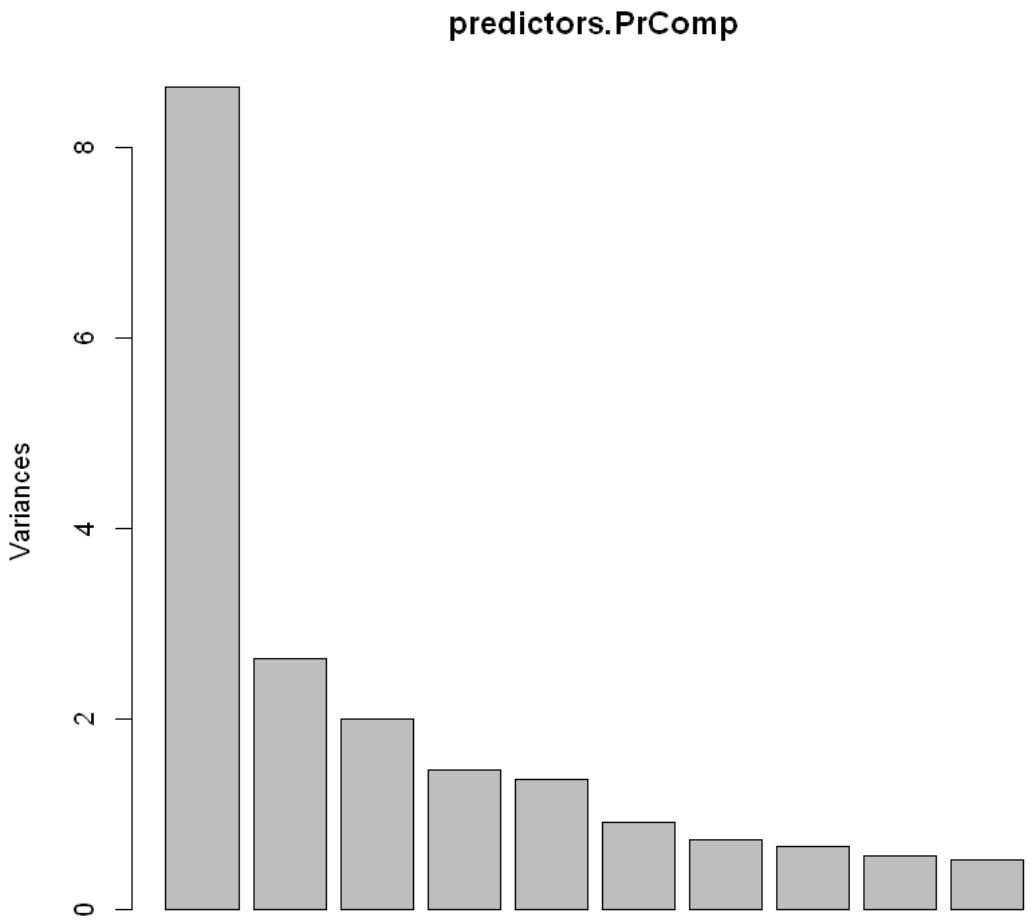
```
In [11]: corrplot::corrplot(correlations[["predictors"]], order = "hclust",
                            main = "correlations between predictors")
```

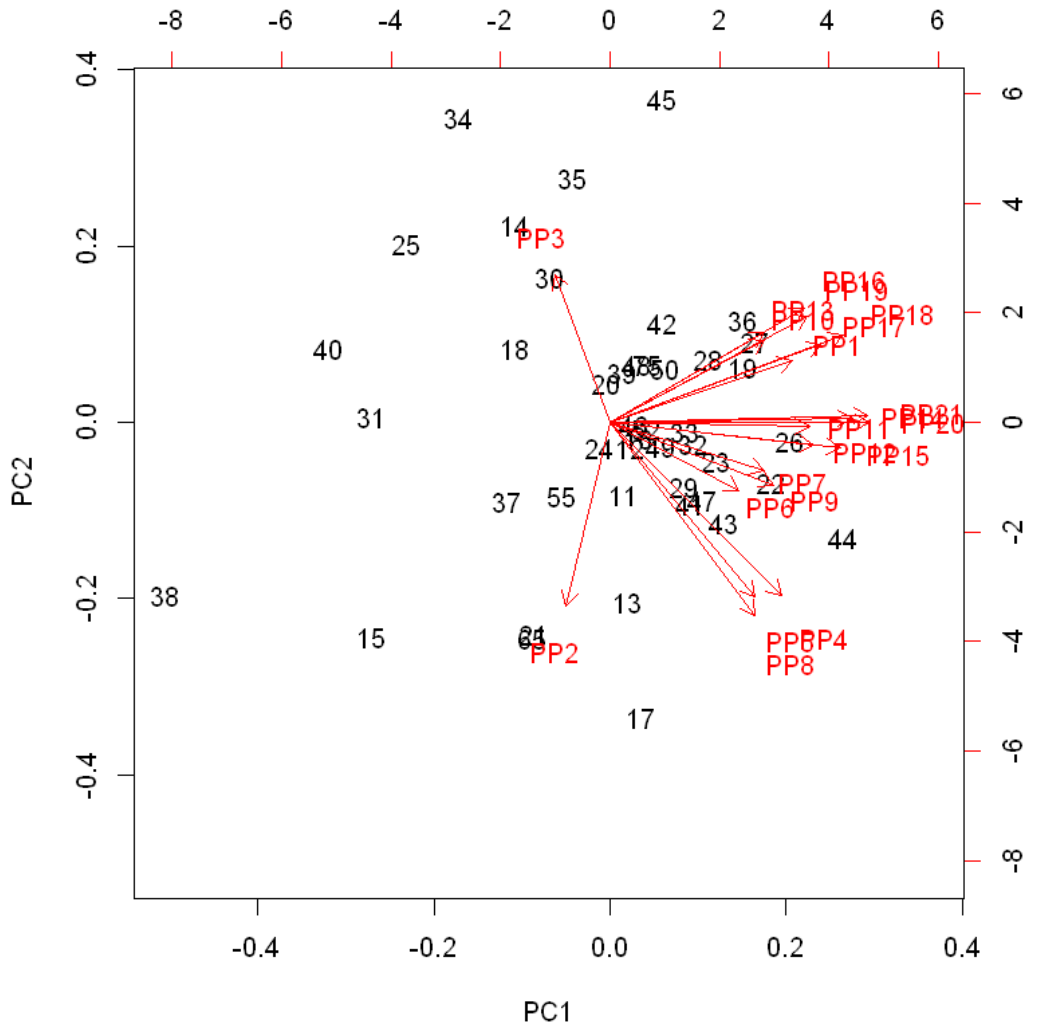


再看看 PCA：

```
In [12]: predictors.PrComp <- prcomp(na.omit(powder.scaled.train[predictors.index]), center = T
         plot(predictors.PrComp)
         head(predictors.PrComp$sd ^ 2 / sum(predictors.PrComp$sd ^ 2))
         biplot(predictors.PrComp)
```

1. 0.411181938879826 2. 0.125313079842069 3. 0.095134812610799 4. 0.0697969244871203 5. 0.0652211611611012 6. 0.0437654092773439


predictors.PrComp
Variances
0
2
4
6
8



与其他自变量相比，PP2 和 PP3 分别代表一种很不同的属性；PP4 和 PP5 很接近，与其他属性不同；其余属性比较相似。

1.1.5 自变量共线性

在拟合模型之前，需要考虑自变量的共线性问题。可以看看设计矩阵的条件数 κ，以及共线性的指标 “VIF”：

```
In [13]: OLS.colin<-lm(reformulate(grep("PP", names(powder), value = T),
                                   response = "factor1"), powder.scaled.train, na.action = na.omi
```

```
kappa(OLS.colin)
head(sort(car::vif(OLS.colin),decreasing = T))
```

32.1468900385313

PP4 27.7745558063742 **PP5** 26.0130998706372 **PP20** 22.917861809114 **PP21** 20.4554266632406 **PP14** 11.7521543932274 **PP19** 10.679650101983

设计矩阵的条件数并不小，这警示我们如果用全模型直接线性拟合会有严重的共线性问题。从各个自变量的 VIF 值中，PP4、PP5、PP20、PP21 的值很大，这从之前的相关矩阵图中已经可以看出来了。

1.1.6 典型变量分析（CCA）

Multiple Outcomes 的情形特别适合用典型变量分析，“CCA” (canonical correlation analysis)，即分别在 **X** 和 **Y** 中各自找一个代表，即各自变量的线性组合 $\mathbf{X}v_m$ 和 $\mathbf{Y}u_m$ 使得相关系数：

$$\text{Corr}^2(\mathbf{Y}u_m, \mathbf{X}v_m) \tag{1}$$

达到最大。

```
In [138]: # powder.scaled.cc.CCA <- CCA::cc(
          #     powder.scaled.train[complete.cases(powder.scaled), predictors.index],
          #     powder.scaled.train[complete.cases(powder.scaled), responses.index])
          # powder.scaled.cc.CCA$cor
          # with(powder.scaled.cc.CCA,
          #      biplot(cbind(scores$xscores[,1],scores$xscores[,2]),cbind(xcoef[,1],xcoef[,2]
          # with(powder.scaled.cc.CCA,
          #      biplot(cbind(scores$yscores[,1],scores$yscores[,2]),cbind(ycoef[,1],ycoef[,2]
          # powder.scaled.cc.CCA
```

1.2 缺失数据填补

在进一步拟合模型之前，还有一个重要的问题——-缺失数据的处理。将缺失数据所在行直接扔掉的做法是难以接受的：即使 MCAR（缺失与结果无关）的假设成立，在样本量已经如此小的情况下也会导致丢失太多信息。所以考虑对缺失数据作填补（imputation）。

众所周知，填补的方法不胜枚举，分别有不同的假设，适应不同的场合。这里主要考虑用 “pmm”（predictive mean matching），做多重填补，最后取多个填补后的数据的推断和预测的平均。

```
In [66]: # library(mice)
         # pred_matrix<- 1 - diag(1, ncol(powder.scaled.train))
         # pred_matrix[,1]<-0
```

```
# pred_matrix[,(1+length(predictors.index)+1):ncol(powder.scaled.train)]<- 0
# powder.imputed.pmm<-mice(powder.scaled.train,predictorMatrix = pred_matrix ,maxit =
# summary(powder.imputed.pmm)
```

In [67]:
```
# powder.imputed.pmm$imp$PP2 # 查看填补的值，多重填补，填补了五次
```

In [68]:
```
# fit<-with(data=powder.imputed.pmm, exp=lm(reformulate(grep("PP", names(powder), val
#                                                    response = "Cluster1")))
# combine<-pool(fit)
# summary(combine)
```

In [69]:
```
# powder.imputed.pmm.2 <- complete(powder.imputed.pmm,2)
```

1.2.1 KNN Impute

第二种方法填补方法是 “k-nearest neighbors”

In [14]:
```
powder.imputed.knn <- powder.scaled.train[predictors.index]
powder.imputed.knn<-t(impute::impute.knn(t(powder.scaled.train[predictors.index]))[['
tail(powder.imputed.knn)
```

	PP1	PP2	PP3	PP4	PP5	PP6	PP7	PP8
81	0.4925387	-0.8045645	-0.3062727	1.2432637	1.2432637	0.7129171	0.3586483	-0.24788446
82	0.2422946	0.5022739	0.8765390	0.1313113	0.1313113	-0.5406288	-0.5086005	-0.20355603
83	-0.4667918	0.1039439	0.4385650	0.4395125	0.4395125	-0.5138944	-1.6079754	0.02982012
84	0.1719019	0.1977186	0.4596904	0.5876713	0.5876713	-0.6208320	0.2522572	-0.01059698
85	-1.4537634	-0.5351133	0.7507080	0.2274169	0.2274169	1.0099659	0.7971087	0.53307818
86	-0.4450475	-0.3532337	0.6601096	0.4115617	0.4115617	-0.8436186	0.7132853	-0.49038705

1.2.2 （非条件）均值填补

In [15]:
```
# 因为 X 已经中心化了，所以填补 0
powder.imputed.mean<-powder.scaled.train[predictors.index]
for (PP in paste("PP",2:5,sep="")){
    powder.imputed.mean[is.na(powder.imputed.mean[PP]),PP]<-0
}
tail(powder.imputed.mean)
```

	PP1	PP2	PP3	PP4	PP5	PP6	PP7	PP8	PP9
81	0.4925387	-0.8045645	-0.3062727	0	0	0.7129171	0.3586483	-0.24788446	-0.4220135
82	0.2422946	0.5022739	0.8765390	0	0	-0.5406288	-0.5086005	-0.20355603	-0.3831310
83	-0.4667918	0.0000000	0.0000000	0	0	-0.5138944	-1.6079754	0.02982012	-0.1687516
84	0.1719019	0.0000000	0.0000000	0	0	-0.6208320	0.2522572	-0.01059698	-0.5470681
85	-1.4537634	-0.5351133	0.7507080	0	0	1.0099659	0.7971087	0.53307818	-0.9421986
86	-0.4450475	-0.3532337	0.6601096	0	0	-0.8436186	0.7132853	-0.49038705	-0.7036491

1.2.3 条件均值填补（线性回归）

In [271]:
```
powder.imputed.reg<-powder.scaled.train[predictors.index]
PP4.impute.reg<-lm(PP4~.-PP2-PP3-PP5 , data=powder.scaled.train[predictors.index],sub
# powder.scaled.train$PP4[which(is.na(powder.scaled.train$PP4))]
# which(is.na(powder.scaled.train$PP4))
powder.imputed.reg$PP4[which(is.na(powder.scaled.train$PP4))]<- predict(
    PP4.impute.reg,newdata=powder.scaled.train[which(is.na(powder.scaled.train$PP4))

PP5.impute.reg<-lm(PP5~.-PP2-PP3-PP4 , data=powder.scaled.train[predictors.index],sub
powder.imputed.reg$PP5[which(is.na(powder.scaled.train$PP5))]<- predict(
    PP5.impute.reg,newdata=powder.scaled.train[which(is.na(powder.scaled.train$PP5))
PP2.impute.reg<-lm(PP2~.-PP3-PP5-PP4 , data=powder.scaled.train[predictors.index],sub
powder.imputed.reg$PP2[which(is.na(powder.scaled.train$PP2))]<- predict(
    PP2.impute.reg,newdata=powder.scaled.train[which(is.na(powder.scaled.train$PP2))
PP3.impute.reg<-lm(PP3~.-PP2-PP5-PP4 , data=powder.scaled.train[predictors.index],sub
powder.imputed.reg$PP3[which(is.na(powder.scaled.train$PP3))]<- predict(
    PP3.impute.reg,newdata=powder.scaled.train[which(is.na(powder.scaled.train$PP3))

tail(powder.imputed.reg)
```

	PP1	PP2	PP3	PP4	PP5	PP6	PP7	PP8
81	0.4925387	-0.8045645	-0.3062727	0.8839818	0.9300455	0.7129171	0.3586483	-0.24788446
82	0.2422946	0.5022739	0.8765390	-0.1622114	0.2227270	-0.5406288	-0.5086005	-0.20355603
83	-0.4667918	-0.1175576	0.1975615	-0.7661209	-0.7383249	-0.5138944	-1.6079754	0.02982012
84	0.1719019	-0.4313325	0.2030472	0.1670399	0.1586194	-0.6208320	0.2522572	-0.01059698
85	-1.4537634	-0.5351133	0.7507080	0.3127630	0.6850024	1.0099659	0.7971087	0.53307818
86	-0.4450475	-0.3532337	0.6601096	0.3972799	0.3924339	-0.8436186	0.7132853	-0.49038705

PP1 0.0337206786632923 **PP2** -0.118542138755545 **PP3** -0.0401502653098411 **PP4** -0.14450841863385 **PP5** -0.211755687730749 **PP6** 0.00566737814382556 **PP7** 0.0718843446512443

PP8 0.0281389304416211 **PP9** 0.0104370813296182 **PP10** -0.0249348123744458 **PP11** 0.0812087832884899 **PP12** 0.0295722777701949 **PP13** -0.0945648000682777 **PP14** 0.0750863777939325 **PP15** 0.0182355242097546 **PP16** 0.0261864497803607 **PP17** 0.109763838900697 **PP18** 0.0793015938121416 **PP19** 0.0499381305937397 **PP20** 0.0561526979171524 **PP21** 0.0835282834068926

为简明起见，后续的分析统一采用中心化、标准化、knn 填补后的数据，即 `powder.imputed.knn`。

响应变量统一采用降维后得到的 `Cluster1` 和 `Cluster2`。

之后我们会看到，通过 CV 评价，在各种方法下 `knn.impute` 的表现都不错。

2 建立统计模型

2.1 评价标准：10-fold CV MSE

由于样本量小，选择了 10-fold CV 的 RMSE（或 MSE）作为评价预测能力好坏的指标。

packagecaret 中的 `train` 函数可以很方便地整合、对比不同方法的训练和预测结果。比如，我们采取对所有自变量的简单线性回归：

```
In [272]: library(caret)
          library(e1071)
          library(MASS)
          library(RANN)

          k<-10
          set.seed(1)
          index<-createFolds(powder.scaled.train$Treatment,k=k,returnTrain = TRUE)
          repeats<- 5
          X<-powder.imputed.reg
          y<-powder.scaled.train$factor1 # 响应函数函数
          trControl<-trainControl(method = "cv",number = k, index = index)
          sd(y)
```

0.890054069226133

2.1.1 Ordinary Least Square

```
In [273]: OLS_knn <- train(x = powder.imputed.knn,y=y,
                           #preProcess="pca",
                           method = "lm",
```

```
                    trControl = trControl)
          OLS <- train(x = powder.imputed.reg,y=y,
                    #preProcess="pca",
                    method = "lm",
                    trControl = trControl)
          OLS_mean <- train(x = powder.imputed.mean,y=y,
                    #preProcess="pca",
                    method = "lm",
                    trControl = trControl)
          OLS_knn;OLS;OLS_mean
          summary(OLS$finalModel)
```

```
Linear Regression

76 samples
21 predictors

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 69, 68, 69, 68, 68, 68, ...
Resampling results:

  RMSE       Rsquared
  0.6777366  0.4894324

Tuning parameter 'intercept' was held constant at a value of TRUE



Linear Regression

76 samples
21 predictors

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 69, 68, 69, 68, 68, 68, ...
Resampling results:
```

```
  RMSE      Rsquared
  0.657831  0.5342063

Tuning parameter 'intercept' was held constant at a value of TRUE



Linear Regression

76 samples
21 predictors

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 69, 68, 69, 68, 68, 68, ...
Resampling results:

  RMSE       Rsquared
  0.6931214  0.4669466

Tuning parameter 'intercept' was held constant at a value of TRUE




Call:
lm(formula = .outcome ~ ., data = dat)

Residuals:
     Min       1Q   Median       3Q      Max
-1.84746 -0.32163  0.06304  0.32345  0.98872

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.040764   0.077183   0.528 0.599564
PP1          0.115661   0.135583   0.853 0.397391
PP2         -0.457052   0.137412  -3.326 0.001589 **
```

```
PP3         -0.493418   0.128812  -3.831 0.000335 ***
PP4          0.354660   0.427724   0.829 0.410653
PP5         -0.330189   0.451687  -0.731 0.467931
PP6         -0.086540   0.085696  -1.010 0.317074
PP7          0.085152   0.087886   0.969 0.336921
PP8         -0.156722   0.134232  -1.168 0.248120
PP9          0.210453   0.154258   1.364 0.178133
PP10         0.361225   0.127497   2.833 0.006466 **
PP11        -0.158126   0.158471  -0.998 0.322817
PP12         0.003543   0.149725   0.024 0.981207
PP13        -0.283981   0.126063  -2.253 0.028365 *
PP14        -0.174153   0.170454  -1.022 0.311476
PP15         0.156737   0.156307   1.003 0.320453
PP16         0.133697   0.191252   0.699 0.487513
PP17         0.172703   0.202311   0.854 0.397069
PP18        -0.047867   0.184314  -0.260 0.796079
PP19         0.068731   0.185671   0.370 0.712701
PP20        -0.276951   0.253274  -1.093 0.279035
PP21         0.072731   0.222794   0.326 0.745347
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.5613 on 54 degrees of freedom
Multiple R-squared:  0.7136,Adjusted R-squared:  0.6023
F-statistic: 6.409 on 21 and 54 DF,  p-value: 1.855e-08
```

```
In [300]: OLS.restricted<-train(x = powder.imputed.reg[,paste("PP",c(2,3,9,10,13),sep="")],y=y
                        #preProcess="pca",
                        method = "lm",
                        trControl = trControl)
          # OLS.restricted<-train(x = powder.imputed.reg[,paste("PP",c(9,11,13),sep="")],y=y,
          #               #preProcess="pca",
          #               method = "lm",
          #               trControl = trControl)
          OLS.restricted
```

```
summary(OLS.restricted$finalModel)
anova(OLS.restricted$finalModel,OLS$finalModel)
```

```
Linear Regression

76 samples
 5 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 69, 68, 69, 68, 68, 68, ...
Resampling results:

  RMSE       Rsquared
  0.5866794  0.5769596

Tuning parameter 'intercept' was held constant at a value of TRUE
```

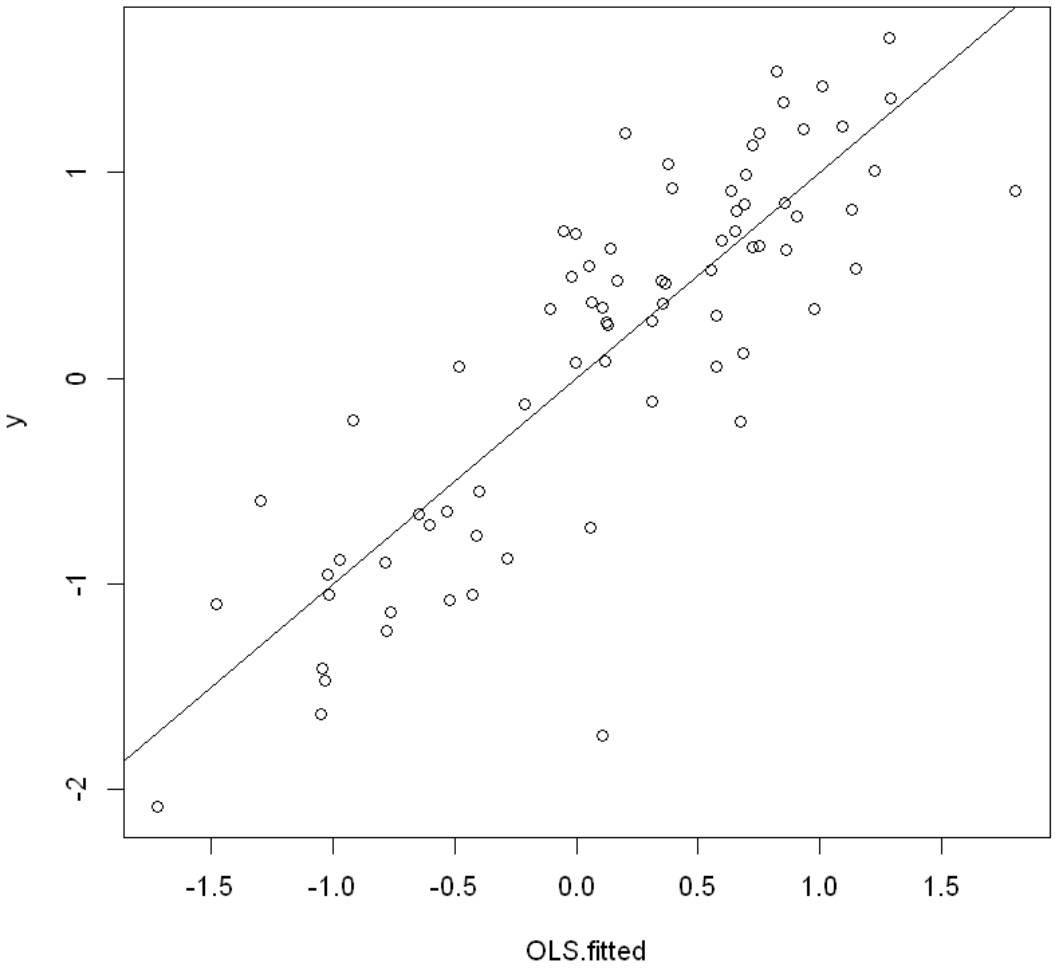
```
Call:
lm(formula = .outcome ~ ., data = dat)

Residuals:
     Min       1Q   Median       3Q      Max
-2.10852 -0.38915  0.04077  0.40667  0.94798

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.03734    0.06944   0.538    0.592
PP2         -0.61588    0.08841  -6.966 1.44e-09 ***
PP3         -0.55709    0.09384  -5.936 1.01e-07 ***
PP9          0.12629    0.08016   1.575    0.120
PP10         0.45000    0.10380   4.335 4.78e-05 ***
PP13        -0.26903    0.10425  -2.580    0.012 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 0.5909 on 70 degrees of freedom
Multiple R-squared:  0.5886,Adjusted R-squared:  0.5592
F-statistic: 20.03 on 5 and 70 DF,  p-value: 2.409e-12
```

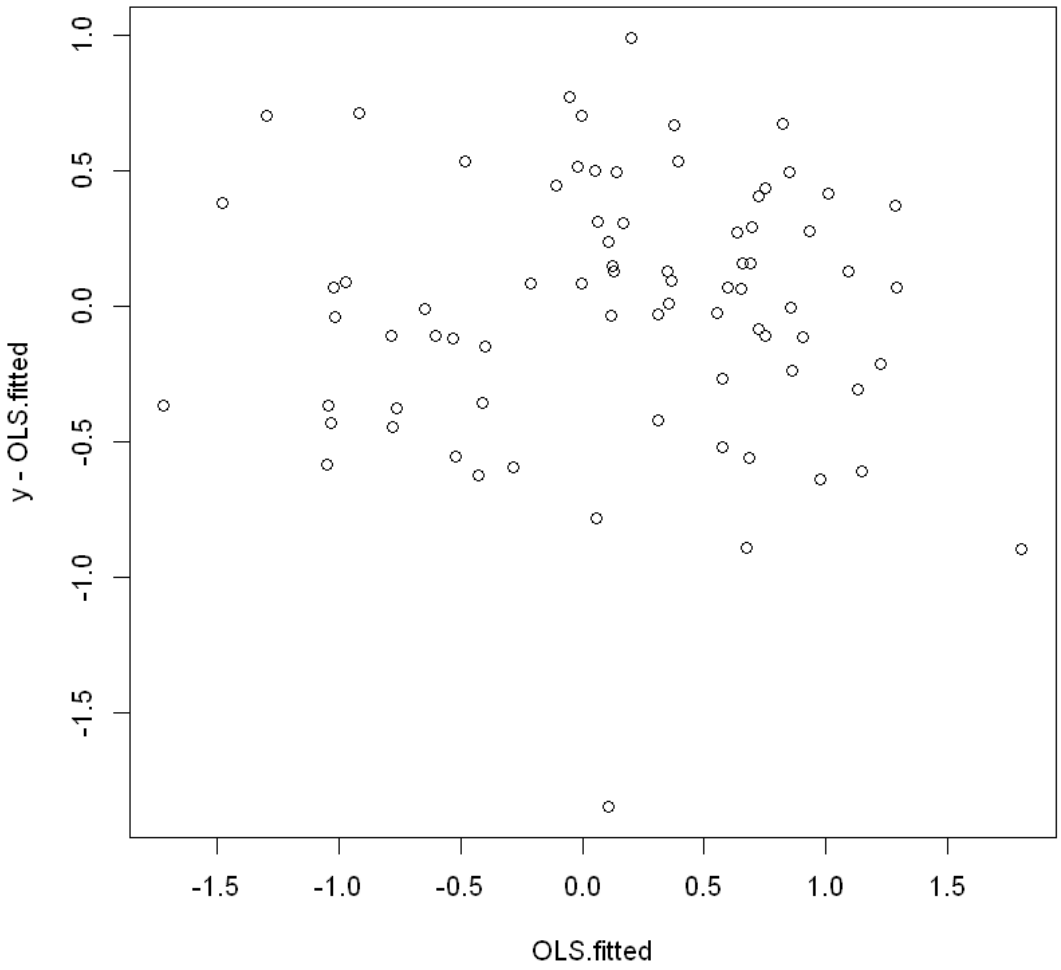
Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
70	24.44319	NA	NA	NA	NA
54	17.01346	16	7.429737	1.473854	0.1443544

残差诊断

```
In [275]: OLS.fitted<-predict(OLS$finalModel,newdata=X[,predictors.index])
          plot(OLS.fitted,y)
          abline(a = 0, b = 1)
          plot(OLS.fitted,y-OLS.fitted) # 残差依然和 y 高度相关
          sqrt(mean((y-OLS.fitted)^2))
```



0.47313982251825



2.1.2 Stepwise Regression

```
In [276]: lmStepAIC <- train(x = X, y = y,
                             method = "lmStepAIC",trace=F,
                             #preProcess = "pca",
                             trControl = trControl)
          lmStepAIC
          summary(lmStepAIC$finalModel)
          lmStepAIC$finalModel$coef
```

```
Linear Regression with Stepwise Selection

76 samples
21 predictors

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 69, 68, 69, 68, 68, 68, ...
Resampling results:

  RMSE       Rsquared
  0.6476043  0.511504




Call:
lm(formula = .outcome ~ PP2 + PP3 + PP6 + PP9 + PP10 + PP11 +
    PP13 + PP16, data = dat)

Residuals:
     Min       1Q   Median       3Q      Max
-1.75027 -0.27904  0.00902  0.34382  1.33366

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.04847    0.06244   0.776 0.440338
PP2         -0.53447    0.08560  -6.244 3.32e-08 ***
PP3         -0.51969    0.08865  -5.862 1.54e-07 ***
PP6         -0.15320    0.06810  -2.250 0.027758 *
PP9          0.20536    0.07939   2.587 0.011869 *
PP10         0.38081    0.10350   3.679 0.000468 ***
PP11        -0.17192    0.07345  -2.341 0.022233 *
PP13        -0.31637    0.09531  -3.320 0.001461 **
PP16         0.29874    0.08059   3.707 0.000428 ***
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.5301 on 67 degrees of freedom
Multiple R-squared:  0.6831,Adjusted R-squared:  0.6453
F-statistic: 18.05 on 8 and 67 DF,  p-value: 4.725e-14
```

(Intercept) 0.0484690181351115 **PP2** -0.534469377510599 **PP3** -0.519689441753931 **PP6** -0.153202095633269 **PP9** 0.205355466500618 **PP10** 0.380810291755392 **PP11** -0.171920427540851 **PP13** -0.316368115768644 **PP16** 0.298735432757274

```
In [277]: lmStepBIC <- train(x = X, y = y,
                             method = "lmStepAIC",trace=F,k=log(nrow(X)),
                             #preProcess = "pca",
                             trControl = trControl)
          lmStepBIC
          summary(lmStepBIC$finalModel)
```

```
Linear Regression with Stepwise Selection

76 samples
21 predictors

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 69, 68, 69, 68, 68, 68, ...
Resampling results:

  RMSE       Rsquared
  0.6644794  0.5038216
```

```
Call:
lm(formula = .outcome ~ PP2 + PP3 + PP6 + PP9 + PP10 + PP11 +
    PP13 + PP16, data = dat)
```

```
Residuals:
     Min       1Q   Median       3Q      Max
-1.75027 -0.27904  0.00902  0.34382  1.33366

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.04847    0.06244   0.776 0.440338
PP2         -0.53447    0.08560  -6.244 3.32e-08 ***
PP3         -0.51969    0.08865  -5.862 1.54e-07 ***
PP6         -0.15320    0.06810  -2.250 0.027758 *
PP9          0.20536    0.07939   2.587 0.011869 *
PP10         0.38081    0.10350   3.679 0.000468 ***
PP11        -0.17192    0.07345  -2.341 0.022233 *
PP13        -0.31637    0.09531  -3.320 0.001461 **
PP16         0.29874    0.08059   3.707 0.000428 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.5301 on 67 degrees of freedom
Multiple R-squared:  0.6831,Adjusted R-squared:  0.6453
F-statistic: 18.05 on 8 and 67 DF,  p-value: 4.725e-14
```

2.1.3 Lasso

```
In [278]: lasso <- train(x = X, y = y,
                         method = "lasso",
                         #preProcess = "pca",
                         tuneLength=10,
                         trControl = trControl)
          lasso_knn <- train(x = powder.imputed.knn, y = y,
                         method = "lasso",
                         #preProcess = "pca",
                         tuneLength=10,
                         trControl = trControl)
```

```
          lasso_mean <- train(x = powder.imputed.mean, y = y,
                              method = "lasso",
                              #preProcess = "pca",
                              tuneLength=10,
                              trControl = trControl)
          lasso;lasso_knn;lasso_mean
```

```
The lasso

76 samples
21 predictors

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 69, 68, 69, 68, 68, 68, ...
Resampling results across tuning parameters:

  fraction   RMSE       Rsquared
  0.1000000  0.7444065  0.2897148
  0.1888889  0.7015711  0.3654956
  0.2777778  0.6642921  0.4534231
  0.3666667  0.6387738  0.5002666
  0.4555556  0.6216709  0.5250539
  0.5444444  0.6117991  0.5474532
  0.6333333  0.6166779  0.5512551
  0.7222222  0.6290299  0.5454848
  0.8111111  0.6396419  0.5437847
  0.9000000  0.6501354  0.5384516

RMSE was used to select the optimal model using  the smallest value.
The final value used for the model was fraction = 0.5444444.


The lasso

76 samples
21 predictors
```

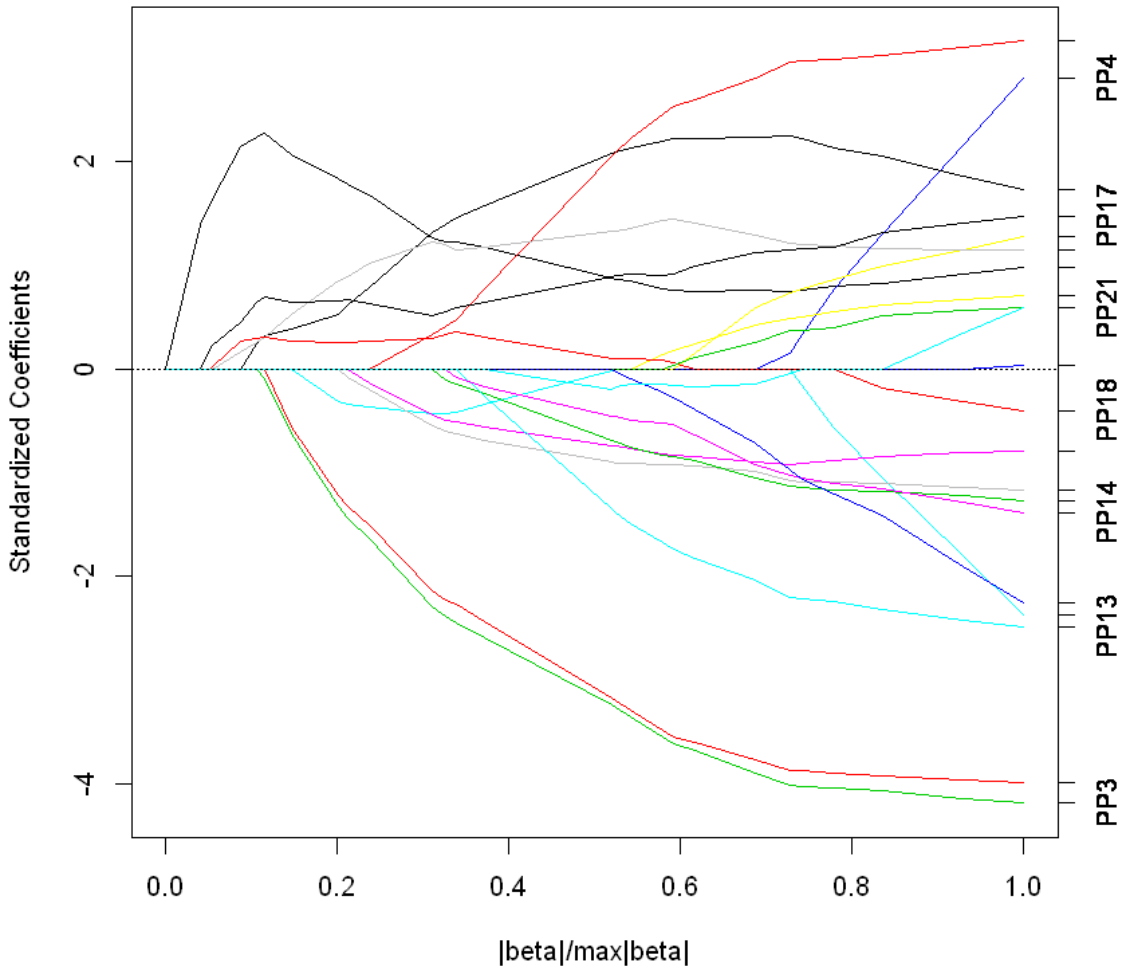
```
No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 69, 68, 69, 68, 68, 68, ...
Resampling results across tuning parameters:

  fraction   RMSE       Rsquared
  0.1000000  0.7488708  0.2909545
  0.1888889  0.7035818  0.3578725
  0.2777778  0.6748671  0.4230219
  0.3666667  0.6527217  0.4625334
  0.4555556  0.6310152  0.4985378
  0.5444444  0.6175990  0.5270704
  0.6333333  0.6215417  0.5378406
  0.7222222  0.6365859  0.5284199
  0.8111111  0.6512579  0.5157387
  0.9000000  0.6658473  0.5009795

RMSE was used to select the optimal model using  the smallest value.
The final value used for the model was fraction = 0.5444444.


The lasso

76 samples
21 predictors

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 69, 68, 69, 68, 68, 68, ...
Resampling results across tuning parameters:

  fraction   RMSE       Rsquared
  0.1000000  0.7484312  0.2921941
  0.1888889  0.7091843  0.3502170
  0.2777778  0.6779893  0.4250487
  0.3666667  0.6570983  0.4649434
  0.4555556  0.6375415  0.4980709
```

```
  0.5444444  0.6285963  0.5204677
  0.6333333  0.6409407  0.5182097
  0.7222222  0.6584152  0.5029009
  0.8111111  0.6737830  0.4854616
  0.9000000  0.6846344  0.4745687

RMSE was used to select the optimal model using  the smallest value.
The final value used for the model was fraction = 0.5444444.
```

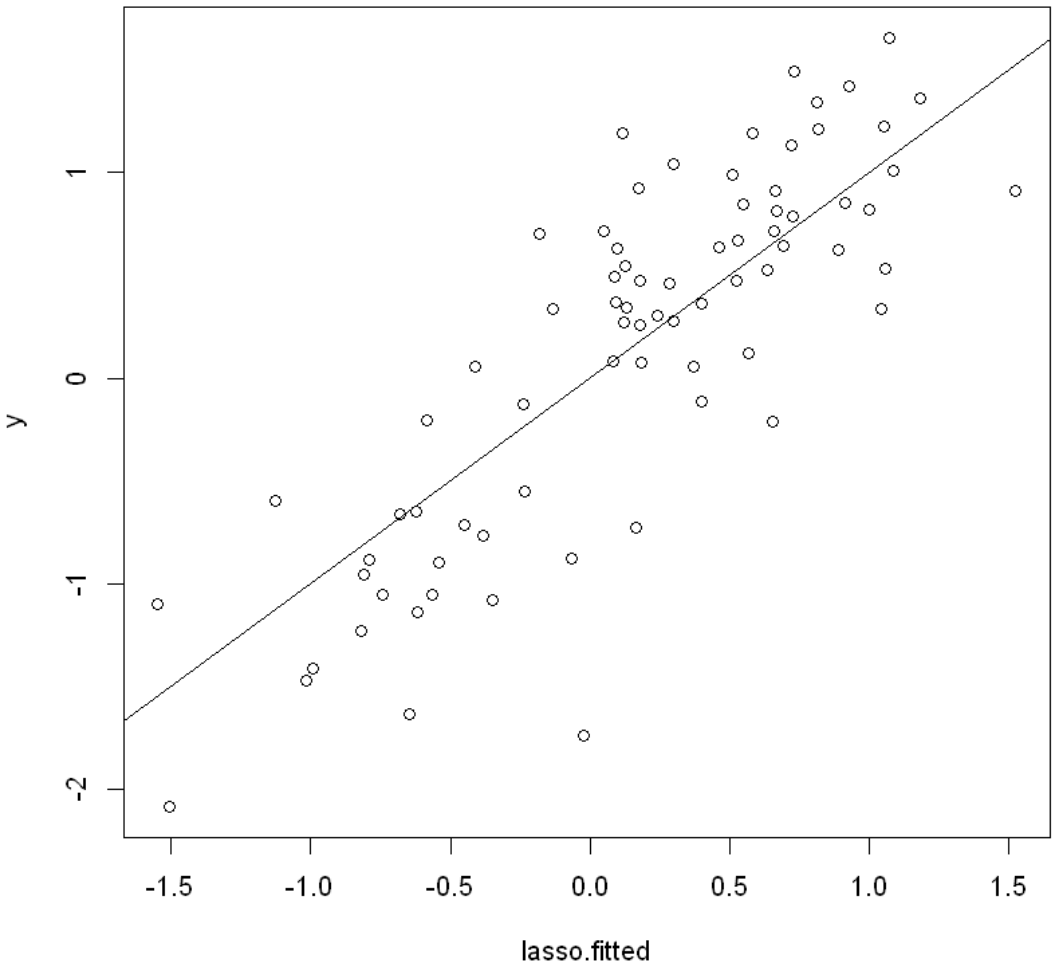
```
In [279]: plot(lasso$finalModel,use.color =T)
          lasso.finalCoef<-predict(lasso$finalModel,type="coefficients",mode="fraction",s=lasso
          lasso.finalCoef[order(abs(lasso.finalCoef),decreasing = T)]
          head(lasso.finalCoef[order(abs(lasso.finalCoef),decreasing = T)])
```

PP3 -0.397736580322586 **PP2** -0.378798145713308 **PP9** 0.260120644141786 **PP10** 0.255429316282108 **PP13** -0.171511292949851 **PP16** 0.157486521259984 **PP8** -0.121721167701723 **PP17** 0.107717774125997 **PP1** 0.0987385031284641 **PP11** -0.0945033130651558 **PP6** -0.0840983788019458 **PP14** -0.0621104551082043 **PP21** -0.0186086857589806 **PP20** -0.0125362494523121 **PP18** 0.0110607322940431 **PP7** 0.00136021833002865 **PP4** 0 **PP5** 0 **PP12** 0 **PP15** 0 **PP19** 0

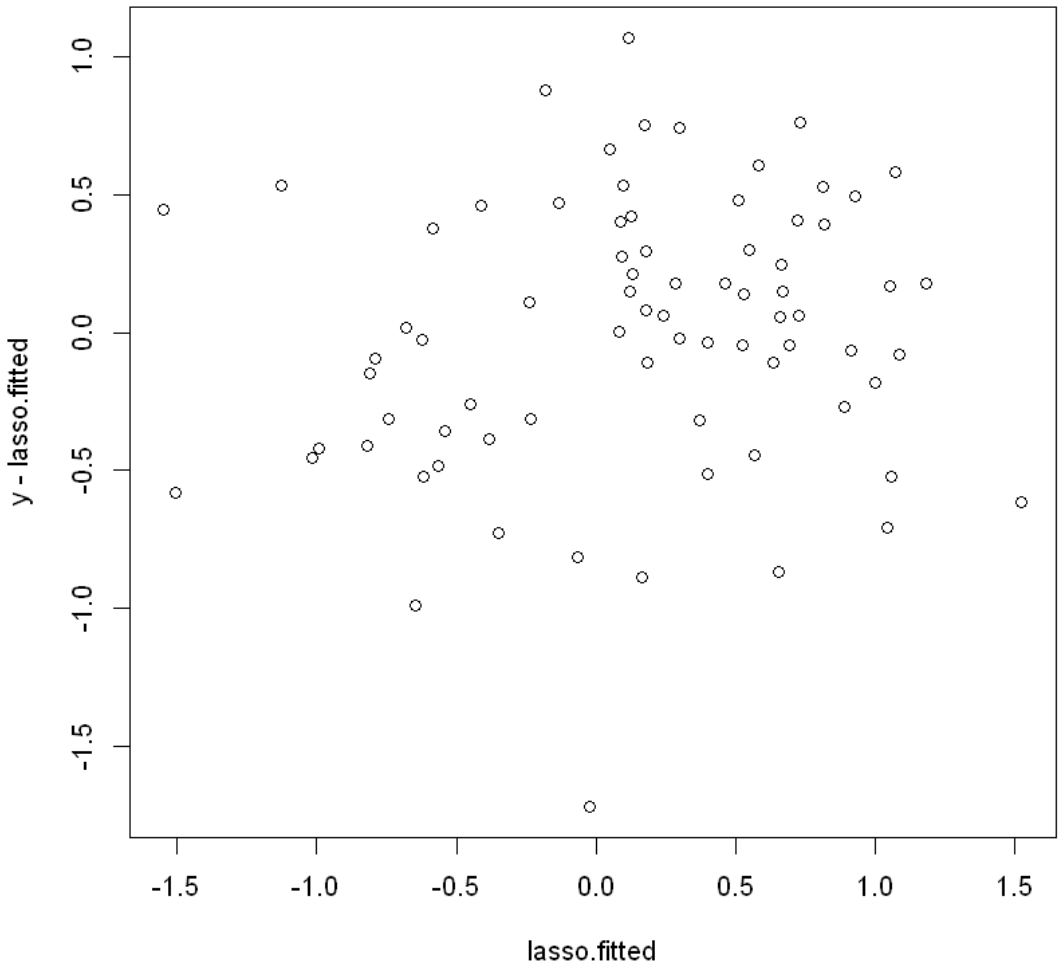
PP3 -0.397736580322586 **PP2** -0.378798145713308 **PP9** 0.260120644141786 **PP10** 0.255429316282108 **PP13** -0.171511292949851 **PP16** 0.157486521259984



In [280]:
```
lasso.fitted<-predict(lasso$finalModel,newx=X, mode="fraction",s=lasso$bestTune$fract
plot(lasso.fitted,y)
abline(a = 0, b = 1)
plot(lasso.fitted,y-lasso.fitted) # 残差依然和 y 高度相关
sqrt(mean((y-lasso.fitted)^2))
```



0.494605618709581



2.1.4 Elastic Net

```
In [281]: enet <- train(x = X, y = y,
                        method = "enet",
                        #preProcess = "pca",
                        tuneLength=20,
                        trControl = trControl)
          enet_knn <- train(x = powder.imputed.knn,y=y,
                        #preProcess="pca",
                        method = "enet",
```

```
                        tuneLength=20,
                    trControl = trControl)
enet_mean <- train(x = powder.imputed.mean,y=y,
                    #preProcess="pca",
                    method = "enet",
                          tuneLength=20,
                    trControl = trControl)
enet;enet_knn;enet_mean
```

```
Elasticnet

76 samples
21 predictors

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 69, 68, 69, 68, 68, 68, ...
Resampling results across tuning parameters:

  lambda        fraction  RMSE       Rsquared
  0.0000000000  0.05      0.7799508  0.3002581
  0.0000000000  0.10      0.7444065  0.2897148
  0.0000000000  0.15      0.7231235  0.3159492
  0.0000000000  0.20      0.6962384  0.3789190
  0.0000000000  0.25      0.6721653  0.4335752
  0.0000000000  0.30      0.6578788  0.4678598
  0.0000000000  0.35      0.6435896  0.4928469
  0.0000000000  0.40      0.6313195  0.5106263
  0.0000000000  0.45      0.6226081  0.5236357
  0.0000000000  0.50      0.6147802  0.5376969
  0.0000000000  0.55      0.6115026  0.5485073
  0.0000000000  0.60      0.6122556  0.5531602
  0.0000000000  0.65      0.6190827  0.5498490
  0.0000000000  0.70      0.6267689  0.5457550
  0.0000000000  0.75      0.6317373  0.5456676
  0.0000000000  0.80      0.6382734  0.5440755
  0.0000000000  0.85      0.6441931  0.5423428
```

0.0000000000	0.90	0.6501354	0.5384516
0.0000000000	0.95	0.6542811	0.5359203
0.0000000000	1.00	0.6578310	0.5342063
0.0001000000	0.05	0.7800744	0.3002690
0.0001000000	0.10	0.7444525	0.2897875
0.0001000000	0.15	0.7232808	0.3156499
0.0001000000	0.20	0.6964286	0.3784869
0.0001000000	0.25	0.6723613	0.4330789
0.0001000000	0.30	0.6580874	0.4674820
0.0001000000	0.35	0.6438592	0.4924614
0.0001000000	0.40	0.6315459	0.5103166
0.0001000000	0.45	0.6228203	0.5233671
0.0001000000	0.50	0.6149252	0.5374458
0.0001000000	0.55	0.6115860	0.5483377
0.0001000000	0.60	0.6121401	0.5532403
0.0001000000	0.65	0.6188831	0.5500694
0.0001000000	0.70	0.6265926	0.5459152
0.0001000000	0.75	0.6315695	0.5458039
0.0001000000	0.80	0.6380650	0.5442305
0.0001000000	0.85	0.6439971	0.5425138
0.0001000000	0.90	0.6499132	0.5386420
0.0001000000	0.95	0.6540671	0.5361093
0.0001000000	1.00	0.6576012	0.5344088
0.0001467799	0.05	0.7801317	0.3002739
0.0001467799	0.10	0.7444739	0.2898213
0.0001467799	0.15	0.7233538	0.3155112
0.0001467799	0.20	0.6965169	0.3782867
0.0001467799	0.25	0.6724524	0.4328486
0.0001467799	0.30	0.6581841	0.4673068
0.0001467799	0.35	0.6439843	0.4922825
0.0001467799	0.40	0.6316552	0.5101603
0.0001467799	0.45	0.6229190	0.5232418
0.0001467799	0.50	0.6149931	0.5373282
0.0001467799	0.55	0.6116254	0.5482576
0.0001467799	0.60	0.6120878	0.5532757
0.0001467799	0.65	0.6187916	0.5501701
0.0001467799	0.70	0.6265009	0.5459998

0.0001467799	0.75	0.6314919	0.5458670
0.0001467799	0.80	0.6379686	0.5443022
0.0001467799	0.85	0.6439065	0.5425930
0.0001467799	0.90	0.6498104	0.5387301
0.0001467799	0.95	0.6539680	0.5361969
0.0001467799	1.00	0.6574949	0.5345026
0.0002154435	0.05	0.7802151	0.3002811
0.0002154435	0.10	0.7445052	0.2898706
0.0002154435	0.15	0.7234601	0.3153093
0.0002154435	0.20	0.6966442	0.3779952
0.0002154435	0.25	0.6725853	0.4325125
0.0002154435	0.30	0.6583251	0.4670512
0.0002154435	0.35	0.6441664	0.4920216
0.0002154435	0.40	0.6318139	0.5099373
0.0002154435	0.45	0.6230631	0.5230584
0.0002154435	0.50	0.6150928	0.5371555
0.0002154435	0.55	0.6116837	0.5481395
0.0002154435	0.60	0.6120129	0.5533252
0.0002154435	0.65	0.6186597	0.5503152
0.0002154435	0.70	0.6263677	0.5461231
0.0002154435	0.75	0.6313791	0.5459588
0.0002154435	0.80	0.6378283	0.5444067
0.0002154435	0.85	0.6437748	0.5427082
0.0002154435	0.90	0.6496609	0.5388583
0.0002154435	0.95	0.6538238	0.5363246
0.0002154435	1.00	0.6573400	0.5346394
0.0003162278	0.05	0.7803362	0.3002915
0.0003162278	0.10	0.7445506	0.2899428
0.0003162278	0.15	0.7236138	0.3150202
0.0003162278	0.20	0.6968217	0.3775733
0.0003162278	0.25	0.6727788	0.4320224
0.0003162278	0.30	0.6585302	0.4666796
0.0003162278	0.35	0.6444311	0.4916423
0.0003162278	0.40	0.6320437	0.5096192
0.0003162278	0.45	0.6232716	0.5227861
0.0003162278	0.50	0.6152394	0.5369017
0.0003162278	0.55	0.6117701	0.5479645

0.0003162278	0.60	0.6119277	0.5533742
0.0003162278	0.65	0.6184707	0.5505225
0.0003162278	0.70	0.6261748	0.5463019
0.0003162278	0.75	0.6312156	0.5460919
0.0003162278	0.80	0.6376251	0.5445583
0.0003162278	0.85	0.6435839	0.5428753
0.0003162278	0.90	0.6494444	0.5390443
0.0003162278	0.95	0.6536146	0.5365099
0.0003162278	1.00	0.6571154	0.5348380
0.0004641589	0.05	0.7805117	0.3003059
0.0004641589	0.10	0.7446163	0.2900484
0.0004641589	0.15	0.7238350	0.3146078
0.0004641589	0.20	0.6970791	0.3769634
0.0004641589	0.25	0.6730596	0.4313119
0.0004641589	0.30	0.6588270	0.4661416
0.0004641589	0.35	0.6448088	0.4911027
0.0004641589	0.40	0.6323748	0.5091622
0.0004641589	0.45	0.6235736	0.5223864
0.0004641589	0.50	0.6154490	0.5365457
0.0004641589	0.55	0.6118989	0.5477048
0.0004641589	0.60	0.6118099	0.5534365
0.0004641589	0.65	0.6182029	0.5508149
0.0004641589	0.70	0.6258973	0.5465602
0.0004641589	0.75	0.6309846	0.5462767
0.0004641589	0.80	0.6373302	0.5447798
0.0004641589	0.85	0.6433091	0.5431164
0.0004641589	0.90	0.6491326	0.5393125
0.0004641589	0.95	0.6533129	0.5367777
0.0004641589	1.00	0.6567913	0.5351252
0.0006812921	0.05	0.7807832	0.3003129
0.0006812921	0.10	0.7447090	0.2902048
0.0006812921	0.15	0.7241464	0.3140291
0.0006812921	0.20	0.6973857	0.3761776
0.0006812921	0.25	0.6734562	0.4303131
0.0006812921	0.30	0.6592543	0.4653632
0.0006812921	0.35	0.6453334	0.4903611
0.0006812921	0.40	0.6328490	0.5085072

0.0006812921	0.45	0.6240120	0.5218017
0.0006812921	0.50	0.6157474	0.5360474
0.0006812921	0.55	0.6120525	0.5473314
0.0006812921	0.60	0.6116487	0.5535187
0.0006812921	0.65	0.6178280	0.5512292
0.0006812921	0.70	0.6255019	0.5469390
0.0006812921	0.75	0.6306858	0.5464790
0.0006812921	0.80	0.6369074	0.5451020
0.0006812921	0.85	0.6429126	0.5434708
0.0006812921	0.90	0.6486830	0.5397061
0.0006812921	0.95	0.6528762	0.5371716
0.0006812921	1.00	0.6563271	0.5355377
0.0010000000	0.05	0.7811678	0.3003227
0.0010000000	0.10	0.7448780	0.2904118
0.0010000000	0.15	0.7245826	0.3132221
0.0010000000	0.20	0.6977625	0.3751456
0.0010000000	0.25	0.6740721	0.4288867
0.0010000000	0.30	0.6598635	0.4642500
0.0010000000	0.35	0.6460714	0.4893108
0.0010000000	0.40	0.6335248	0.5075717
0.0010000000	0.45	0.6246428	0.5209528
0.0010000000	0.50	0.6162277	0.5353212
0.0010000000	0.55	0.6122011	0.5468980
0.0010000000	0.60	0.6115470	0.5535073
0.0010000000	0.65	0.6173075	0.5518150
0.0010000000	0.70	0.6249451	0.5474836
0.0010000000	0.75	0.6302640	0.5467882
0.0010000000	0.80	0.6363086	0.5455633
0.0010000000	0.85	0.6423496	0.5439806
0.0010000000	0.90	0.6480448	0.5402716
0.0010000000	0.95	0.6522536	0.5377397
0.0010000000	1.00	0.6556691	0.5361251
0.0014677993	0.05	0.7817056	0.3003426
0.0014677993	0.10	0.7451270	0.2906996
0.0014677993	0.15	0.7251676	0.3121568
0.0014677993	0.20	0.6982984	0.3736972
0.0014677993	0.25	0.6749711	0.4268662

0.0014677993	0.30	0.6606939	0.4625109
0.0014677993	0.35	0.6470683	0.4878589
0.0014677993	0.40	0.6344746	0.5062705
0.0014677993	0.45	0.6255258	0.5197245
0.0014677993	0.50	0.6169529	0.5341772
0.0014677993	0.55	0.6124370	0.5462373
0.0014677993	0.60	0.6115898	0.5532848
0.0014677993	0.65	0.6165560	0.5527049
0.0014677993	0.70	0.6241661	0.5482561
0.0014677993	0.75	0.6296778	0.5472295
0.0014677993	0.80	0.6354761	0.5462050
0.0014677993	0.85	0.6415679	0.5446933
0.0014677993	0.90	0.6471578	0.5410619
0.0014677993	0.95	0.6513827	0.5385379
0.0014677993	1.00	0.6547490	0.5369521
0.0021544347	0.05	0.7824433	0.3003751
0.0021544347	0.10	0.7453834	0.2912562
0.0021544347	0.15	0.7259560	0.3107385
0.0021544347	0.20	0.6990484	0.3717056
0.0021544347	0.25	0.6762200	0.4240743
0.0021544347	0.30	0.6615384	0.4605398
0.0021544347	0.35	0.6484420	0.4858464
0.0021544347	0.40	0.6357750	0.5044860
0.0021544347	0.45	0.6267598	0.5179790
0.0021544347	0.50	0.6180026	0.5325241
0.0021544347	0.55	0.6128703	0.5452007
0.0021544347	0.60	0.6117239	0.5528423
0.0021544347	0.65	0.6155485	0.5538467
0.0021544347	0.70	0.6230349	0.5492746
0.0021544347	0.75	0.6288748	0.5478421
0.0021544347	0.80	0.6343459	0.5470744
0.0021544347	0.85	0.6405074	0.5456449
0.0021544347	0.90	0.6459513	0.5421441
0.0021544347	0.95	0.6501692	0.5396547
0.0021544347	1.00	0.6534857	0.5380995
0.0031622777	0.05	0.7834298	0.3004193
0.0031622777	0.10	0.7455960	0.2922358

0.0031622777	0.15	0.7269568	0.3089506
0.0031622777	0.20	0.7000863	0.3690001
0.0031622777	0.25	0.6779024	0.4202937
0.0031622777	0.30	0.6626214	0.4579654
0.0031622777	0.35	0.6502108	0.4831128
0.0031622777	0.40	0.6376223	0.5019803
0.0031622777	0.45	0.6284063	0.5156705
0.0031622777	0.50	0.6197395	0.5300474
0.0031622777	0.55	0.6134838	0.5436911
0.0031622777	0.60	0.6120375	0.5520132
0.0031622777	0.65	0.6142705	0.5548377
0.0031622777	0.70	0.6215922	0.5504950
0.0031622777	0.75	0.6278295	0.5486262
0.0031622777	0.80	0.6328930	0.5482318
0.0031622777	0.85	0.6390733	0.5468113
0.0031622777	0.90	0.6443079	0.5436458
0.0031622777	0.95	0.6485156	0.5411815
0.0031622777	1.00	0.6517897	0.5396636
0.0046415888	0.05	0.7847243	0.3005353
0.0046415888	0.10	0.7459351	0.2935090
0.0046415888	0.15	0.7282654	0.3066492
0.0046415888	0.20	0.7014895	0.3654178
0.0046415888	0.25	0.6800412	0.4153295
0.0046415888	0.30	0.6635613	0.4552795
0.0046415888	0.35	0.6523259	0.4792973
0.0046415888	0.40	0.6400169	0.4986923
0.0046415888	0.45	0.6305164	0.5127260
0.0046415888	0.50	0.6223391	0.5263755
0.0046415888	0.55	0.6145046	0.5415477
0.0046415888	0.60	0.6126694	0.5505772
0.0046415888	0.65	0.6132879	0.5554225
0.0046415888	0.70	0.6195362	0.5527614
0.0046415888	0.75	0.6265173	0.5496198
0.0046415888	0.80	0.6310480	0.5499065
0.0046415888	0.85	0.6370156	0.5484396
0.0046415888	0.90	0.6421015	0.5457148
0.0046415888	0.95	0.6463524	0.5432142

0.0046415888	1.00	0.6495727	0.5417525
0.0068129207	0.05	0.7863435	0.3006951
0.0068129207	0.10	0.7469777	0.2942975
0.0068129207	0.15	0.7298983	0.3038552
0.0068129207	0.20	0.7033420	0.3608264
0.0068129207	0.25	0.6825500	0.4094000
0.0068129207	0.30	0.6646396	0.4518440
0.0068129207	0.35	0.6545756	0.4752730
0.0068129207	0.40	0.6430588	0.4945755
0.0068129207	0.45	0.6332036	0.5088847
0.0068129207	0.50	0.6250830	0.5221953
0.0068129207	0.55	0.6163008	0.5382542
0.0068129207	0.60	0.6132748	0.5485629
0.0068129207	0.65	0.6131967	0.5547616
0.0068129207	0.70	0.6171569	0.5552123
0.0068129207	0.75	0.6240810	0.5516812
0.0068129207	0.80	0.6286288	0.5522336
0.0068129207	0.85	0.6343343	0.5506222
0.0068129207	0.90	0.6393952	0.5483154
0.0068129207	0.95	0.6436021	0.5458598
0.0068129207	1.00	0.6467606	0.5444774
0.0100000000	0.05	0.7882802	0.3009136
0.0100000000	0.10	0.7488592	0.2944905
0.0100000000	0.15	0.7318657	0.3006396
0.0100000000	0.20	0.7057758	0.3550107
0.0100000000	0.25	0.6853036	0.4025481
0.0100000000	0.30	0.6666266	0.4464662
0.0100000000	0.35	0.6572430	0.4705401
0.0100000000	0.40	0.6464841	0.4899546
0.0100000000	0.45	0.6367256	0.5038456
0.0100000000	0.50	0.6280279	0.5179092
0.0100000000	0.55	0.6192467	0.5334762
0.0100000000	0.60	0.6142009	0.5456880
0.0100000000	0.65	0.6138086	0.5528694
0.0100000000	0.70	0.6152291	0.5563757
0.0100000000	0.75	0.6208096	0.5548163
0.0100000000	0.80	0.6257728	0.5549069

0.0100000000	0.85	0.6309106	0.5538759
0.0100000000	0.90	0.6361436	0.5515540
0.0100000000	0.95	0.6402370	0.5491905
0.0100000000	1.00	0.6433116	0.5479294
0.0146779927	0.05	0.7903685	0.3020628
0.0146779927	0.10	0.7511179	0.2949509
0.0146779927	0.15	0.7339768	0.2983396
0.0146779927	0.20	0.7088915	0.3478707
0.0146779927	0.25	0.6885618	0.3947977
0.0146779927	0.30	0.6695577	0.4388986
0.0146779927	0.35	0.6597700	0.4653073
0.0146779927	0.40	0.6503168	0.4839268
0.0146779927	0.45	0.6406334	0.4984637
0.0146779927	0.50	0.6315448	0.5127458
0.0146779927	0.55	0.6231868	0.5270294
0.0146779927	0.60	0.6162044	0.5412340
0.0146779927	0.65	0.6146177	0.5497816
0.0146779927	0.70	0.6151359	0.5551208
0.0146779927	0.75	0.6177096	0.5571811
0.0146779927	0.80	0.6227266	0.5572390
0.0146779927	0.85	0.6270522	0.5576685
0.0146779927	0.90	0.6323056	0.5558189
0.0146779927	0.95	0.6363238	0.5531727
0.0146779927	1.00	0.6392440	0.5521306
0.0215443469	0.05	0.7927072	0.3033873
0.0215443469	0.10	0.7535333	0.2956302
0.0215443469	0.15	0.7354682	0.2966924
0.0215443469	0.20	0.7124700	0.3400203
0.0215443469	0.25	0.6918867	0.3869581
0.0215443469	0.30	0.6732114	0.4301242
0.0215443469	0.35	0.6609891	0.4612653
0.0215443469	0.40	0.6538976	0.4772369
0.0215443469	0.45	0.6446551	0.4931021
0.0215443469	0.50	0.6359221	0.5065967
0.0215443469	0.55	0.6276131	0.5201503
0.0215443469	0.60	0.6198783	0.5345890
0.0215443469	0.65	0.6158335	0.5454147

0.0215443469	0.70	0.6157186	0.5520982
0.0215443469	0.75	0.6164960	0.5567122
0.0215443469	0.80	0.6190483	0.5596157
0.0215443469	0.85	0.6234843	0.5601498
0.0215443469	0.90	0.6276601	0.5604701
0.0215443469	0.95	0.6317278	0.5582701
0.0215443469	1.00	0.6346857	0.5569445
0.0316227766	0.05	0.7950860	0.3048769
0.0316227766	0.10	0.7561182	0.2961990
0.0316227766	0.15	0.7366454	0.2954742
0.0316227766	0.20	0.7160091	0.3325421
0.0316227766	0.25	0.6946589	0.3796172
0.0316227766	0.30	0.6770143	0.4204454
0.0316227766	0.35	0.6630415	0.4544040
0.0316227766	0.40	0.6570585	0.4713011
0.0316227766	0.45	0.6486322	0.4870041
0.0316227766	0.50	0.6403751	0.5002401
0.0316227766	0.55	0.6324765	0.5132092
0.0316227766	0.60	0.6240028	0.5277344
0.0316227766	0.65	0.6184736	0.5393799
0.0316227766	0.70	0.6167626	0.5479163
0.0316227766	0.75	0.6172215	0.5534153
0.0316227766	0.80	0.6172126	0.5592527
0.0316227766	0.85	0.6204256	0.5612633
0.0316227766	0.90	0.6236096	0.5629019
0.0316227766	0.95	0.6271167	0.5631963
0.0316227766	1.00	0.6299389	0.5619666
0.0464158883	0.05	0.7973360	0.3064937
0.0464158883	0.10	0.7586139	0.2969085
0.0464158883	0.15	0.7376150	0.2951641
0.0464158883	0.20	0.7189965	0.3261453
0.0464158883	0.25	0.6976359	0.3723467
0.0464158883	0.30	0.6807314	0.4106133
0.0464158883	0.35	0.6658183	0.4464138
0.0464158883	0.40	0.6580540	0.4684238
0.0464158883	0.45	0.6514706	0.4827214
0.0464158883	0.50	0.6439358	0.4954359

0.0464158883	0.55	0.6362107	0.5081298
0.0464158883	0.60	0.6285165	0.5211414
0.0464158883	0.65	0.6216849	0.5340704
0.0464158883	0.70	0.6182361	0.5433220
0.0464158883	0.75	0.6179214	0.5501564
0.0464158883	0.80	0.6176713	0.5569953
0.0464158883	0.85	0.6179260	0.5621818
0.0464158883	0.90	0.6208656	0.5637525
0.0464158883	0.95	0.6234485	0.5653659
0.0464158883	1.00	0.6255303	0.5664601
0.0681292069	0.05	0.7995795	0.3081758
0.0681292069	0.10	0.7617572	0.2967292
0.0681292069	0.15	0.7384527	0.2961280
0.0681292069	0.20	0.7215819	0.3214417
0.0681292069	0.25	0.7009467	0.3649911
0.0681292069	0.30	0.6843791	0.4019319
0.0681292069	0.35	0.6690341	0.4379495
0.0681292069	0.40	0.6587168	0.4652364
0.0681292069	0.45	0.6541773	0.4785273
0.0681292069	0.50	0.6472607	0.4911355
0.0681292069	0.55	0.6397398	0.5033126
0.0681292069	0.60	0.6332594	0.5145688
0.0681292069	0.65	0.6256980	0.5281977
0.0681292069	0.70	0.6204959	0.5394360
0.0681292069	0.75	0.6188383	0.5476880
0.0681292069	0.80	0.6183604	0.5547773
0.0681292069	0.85	0.6186299	0.5606514
0.0681292069	0.90	0.6195155	0.5642184
0.0681292069	0.95	0.6210101	0.5669789
0.0681292069	1.00	0.6222012	0.5694421
0.1000000000	0.05	0.8018065	0.3097719
0.1000000000	0.10	0.7649646	0.2966620
0.1000000000	0.15	0.7395052	0.2980645
0.1000000000	0.20	0.7237005	0.3190748
0.1000000000	0.25	0.7039111	0.3584968
0.1000000000	0.30	0.6877778	0.3940358
0.1000000000	0.35	0.6724611	0.4298348

```
  0.1000000000  0.40      0.6603833  0.4598209
  0.1000000000  0.45      0.6553794  0.4761933
  0.1000000000  0.50      0.6503771  0.4872728
  0.1000000000  0.55      0.6439926  0.4980022
  0.1000000000  0.60      0.6366522  0.5105946
  0.1000000000  0.65      0.6302162  0.5225469
  0.1000000000  0.70      0.6238846  0.5348273
  0.1000000000  0.75      0.6204830  0.5443855
  0.1000000000  0.80      0.6195641  0.5519200
  0.1000000000  0.85      0.6194219  0.5581312
  0.1000000000  0.90      0.6194874  0.5635599
  0.1000000000  0.95      0.6201888  0.5666499
  0.1000000000  1.00      0.6208205  0.5699748

RMSE was used to select the optimal model using  the smallest value.
The final values used for the model were fraction = 0.55 and lambda = 0.


Elasticnet

76 samples
21 predictors

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 69, 68, 69, 68, 68, 68, ...
Resampling results across tuning parameters:

  lambda        fraction  RMSE       Rsquared
  0.0000000000  0.05      0.7859920  0.2987827
  0.0000000000  0.10      0.7488708  0.2909545
  0.0000000000  0.15      0.7252899  0.3156452
  0.0000000000  0.20      0.6988259  0.3670982
  0.0000000000  0.25      0.6807484  0.4071992
  0.0000000000  0.30      0.6700407  0.4330112
  0.0000000000  0.35      0.6571229  0.4549977
  0.0000000000  0.40      0.6437464  0.4773885
```

0.0000000000	0.45	0.6321943	0.4964394
0.0000000000	0.50	0.6223355	0.5139852
0.0000000000	0.55	0.6174379	0.5283194
0.0000000000	0.60	0.6178921	0.5367062
0.0000000000	0.65	0.6242983	0.5366940
0.0000000000	0.70	0.6331836	0.5312607
0.0000000000	0.75	0.6406562	0.5248659
0.0000000000	0.80	0.6491212	0.5177923
0.0000000000	0.85	0.6585316	0.5088779
0.0000000000	0.90	0.6658473	0.5009795
0.0000000000	0.95	0.6721322	0.4946767
0.0000000000	1.00	0.6777366	0.4894324
0.0001000000	0.05	0.7860395	0.2988113
0.0001000000	0.10	0.7488984	0.2909997
0.0001000000	0.15	0.7253730	0.3154960
0.0001000000	0.20	0.6989066	0.3669386
0.0001000000	0.25	0.6808042	0.4070701
0.0001000000	0.30	0.6701193	0.4328991
0.0001000000	0.35	0.6572322	0.4548324
0.0001000000	0.40	0.6438668	0.4772162
0.0001000000	0.45	0.6323113	0.4962636
0.0001000000	0.50	0.6224469	0.5138269
0.0001000000	0.55	0.6174559	0.5282321
0.0001000000	0.60	0.6178554	0.5366868
0.0001000000	0.65	0.6241854	0.5367938
0.0001000000	0.70	0.6330641	0.5314047
0.0001000000	0.75	0.6405333	0.5250262
0.0001000000	0.80	0.6489628	0.5179871
0.0001000000	0.85	0.6583593	0.5090967
0.0001000000	0.90	0.6656789	0.5012063
0.0001000000	0.95	0.6719397	0.4949342
0.0001000000	1.00	0.6775327	0.4897027
0.0001467799	0.05	0.7860617	0.2988246
0.0001467799	0.10	0.7489113	0.2910208
0.0001467799	0.15	0.7254117	0.3154264
0.0001467799	0.20	0.6989442	0.3668643
0.0001467799	0.25	0.6808303	0.4070099

0.0001467799	0.30	0.6701560	0.4328468
0.0001467799	0.35	0.6572835	0.4547538
0.0001467799	0.40	0.6439229	0.4771359
0.0001467799	0.45	0.6323659	0.4961816
0.0001467799	0.50	0.6224989	0.5137529
0.0001467799	0.55	0.6174644	0.5281911
0.0001467799	0.60	0.6178384	0.5366774
0.0001467799	0.65	0.6241330	0.5368400
0.0001467799	0.70	0.6330025	0.5314715
0.0001467799	0.75	0.6404761	0.5251007
0.0001467799	0.80	0.6488891	0.5180776
0.0001467799	0.85	0.6582791	0.5091984
0.0001467799	0.90	0.6656005	0.5013119
0.0001467799	0.95	0.6718501	0.4950541
0.0001467799	1.00	0.6774378	0.4898285
0.0002154435	0.05	0.7860941	0.2988440
0.0002154435	0.10	0.7489302	0.2910517
0.0002154435	0.15	0.7254683	0.3153248
0.0002154435	0.20	0.6989992	0.3667555
0.0002154435	0.25	0.6808668	0.4069195
0.0002154435	0.30	0.6702096	0.4327703
0.0002154435	0.35	0.6573590	0.4546375
0.0002154435	0.40	0.6440050	0.4770183
0.0002154435	0.45	0.6324458	0.4960614
0.0002154435	0.50	0.6225752	0.5136444
0.0002154435	0.55	0.6174772	0.5281306
0.0002154435	0.60	0.6178244	0.5366551
0.0002154435	0.65	0.6240565	0.5369073
0.0002154435	0.70	0.6329127	0.5315689
0.0002154435	0.75	0.6403926	0.5252095
0.0002154435	0.80	0.6487814	0.5182099
0.0002154435	0.85	0.6581617	0.5093472
0.0002154435	0.90	0.6654859	0.5014662
0.0002154435	0.95	0.6717191	0.4952292
0.0002154435	1.00	0.6772990	0.4900124
0.0003162278	0.05	0.7861415	0.2988725
0.0003162278	0.10	0.7489579	0.2910968

0.0003162278	0.15	0.7255510	0.3151764
0.0003162278	0.20	0.6990796	0.3665966
0.0003162278	0.25	0.6809200	0.4067873
0.0003162278	0.30	0.6702879	0.4326585
0.0003162278	0.35	0.6574692	0.4544675
0.0003162278	0.40	0.6441251	0.4768463
0.0003162278	0.45	0.6325627	0.4958855
0.0003162278	0.50	0.6226868	0.5134854
0.0003162278	0.55	0.6174964	0.5280415
0.0003162278	0.60	0.6178161	0.5366124
0.0003162278	0.65	0.6239452	0.5370049
0.0003162278	0.70	0.6327817	0.5317107
0.0003162278	0.75	0.6402709	0.5253680
0.0003162278	0.80	0.6486242	0.5184028
0.0003162278	0.85	0.6579904	0.5095642
0.0003162278	0.90	0.6653185	0.5016913
0.0003162278	0.95	0.6715279	0.4954848
0.0003162278	1.00	0.6770964	0.4902807
0.0004641589	0.05	0.7862106	0.2989140
0.0004641589	0.10	0.7489985	0.2911625
0.0004641589	0.15	0.7256715	0.3149603
0.0004641589	0.20	0.6991967	0.3663650
0.0004641589	0.25	0.6809977	0.4065944
0.0004641589	0.30	0.6704021	0.4324955
0.0004641589	0.35	0.6576298	0.4542197
0.0004641589	0.40	0.6443002	0.4765952
0.0004641589	0.45	0.6327337	0.4956281
0.0004641589	0.50	0.6228502	0.5132524
0.0004641589	0.55	0.6175253	0.5279096
0.0004641589	0.60	0.6178048	0.5365486
0.0004641589	0.65	0.6237839	0.5371459
0.0004641589	0.70	0.6325917	0.5319160
0.0004641589	0.75	0.6400942	0.5255982
0.0004641589	0.80	0.6483955	0.5186831
0.0004641589	0.85	0.6577410	0.5098798
0.0004641589	0.90	0.6650748	0.5020189
0.0004641589	0.95	0.6712495	0.4958566

0.0004641589	1.00	0.6768014	0.4906711
0.0006812921	0.05	0.7863110	0.2989744
0.0006812921	0.10	0.7490698	0.2912381
0.0006812921	0.15	0.7258369	0.3146629
0.0006812921	0.20	0.6993668	0.3660285
0.0006812921	0.25	0.6811107	0.4063140
0.0006812921	0.30	0.6705681	0.4322584
0.0006812921	0.35	0.6578632	0.4538596
0.0006812921	0.40	0.6445550	0.4762295
0.0006812921	0.45	0.6329842	0.4952511
0.0006812921	0.50	0.6230888	0.5129114
0.0006812921	0.55	0.6175697	0.5277139
0.0006812921	0.60	0.6177902	0.5364524
0.0006812921	0.65	0.6235516	0.5373479
0.0006812921	0.70	0.6323172	0.5322116
0.0006812921	0.75	0.6398385	0.5259306
0.0006812921	0.80	0.6480641	0.5190884
0.0006812921	0.85	0.6573791	0.5103371
0.0006812921	0.90	0.6647212	0.5024938
0.0006812921	0.95	0.6708458	0.4963952
0.0006812921	1.00	0.6763736	0.4912368
0.0010000000	0.05	0.7864572	0.2990634
0.0010000000	0.10	0.7491742	0.2913468
0.0010000000	0.15	0.7260761	0.3142336
0.0010000000	0.20	0.6996129	0.3655417
0.0010000000	0.25	0.6812741	0.4059076
0.0010000000	0.30	0.6708083	0.4319152
0.0010000000	0.35	0.6581924	0.4533628
0.0010000000	0.40	0.6449242	0.4756990
0.0010000000	0.45	0.6333477	0.4947026
0.0010000000	0.50	0.6234439	0.5123932
0.0010000000	0.55	0.6176385	0.5274227
0.0010000000	0.60	0.6177666	0.5363169
0.0010000000	0.65	0.6232197	0.5376339
0.0010000000	0.70	0.6319238	0.5326335
0.0010000000	0.75	0.6394707	0.5264071
0.0010000000	0.80	0.6475874	0.5196733

0.0010000000	0.85	0.6568568	0.5109956
0.0010000000	0.90	0.6642110	0.5031780
0.0010000000	0.95	0.6702635	0.4971710
0.0010000000	1.00	0.6757564	0.4920522
0.0014677993	0.05	0.7866689	0.2991946
0.0014677993	0.10	0.7493263	0.2915031
0.0014677993	0.15	0.7264196	0.3136181
0.0014677993	0.20	0.6999639	0.3648607
0.0014677993	0.25	0.6815092	0.4053224
0.0014677993	0.30	0.6711535	0.4314211
0.0014677993	0.35	0.6586531	0.4526833
0.0014677993	0.40	0.6454561	0.4749333
0.0014677993	0.45	0.6338728	0.4939082
0.0014677993	0.50	0.6239760	0.5116236
0.0014677993	0.55	0.6177468	0.5269874
0.0014677993	0.60	0.6177330	0.5361377
0.0014677993	0.65	0.6227721	0.5380037
0.0014677993	0.70	0.6313658	0.5332277
0.0014677993	0.75	0.6389502	0.5270728
0.0014677993	0.80	0.6469076	0.5205090
0.0014677993	0.85	0.6561087	0.5119357
0.0014677993	0.90	0.6634800	0.5041559
0.0014677993	0.95	0.6694301	0.4982792
0.0014677993	1.00	0.6748726	0.4932175
0.0021544347	0.05	0.7869709	0.2993820
0.0021544347	0.10	0.7495466	0.2917258
0.0021544347	0.15	0.7269082	0.3127449
0.0021544347	0.20	0.7004561	0.3639449
0.0021544347	0.25	0.6818429	0.4044957
0.0021544347	0.30	0.6716453	0.4307161
0.0021544347	0.35	0.6593069	0.4517212
0.0021544347	0.40	0.6462162	0.4738364
0.0021544347	0.45	0.6346257	0.4927643
0.0021544347	0.50	0.6247453	0.5105057
0.0021544347	0.55	0.6179201	0.5263238
0.0021544347	0.60	0.6176897	0.5358431
0.0021544347	0.65	0.6221903	0.5384515

0.0021544347	0.70	0.6305868	0.5340487
0.0021544347	0.75	0.6382318	0.5279643
0.0021544347	0.80	0.6459558	0.5216685
0.0021544347	0.85	0.6550441	0.5132681
0.0021544347	0.90	0.6624442	0.5055374
0.0021544347	0.95	0.6682502	0.4998433
0.0021544347	1.00	0.6736208	0.4948641
0.0031622777	0.05	0.7873967	0.2996469
0.0031622777	0.10	0.7498643	0.2920392
0.0031622777	0.15	0.7275944	0.3115236
0.0031622777	0.20	0.7011480	0.3626560
0.0031622777	0.25	0.6824662	0.4030260
0.0031622777	0.30	0.6723348	0.4297219
0.0031622777	0.35	0.6602072	0.4504180
0.0031622777	0.40	0.6472902	0.4722817
0.0031622777	0.45	0.6356944	0.4911317
0.0031622777	0.50	0.6258474	0.5088910
0.0031622777	0.55	0.6182048	0.5253149
0.0031622777	0.60	0.6176359	0.5353786
0.0031622777	0.65	0.6214382	0.5389877
0.0031622777	0.70	0.6295195	0.5351498
0.0031622777	0.75	0.6372414	0.5291807
0.0031622777	0.80	0.6446446	0.5232481
0.0031622777	0.85	0.6534935	0.5152084
0.0031622777	0.90	0.6609981	0.5074569
0.0031622777	0.95	0.6666053	0.5020117
0.0031622777	1.00	0.6718736	0.4971537
0.0046415888	0.05	0.7879857	0.3000156
0.0046415888	0.10	0.7503156	0.2924747
0.0046415888	0.15	0.7285423	0.3098472
0.0046415888	0.20	0.7020086	0.3610534
0.0046415888	0.25	0.6834120	0.4008071
0.0046415888	0.30	0.6728578	0.4287036
0.0046415888	0.35	0.6614080	0.4486621
0.0046415888	0.40	0.6488567	0.4698833
0.0046415888	0.45	0.6371797	0.4888544
0.0046415888	0.50	0.6274087	0.5065805

0.0046415888	0.55	0.6186876	0.5237698
0.0046415888	0.60	0.6176498	0.5345753
0.0046415888	0.65	0.6206291	0.5392761
0.0046415888	0.70	0.6281749	0.5363608
0.0046415888	0.75	0.6358112	0.5309429
0.0046415888	0.80	0.6429053	0.5253027
0.0046415888	0.85	0.6513837	0.5178193
0.0046415888	0.90	0.6590174	0.5100646
0.0046415888	0.95	0.6643598	0.5049422
0.0046415888	1.00	0.6694840	0.5002687
0.0068129207	0.05	0.7887675	0.3005193
0.0068129207	0.10	0.7509261	0.2930799
0.0068129207	0.15	0.7298610	0.3078284
0.0068129207	0.20	0.7032010	0.3588474
0.0068129207	0.25	0.6847327	0.3977912
0.0068129207	0.30	0.6732575	0.4276549
0.0068129207	0.35	0.6630116	0.4463444
0.0068129207	0.40	0.6510093	0.4665392
0.0068129207	0.45	0.6390519	0.4860797
0.0068129207	0.50	0.6293456	0.5036148
0.0068129207	0.55	0.6196030	0.5214953
0.0068129207	0.60	0.6177459	0.5333319
0.0068129207	0.65	0.6198140	0.5393030
0.0068129207	0.70	0.6265035	0.5377728
0.0068129207	0.75	0.6338749	0.5332428
0.0068129207	0.80	0.6406923	0.5278312
0.0068129207	0.85	0.6486471	0.5211809
0.0068129207	0.90	0.6563035	0.5135477
0.0068129207	0.95	0.6613856	0.5087622
0.0068129207	1.00	0.6663024	0.5043852
0.0100000000	0.05	0.7898032	0.3011907
0.0100000000	0.10	0.7517720	0.2938849
0.0100000000	0.15	0.7315741	0.3053296
0.0100000000	0.20	0.7047636	0.3559959
0.0100000000	0.25	0.6866809	0.3934530
0.0100000000	0.30	0.6740467	0.4253996
0.0100000000	0.35	0.6650274	0.4436258

0.0100000000	0.40	0.6535404	0.4625181
0.0100000000	0.45	0.6415473	0.4824155
0.0100000000	0.50	0.6317873	0.4997642
0.0100000000	0.55	0.6216062	0.5178532
0.0100000000	0.60	0.6180706	0.5312578
0.0100000000	0.65	0.6190680	0.5388619
0.0100000000	0.70	0.6243623	0.5394714
0.0100000000	0.75	0.6314640	0.5358431
0.0100000000	0.80	0.6379445	0.5308433
0.0100000000	0.85	0.6451845	0.5253822
0.0100000000	0.90	0.6526204	0.5182151
0.0100000000	0.95	0.6575869	0.5135724
0.0100000000	1.00	0.6622118	0.5096222
0.0146779927	0.05	0.7911354	0.3020604
0.0146779927	0.10	0.7527240	0.2948163
0.0146779927	0.15	0.7327536	0.3034032
0.0146779927	0.20	0.7068201	0.3522135
0.0146779927	0.25	0.6890676	0.3881154
0.0146779927	0.30	0.6751544	0.4222913
0.0146779927	0.35	0.6676240	0.4398135
0.0146779927	0.40	0.6566202	0.4576346
0.0146779927	0.45	0.6447622	0.4777067
0.0146779927	0.50	0.6350037	0.4946290
0.0146779927	0.55	0.6247990	0.5126364
0.0146779927	0.60	0.6189910	0.5276532
0.0146779927	0.65	0.6186518	0.5373721
0.0146779927	0.70	0.6221480	0.5406903
0.0146779927	0.75	0.6287042	0.5383259
0.0146779927	0.80	0.6348187	0.5341955
0.0146779927	0.85	0.6411626	0.5298307
0.0146779927	0.90	0.6481079	0.5238657
0.0146779927	0.95	0.6529299	0.5193724
0.0146779927	1.00	0.6571826	0.5159639
0.0215443469	0.05	0.7927922	0.3031510
0.0215443469	0.10	0.7539691	0.2959266
0.0215443469	0.15	0.7339579	0.3016593
0.0215443469	0.20	0.7096292	0.3469347

0.0215443469	0.25	0.6919648	0.3816961
0.0215443469	0.30	0.6766339	0.4181285
0.0215443469	0.35	0.6702830	0.4355705
0.0215443469	0.40	0.6602534	0.4520655
0.0215443469	0.45	0.6488521	0.4715906
0.0215443469	0.50	0.6390924	0.4880857
0.0215443469	0.55	0.6294138	0.5055035
0.0215443469	0.60	0.6208277	0.5224188
0.0215443469	0.65	0.6189095	0.5340455
0.0215443469	0.70	0.6208164	0.5396790
0.0215443469	0.75	0.6256736	0.5401411
0.0215443469	0.80	0.6314789	0.5375959
0.0215443469	0.85	0.6368496	0.5339890
0.0215443469	0.90	0.6427297	0.5302669
0.0215443469	0.95	0.6471535	0.5264358
0.0215443469	1.00	0.6513439	0.5231598
0.0316227766	0.05	0.7947426	0.3044155
0.0316227766	0.10	0.7558605	0.2965761
0.0316227766	0.15	0.7353023	0.2997821
0.0316227766	0.20	0.7130471	0.3399719
0.0316227766	0.25	0.6950480	0.3749436
0.0316227766	0.30	0.6790681	0.4116041
0.0316227766	0.35	0.6711329	0.4328942
0.0316227766	0.40	0.6638897	0.4470243
0.0316227766	0.45	0.6538210	0.4640068
0.0316227766	0.50	0.6437117	0.4809668
0.0316227766	0.55	0.6350795	0.4969491
0.0316227766	0.60	0.6249666	0.5151714
0.0316227766	0.65	0.6202356	0.5287704
0.0316227766	0.70	0.6200484	0.5369130
0.0316227766	0.75	0.6233510	0.5402981
0.0316227766	0.80	0.6283192	0.5396299
0.0316227766	0.85	0.6327099	0.5373813
0.0316227766	0.90	0.6374803	0.5350014
0.0316227766	0.95	0.6411760	0.5335256
0.0316227766	1.00	0.6450579	0.5306251
0.0464158883	0.05	0.7968705	0.3058294

0.0464158883	0.10	0.7583129	0.2968166
0.0464158883	0.15	0.7367452	0.2982225
0.0464158883	0.20	0.7163959	0.3329053
0.0464158883	0.25	0.6975849	0.3690528
0.0464158883	0.30	0.6813880	0.4047213
0.0464158883	0.35	0.6713985	0.4308385
0.0464158883	0.40	0.6669210	0.4433544
0.0464158883	0.45	0.6580169	0.4575380
0.0464158883	0.50	0.6482472	0.4744280
0.0464158883	0.55	0.6396628	0.4897686
0.0464158883	0.60	0.6304753	0.5064667
0.0464158883	0.65	0.6229704	0.5220891
0.0464158883	0.70	0.6207140	0.5325320
0.0464158883	0.75	0.6218942	0.5384692
0.0464158883	0.80	0.6256624	0.5397815
0.0464158883	0.85	0.6295917	0.5392348
0.0464158883	0.90	0.6330393	0.5377706
0.0464158883	0.95	0.6359235	0.5377106
0.0464158883	1.00	0.6389635	0.5374071
0.0681292069	0.05	0.7989963	0.3074219
0.0681292069	0.10	0.7610451	0.2966228
0.0681292069	0.15	0.7381199	0.2981210
0.0681292069	0.20	0.7197360	0.3264919
0.0681292069	0.25	0.7001594	0.3637755
0.0681292069	0.30	0.6849751	0.3959324
0.0681292069	0.35	0.6727227	0.4262127
0.0681292069	0.40	0.6686028	0.4406536
0.0681292069	0.45	0.6617973	0.4525164
0.0681292069	0.50	0.6533523	0.4672721
0.0681292069	0.55	0.6447874	0.4824446
0.0681292069	0.60	0.6367919	0.4974927
0.0681292069	0.65	0.6277509	0.5145482
0.0681292069	0.70	0.6229760	0.5269137
0.0681292069	0.75	0.6220620	0.5350745
0.0681292069	0.80	0.6246355	0.5382727
0.0681292069	0.85	0.6276158	0.5395075
0.0681292069	0.90	0.6300028	0.5399337

```
  0.0681292069  0.95      0.6317253  0.5409737
  0.0681292069  1.00      0.6339483  0.5423243
  0.1000000000  0.05      0.8012728  0.3091181
  0.1000000000  0.10      0.7642330  0.2964494
  0.1000000000  0.15      0.7400806  0.2976617
  0.1000000000  0.20      0.7218946  0.3224987
  0.1000000000  0.25      0.7029864  0.3583985
  0.1000000000  0.30      0.6887508  0.3871865
  0.1000000000  0.35      0.6750340  0.4196762
  0.1000000000  0.40      0.6685667  0.4392981
  0.1000000000  0.45      0.6652557  0.4492352
  0.1000000000  0.50      0.6578716  0.4609926
  0.1000000000  0.55      0.6504983  0.4745340
  0.1000000000  0.60      0.6425933  0.4894894
  0.1000000000  0.65      0.6343535  0.5057501
  0.1000000000  0.70      0.6268160  0.5203112
  0.1000000000  0.75      0.6239333  0.5298932
  0.1000000000  0.80      0.6240224  0.5361543
  0.1000000000  0.85      0.6269301  0.5381568
  0.1000000000  0.90      0.6289514  0.5396175
  0.1000000000  0.95      0.6297565  0.5420011
  0.1000000000  1.00      0.6310288  0.5443299

RMSE was used to select the optimal model using  the smallest value.
The final values used for the model were fraction = 0.55 and lambda = 0.


Elasticnet

76 samples
21 predictors

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 69, 68, 69, 68, 68, 68, ...
Resampling results across tuning parameters:
```

lambda	fraction	RMSE	Rsquared
0.0000000000	0.05	0.7863112	0.2987688
0.0000000000	0.10	0.7484312	0.2921941
0.0000000000	0.15	0.7301101	0.3068085
0.0000000000	0.20	0.7045271	0.3604548
0.0000000000	0.25	0.6851295	0.4069650
0.0000000000	0.30	0.6724365	0.4383811
0.0000000000	0.35	0.6610885	0.4584559
0.0000000000	0.40	0.6496193	0.4777686
0.0000000000	0.45	0.6385986	0.4962119
0.0000000000	0.50	0.6300702	0.5116547
0.0000000000	0.55	0.6287965	0.5211609
0.0000000000	0.60	0.6344863	0.5222438
0.0000000000	0.65	0.6444207	0.5154442
0.0000000000	0.70	0.6541742	0.5070682
0.0000000000	0.75	0.6641773	0.4966173
0.0000000000	0.80	0.6722613	0.4870991
0.0000000000	0.85	0.6791410	0.4798691
0.0000000000	0.90	0.6846344	0.4745687
0.0000000000	0.95	0.6887642	0.4708663
0.0000000000	1.00	0.6931214	0.4669466
0.0001000000	0.05	0.7863544	0.2987929
0.0001000000	0.10	0.7484588	0.2922295
0.0001000000	0.15	0.7301871	0.3066647
0.0001000000	0.20	0.7046076	0.3602746
0.0001000000	0.25	0.6851876	0.4068160
0.0001000000	0.30	0.6725224	0.4382187
0.0001000000	0.35	0.6611860	0.4582973
0.0001000000	0.40	0.6497349	0.4775775
0.0001000000	0.45	0.6387078	0.4960433
0.0001000000	0.50	0.6301482	0.5115082
0.0001000000	0.55	0.6287915	0.5211034
0.0001000000	0.60	0.6344110	0.5222914
0.0001000000	0.65	0.6443113	0.5155499
0.0001000000	0.70	0.6540604	0.5072028
0.0001000000	0.75	0.6640322	0.4968122
0.0001000000	0.80	0.6721191	0.4872861

0.0001000000	0.85	0.6789821	0.4800736
0.0001000000	0.90	0.6844737	0.4747760
0.0001000000	0.95	0.6885903	0.4710893
0.0001000000	1.00	0.6929391	0.4671842
0.0001467799	0.05	0.7863746	0.2988041
0.0001467799	0.10	0.7484717	0.2922460
0.0001467799	0.15	0.7302230	0.3065978
0.0001467799	0.20	0.7046451	0.3601907
0.0001467799	0.25	0.6852147	0.4067465
0.0001467799	0.30	0.6725624	0.4381430
0.0001467799	0.35	0.6612314	0.4582233
0.0001467799	0.40	0.6497888	0.4774883
0.0001467799	0.45	0.6387588	0.4959646
0.0001467799	0.50	0.6301847	0.5114397
0.0001467799	0.55	0.6287894	0.5210763
0.0001467799	0.60	0.6343761	0.5223133
0.0001467799	0.65	0.6442605	0.5155989
0.0001467799	0.70	0.6540074	0.5072653
0.0001467799	0.75	0.6639646	0.4969030
0.0001467799	0.80	0.6720529	0.4873731
0.0001467799	0.85	0.6789080	0.4801688
0.0001467799	0.90	0.6843988	0.4748725
0.0001467799	0.95	0.6885093	0.4711931
0.0001467799	1.00	0.6928542	0.4672948
0.0002154435	0.05	0.7864039	0.2988213
0.0002154435	0.10	0.7484906	0.2922702
0.0002154435	0.15	0.7302755	0.3064998
0.0002154435	0.20	0.7047000	0.3600678
0.0002154435	0.25	0.6852543	0.4066447
0.0002154435	0.30	0.6726210	0.4380321
0.0002154435	0.35	0.6612979	0.4581150
0.0002154435	0.40	0.6498678	0.4773577
0.0002154435	0.45	0.6388334	0.4958492
0.0002154435	0.50	0.6302384	0.5113391
0.0002154435	0.55	0.6287864	0.5210363
0.0002154435	0.60	0.6343252	0.5223452
0.0002154435	0.65	0.6441862	0.5156705

0.0002154435	0.70	0.6539301	0.5073566
0.0002154435	0.75	0.6638658	0.4970356
0.0002154435	0.80	0.6719565	0.4875008
0.0002154435	0.85	0.6787997	0.4803081
0.0002154435	0.90	0.6842893	0.4750136
0.0002154435	0.95	0.6883908	0.4713450
0.0002154435	1.00	0.6927300	0.4674566
0.0003162278	0.05	0.7864465	0.2988477
0.0003162278	0.10	0.7485183	0.2923056
0.0003162278	0.15	0.7303522	0.3063569
0.0003162278	0.20	0.7047802	0.3598883
0.0003162278	0.25	0.6853123	0.4064958
0.0003162278	0.30	0.6727067	0.4378700
0.0003162278	0.35	0.6613951	0.4579566
0.0003162278	0.40	0.6499834	0.4771664
0.0003162278	0.45	0.6389427	0.4956801
0.0003162278	0.50	0.6303172	0.5111914
0.0003162278	0.55	0.6287826	0.5209771
0.0003162278	0.60	0.6342513	0.5223911
0.0003162278	0.65	0.6440780	0.5157747
0.0003162278	0.70	0.6538174	0.5074896
0.0003162278	0.75	0.6637215	0.4972293
0.0003162278	0.80	0.6718160	0.4876875
0.0003162278	0.85	0.6786416	0.4805115
0.0003162278	0.90	0.6841295	0.4752195
0.0003162278	0.95	0.6882177	0.4715668
0.0003162278	1.00	0.6925485	0.4676929
0.0004641589	0.05	0.7865086	0.2988863
0.0004641589	0.10	0.7485590	0.2923572
0.0004641589	0.15	0.7304641	0.3061488
0.0004641589	0.20	0.7048973	0.3596265
0.0004641589	0.25	0.6853968	0.4062785
0.0004641589	0.30	0.6728318	0.4376334
0.0004641589	0.35	0.6615370	0.4577254
0.0004641589	0.40	0.6501521	0.4768869
0.0004641589	0.45	0.6391020	0.4954350
0.0004641589	0.50	0.6304329	0.5109745

0.0004641589	0.55	0.6287778	0.5208893
0.0004641589	0.60	0.6341444	0.5224570
0.0004641589	0.65	0.6439208	0.5159257
0.0004641589	0.70	0.6536536	0.5076826
0.0004641589	0.75	0.6635114	0.4975112
0.0004641589	0.80	0.6716114	0.4879594
0.0004641589	0.85	0.6784111	0.4808077
0.0004641589	0.90	0.6838966	0.4755194
0.0004641589	0.95	0.6879655	0.4718897
0.0004641589	1.00	0.6922840	0.4680371
0.0006812921	0.05	0.7865990	0.2989425
0.0006812921	0.10	0.7486185	0.2924325
0.0006812921	0.15	0.7306267	0.3058467
0.0006812921	0.20	0.7050606	0.3592564
0.0006812921	0.25	0.6855271	0.4059406
0.0006812921	0.30	0.6730004	0.4372798
0.0006812921	0.35	0.6617436	0.4573887
0.0006812921	0.40	0.6503980	0.4764791
0.0006812921	0.45	0.6393324	0.4950868
0.0006812921	0.50	0.6306030	0.5106560
0.0006812921	0.55	0.6287729	0.5207585
0.0006812921	0.60	0.6339906	0.5225503
0.0006812921	0.65	0.6436936	0.5161431
0.0006812921	0.70	0.6534167	0.5079611
0.0006812921	0.75	0.6632063	0.4979201
0.0006812921	0.80	0.6713141	0.4883539
0.0006812921	0.85	0.6780763	0.4812375
0.0006812921	0.90	0.6835586	0.4759544
0.0006812921	0.95	0.6875992	0.4723584
0.0006812921	1.00	0.6918998	0.4685366
0.0010000000	0.05	0.7867301	0.2990241
0.0010000000	0.10	0.7487195	0.2925243
0.0010000000	0.15	0.7308622	0.3054108
0.0010000000	0.20	0.7052868	0.3587365
0.0010000000	0.25	0.6857310	0.4054061
0.0010000000	0.30	0.6732379	0.4367622
0.0010000000	0.35	0.6620431	0.4569001

0.0010000000	0.40	0.6507551	0.4758857
0.0010000000	0.45	0.6396676	0.4945784
0.0010000000	0.50	0.6308528	0.5101886
0.0010000000	0.55	0.6287699	0.5205623
0.0010000000	0.60	0.6337806	0.5226364
0.0010000000	0.65	0.6433674	0.5164537
0.0010000000	0.70	0.6530765	0.5083602
0.0010000000	0.75	0.6627658	0.4985101
0.0010000000	0.80	0.6708843	0.4889236
0.0010000000	0.85	0.6775925	0.4818580
0.0010000000	0.90	0.6830703	0.4765820
0.0010000000	0.95	0.6870696	0.4730352
0.0010000000	1.00	0.6913443	0.4692581
0.0014677993	0.05	0.7869187	0.2991440
0.0014677993	0.10	0.7488672	0.2926551
0.0014677993	0.15	0.7311979	0.3047964
0.0014677993	0.20	0.7056110	0.3579981
0.0014677993	0.25	0.6860241	0.4046428
0.0014677993	0.30	0.6735676	0.4360528
0.0014677993	0.35	0.6624720	0.4562018
0.0014677993	0.40	0.6512688	0.4750312
0.0014677993	0.45	0.6401529	0.4938385
0.0014677993	0.50	0.6312216	0.5095011
0.0014677993	0.55	0.6287786	0.5202636
0.0014677993	0.60	0.6334727	0.5227505
0.0014677993	0.65	0.6429088	0.5168847
0.0014677993	0.70	0.6525931	0.5089250
0.0014677993	0.75	0.6621416	0.4993431
0.0014677993	0.80	0.6702752	0.4897267
0.0014677993	0.85	0.6769047	0.4827363
0.0014677993	0.90	0.6823774	0.4774688
0.0014677993	0.95	0.6863169	0.4739925
0.0014677993	1.00	0.6905464	0.4702928
0.0021544347	0.05	0.7871899	0.2993221
0.0021544347	0.10	0.7490812	0.2928419
0.0021544347	0.15	0.7316735	0.3039367
0.0021544347	0.20	0.7060724	0.3569554

```
0.0021544347  0.25      0.6864425  0.4035575
0.0021544347  0.30      0.6740137  0.4351059
0.0021544347  0.35      0.6630780  0.4552163
0.0021544347  0.40      0.6519922  0.4738205
0.0021544347  0.45      0.6408515  0.4927662
0.0021544347  0.50      0.6317907  0.5084995
0.0021544347  0.55      0.6288133  0.5198061
0.0021544347  0.60      0.6330694  0.5228710
0.0021544347  0.65      0.6422719  0.5174739
0.0021544347  0.70      0.6519159  0.5097133
0.0021544347  0.75      0.6612638  0.5005105
0.0021544347  0.80      0.6694350  0.4908333
0.0021544347  0.85      0.6759341  0.4839709
0.0021544347  0.90      0.6814009  0.4787138
0.0021544347  0.95      0.6852476  0.4753651
0.0021544347  1.00      0.6894114  0.4717615
0.0031622777  0.05      0.7875742  0.2995753
0.0031622777  0.10      0.7493900  0.2931078
0.0031622777  0.15      0.7322887  0.3028417
0.0031622777  0.20      0.7067079  0.3555814
0.0031622777  0.25      0.6870350  0.4020168
0.0031622777  0.30      0.6745742  0.4338559
0.0031622777  0.35      0.6639016  0.4539311
0.0031622777  0.40      0.6529017  0.4722418
0.0031622777  0.45      0.6418230  0.4912578
0.0031622777  0.50      0.6326456  0.5071162
0.0031622777  0.55      0.6288896  0.5190986
0.0031622777  0.60      0.6325099  0.5230226
0.0031622777  0.65      0.6414312  0.5182226
0.0031622777  0.70      0.6509832  0.5107896
0.0031622777  0.75      0.6599757  0.5021753
0.0031622777  0.80      0.6682626  0.4923745
0.0031622777  0.85      0.6745689  0.4857118
0.0031622777  0.90      0.6799565  0.4805867
0.0031622777  0.95      0.6838003  0.4772466
0.0031622777  1.00      0.6878181  0.4738173
0.0046415888  0.05      0.7881113  0.2999302
```

0.0046415888	0.10	0.7498324	0.2934812
0.0046415888	0.15	0.7331070	0.3014100
0.0046415888	0.20	0.7075859	0.3537159
0.0046415888	0.25	0.6878530	0.3998197
0.0046415888	0.30	0.6754091	0.4318478
0.0046415888	0.35	0.6650322	0.4522176
0.0046415888	0.40	0.6541773	0.4700166
0.0046415888	0.45	0.6432498	0.4889494
0.0046415888	0.50	0.6339208	0.5051853
0.0046415888	0.55	0.6289813	0.5181197
0.0046415888	0.60	0.6317347	0.5232080
0.0046415888	0.65	0.6404569	0.5189689
0.0046415888	0.70	0.6496539	0.5121970
0.0046415888	0.75	0.6583566	0.5040801
0.0046415888	0.80	0.6666449	0.4944914
0.0046415888	0.85	0.6726638	0.4881722
0.0046415888	0.90	0.6779217	0.4832071
0.0046415888	0.95	0.6819317	0.4797036
0.0046415888	1.00	0.6856217	0.4766400
0.0068129207	0.05	0.7888481	0.3004192
0.0068129207	0.10	0.7504587	0.2939978
0.0068129207	0.15	0.7342085	0.2995924
0.0068129207	0.20	0.7087852	0.3511762
0.0068129207	0.25	0.6889414	0.3968079
0.0068129207	0.30	0.6765825	0.4289834
0.0068129207	0.35	0.6665071	0.4498815
0.0068129207	0.40	0.6559081	0.4670193
0.0068129207	0.45	0.6452801	0.4855860
0.0068129207	0.50	0.6357702	0.5024202
0.0068129207	0.55	0.6292384	0.5165998
0.0068129207	0.60	0.6307399	0.5233373
0.0068129207	0.65	0.6391430	0.5198549
0.0068129207	0.70	0.6477972	0.5139711
0.0068129207	0.75	0.6562648	0.5064159
0.0068129207	0.80	0.6642174	0.4977127
0.0068129207	0.85	0.6700866	0.4914978
0.0068129207	0.90	0.6751888	0.4866984

0.0068129207	0.95	0.6794014	0.4830065
0.0068129207	1.00	0.6826676	0.4804182
0.0100000000	0.05	0.7898396	0.3010783
0.0100000000	0.10	0.7513098	0.2946908
0.0100000000	0.15	0.7351856	0.2986417
0.0100000000	0.20	0.7104671	0.3476342
0.0100000000	0.25	0.6906596	0.3924382
0.0100000000	0.30	0.6779526	0.4254602
0.0100000000	0.35	0.6683169	0.4467324
0.0100000000	0.40	0.6579519	0.4636765
0.0100000000	0.45	0.6477506	0.4815464
0.0100000000	0.50	0.6383110	0.4985509
0.0100000000	0.55	0.6304632	0.5137274
0.0100000000	0.60	0.6301712	0.5224178
0.0100000000	0.65	0.6371351	0.5213306
0.0100000000	0.70	0.6453585	0.5161752
0.0100000000	0.75	0.6535060	0.5094525
0.0100000000	0.80	0.6612688	0.5014683
0.0100000000	0.85	0.6667121	0.4958246
0.0100000000	0.90	0.6715696	0.4912556
0.0100000000	0.95	0.6758543	0.4875325
0.0100000000	1.00	0.6788203	0.4853094
0.0146779927	0.05	0.7911319	0.3019426
0.0146779927	0.10	0.7524762	0.2955895
0.0146779927	0.15	0.7353720	0.2985618
0.0146779927	0.20	0.7126400	0.3430957
0.0146779927	0.25	0.6932107	0.3862372
0.0146779927	0.30	0.6790875	0.4216661
0.0146779927	0.35	0.6704561	0.4428604
0.0146779927	0.40	0.6605757	0.4594067
0.0146779927	0.45	0.6510100	0.4761356
0.0146779927	0.50	0.6413536	0.4935363
0.0146779927	0.55	0.6325862	0.5093621
0.0146779927	0.60	0.6298878	0.5205428
0.0146779927	0.65	0.6347037	0.5225717
0.0146779927	0.70	0.6422462	0.5188113
0.0146779927	0.75	0.6500369	0.5131482

0.0146779927	0.80	0.6573251	0.5063888
0.0146779927	0.85	0.6624441	0.5011266
0.0146779927	0.90	0.6668270	0.4971179
0.0146779927	0.95	0.6711292	0.4934488
0.0146779927	1.00	0.6740148	0.4913768
0.0215443469	0.05	0.7927499	0.3030320
0.0215443469	0.10	0.7538352	0.2966056
0.0215443469	0.15	0.7356961	0.2983643
0.0215443469	0.20	0.7152897	0.3376359
0.0215443469	0.25	0.6963595	0.3787123
0.0215443469	0.30	0.6808269	0.4164960
0.0215443469	0.35	0.6729241	0.4381750
0.0215443469	0.40	0.6637900	0.4542338
0.0215443469	0.45	0.6548901	0.4693788
0.0215443469	0.50	0.6452708	0.4866957
0.0215443469	0.55	0.6359391	0.5034160
0.0215443469	0.60	0.6304256	0.5167963
0.0215443469	0.65	0.6323693	0.5226643
0.0215443469	0.70	0.6392117	0.5207401
0.0215443469	0.75	0.6459995	0.5168896
0.0215443469	0.80	0.6529946	0.5113083
0.0215443469	0.85	0.6574466	0.5072102
0.0215443469	0.90	0.6615630	0.5034641
0.0215443469	0.95	0.6652784	0.5006892
0.0215443469	1.00	0.6683261	0.4984979
0.0316227766	0.05	0.7946439	0.3042760
0.0316227766	0.10	0.7556674	0.2970396
0.0316227766	0.15	0.7360931	0.2981722
0.0316227766	0.20	0.7182498	0.3313215
0.0316227766	0.25	0.6998620	0.3700937
0.0316227766	0.30	0.6830881	0.4099523
0.0316227766	0.35	0.6747682	0.4334845
0.0316227766	0.40	0.6674272	0.4481363
0.0316227766	0.45	0.6588073	0.4621430
0.0316227766	0.50	0.6501824	0.4778982
0.0316227766	0.55	0.6412612	0.4944586
0.0316227766	0.60	0.6333283	0.5098033

```
0.0316227766  0.65      0.6312645  0.5202836
0.0316227766  0.70      0.6358947  0.5222316
0.0316227766  0.75      0.6423816  0.5190889
0.0316227766  0.80      0.6482969  0.5155265
0.0316227766  0.85      0.6524929  0.5126570
0.0316227766  0.90      0.6557963  0.5105057
0.0316227766  0.95      0.6591503  0.5080651
0.0316227766  1.00      0.6620490  0.5062608
0.0464158883  0.05      0.7968142  0.3056961
0.0464158883  0.10      0.7581376  0.2970072
0.0464158883  0.15      0.7369137  0.2977928
0.0464158883  0.20      0.7218241  0.3236092
0.0464158883  0.25      0.7033586  0.3613420
0.0464158883  0.30      0.6863644  0.4001240
0.0464158883  0.35      0.6761589  0.4269302
0.0464158883  0.40      0.6710229  0.4416486
0.0464158883  0.45      0.6626925  0.4555255
0.0464158883  0.50      0.6553390  0.4688059
0.0464158883  0.55      0.6469466  0.4844990
0.0464158883  0.60      0.6381579  0.5006926
0.0464158883  0.65      0.6329884  0.5135158
0.0464158883  0.70      0.6338596  0.5204903
0.0464158883  0.75      0.6391595  0.5200658
0.0464158883  0.80      0.6441478  0.5180030
0.0464158883  0.85      0.6480853  0.5161312
0.0464158883  0.90      0.6508110  0.5157561
0.0464158883  0.95      0.6531644  0.5151425
0.0464158883  1.00      0.6557582  0.5138896
0.0681292069  0.05      0.7992179  0.3076547
0.0681292069  0.10      0.7612224  0.2967863
0.0681292069  0.15      0.7381388  0.2977672
0.0681292069  0.20      0.7249339  0.3166951
0.0681292069  0.25      0.7066335  0.3532787
0.0681292069  0.30      0.6904592  0.3885450
0.0681292069  0.35      0.6778242  0.4195483
0.0681292069  0.40      0.6727239  0.4369870
0.0681292069  0.45      0.6666227  0.4491055
```

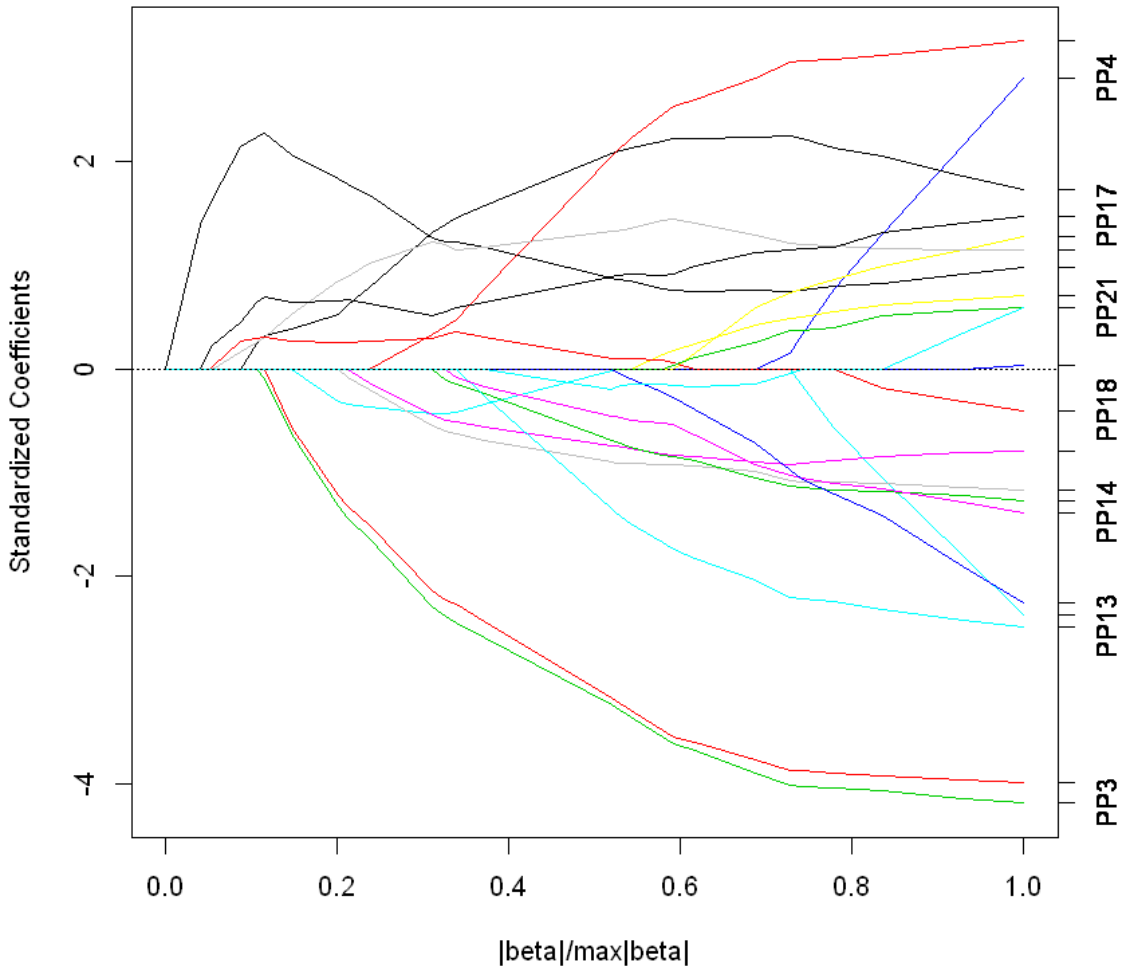
```
  0.0681292069  0.50      0.6597374  0.4609445
  0.0681292069  0.55      0.6526315  0.4745415
  0.0681292069  0.60      0.6451920  0.4892496
  0.0681292069  0.65      0.6373665  0.5040358
  0.0681292069  0.70      0.6342943  0.5151278
  0.0681292069  0.75      0.6366221  0.5191195
  0.0681292069  0.80      0.6415054  0.5180989
  0.0681292069  0.85      0.6448687  0.5177761
  0.0681292069  0.90      0.6466668  0.5187174
  0.0681292069  0.95      0.6485867  0.5195957
  0.0681292069  1.00      0.6503091  0.5202956
  0.1000000000  0.05      0.8015827  0.3095832
  0.1000000000  0.10      0.7645900  0.2966834
  0.1000000000  0.15      0.7401085  0.2976947
  0.1000000000  0.20      0.7255750  0.3146812
  0.1000000000  0.25      0.7093804  0.3475503
  0.1000000000  0.30      0.6943182  0.3790907
  0.1000000000  0.35      0.6804017  0.4115016
  0.1000000000  0.40      0.6732012  0.4329652
  0.1000000000  0.45      0.6699539  0.4443185
  0.1000000000  0.50      0.6632782  0.4555728
  0.1000000000  0.55      0.6574427  0.4671126
  0.1000000000  0.60      0.6506821  0.4807303
  0.1000000000  0.65      0.6439533  0.4940384
  0.1000000000  0.70      0.6377112  0.5073740
  0.1000000000  0.75      0.6365179  0.5155043
  0.1000000000  0.80      0.6400094  0.5170002
  0.1000000000  0.85      0.6433084  0.5174322
  0.1000000000  0.90      0.6447891  0.5197599
  0.1000000000  0.95      0.6453028  0.5227230
  0.1000000000  1.00      0.6467518  0.5243428

RMSE was used to select the optimal model using  the smallest value.
The final values used for the model were fraction = 0.55 and lambda = 0.001.
```

```
In [282]: plot(enet$finalModel,use.color =T)
```

```
enet.finalCoef<-predict(enet$finalModel,type="coefficients",mode="fraction",s=enet$be
enet.finalCoef[order(abs(enet.finalCoef),decreasing = T)]
head(enet.finalCoef[order(abs(enet.finalCoef),decreasing = T)])
```

PP3 -0.401111380596764 **PP2** -0.382145081056234 **PP9** 0.261340180206403 **PP10** 0.259469745320995 **PP13** -0.174675280631816 **PP16** 0.159011882271069 **PP8** -0.121963591366658 **PP17** 0.107326232665419 **PP1** 0.0975200424922241 **PP11** -0.0958769161807633 **PP6** -0.0849132955932825 **PP14** -0.0627631795223256 **PP21** -0.0185568149454934 **PP20** -0.0149870524315067 **PP18** 0.0108617017374061 **PP7** 0.0037251538032139 **PP4** 0 **PP5** 0 **PP12** 0 **PP15** 0 **PP19** 0

PP3 -0.401111380596764 **PP2** -0.382145081056234 **PP9** 0.261340180206403 **PP10** 0.259469745320995 **PP13** -0.174675280631816 **PP16** 0.159011882271069



通过 lasso 乃至 elastic net 等方法的变量选择特点，我们看到了自变量中哪些是重要的，哪些不是。由于所有自变量都是归一化的，这大致可以作为重要性的排序。

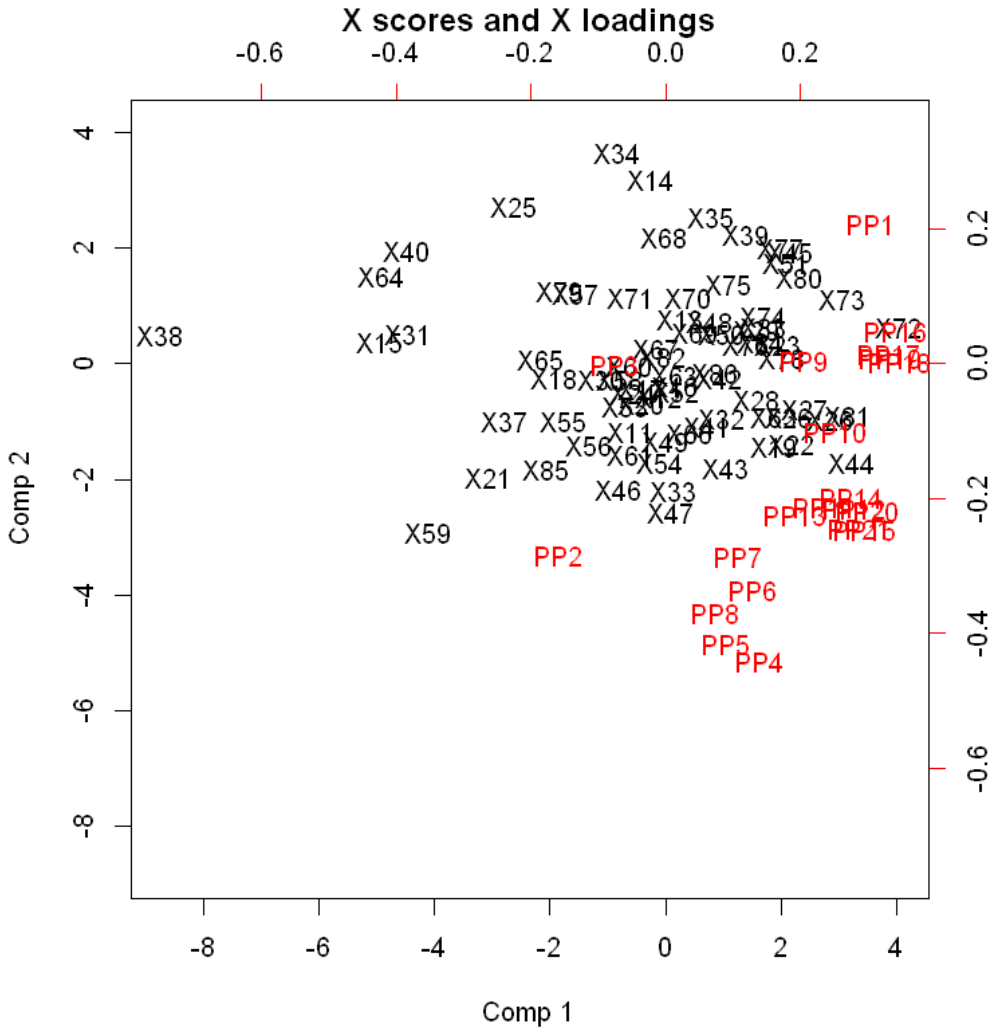
2.1.5 Partial Least Square

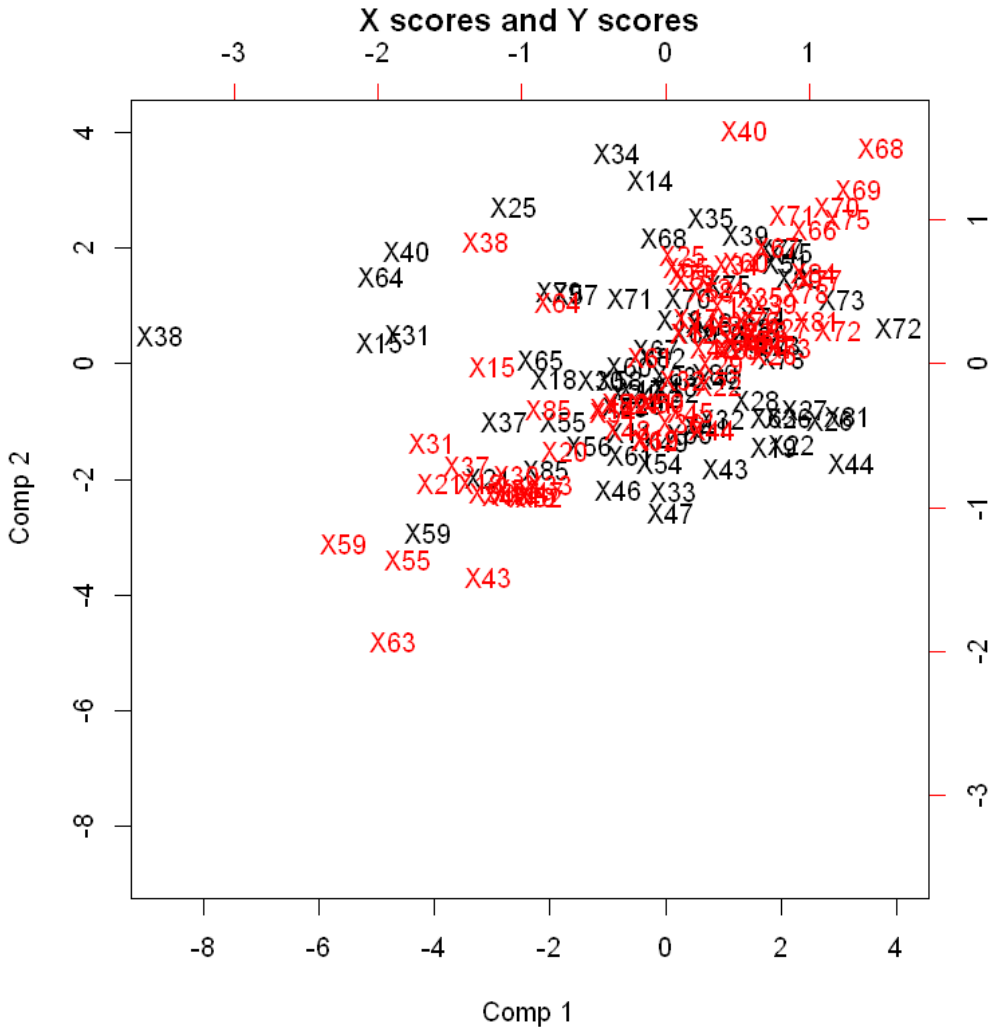
```
In [283]: pls <- train(x = X, y = y,
                       method = "pls",
                       tuneLength=10,
                       trControl = trControl)
          pls
```

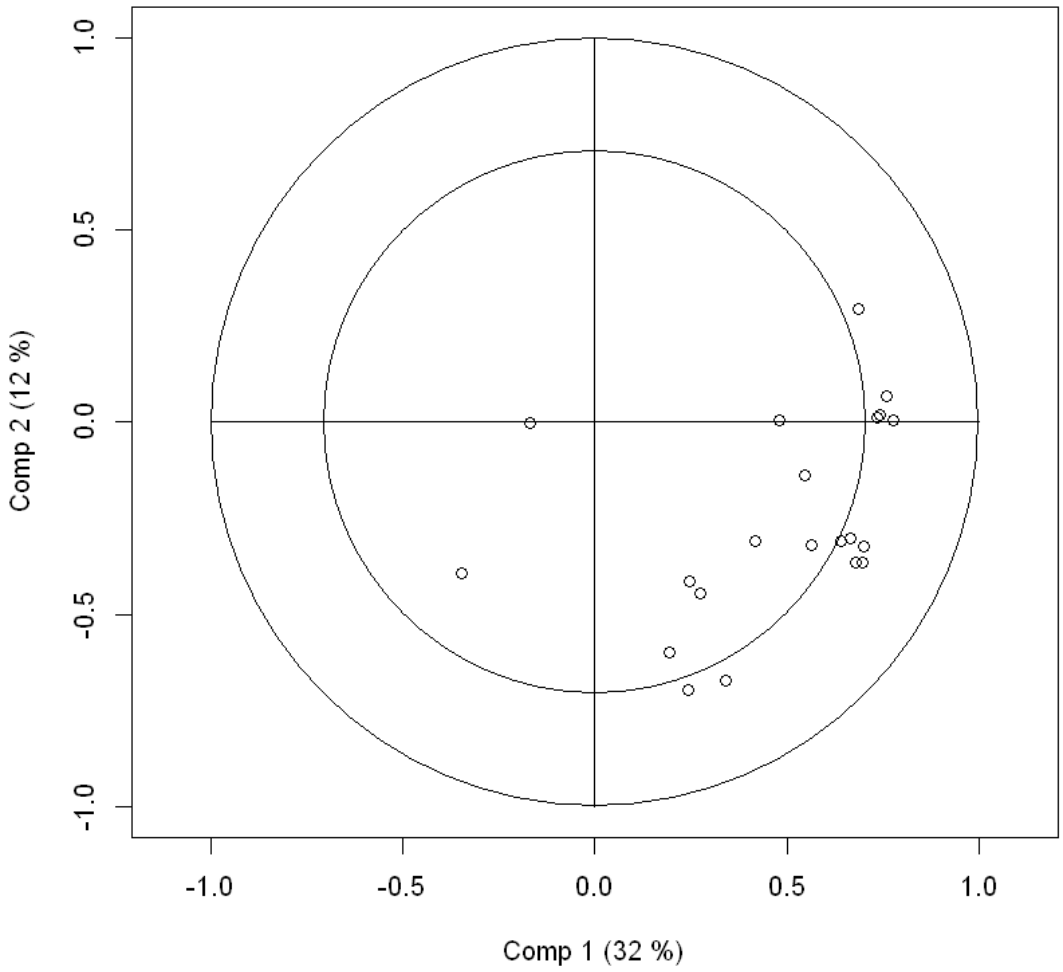
```
Partial Least Squares

76 samples
21 predictors

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 69, 68, 69, 68, 68, 68, ...
Resampling results across tuning parameters:

  ncomp  RMSE       Rsquared
   1     0.7363147  0.3715763
   2     0.6666783  0.4866863
   3     0.6169916  0.5359418
   4     0.6247479  0.5431368
   5     0.6268527  0.5665616
   6     0.6304428  0.5699974
   7     0.6390508  0.5588724
   8     0.6447784  0.5496344
   9     0.6436157  0.5488394
  10     0.6484118  0.5382332

RMSE was used to select the optimal model using  the smallest value.
The final value used for the model was ncomp = 3.
```

```
In [284]: explvar(pls$finalModel)
          biplot(pls$finalModel,which="x")
          biplot(pls$finalModel,which="scores")
          plot(pls$finalModel,plottype="correlation")
```

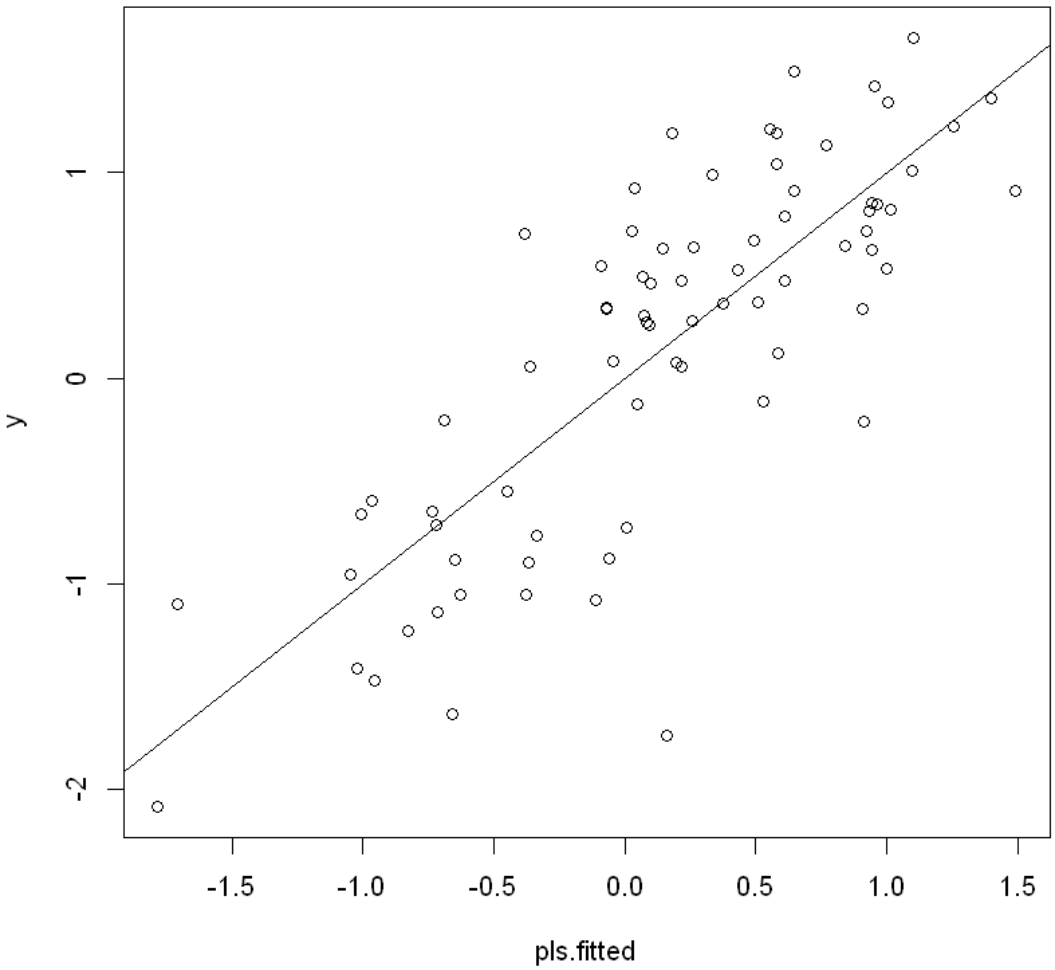
Comp 1 31.5576018451974 **Comp 2** 11.8273390844566 **Comp 3** 8.96483205193238




X scores and Y scores
Comp 1
Comp 2
-3
-2
-1
0
1
-8
-6
-4
-2
0
2
4
X40
X68
X34
X14
X69
X25
X35
X38
X68
X40
X64
X38
X31
X15
X72
X72
X15
X65
X37
X55
X31
X43
X44
X46
X33
X47
X59
X55
X43
X63

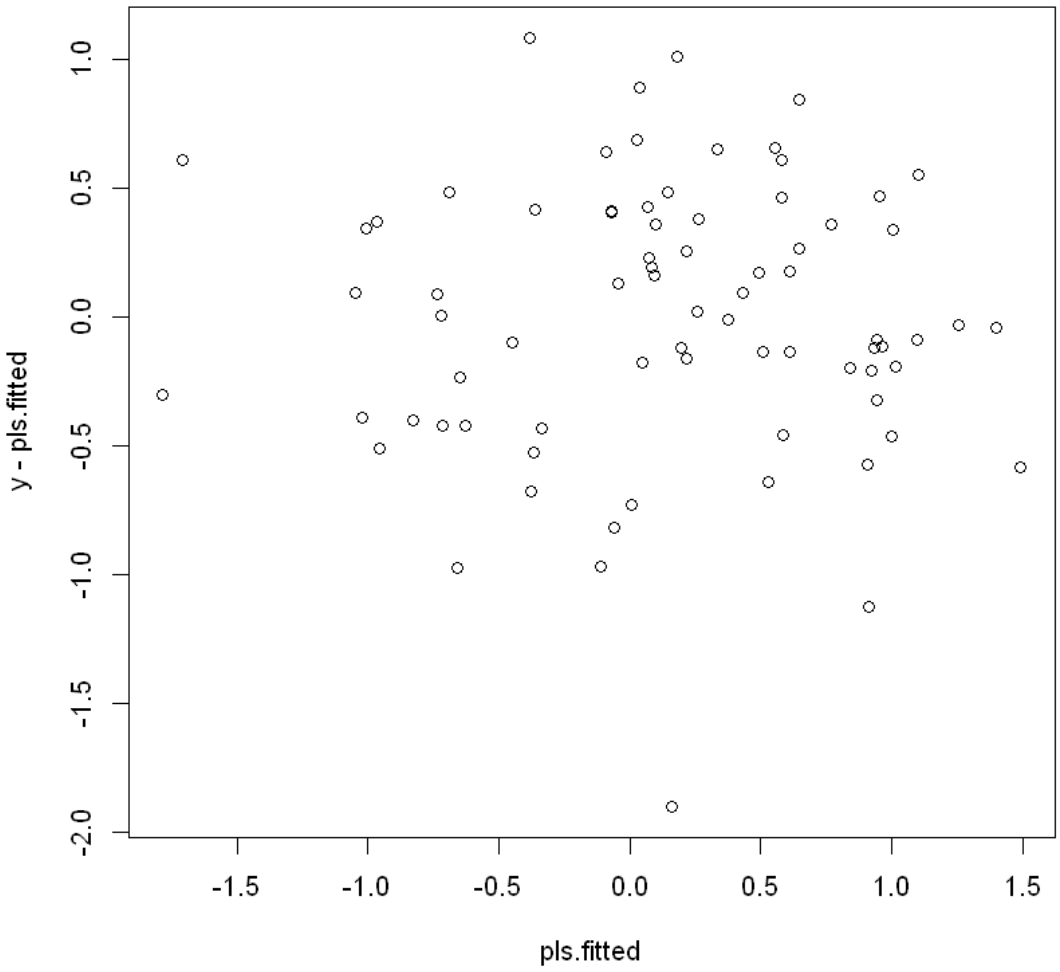


In [418]:
```
pls.fitted<-predict(pls$finalModel,ncomp=pls$finalModel$ncomp)
plot(pls.fitted,y)
abline(a = 0, b = 1)
plot(pls.fitted,y-pls.fitted)
sqrt(mean((y-pls.fitted)^2))
cor(pls.fitted,y)
```



0.526525718211619

0.803357749107699



2.1.6 SVM with Linear Kernel

```
In [286]: svmLinear <- train(x = X, y = y,
                             method = "svmLinear",
                             #preProcess = "pca",
                             trControl = trControl,
                            tuneGrid=data.frame(C=seq(0.01,0.1,0.01)))
          svmLinear
```

```
Support Vector Machines with Linear Kernel
```

```
76 samples
21 predictors

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 69, 68, 69, 68, 68, 68, ...
Resampling results across tuning parameters:

  C     RMSE       Rsquared
  0.01  0.6393900  0.5485496
  0.02  0.6343082  0.5403752
  0.03  0.6425667  0.5345902
  0.04  0.6349063  0.5535948
  0.05  0.6344177  0.5557623
  0.06  0.6358040  0.5553976
  0.07  0.6392375  0.5572223
  0.08  0.6409081  0.5564324
  0.09  0.6412407  0.5561485
  0.10  0.6434297  0.5528126

RMSE was used to select the optimal model using  the smallest value.
The final value used for the model was C = 0.02.
```

2.1.7 SVM with Radial Kernel

```
In [287]: set.seed(1)
          svmRadial <- train(x = X, y = y,
                             method = "svmRadial",
                             #preProcess = "pca",
                             tuneLength = 10,
                             trControl = trControl)
          svmRadial_knn <- train(x = powder.imputed.knn, y = y,
                             method = "svmRadial",
                             #preProcess = "pca",
                             tuneLength = 10,
```

```
                        trControl = trControl)
svmRadial_mean <- train(x = powder.imputed.mean, y = y,
                        method = "svmRadial",
                        #preProcess = "pca",
                        tuneLength = 10,
                        trControl = trControl)
svmRadial
svmRadial_knn
svmRadial_mean
```

```
Support Vector Machines with Radial Basis Function Kernel

76 samples
21 predictors

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 69, 68, 69, 68, 68, 68, ...
Resampling results across tuning parameters:

  C       RMSE       Rsquared
    0.25  0.7356526  0.5128641
    0.50  0.6786831  0.5385720
    1.00  0.6231002  0.5792917
    2.00  0.6087199  0.5751504
    4.00  0.6140620  0.5463536
    8.00  0.6263417  0.4956231
   16.00  0.6249902  0.4957795
   32.00  0.6249902  0.4957795
   64.00  0.6249902  0.4957795
  128.00  0.6249902  0.4957795

Tuning parameter 'sigma' was held constant at a value of 0.03078952
RMSE was used to select the optimal model using  the smallest value.
The final values used for the model were sigma = 0.03078952 and C = 2.


Support Vector Machines with Radial Basis Function Kernel
```

```
76 samples
21 predictors

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 69, 68, 69, 68, 68, 68, ...
Resampling results across tuning parameters:

  C       RMSE       Rsquared
    0.25  0.7386020  0.4810363
    0.50  0.6778591  0.5115831
    1.00  0.6244066  0.5581174
    2.00  0.6199040  0.5429144
    4.00  0.6238938  0.5006477
    8.00  0.6460970  0.4368309
   16.00  0.6466659  0.4357171
   32.00  0.6466659  0.4357171
   64.00  0.6466659  0.4357171
  128.00  0.6466659  0.4357171

Tuning parameter 'sigma' was held constant at a value of 0.0337301
RMSE was used to select the optimal model using  the smallest value.
The final values used for the model were sigma = 0.0337301 and C = 2.


Support Vector Machines with Radial Basis Function Kernel

76 samples
21 predictors

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 69, 68, 69, 68, 68, 68, ...
Resampling results across tuning parameters:

  C       RMSE       Rsquared
```

```
    0.25  0.7446500  0.4649193
    0.50  0.6864499  0.4913826
    1.00  0.6492914  0.5204014
    2.00  0.6400298  0.5009061
    4.00  0.6398167  0.4650793
    8.00  0.6459632  0.4491844
   16.00  0.6459632  0.4491844
   32.00  0.6459632  0.4491844
   64.00  0.6459632  0.4491844
  128.00  0.6459632  0.4491844

Tuning parameter 'sigma' was held constant at a value of 0.04772906
RMSE was used to select the optimal model using  the smallest value.
The final values used for the model were sigma = 0.04772906 and C = 4.


In [288]: svmRadial$finalModel
          svmRadial$resample

Support Vector Machine object of class "ksvm"

SV type: eps-svr  (regression)
 parameter : epsilon = 0.1  cost C = 2

Gaussian Radial Basis kernel function.
 Hyperparameter : sigma =  0.0307895168950236

Number of Support Vectors : 70

Objective Function Value : -39.9296
Training error : 0.136575
```

RMSE	Rsquared	Resample
0.6261961	0.4181973	Fold03
0.3618959	0.8773446	Fold02
0.5654572	0.4066675	Fold06
0.5819365	0.4027495	Fold07
0.4926020	0.6360614	Fold01
0.7029894	0.6634662	Fold05
0.5444340	0.7929325	Fold09
0.6320659	0.4246423	Fold10
0.9384870	0.2531872	Fold04
0.6411350	0.8762554	Fold08

2.2 Comparing Models According to 10-fold CV

反映模型评价能力的准则：10-fold CV MSE（或 RMSE），认为该值越低的模型预测能力越强。

```
In [302]: methods<-list(OLS = OLS, OLS_mean=OLS_mean,OLS_knn=OLS_knn, OLS.restricted=OLS.restri
                        lmStepAIC=lmStepAIC,lmStepBIC=lmStepBIC, svmLinear = svmLinear,
                                  svmRadial=svmRadial, svmRadial_mean=svmRadial_mean,svmRadial
                                  lasso = lasso,pls = pls,enet=enet,enet_mean=enet_mean,enet_l
                                  )
          resamp <- resamples(methods)
          summary(resamp)$statistics$RMSE
          summary(resamp)$statistics$Rsq
```

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
OLS	0.4245	0.5606	0.6383	0.6578	0.7554	1.0080	0
OLS_mean	0.4801	0.5416	0.6432	0.6931	0.8325	1.0360	0
OLS_knn	0.4973	0.5362	0.6238	0.6777	0.7807	1.0040	0
OLS.restricted	0.3115	0.4192	0.5976	0.5867	0.7260	0.9356	0
lmStepAIC	0.3949	0.5870	0.6264	0.6476	0.6745	1.0080	0
lmStepBIC	0.3870	0.6013	0.7014	0.6645	0.7308	1.0240	0
svmLinear	0.3933	0.5001	0.6226	0.6343	0.7592	0.9078	0
svmRadial	0.3619	0.5497	0.6041	0.6087	0.6389	0.9385	0
svmRadial_mean	0.3616	0.5962	0.6203	0.6398	0.6694	1.0000	0
svmRadial_knn	0.3651	0.5194	0.6299	0.6199	0.6700	0.9584	0
lasso	0.3790	0.5219	0.6052	0.6118	0.6986	0.9280	0
pls	0.3513	0.5131	0.5623	0.6170	0.7319	0.9749	0
enet	0.3796	0.5222	0.6036	0.6115	0.6987	0.9296	0
enet_mean	0.3856	0.4883	0.6194	0.6288	0.7078	0.9463	0
enet_knn	0.3764	0.5040	0.6123	0.6174	0.6888	0.9118	0
	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
OLS	0.2299	0.3752	0.5851	0.5342	0.6553	0.7722	0
OLS_mean	0.1882	0.3287	0.4705	0.4669	0.6050	0.7436	0
OLS_knn	0.2257	0.3771	0.5078	0.4894	0.6186	0.7493	0
OLS.restricted	0.2300	0.4518	0.5725	0.5770	0.7548	0.9174	0
lmStepAIC	0.1960	0.3366	0.5907	0.5115	0.6199	0.8051	0
lmStepBIC	0.1768	0.3259	0.5359	0.5038	0.6054	0.8570	0
svmLinear	0.2449	0.3620	0.6086	0.5404	0.6726	0.8298	0
svmRadial	0.2532	0.4095	0.5304	0.5752	0.7606	0.8773	0
svmRadial_mean	0.1393	0.2802	0.4092	0.4651	0.6506	0.8848	0
svmRadial_knn	0.1961	0.3247	0.5313	0.5429	0.7905	0.8821	0
lasso	0.2362	0.3315	0.5999	0.5475	0.7259	0.8030	0
pls	0.1614	0.3365	0.6082	0.5359	0.7295	0.8216	0
enet	0.2379	0.3314	0.6006	0.5485	0.7254	0.8078	0
enet_mean	0.2204	0.3372	0.5267	0.5206	0.7426	0.7531	0
enet_knn	0.2615	0.2983	0.5858	0.5283	0.7170	0.7602	0

2.3 对前 10 个样本的预测

```
In [401]: newx<- powder.scaled.test[,predictors.index]
          newy<-powder.scaled.test$factor1
```

```
sd(newy)
RMSE.record<-c()
yhat.record<-data.frame(id=1:10,ytrue=newy)
yhat.record["OLS"]<-predict(OLS$finalModel,newdata=newx)
yhat.record["OLS_mean"]<-predict(OLS_mean$finalModel,newdata=newx)
yhat.record["OLS_knn"]<-predict(OLS_knn$finalModel,newdata=newx)
yhat.record["OLS.restricted"]<-predict(OLS.restricted$finalModel,newdata=newx)
yhat.record["pls"]<-predict(pls$finalModel,newdata=newx, ncomp = pls$finalModel$ncomp
yhat.record["lasso"]<-predict(lasso$finalModel,newx=newx,s=lasso$bestTune$fraction,mo
yhat.record["enet"]<-predict(enet$finalModel,newx=newx,s=enet$bestTune$fraction,mode=
yhat.record["svmLinear"]<-predict(svmLinear$finalModel,newdata=newx)
yhat.record["svmRadial"]<-predict(svmRadial$finalModel,newdata=newx)
yhat.record["svmRadial_mean"]<-predict(svmRadial_mean$finalModel,newdata=newx)
yhat.record["svmRadial_knn"]<-predict(svmRadial_knn$finalModel,newdata=newx)

#yhat.record["mean"]<-rowMeans(yhat.record[,-c(1,2)])
for (method in colnames(yhat.record)[-c(1,2)]){
    RMSE.record[method]<-sqrt(mean((yhat.record[method]-newy)^2))
}
yhat.record
RMSE.record[order(RMSE.record)]
# for (method in colnames(yhat.record)[-c(1,2)]){
#     RMSE.record[method]<-sqrt(mean((yhat.record[-8,method]-newy[-8])^2))
# }
# RMSE.record[order(RMSE.record)]
```

1.02941708089028

id	ytrue	OLS	OLS_mean	OLS_knn	OLS.restricted	pls	lasso	en
1	-1.5855122	-1.235586547	-1.2702002	-1.33806520	-1.2735787	-1.26228901	-1.0794214	-1.
2	-2.3714088	-1.150952322	-1.0650698	-1.11977439	-1.0020097	-1.08654185	-1.0878594	-1.
3	-0.2747798	0.001373642	0.1986472	0.23418961	0.5773681	0.03459549	0.3132947	0.3
4	-1.0845762	-1.090128847	-0.9983714	-1.07490514	-1.9429371	-1.00317991	-0.9603596	-0.
5	-2.1562664	-2.210344863	-2.1558682	-2.18362914	-2.6699820	-2.14980363	-1.9677296	-1.
6	-0.9666176	0.003946920	-0.0435570	-0.06115095	-0.5695284	-0.10905434	-0.2429749	-0.
7	0.4459457	0.577643914	0.6286399	0.76970469	0.7959296	0.71360882	0.7914424	0.7
8	-1.5368518	0.411712896	0.5612333	0.55505744	0.2084053	0.60938670	0.6076258	0.6
9	-2.1313065	-0.941475656	-0.9143817	-0.92239325	-0.8548945	-0.59622142	-0.5532731	-0.
10	0.3943648	0.570936897	0.5388808	0.52756880	1.0405667	0.64252885	0.7463014	0.7

OLS 0.888498574010219 **OLS_knn** 0.931004784256062 **OLS_mean** 0.939504003531406 **svmRadial** 0.949728593254381 **OLS.restricted** 0.952485959846832 **pls** 0.984337636871385 **enet** 1.00603535571914 **lasso** 1.00778440569384 **svmRadial_mean** 1.01721789658544 **svmRadial_knn** 1.01815496024335 **svmLinear** 1.09452607343546 **svmPoly** 1.30811761494288

2.4 考虑平方项或交互项

自变量总共有 21 个，如果考虑平方项则又有 21 个，如果考虑交互项则有 $\binom{21}{2} = 210$ 个，甚至已经超出了样本得个数（$n = 86$），这就要求只考虑有显著影响的变量。（注：之后一律采用标准化、impute.knn 的数据）

```
In [303]: powder.augmented<-powder[,predictors.index]
          for (PP in (paste("PP",1:21,sep=""))){
              powder.augmented[paste(PP,"_sqrd",sep = "")]<-powder[PP]^2
          }
          # head(powder.augmented)
          # colnames(powder.augmented)
```

```
In [304]: predictors.sqrt.index<-grep("PP\\d+[_]sqrd", colnames(powder.augmented),value=T)
          #powder.augmented[,predictors.sqrt.index]<-scale(powder.augmented[,predictors.sqrt.i
          head(powder.augmented[,predictors.sqrt.index])
```

PP1_sqrd	PP2_sqrd	PP3_sqrd	PP4_sqrd	PP5_sqrd	PP6_sqrd	PP7_sqrd	PP8_sqrd	PP9_sqrd
11940480	2.9584	22.7529	10555.508	35160.000	2304.960	2465.122	1812.2049	272.9104
10572252	0.0225	48.5809	8233.748	24973.481	2191.176	2478.048	3264.9796	479.6100
20299530	3.2761	52.2729	9217.920	28180.337	2674.958	2747.856	5077.9876	2288.6656
10650432	4.6656	49.9849	7421.823	23941.373	2247.708	1830.984	910.8324	326.8864
2563201	3.5344	63.8401	10633.734	34032.870	2320.349	1848.140	2257.2001	137.1241
22198232	0.0121	14.2884	3153.946	7633.517	2249.605	1928.088	1946.5744	125.2161

```
In [305]: combni<-combn(1:21,2)
          for (i in (1:ncol(combni))){
              powder.augmented[paste("PP_",combni[1,i],"_",combni[2,i],sep = "")]<-powder[,comb
          }
          #head(powder.augmented)
          #colnames(powder.augmented);ncol(powder.augmented)
```

```
In [306]: predictors.ia.index<-grep("PP[_]", colnames(powder.augmented),value=T) # 交互项 index
          predictors.tot.index<-c(predictors.index,predictors.sqrt.index,predictors.ia.index)
```

在加入了这些新的变量之后，仍然采用 knn 填补

```
In [307]: powder.augmented.imputed.knn <-scale(powder.augmented)
          powder.augmented.imputed.knn[,predictors.index]<-t(impute::impute.knn(t(powder.augmer
          powder.augmented.imputed.knn<-t(impute::impute.knn(t(powder.augmented.imputed.knn))[
```

2.5 仍然分训练集和测试集

前 10 个样本为测试集，其余为训练集。

```
In [308]: X.aug<-powder.augmented.imputed.knn.train<-powder.augmented.imputed.knn[-(1:10),]
          powder.augmented.imputed.knn.test<-powder.augmented.imputed.knn[1:10,]
          head(X.aug);ncol(X.aug)
```

	PP1	PP2	PP3	PP4	PP5	PP6	PP7	PP8
11	-0.1000865	1.5261886	-0.90522847	-0.5449897	-0.57851441	0.77232684	0.23936127	0.06241
12	0.1947519	-0.3464974	0.04605416	0.3845061	0.54192259	0.58518611	-0.05079632	-0.1918
13	0.3425397	0.1991413	-1.54445012	0.1125063	-0.03167724	1.86843686	-0.78263826	1.01417
14	1.2233695	-2.7176683	1.46542823	-2.3726719	-1.77804108	-0.05940976	0.50372708	-0.8032
15	-0.1999630	-0.7574106	-0.15024226	0.7002085	0.85708295	-0.22872756	-0.63433549	-0.1944
16	0.7361490	-1.4714563	1.49562768	0.4582135	0.10453275	1.37236539	0.23291333	0.94116

252

```
In [402]: head(colnames(X.aug)[order(abs(cor(X.aug,y)),decreasing = T)],20)
          head(cor(X.aug,y)[order(abs(cor(X.aug,y)),decreasing = T)],20)
```

1. 'PP_2_17' 2. 'PP_2_18' 3. 'PP_2_19' 4. 'PP_2_20' 5. 'PP_2_11' 6. 'PP_2_13' 7. 'PP_2_21' 8. 'PP_2_16' 9. 'PP1' 10. 'PP_2_12' 11. 'PP_2_15' 12. 'PP_2_14' 13. 'PP_2_8' 14. 'PP_1_2' 15. 'PP1_sqrd' 16. 'PP_2_7' 17. 'PP_2_10' 18. 'PP_18_20' 19. 'PP_17_18' 20. 'PP_1_10'

1. 0.591914130263642 2. 0.590494003665169 3. 0.588101976331521 4. 0.580718721110877 5. 0.574823467066635 6. 0.571234325013454 7. 0.566034413417578 8. 0.565139835195405 9. 0.563516651538441 10. 0.559612766741098 11. 0.555474007590351 12. 0.549282671340631 13. 0.544021393580157 14. 0.534924180302549 15. 0.524768634087107 16. 0.524153892074977 17. 0.523199136770847 18. 0.489031800318273 19. 0.476351237911445 20. 0.473652006624249

2.5.1 Stepwise Regression

```
In [354]: lmStepBIC.aug <- train(x = X.aug[,c(grep("PP[_]2",colnames(X.aug),value = T),predict
                                  y = y,
                                  method = "lmStepAIC",trace=F,k=log(nrow(X.aug)),
```

```
                                #preProcess = "pca",
                                trControl = trControl)
          lmStepBIC.aug
          summary(lmStepBIC.aug$finalModel)
```

```
Linear Regression with Stepwise Selection

76 samples
41 predictors

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 69, 68, 69, 68, 68, 68, ...
Resampling results:

  RMSE       Rsquared
  0.8183479  0.3905227




Call:
lm(formula = .outcome ~ PP_2_4 + PP_2_5 + PP_2_6 + PP_2_8 + PP_2_11 +
    PP_2_12 + PP_2_14 + PP_2_15 + PP_2_16 + PP_2_17 + PP_2_19 +
    PP_2_20 + PP_20_21 + PP2 + PP4 + PP5 + PP11 + PP13 + PP14 +
    PP16 + PP18 + PP19, data = dat)

Residuals:
     Min       1Q   Median       3Q      Max
-1.22866 -0.25288  0.03902  0.21697  0.85571

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.05896    0.06261  -0.942 0.350654
PP_2_4       -0.73082    0.10091  -7.242 1.84e-09 ***
PP_2_5       -2.84844    1.19505  -2.384 0.020762 *
PP_2_6        2.95227    1.36970   2.155 0.035691 *
```

```
PP_2_8        0.75014    0.33382   2.247 0.028819 *
PP_2_11       2.82678    0.76313   3.704 0.000507 ***
PP_2_12     -22.61118    5.29503  -4.270 8.13e-05 ***
PP_2_14       5.34036    2.44808   2.181 0.033607 *
PP_2_15      11.45102    4.31590   2.653 0.010498 *
PP_2_16       2.96623    1.15983   2.557 0.013441 *
PP_2_17      -7.94470    2.71796  -2.923 0.005087 **
PP_2_19      16.03747    5.13554   3.123 0.002901 **
PP_2_20      -8.78415    3.36342  -2.612 0.011694 *
PP_20_21     -0.74966    0.27176  -2.759 0.007951 **
PP2          -0.63322    0.10536  -6.010 1.75e-07 ***
PP4           1.95263    0.69609   2.805 0.007017 **
PP5          -2.21596    0.75690  -2.928 0.005023 **
PP11          2.44014    0.61098   3.994 0.000202 ***
PP13         -1.09787    0.40107  -2.737 0.008412 **
PP14         -1.55720    0.49985  -3.115 0.002964 **
PP16          1.24889    0.38793   3.219 0.002195 **
PP18         -1.33604    0.49582  -2.695 0.009419 **
PP19          1.59312    0.49557   3.215 0.002225 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4645 on 53 degrees of freedom
Multiple R-squared:  0.8076,Adjusted R-squared:  0.7277
F-statistic: 10.11 on 22 and 53 DF,  p-value: 4.15e-12
```

似乎在加入了更多的自变量（平方项）后，逐步回归的效果反而变差了。（RMSE 不降反增）。

2.5.2 Partial Least Square

```
In [380]: pls.aug <- train(x = X.aug,
                           #x = powder.augmented.imputed.knn[,predictors.index],
                                       y = y,
                                       method = "pls",
                                       #preProcess = "pca",
                                       tuneLength=10,
```

```
                          trControl = trControl)
          pls.aug
```

```
Partial Least Squares

 76 samples
252 predictors

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 69, 68, 69, 68, 68, 68, ...
Resampling results across tuning parameters:

  ncomp  RMSE       Rsquared
   1     0.7049500  0.4168898
   2     0.5882099  0.5892351
   3     0.5488371  0.6390946
   4     0.5494725  0.6525460
   5     0.5601271  0.6330738
   6     0.5537498  0.6378837
   7     0.5636878  0.6379009
   8     0.5662743  0.6247688
   9     0.5779131  0.6032797
  10     0.5886000  0.5865689

RMSE was used to select the optimal model using  the smallest value.
The final value used for the model was ncomp = 3.
```

```
In [381]: summary(pls.aug$finalModel)
          class(pls.aug$finalModel)
          pls.aug$finalModel$loadings
          pls.aug$finalModel$loading.weights
          pls.aug$finalModel$projection
```

```
Data:         X dimension: 76 252
        Y dimension: 76 1
Fit method: oscorespls
```

```
Number of components considered: 3
TRAINING: % variance explained
          1 comps  2 comps  3 comps
X           30.96    44.28    55.37
.outcome    40.72    63.11    67.66
```

'mvr'

```
Loadings:
          Comp 1 Comp 2 Comp 3
PP1        0.103
PP2                      0.160
PP3                     -0.206
PP4              -0.102
PP5
PP6              -0.107
PP7
PP8              -0.104
PP9
PP10
PP11
PP12
PP13
PP14
PP15
PP16
PP17
PP18
PP19
PP20
PP21
PP1_sqrd
PP2_sqrd                -0.118
PP3_sqrd                -0.201
PP4_sqrd
PP5_sqrd
```

PP6_sqrd	-0.109		
PP7_sqrd			
PP8_sqrd	-0.104		
PP9_sqrd			
PP10_sqrd			
PP11_sqrd			
PP12_sqrd			
PP13_sqrd			
PP14_sqrd			
PP15_sqrd			
PP16_sqrd			
PP17_sqrd			
PP18_sqrd			
PP19_sqrd			
PP20_sqrd			
PP21_sqrd			
PP_1_2			
PP_1_3		0.150	
PP_1_4		-0.111	
PP_1_5			
PP_1_6			
PP_1_7			
PP_1_8			
PP_1_9			
PP_1_10			
PP_1_11			
PP_1_12			
PP_1_13			
PP_1_14			
PP_1_15			
PP_1_16			
PP_1_17			
PP_1_18			
PP_1_19			
PP_1_20			
PP_1_21			
PP_2_3		0.166	

PP_2_4		-0.202
PP_2_5	0.101	
PP_2_6		
PP_2_7	0.104	
PP_2_8	0.105	
PP_2_9		
PP_2_10		
PP_2_11	0.106	
PP_2_12	0.106	
PP_2_13	0.108	
PP_2_14	0.106	
PP_2_15	0.106	
PP_2_16	0.106	
PP_2_17	0.109	
PP_2_18	0.108	
PP_2_19	0.108	
PP_2_20	0.107	
PP_2_21	0.106	
PP_3_4		0.161
PP_3_5		0.136
PP_3_6		0.136
PP_3_7		0.161
PP_3_8		0.161
PP_3_9		0.159
PP_3_10		0.160
PP_3_11		0.160
PP_3_12		0.162
PP_3_13		0.161
PP_3_14		0.160
PP_3_15		0.161
PP_3_16		0.161
PP_3_17		0.161
PP_3_18		0.161
PP_3_19		0.161
PP_3_20		0.161
PP_3_21		0.161
PP_4_5		-0.164

PP_4_6		-0.170
PP_4_7		-0.203
PP_4_8		-0.203
PP_4_9		-0.182
PP_4_10		-0.135
PP_4_11		-0.207
PP_4_12		-0.211
PP_4_13		-0.211
PP_4_14		-0.208
PP_4_15		-0.210
PP_4_16		-0.209
PP_4_17		-0.209
PP_4_18		-0.210
PP_4_19		-0.209
PP_4_20		-0.209
PP_4_21		-0.209
PP_5_6		
PP_5_7	-0.114	
PP_5_8	-0.110	
PP_5_9		
PP_5_10		
PP_5_11	-0.105	
PP_5_12	-0.103	
PP_5_13	-0.102	
PP_5_14	-0.105	
PP_5_15	-0.103	
PP_5_16	-0.103	
PP_5_17	-0.105	
PP_5_18	-0.103	
PP_5_19	-0.102	
PP_5_20	-0.104	
PP_5_21	-0.102	
PP_6_7	-0.113	
PP_6_8	-0.109	
PP_6_9		
PP_6_10		
PP_6_11	-0.104	

PP_6_12	-0.102
PP_6_13	-0.102
PP_6_14	-0.103
PP_6_15	-0.102
PP_6_16	-0.102
PP_6_17	-0.105
PP_6_18	-0.102
PP_6_19	-0.101
PP_6_20	-0.104
PP_6_21	-0.102
PP_7_8	-0.115
PP_7_9	-0.120
PP_7_10	
PP_7_11	-0.109
PP_7_12	-0.110
PP_7_13	-0.108
PP_7_14	-0.120
PP_7_15	-0.109
PP_7_16	-0.112
PP_7_17	
PP_7_18	
PP_7_19	
PP_7_20	
PP_7_21	-0.107
PP_8_9	-0.111
PP_8_10	
PP_8_11	
PP_8_12	
PP_8_13	
PP_8_14	
PP_8_15	
PP_8_16	
PP_8_17	
PP_8_18	
PP_8_19	
PP_8_20	
PP_8_21	

PP_9_10	
PP_9_11	-0.106
PP_9_12	-0.104
PP_9_13	-0.104
PP_9_14	-0.115
PP_9_15	-0.105
PP_9_16	-0.105
PP_9_17	
PP_9_18	
PP_9_19	
PP_9_20	-0.100
PP_9_21	-0.103
PP_10_11	
PP_10_12	
PP_10_13	
PP_10_14	
PP_10_15	
PP_10_16	
PP_10_17	
PP_10_18	
PP_10_19	
PP_10_20	
PP_10_21	
PP_11_12	
PP_11_13	
PP_11_14	
PP_11_15	
PP_11_16	
PP_11_17	
PP_11_18	
PP_11_19	
PP_11_20	
PP_11_21	
PP_12_13	
PP_12_14	
PP_12_15	
PP_12_16	

PP_12_17 0.102
PP_12_18
PP_12_19
PP_12_20 0.105
PP_12_21
PP_13_14
PP_13_15
PP_13_16
PP_13_17 0.102
PP_13_18
PP_13_19
PP_13_20 0.101
PP_13_21
PP_14_15
PP_14_16
PP_14_17
PP_14_18
PP_14_19
PP_14_20
PP_14_21
PP_15_16
PP_15_17 0.113
PP_15_18 0.106
PP_15_19 0.109
PP_15_20 0.108
PP_15_21
PP_16_17 0.108
PP_16_18 0.109
PP_16_19 0.108
PP_16_20 0.105
PP_16_21
PP_17_18
PP_17_19
PP_17_20
PP_17_21 0.104
PP_18_19
PP_18_20 0.105

```
PP_18_21   0.105
PP_19_20   0.102
PP_19_21   0.105
PP_20_21

               Comp 1 Comp 2 Comp 3
SS loadings     1.269  1.117  1.610
Proportion Var  0.005  0.004  0.006
Cumulative Var  0.005  0.009  0.016



Loadings:
          Comp 1 Comp 2 Comp 3
PP1        0.119
PP2
PP3                    -0.123
PP4
PP5
PP6
PP7
PP8
PP9
PP10
PP11                   -0.127
PP12
PP13
PP14                   -0.122
PP15
PP16
PP17
PP18
PP19
PP20
PP21
PP1_sqrd   0.110
PP2_sqrd               -0.185
```

```
PP3_sqrd                -0.124
PP4_sqrd
PP5_sqrd
PP6_sqrd
PP7_sqrd
PP8_sqrd
PP9_sqrd
PP10_sqrd
PP11_sqrd               -0.127
PP12_sqrd
PP13_sqrd
PP14_sqrd               -0.121
PP15_sqrd
PP16_sqrd
PP17_sqrd
PP18_sqrd
PP19_sqrd
PP20_sqrd
PP21_sqrd
PP_1_2     0.107
PP_1_3
PP_1_4
PP_1_5
PP_1_6
PP_1_7
PP_1_8
PP_1_9
PP_1_10
PP_1_11
PP_1_12
PP_1_13
PP_1_14
PP_1_15
PP_1_16
PP_1_17
PP_1_18
PP_1_19
```

PP_1_20		
PP_1_21		
PP_2_3		0.108
PP_2_4		-0.179
PP_2_5		
PP_2_6		
PP_2_7	0.111	
PP_2_8	0.115	
PP_2_9		
PP_2_10	0.107	
PP_2_11	0.120	
PP_2_12	0.117	
PP_2_13	0.119	
PP_2_14	0.115	
PP_2_15	0.117	
PP_2_16	0.119	
PP_2_17	0.124	
PP_2_18	0.124	
PP_2_19	0.123	
PP_2_20	0.122	
PP_2_21	0.119	
PP_3_4		0.122
PP_3_5		
PP_3_6		
PP_3_7		
PP_3_8		
PP_3_9		
PP_3_10		
PP_3_11		
PP_3_12		
PP_3_13		
PP_3_14		
PP_3_15		
PP_3_16		
PP_3_17		
PP_3_18		
PP_3_19		

PP_3_20	
PP_3_21	
PP_4_5	
PP_4_6	
PP_4_7	-0.114
PP_4_8	-0.113
PP_4_9	
PP_4_10	
PP_4_11	-0.116
PP_4_12	-0.137
PP_4_13	-0.132
PP_4_14	-0.122
PP_4_15	-0.136
PP_4_16	-0.128
PP_4_17	-0.124
PP_4_18	-0.132
PP_4_19	-0.130
PP_4_20	-0.123
PP_4_21	-0.128
PP_5_6	
PP_5_7	
PP_5_8	
PP_5_9	
PP_5_10	
PP_5_11	
PP_5_12	
PP_5_13	
PP_5_14	
PP_5_15	
PP_5_16	
PP_5_17	
PP_5_18	
PP_5_19	
PP_5_20	
PP_5_21	
PP_6_7	
PP_6_8	

PP_6_9	
PP_6_10	
PP_6_11	
PP_6_12	
PP_6_13	
PP_6_14	
PP_6_15	
PP_6_16	
PP_6_17	
PP_6_18	
PP_6_19	
PP_6_20	
PP_6_21	
PP_7_8	
PP_7_9	-0.104
PP_7_10	
PP_7_11	
PP_7_12	-0.113
PP_7_13	-0.104
PP_7_14	-0.111
PP_7_15	-0.110
PP_7_16	
PP_7_17	
PP_7_18	
PP_7_19	
PP_7_20	
PP_7_21	
PP_8_9	
PP_8_10	
PP_8_11	
PP_8_12	
PP_8_13	
PP_8_14	
PP_8_15	
PP_8_16	
PP_8_17	
PP_8_18	

PP_8_19	
PP_8_20	
PP_8_21	
PP_9_10	
PP_9_11	
PP_9_12	-0.100
PP_9_13	
PP_9_14	-0.107
PP_9_15	-0.101
PP_9_16	
PP_9_17	
PP_9_18	
PP_9_19	
PP_9_20	
PP_9_21	
PP_10_11	
PP_10_12	
PP_10_13	
PP_10_14	
PP_10_15	
PP_10_16	
PP_10_17	
PP_10_18	
PP_10_19	
PP_10_20	
PP_10_21	
PP_11_12	
PP_11_13	
PP_11_14	
PP_11_15	
PP_11_16	
PP_11_17	
PP_11_18	
PP_11_19	
PP_11_20	
PP_11_21	
PP_12_13	

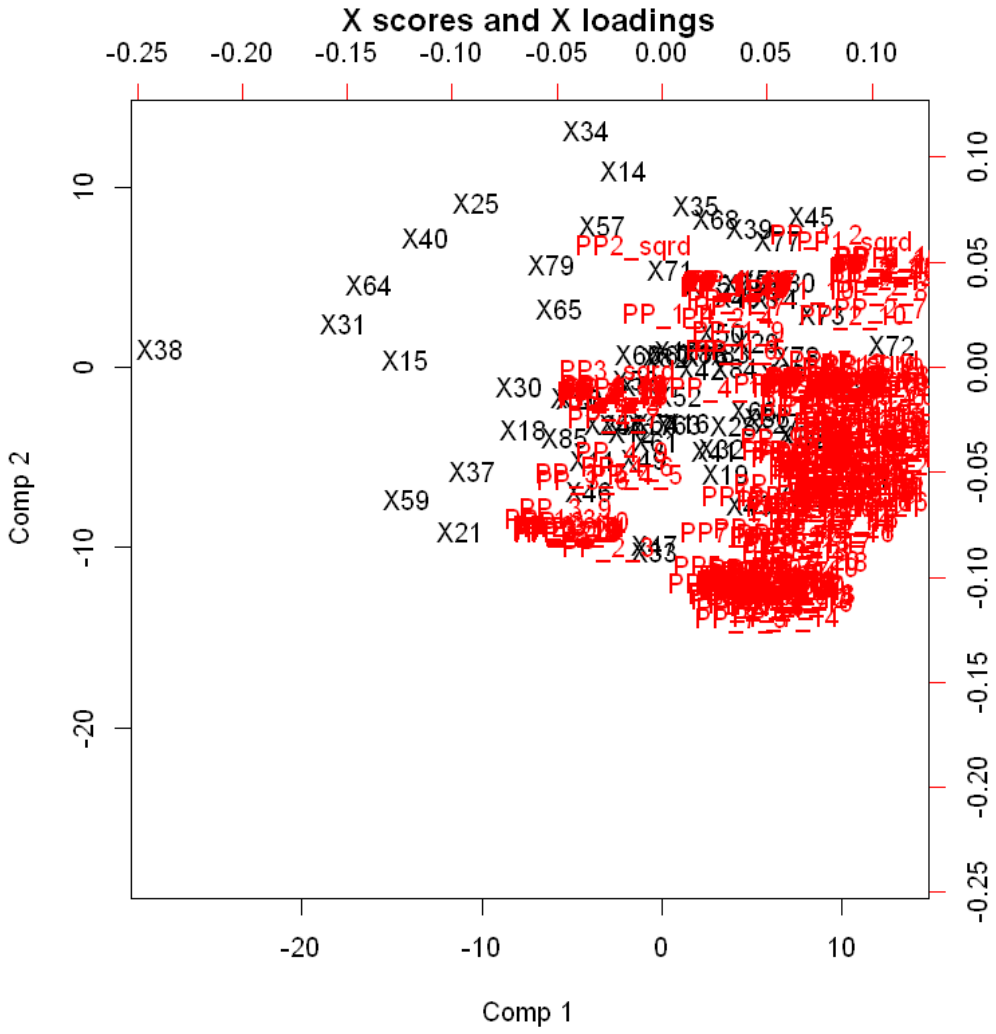
PP_12_14 -0.104
PP_12_15 -0.136
PP_12_16
PP_12_17
PP_12_18
PP_12_19
PP_12_20
PP_12_21
PP_13_14
PP_13_15
PP_13_16
PP_13_17
PP_13_18
PP_13_19
PP_13_20
PP_13_21
PP_14_15 -0.102
PP_14_16
PP_14_17
PP_14_18
PP_14_19
PP_14_20
PP_14_21
PP_15_16
PP_15_17
PP_15_18
PP_15_19
PP_15_20
PP_15_21
PP_16_17
PP_16_18
PP_16_19
PP_16_20
PP_16_21
PP_17_18 0.102
PP_17_19 0.100
PP_17_20

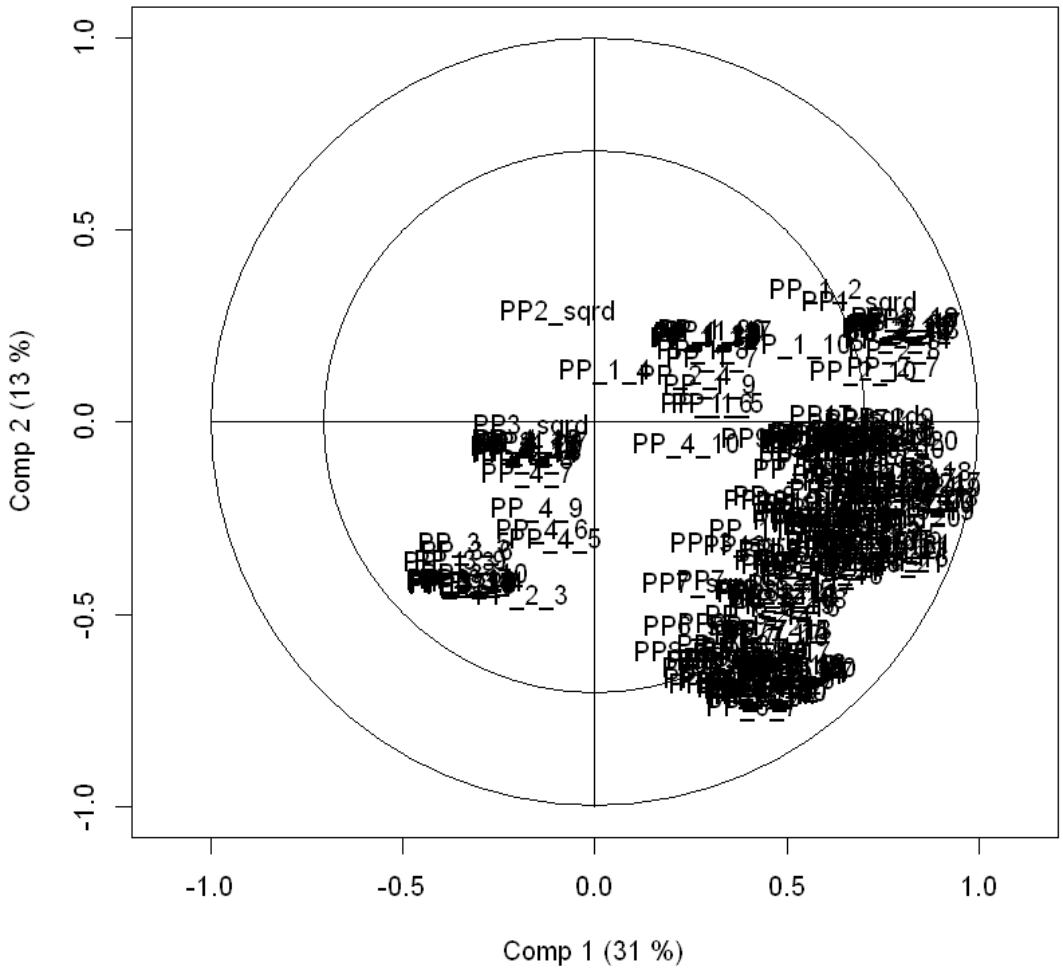
```
PP_17_21
PP_18_19
PP_18_20   0.103
PP_18_21
PP_19_20
PP_19_21
PP_20_21

                Comp 1 Comp 2 Comp 3
SS loadings      1.000  1.000  1.000
Proportion Var   0.004  0.004  0.004
Cumulative Var   0.004  0.008  0.012
```

	Comp 1	Comp 2	Comp 3
PP1	0.118759210	0.09149002	-0.035766133
PP2	-0.072409509	-0.09061438	0.042277453
PP3	-0.055474404	-0.08005158	-0.150769805
PP4	0.009646413	-0.07518959	0.036989695
PP5	-0.010200806	-0.09426079	-0.002032959
PP6	-0.006936449	-0.09444220	0.013558782
PP7	0.006389051	-0.05483053	0.034255844
PP8	-0.017556883	-0.09944592	0.004456789
PP9	0.082713952	0.06096664	0.090159742
PP10	0.059825142	0.01285354	0.067609998
PP11	0.028248507	-0.07271316	-0.152009192
PP12	0.047246894	-0.04498210	-0.073874483
PP13	0.012796823	-0.06554987	-0.050972228
PP14	0.041891221	-0.06583521	-0.144832909
PP15	0.054676541	-0.03739935	-0.016162240
PP16	0.094966066	0.05773261	0.072468186
PP17	0.091678197	0.04706831	0.008517196
PP18	0.090570072	0.03847118	0.006273106
PP19	0.082658086	0.03128104	0.029265391
PP20	0.052184559	-0.03861413	-0.042155469
PP21	0.040099784	-0.06039249	-0.065947953
PP1_sqrd	0.109750194	0.09132392	-0.042114447
PP2_sqrd	-0.016191825	-0.01401890	-0.189888503
PP3_sqrd	-0.044230936	-0.06728901	-0.147366952
PP4_sqrd	0.008673256	-0.07532407	0.029980954
PP5_sqrd	-0.011842822	-0.09324287	-0.009215439
PP6_sqrd	-0.007647740	-0.09589922	0.016639546
PP7_sqrd	0.004509228	-0.05876404	0.033520564
PP8_sqrd	-0.016234346	-0.09912730	0.003892938
PP9_sqrd	0.072309340	0.05770022	0.102671900
...	...	...	...
223	0.07746043	-0.0056675124	-0.021043936
224	0.05700647	-0.0477947052	-0.066116482
225	0.03087135	-0.0858029477	-0.113717055
226	0.03699514	-0.0669447847	-0.043832646
227	0.06411484	-0.0039003237	0.014988139
228	0.05996070	-0.0239384003	-0.033957176
229	0.05590734	-0.0329415529	-0.036996041
230	0.05230978	-0.0263921205	-0.014819007
231	0.03545315	-0.0706030108	-0.058612611

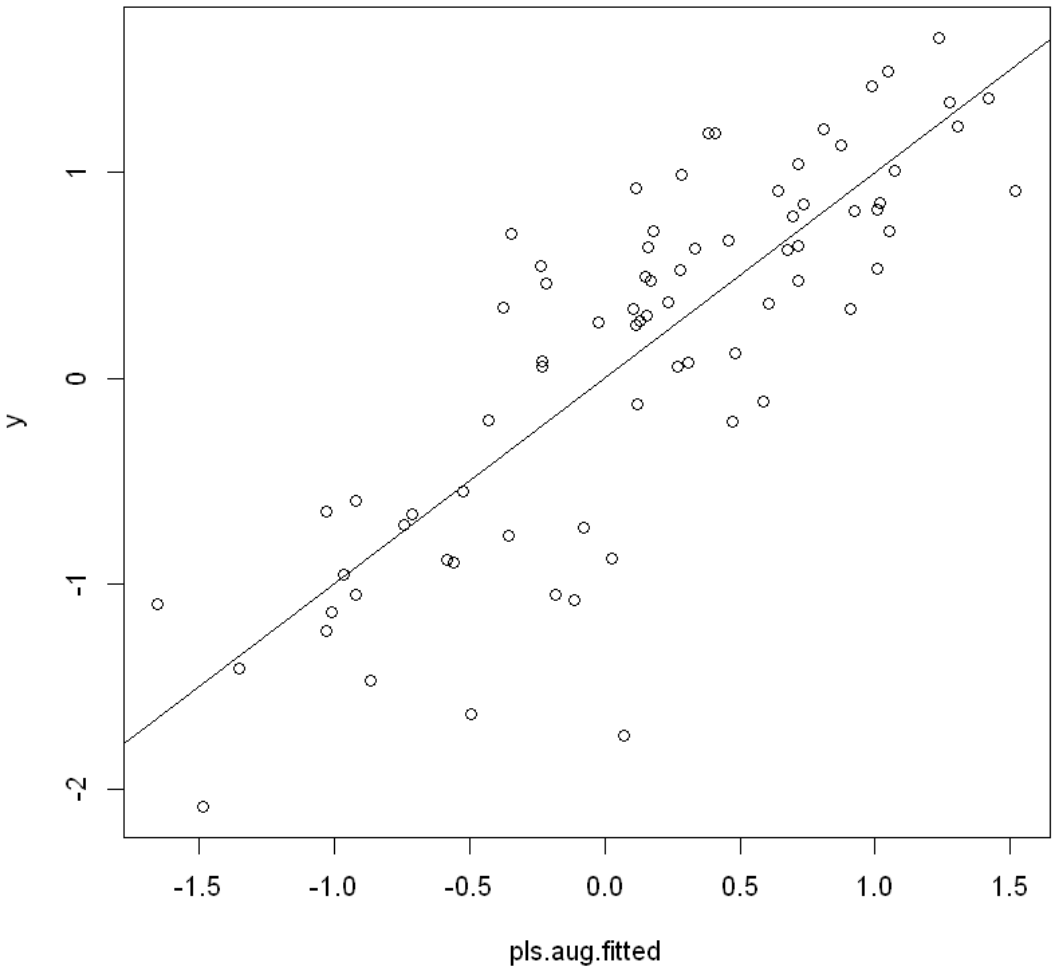
```
In [414]: explvar(pls.aug$finalModel)
          biplot(pls.aug$finalModel,which="x")
          #coef(pls.aug.finalModel)  # 252 个
          corrplot(pls.aug$finalModel,labels=colnames(X.aug))
```

Comp 1 30.9554231789298 **Comp 2** 13.3277593397863 **Comp 3** 11.0889187762478



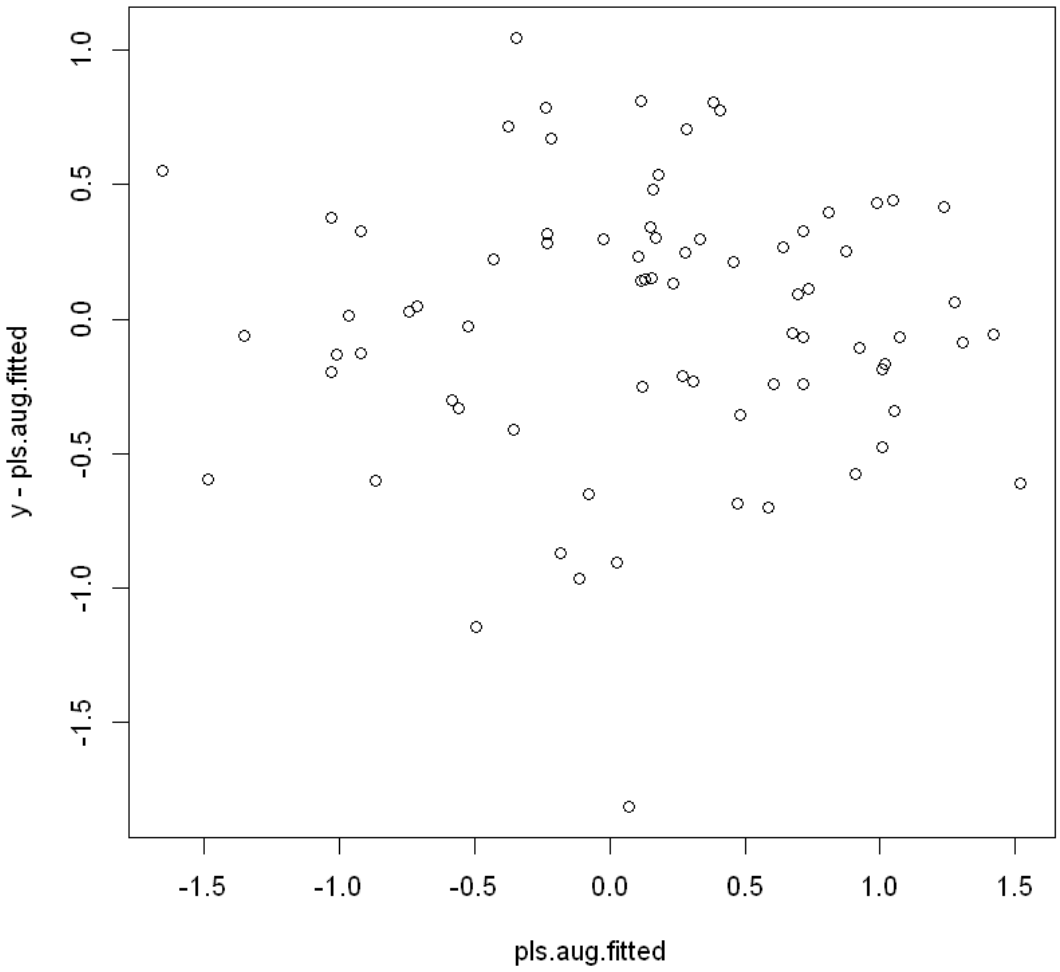



```
In [419]: pls.aug.fitted<-predict(pls.aug$finalModel,ncomp=pls.aug$bestTune$ncomp)
          plot(pls.aug.fitted,y)
          abline(a = 0, b = 1)
          plot(pls.aug.fitted,y-pls.aug.fitted)
          sqrt(mean((y-pls.aug.fitted)^2))
          cor(pls.aug.fitted,y)
```



0.502778736211973

0.822587160785388



2.5.3 Lasso

```
In [384]: lasso.aug <- train(x =X.aug, y = y,
                             method = "lasso",
                             #preProcess = "pca",
                             tuneGrid=data.frame(fraction=seq(0.01,0.1,0.01)),
                             trControl = trControl)
          lasso.aug
```

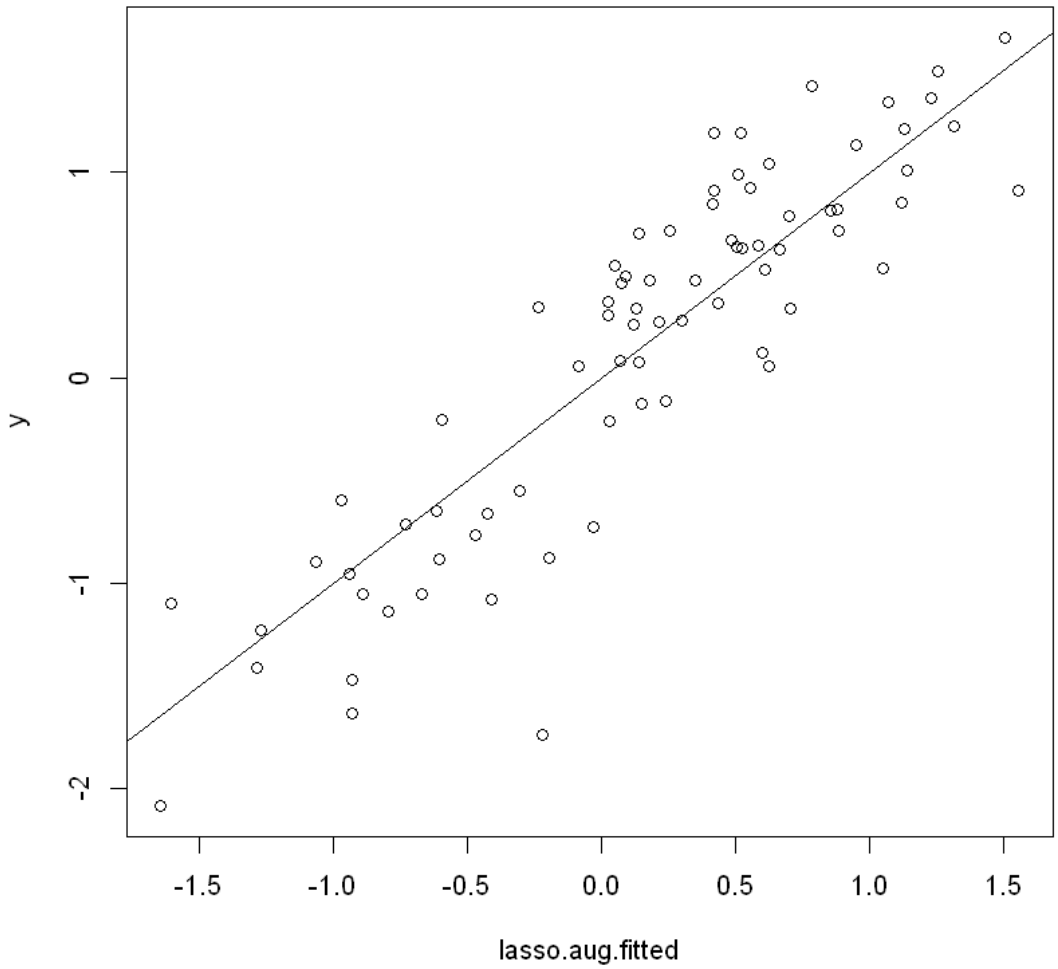
```
The lasso
```

```
 76 samples
252 predictors

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 69, 68, 69, 68, 68, 68, ...
Resampling results across tuning parameters:

  fraction  RMSE       Rsquared
  0.01      0.6513141  0.4614004
  0.02      0.5845682  0.5818238
  0.03      0.5759930  0.6128123
  0.04      0.5921489  0.6010864
  0.05      0.6144012  0.5753466
  0.06      0.6515288  0.5276755
  0.07      0.6808084  0.4896541
  0.08      0.7055652  0.4631316
  0.09      0.7293999  0.4361376
  0.10      0.7448612  0.4196183

RMSE was used to select the optimal model using  the smallest value.
The final value used for the model was fraction = 0.03.
```

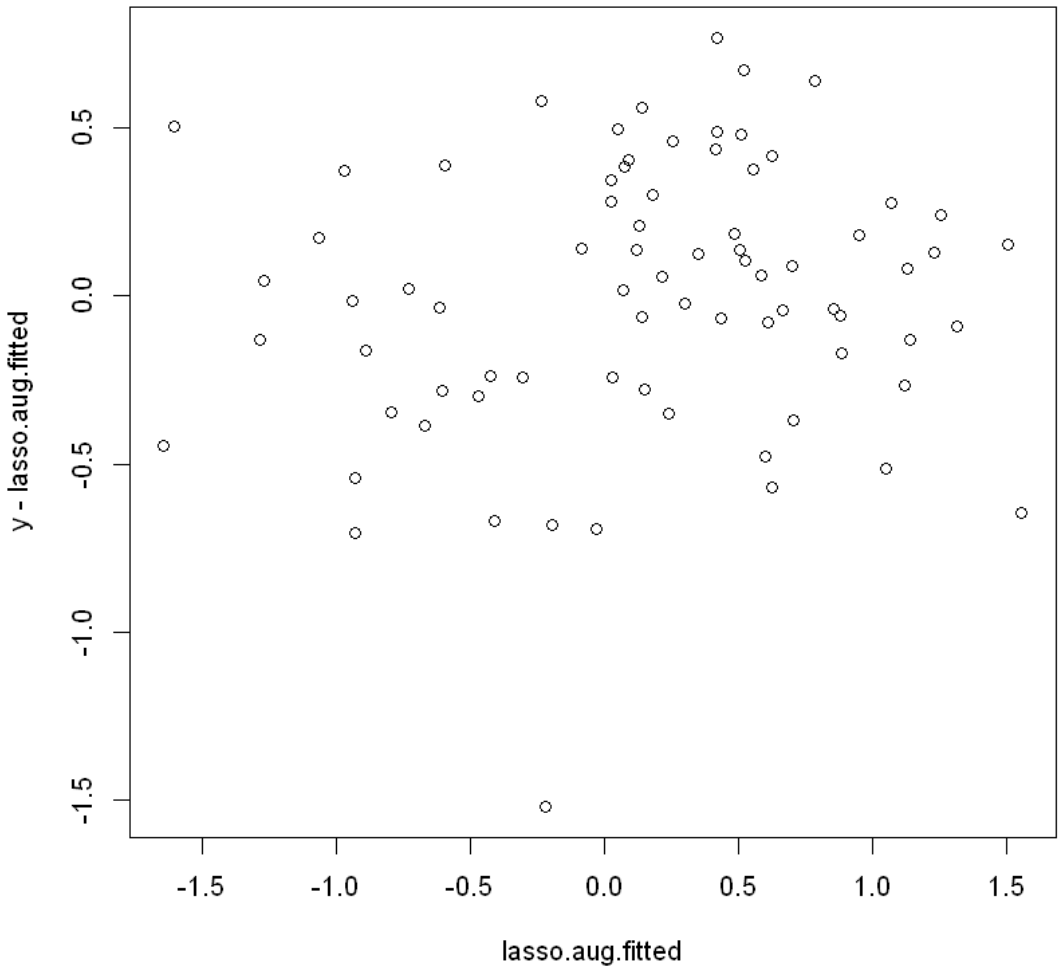
```
In [385]: # 根据我们 bestTune 的参数 lambda
          lasso.aug.finalCoef<-predict(lasso.aug$finalModel,type="coefficients",mode="fraction"
          lasso.aug.finalCoef.nonzero<-lasso.aug.finalCoef[which(lasso.aug.finalCoef!=0)]
          lasso.aug.finalCoef.nonzero[order(abs(lasso.aug.finalCoef.nonzero),decreasing = T)]
          lasso.aug.fitted<-predict(lasso.aug$finalModel,mode="fraction",s=lasso.aug$bestTune$f
          plot(lasso.aug.fitted,y)
          abline(a = 0, b = 1)
          plot(lasso.aug.fitted,y-lasso.aug.fitted) # 残差依然和 y 高度相关
          sqrt(mean((y-lasso.aug.fitted)^2))
```

PP_3_8 -0.440234846269078 **PP_4_15** -0.283769134873081 **PP_1_2** 0.249684692275364 **PP_11_17** 0.196207876135372 **PP_8_10** 0.169678046255624 **PP_12_15** -0.160999487104812 **PP_11_16** 0.155864333750181 **PP9_sqrd** 0.150712637915911 **PP_9_14** -0.147286568122991

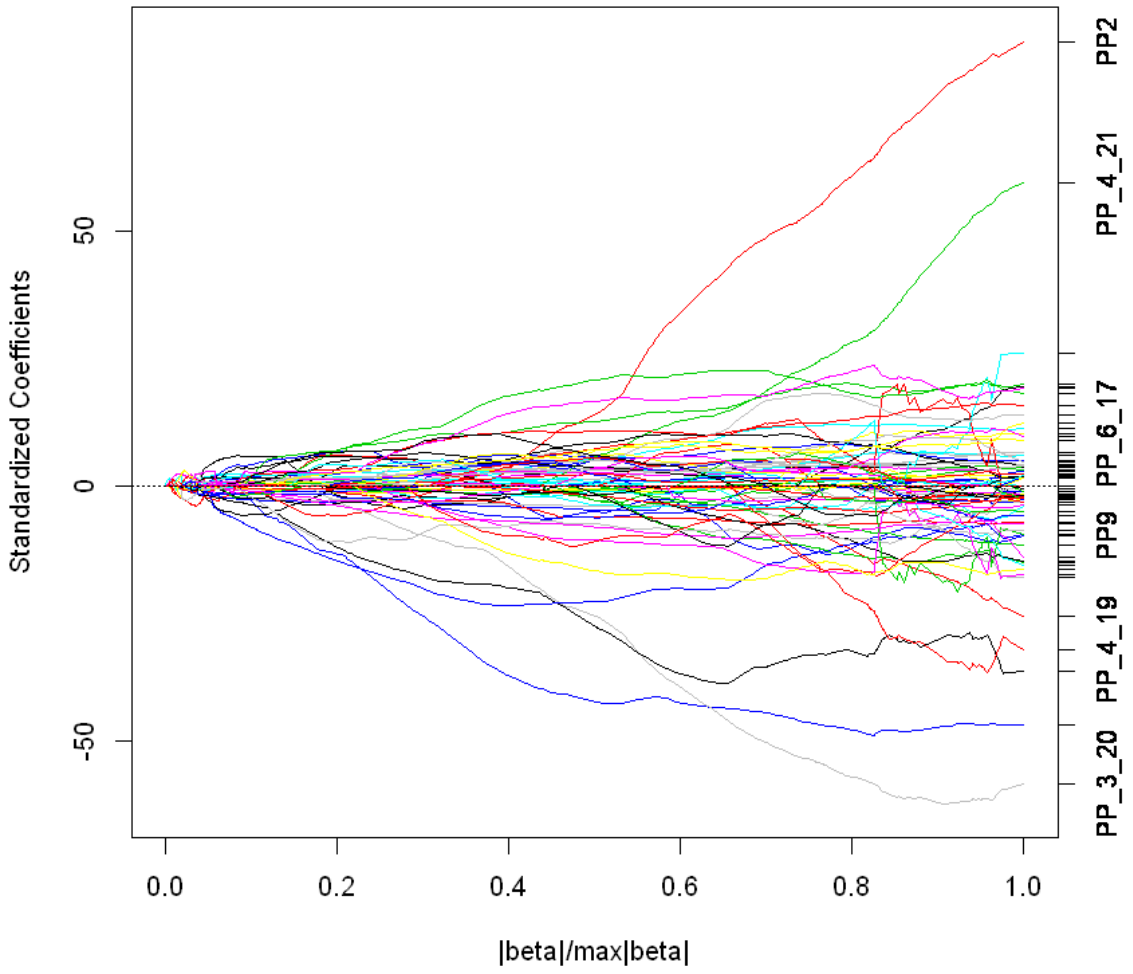
PP_1_17 0.141026267686551 **PP_1_6** -0.116685629305708 **PP2_sqrd** -0.110399619810336 **PP_2_4** -0.108348204667638 **PP5_sqrd** -0.107728680330833 **PP_14_21** -0.09086356925943 **PP14** -0.0780257353336671 **PP_11_18** 0.0754888409224354 **PP_16_17** 0.070499163030641 **PP_7_13** -0.0582623104201397 **PP_8_17** 0.0485411688860544 **PP_2_5** 0.0312992571930578 **PP_12_21** -0.0244854578672758 **PP_1_3** -0.0237443033026904 **PP_8_11** 6.58559698439825e-18



0.399642992599902



In [386]: plot(lasso.aug$finalModel,use.color =T)



2.5.4 SVM with Linear Kernel

```
In [387]: set.seed(1)
          svmLinear.aug <- train(x=X.aug,y=y,
              #x = X.aug[,names(lasso.aug.finalCoef.nonzero)], y = y,

                                 method = "svmLinear",
                                 #preProcess = "pca",
                                 tuneGrid=data.frame(C=seq(0.01,0.1,0.01)),
```

```
                                trControl = trControl)
          svmLinear.aug
```

```
Support Vector Machines with Linear Kernel

 76 samples
252 predictors

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 69, 68, 69, 68, 68, 68, ...
Resampling results across tuning parameters:

  C     RMSE       Rsquared
  0.01  0.5449292  0.6697547
  0.02  0.5440998  0.6668167
  0.03  0.5520797  0.6555191
  0.04  0.5577137  0.6487326
  0.05  0.5671280  0.6361169
  0.06  0.5768440  0.6206023
  0.07  0.5852009  0.6083794
  0.08  0.5948885  0.5953382
  0.09  0.6023343  0.5865995
  0.10  0.6072084  0.5793817

RMSE was used to select the optimal model using  the smallest value.
The final value used for the model was C = 0.02.
```

2.5.5 SVM with Radial Kernel

```
In [388]: set.seed(1)
          svmRadial.aug <- train(x = X.aug, y = y,
                                 method = "svmRadial",
                                 #preProcess = "pca",
                                 tuneLength = 10,
                                 trControl = trControl)
          svmRadial.aug
```

```
Support Vector Machines with Radial Basis Function Kernel

 76 samples
252 predictors

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 69, 68, 69, 68, 68, 68, ...
Resampling results across tuning parameters:

  C       RMSE       Rsquared
    0.25  0.7098923  0.5744243
    0.50  0.6168695  0.5974411
    1.00  0.5658658  0.6304510
    2.00  0.5721891  0.6081474
    4.00  0.5705370  0.5771910
    8.00  0.5838142  0.5401925
   16.00  0.5859070  0.5391797
   32.00  0.5859070  0.5391797
   64.00  0.5859070  0.5391797
  128.00  0.5859070  0.5391797

Tuning parameter 'sigma' was held constant at a value of 0.003132951
RMSE was used to select the optimal model using  the smallest value.
The final values used for the model were sigma = 0.003132951 and C = 1.
```

可以看到，加入线性项之后，之前表现最好的 SVM with Radial Kernel 的方法预测效果也有不小的提升。

3 模型总结

3.1 污渍 1 `powdow.scaled['factor1']`

3.1.1 重要自变量

注：认为 Elastic Net 回归中系数较大的自变量为对响应变量有显著的。

```
In [389]: lasso.aug.finalCoef.nonzero<-lasso.aug.finalCoef[which(lasso.aug.finalCoef!=0)]
          lasso.aug.finalCoef.nonzero[order(abs(lasso.aug.finalCoef.nonzero),decreasing = T)]
          names(lasso.aug.finalCoef.nonzero)
```

PP_3_8 -0.440234846269078 **PP_4_15** -0.283769134873081 **PP_1_2** 0.249684692275364 **PP_11_17** 0.196207876135372 **PP_8_10** 0.169678046255624 **PP_12_15** -0.160999487104812 **PP_11_16** 0.155864333750181 **PP9_sqrd** 0.150712637915911 **PP_9_14** -0.147286568122991 **PP_1_17** 0.141026267686551 **PP_1_6** -0.116685629305708 **PP2_sqrd** -0.110399619810336 **PP_2_4** -0.108348204667638 **PP5_sqrd** -0.107728680330833 **PP_14_21** -0.09086356925943 **PP14** -0.0780257353336671 **PP_11_18** 0.0754888409224354 **PP_16_17** 0.070499163030641 **PP_7_13** -0.0582623104201397 **PP_8_17** 0.0485411688860544 **PP_2_5** 0.0312992571930578 **PP_12_21** -0.0244854578672758 **PP_1_3** -0.0237443033026904 **PP_8_11** 6.58559698439825e-18

1. 'PP14' 2. 'PP2_sqrd' 3. 'PP5_sqrd' 4. 'PP9_sqrd' 5. 'PP_1_2' 6. 'PP_1_3' 7. 'PP_1_6' 8. 'PP_1_17' 9. 'PP_2_4' 10. 'PP_2_5' 11. 'PP_3_8' 12. 'PP_4_15' 13. 'PP_7_13' 14. 'PP_8_10' 15. 'PP_8_11' 16. 'PP_8_17' 17. 'PP_9_14' 18. 'PP_11_16' 19. 'PP_11_17' 20. 'PP_11_18' 21. 'PP_12_15' 22. 'PP_12_21' 23. 'PP_14_21' 24. 'PP_16_17'

3.1.2 预测

10-fold CV

```
In [390]: methods<-list(OLS = OLS, OLS.restricted=OLS.restricted, lmStepAIC=lmStepAIC, svmLinea
                                  svmRadial=svmRadial,
                                  lasso = lasso, pls = pls,enet=enet,
                                  svmRadial.aug=svmRadial.aug,svmLinear.aug=svmLinear.aug,
                                  lasso.aug=lasso.aug, pls.aug=pls.aug)
          resamp <- resamples(methods)
          summary(resamp)$statistics$RMSE
```

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
OLS	0.4245	0.5606	0.6383	0.6578	0.7554	1.0080	0
OLS.restricted	0.3115	0.4192	0.5976	0.5867	0.7260	0.9356	0
lmStepAIC	0.3949	0.5870	0.6264	0.6476	0.6745	1.0080	0
svmLinear	0.3933	0.5001	0.6226	0.6343	0.7592	0.9078	0
svmRadial	0.3619	0.5497	0.6041	0.6087	0.6389	0.9385	0
lasso	0.3790	0.5219	0.6052	0.6118	0.6986	0.9280	0
pls	0.3513	0.5131	0.5623	0.6170	0.7319	0.9749	0
enet	0.3796	0.5222	0.6036	0.6115	0.6987	0.9296	0
svmRadial.aug	0.3639	0.4644	0.5752	0.5659	0.6453	0.8672	0
svmLinear.aug	0.3758	0.4061	0.5215	0.5441	0.6397	0.8623	0
lasso.aug	0.3239	0.4648	0.5387	0.5760	0.6424	0.9332	0
pls.aug	0.3761	0.4199	0.4352	0.5488	0.6551	0.9169	0

对前 10 个样本的预测

In [391]:
```
newx<- powder.scaled.test[,predictors.index]
newy<-powder.scaled.test$factor1
RMSE.record<-c()
yhat.record<-data.frame(id=1:10,ytrue=newy)
yhat.record["OLS"]<-predict(OLS$finalModel,newdata=newx)
yhat.record["pls"]<-predict(pls$finalModel,newdata=newx, ncomp = pls$finalModel$ncom
yhat.record["pls.aug"]<-predict(pls.aug$finalModel,newdata=powder.augmented.imputed.
yhat.record["lasso"]<-predict(lasso$finalModel,newx=newx,s=lasso$bestTune$fraction,m
yhat.record["lasso.aug"]<-predict(lasso.aug$finalModel,newx=powder.augmented.imputed
yhat.record["enet"]<-predict(enet$finalModel,newx=newx,s=enet$bestTune$fraction,mode
yhat.record["svmLinear"]<-predict(svmLinear$finalModel,newdata=newx)
yhat.record["svmLinear.aug"]<-predict(svmLinear.aug$finalModel,newdata=powder.augmen
yhat.record["svmRadial"]<-predict(svmRadial$finalModel,newdata=newx)
yhat.record["svmRadial.aug"]<-predict(svmRadial.aug$finalModel,newdata=powder.augmen

#yhat.record["mean"]<-rowMeans(yhat.record[,-c(1,2)])
for (method in colnames(yhat.record)[-c(1,2)]){
    RMSE.record[method]<-sqrt(mean((yhat.record[method]-newy)^2))
}
yhat.record
RMSE.record[order(RMSE.record)]
```

```
for (method in colnames(yhat.record)[-c(1,2)]){
    RMSE.record[method]<-sqrt(mean((yhat.record[-8,method]-newy[-8])^2))
}
RMSE.record[order(RMSE.record)]
```

id	ytrue	OLS	pls	pls.aug	lasso	lasso.aug	enet	svmLin
1	-1.5855122	-1.235586547	-1.26228901	-1.5898191	-1.0794214	-1.65089485	-1.0855943	-0.7046
2	-2.3714088	-1.150952322	-1.08654185	-1.5856431	-1.0878594	-1.31247786	-1.0935166	-0.5961
3	-0.2747798	0.001373642	0.03459549	-0.4910662	0.3132947	0.24028108	0.3164747	0.14662
4	-1.0845762	-1.090128847	-1.00317991	-1.3115291	-0.9603596	-1.19699602	-0.9660995	-0.6800
5	-2.1562664	-2.210344863	-2.14980363	-2.5096955	-1.9677296	-2.01033885	-1.9784179	-1.5891
6	-0.9666176	0.003946920	-0.10905434	-0.1984712	-0.2429749	-0.15351906	-0.2455679	-0.2095
7	0.4459457	0.577643914	0.71360882	0.3512938	0.7914424	0.47569562	0.7923066	0.50095
8	-1.5368518	0.411712896	0.60938670	0.2226336	0.6076258	0.38929570	0.6090758	0.37245
9	-2.1313065	-0.941475656	-0.59622142	-0.9975191	-0.5532731	-1.12587152	-0.5574353	-0.3583
10	0.3943648	0.570936897	0.64252885	0.3890659	0.7463014	0.03921184	0.7421322	0.55105

pls.aug 0.762951143468692 **svmLinear.aug** 0.797912869019964 **lasso.aug** 0.832701737371018 **OLS** 0.888498574010219 **svmRadial** 0.949728593254381 **pls** 0.984337636871385 **svmRadial.aug** 0.996512432191044 **enet** 1.00603535571914 **lasso** 1.00778440569384 **svmLinear** 1.09452607343546

pls.aug 0.550268174658018 **svmLinear.aug** 0.587980864133521 **lasso.aug** 0.598505331586358 **OLS** 0.674733926652801 **pls** 0.751507093025621 **enet** 0.782876900008961 **lasso** 0.78581237882019 **svmRadial** 0.852869860317964 **svmRadial.aug** 0.89384591795008 **svmLinear** 0.96231367621451