# 应用回归分析 Final Project

邵智轩　　1400012141　　物理学院

## 初步分析

一共 86 个案例，其中自变量 21 个，因变量 18 个。

```
powder <- read.csv("powder.csv")
rownames(powder) <- powder$Treatment
predictors.index <- grep("PP", colnames(powder)) # predictors
responses.index <- grep("O", colnames(powder)) # responses
powder.scaled<-powder
powder.scaled[-1]<-scale(powder[-1]) # normalized data
```

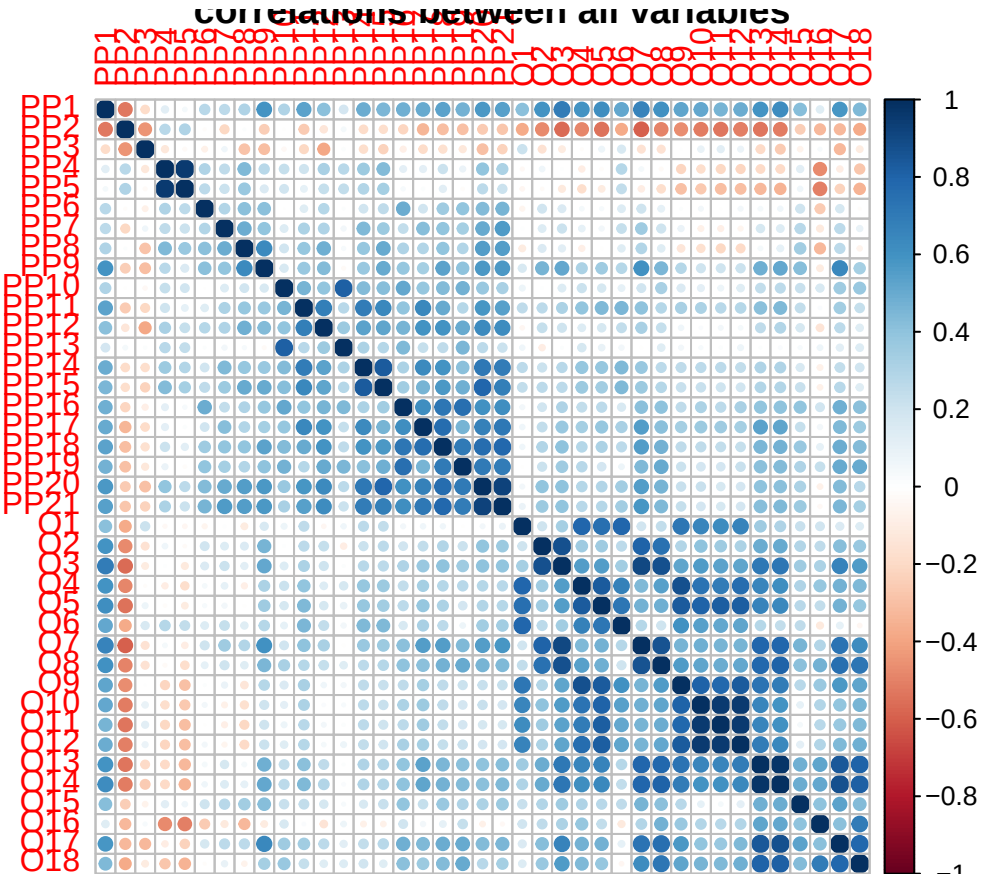
### 缺失情况

```
mice::md.pattern(powder) # 查看缺失情况
```

可以看到，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。

### 数据相关性

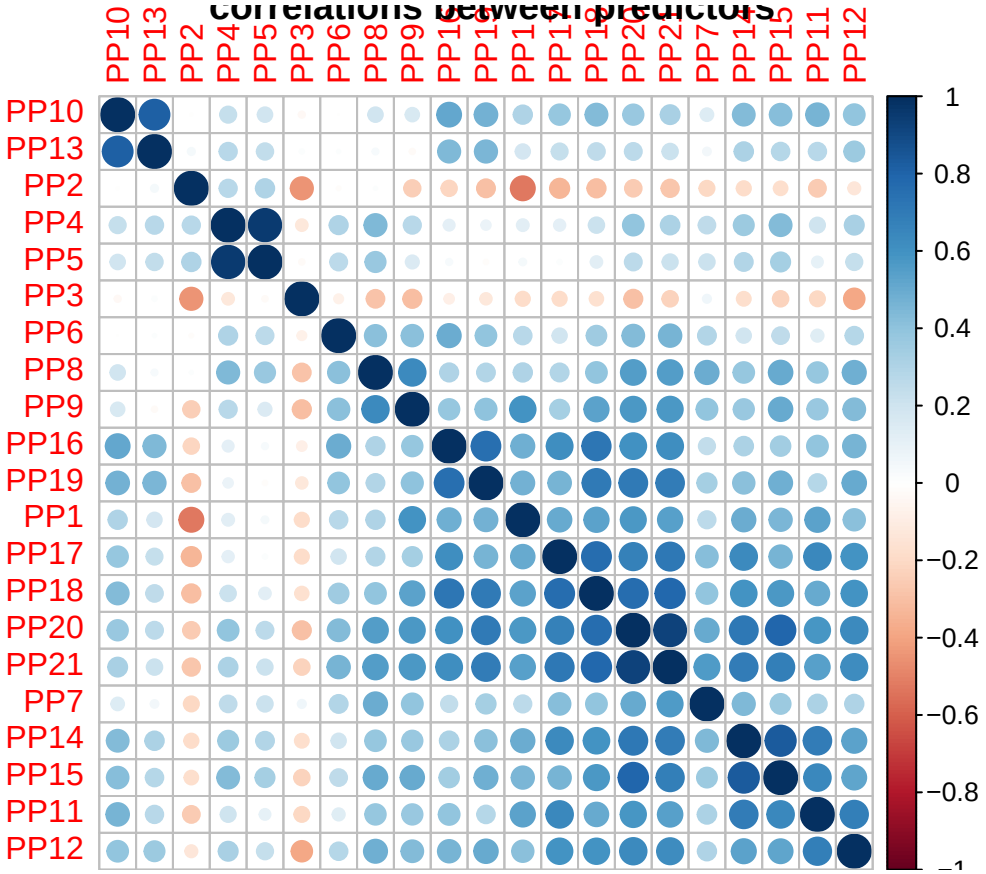
```
correlations <- list("predictors" = cor(powder[predictors.index],use = "complete"),
                     "responses" = cor(powder[responses.index], use = "all.obs"),
                     "all" = cor(powder[-1], use = "complete"))
corrplot::corrplot(correlations[["all"]], order = "original",
    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

我们再来看看自变量之间的相关性（按照 hierachical 聚类排序）：

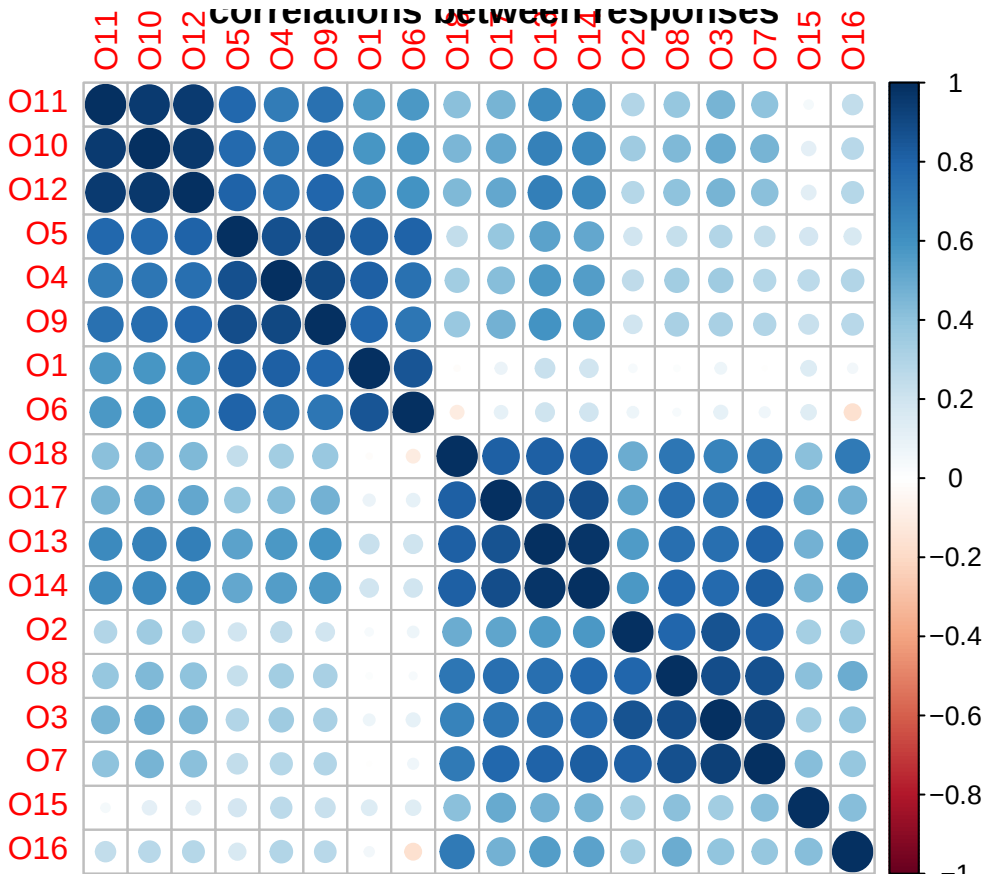
```
corrplot::corrplot(correlations[["predictors"]], order = "hclust",
                   main = "correlations between predictors")
```




可以看到，物理属性大致分为几个聚类：PP10 与 PP13 非常相似；PP2 和 PP3 与其他物理属性轻度负相关，但它们互相也负相关；PP4 与 PP5 基本相同；等等……

再看看因变量之间的相关性：

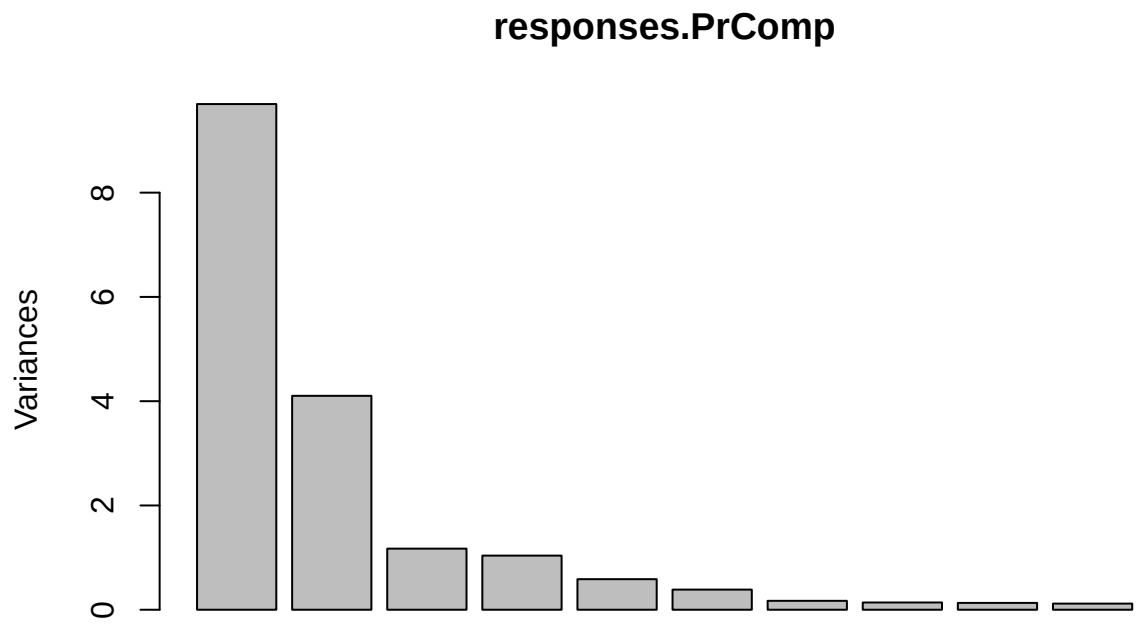
```
corrplot::corrplot(correlations[["responses"]], order = "hclust",
                   main = "correlations between responses")
```




污渍大致可分为三类。

我们可以用主成分分析（PCA）来看一看：

```
responses.PrComp <- prcomp(powder.scaled[responses.index], center = TRUE, scale. = TRUE)
plot(responses.PrComp)
```

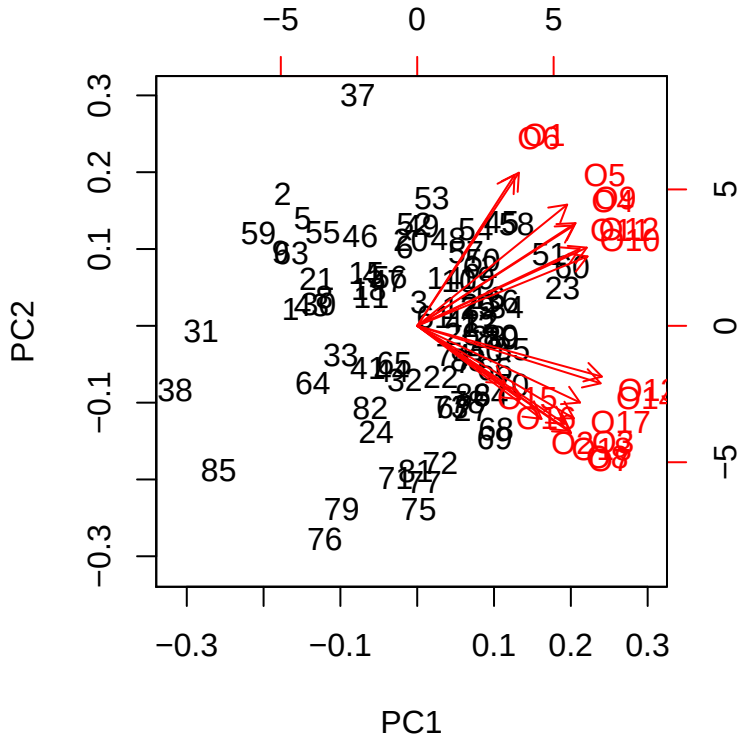



```
head(responses.PrComp$sd ^ 2 / sum(responses.PrComp$sd ^ 2))
```

```
## [1] 0.53886004 0.22794400 0.06511166 0.05770125 0.03256965 0.02144202
```

可见虽然因变量有 18 个，但是其前两个主成分已基本涵盖了大部分变异性。

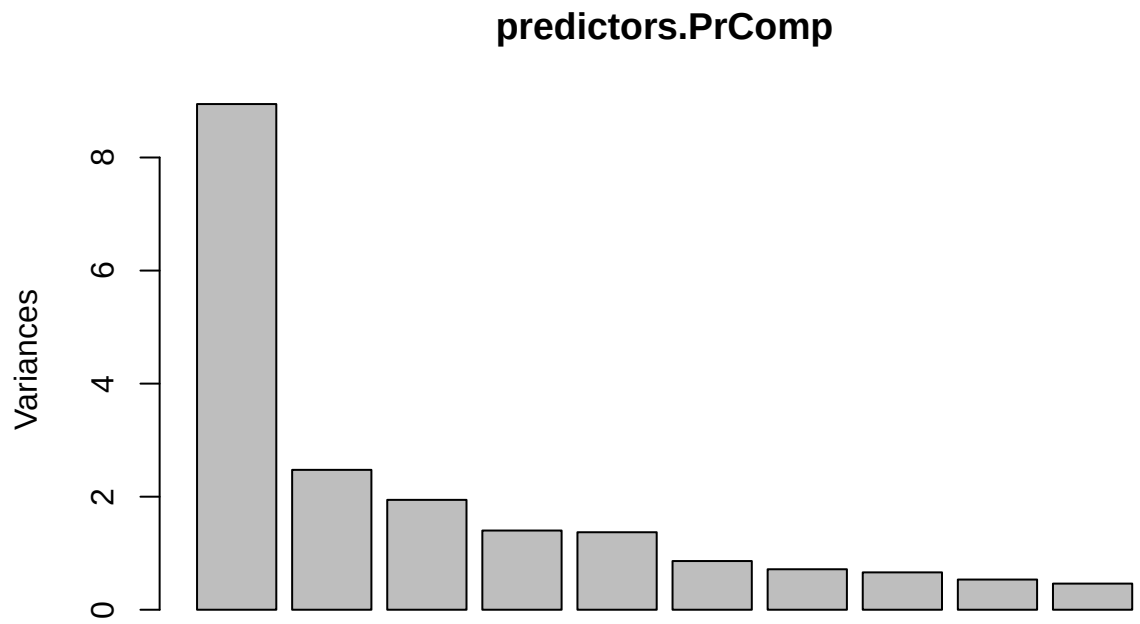
```
biplot(responses.PrComp)
```




从前两个主成分的图上来看，因变量主要分为两类。因子分析（factor analysis）当 `factors=2` 时给出与 PCA 相似的结果）。

再用同样的方法来看看自变量之间的相似性，先看看 PCA（由于自变量中数据有缺失，暂时先考虑完全数据）：

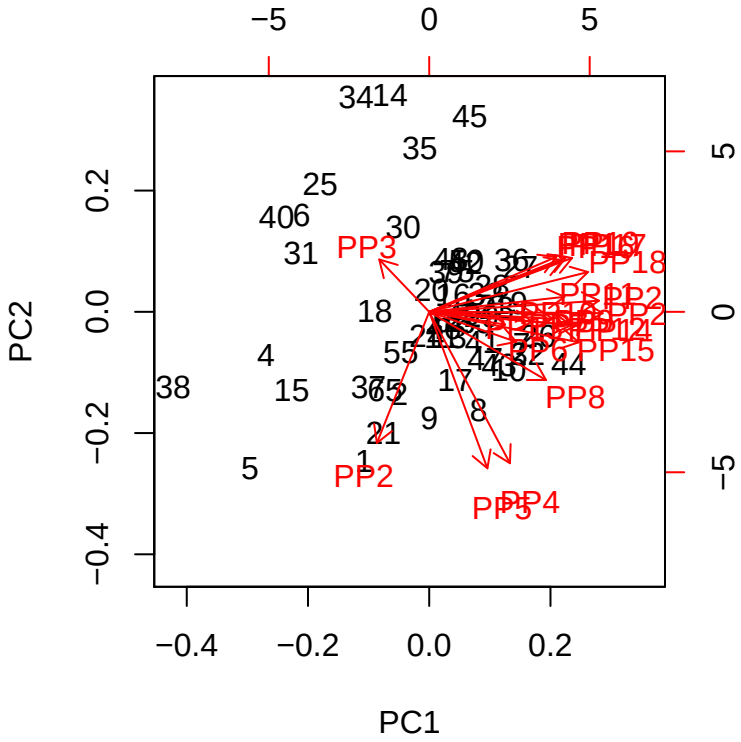
```
predictors.PrComp <- prcomp(na.omit(powder.scaled[predictors.index]), center = TRUE, scale. = TRUE)
plot(predictors.PrComp)
```




```
head(predictors.PrComp$sd ^ 2 / sum(predictors.PrComp$sd ^ 2))
```

```
## [1] 0.42597364 0.11782330 0.09254307 0.06672087 0.06527695 0.04101127
```

```
biplot(predictors.PrComp)
```




与我们之前从相关性得到的结论类似，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。

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

```
OLS.O1<-lm(reformulate(grep("PP", names(powder), value = T),
                       response = "O1"), powder, na.action = na.omit)
kappa(OLS.O1)
```

```
## [1] 124317.7
```

```
car::vif(OLS.O1)
```

```
##       PP1       PP2       PP3       PP4       PP5       PP6       PP7
##  4.101272  5.494771  4.427349 18.712175 18.466339  2.243281  2.157674
##       PP8       PP9      PP10      PP11      PP12      PP13      PP14
##  2.748062  3.628309  5.136966  5.424715  4.594037  5.043519  7.525487
##      PP15      PP16      PP17      PP18      PP19      PP20      PP21
##  8.986843  6.647702  6.778051  5.911741  7.024333 17.893790 12.812533
```

设计矩阵的条件数相当大，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。

Multiple Outcomes 的情形特别适合用典型变量分析，“CCA” (canonical correlation analysis)，即分别在

$\boldsymbol{X}$ 和 $\boldsymbol{Y}$ 中各自找一个代表，即各自变量的线性组合 $\boldsymbol{X}v_m$ 和 $\boldsymbol{Y}u_m$ 使得相关系数：

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

达到最大。

```
powder.scaled.cc.CCA <- CCA::cc(
  powder.scaled[complete.cases(powder.scaled), predictors.index],
                    powder.scaled[complete.cases(powder.scaled), responses.index])
powder.scaled.cc.CCA$cor
```

```
##  [1] 0.97147694 0.94907858 0.91303177 0.85951184 0.83269828 0.81560345
##  [7] 0.73779020 0.71871802 0.62049217 0.58987887 0.56564480 0.53657909
## [13] 0.53232824 0.39777174 0.30368700 0.26678246 0.24419956 0.06880078
```

### 一个 naive 的尝试

我们不妨先取 O1 为因变量，尝试对全模型线性拟合一下。

```
summary(OLS.O1)
```

```
##
## Call:
## lm(formula = reformulate(grep("PP", names(powder), value = T),
##     response = "O1"), data = powder, na.action = na.omit)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -32.505  -6.641   1.051   8.020  31.471
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 141.086527 137.050772   1.029    0.311
## PP1           0.006597   0.002960   2.229    0.033 *
## PP2          -0.208967   2.809243  -0.074    0.941
## PP3           1.696600   2.042597   0.831    0.412
## PP4           0.358387   0.564965   0.634    0.530
## PP5          -0.255750   0.345117  -0.741    0.464
## PP6          -0.728266   0.824047  -0.884    0.383
## PP7           0.129493   0.888468   0.146    0.885
## PP8          -0.572863   0.390387  -1.467    0.152
## PP9           0.128151   0.384333   0.333    0.741
## PP10         -1.530486   2.683386  -0.570    0.572
## PP11         -0.140306   3.399971  -0.041    0.967
```

```
## PP12          0.098025   2.695447   0.036    0.971
## PP13          0.698251   2.370956   0.295    0.770
## PP14          1.489352   3.747124   0.397    0.694
## PP15          6.178933   3.804335   1.624    0.114
## PP16          3.694175   2.861969   1.291    0.206
## PP17         -4.267553   4.163225  -1.025    0.313
## PP18         -1.556638   4.669828  -0.333    0.741
## PP19         -3.837884   2.889605  -1.328    0.194
## PP20         -9.299741   5.543803  -1.678    0.103
## PP21          7.238891   4.629805   1.564    0.128
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.73 on 32 degrees of freedom
##   (32 observations deleted due to missingness)
## Multiple R-squared:  0.5204, Adjusted R-squared:  0.2057
## F-statistic: 1.653 on 21 and 32 DF,  p-value: 0.09728
```

很不幸，绝大多数系数都不显著。

### 缺失数据填补

再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）。