应用回归分析第 9 章

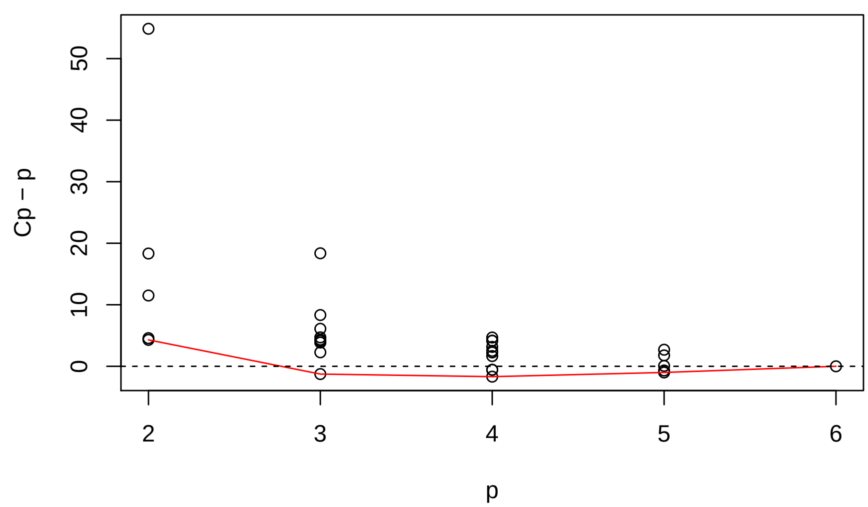
邵智轩　　1400012141　　物理学院

8.7

8.7.1

```
library(car)
data<-alr4::dwaste
n<-20
data['O2UP.log']<-log10(data$O2UP)
predictors<-c('BOD','TKN','TS','TVS','COD')
predictors.powerset<-ggm::powerset(predictors)
lm.tot<-lm(reformulate(predictors,response = "O2UP.log"),data=data)
RSS.tot<-sum(lm.tot$residuals^2)
sigma2<-RSS.tot/lm.tot$df.residual
SYY<-with(data,sum((O2UP.log-mean(O2UP.log))^2))
Cp_p<-data.frame('p'=integer(),
                 'Cp'=numeric(),
                 'R2'=numeric(),
                 'RSS'=numeric(),
                 'Model'=character(),stringsAsFactors = F)
for (i in 1:length(predictors.powerset)){# 计算出书上的表 8.10
  terms<-predictors.powerset[[i]]
  lm.p<-lm(reformulate(terms,response = "O2UP.log"),data=data)
  Cp_p[i,'Model']<-paste(terms,collapse = " ")
  Cp_p[i,'p']<-lm.p$rank
  Cp_p[i,'RSS']<-sum(lm.p$residuals^2)
```

```
  Cp_p[i,"Cp"]<-Cp_p[i,"RSS"]/sigma2+2*Cp_p[i,'p']-n
  Cp_p[i,"R2"]<-1-Cp_p[i,"RSS"]/SYY
}
Cp_p<-dplyr::arrange(Cp_p,p,Cp)
with(Cp_p,plot(p,Cp-p))
with(Cp_p,lines(p[c(1,6,16,26,31)],(Cp-p)[c(1,6,16,26,31)],col="red"))
abline(h=0,lty=2)
```



由 C_p 统计量的公式：

$$
\begin{aligned}
C_p &= \frac{RSS_p}{\hat{\sigma}^2} + 2p - n && (1) \\
&= (k' - p)(F_p - 1) + p && (2)
\end{aligned}
$$

$F_p \leq 2$ 等价于 $C_p \leq k' = 6$。我们找出所有 $C_p \leq 6$ 的模型

```
Cp_p[Cp_p$Cp<=6,]
```

```
##     p       Cp        R2       RSS            Model
## 6   3 1.739348 0.7857096 1.0850838           TS  COD
## 7   3 5.273437 0.7375931 1.3287269          TVS  COD
```

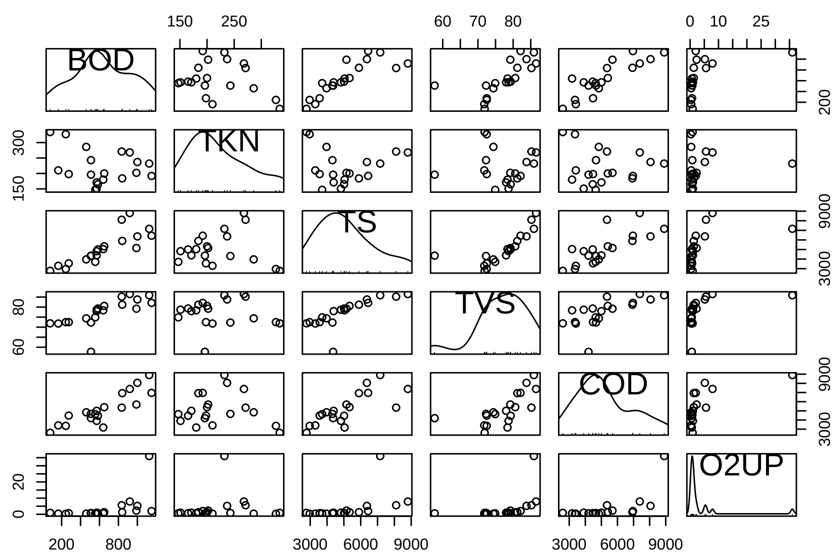
```
## 16 4 2.318918 0.8050487 0.9871583         TKN TS COD
## 17 4 3.424455 0.7899968 1.0633749         TS TVS COD
## 18 4 3.439168 0.7897965 1.0643892         BOD TS COD
## 19 4 5.664793 0.7594947 1.2178256        TKN TVS COD
## 26 5 4.001289 0.8093732 0.9652606     TKN TS TVS COD
## 27 5 4.318644 0.8050524 0.9871393     BOD TKN TS COD
## 28 5 5.068750 0.7948397 1.0388523     BOD TS TVS COD
## 31 6 6.000000 0.8093907 0.9651718 BOD TKN TS TVS COD
```

8.7.2

```
attach(data)
```

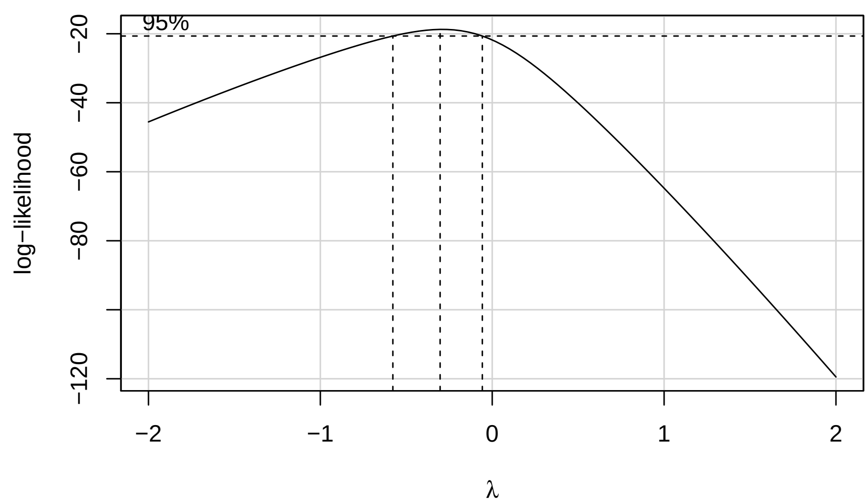
```
## The following object is masked from package:datasets:
##
##     BOD
```

```
scatterplotMatrix(~BOD+TKN+TS+TVS+COD+O2UP,data=data,
                  smoother=FALSE,reg.line=FALSE)
```



由于 O2UP 尺度从 0.3 到 36.0，横跨了几个数量级，应对它做对数变换。下面用 Box-Cox 似然比检验。

```r
boxCox(O2UP~BOD+TKN+TS+TVS+COD)
```

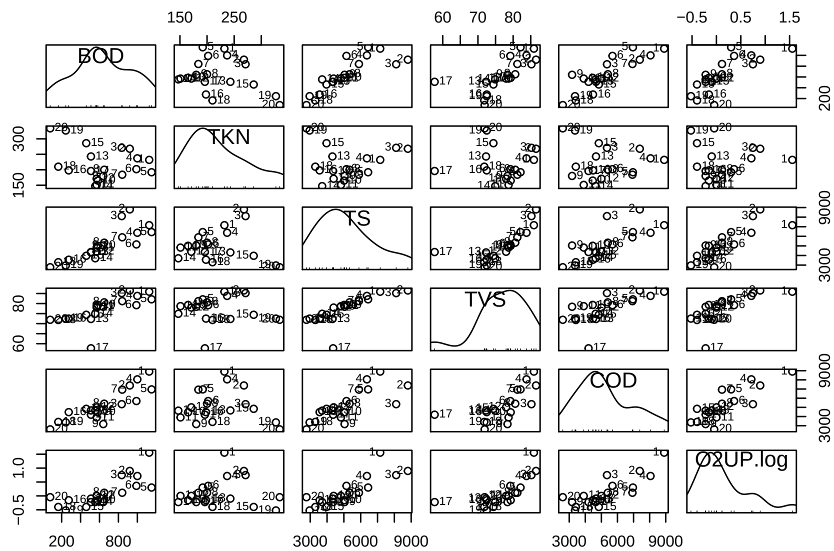



```r
summary(powerTransform(O2UP~BOD+TKN+TS+TVS+COD))
```

```
## bcPower Transformation to Normality
##
##    Est.Power Std.Err. Wald Lower Bound Wald Upper Bound
## Y1   -0.2901   0.1283          -0.5415          -0.0387
##
## Likelihood ratio tests about transformation parameters
##                               LRT df      pval
## LR test, lambda = (0)  6.115335  1 0.0134014
## LR test, lambda = (1) 92.072726  1 0.0000000
```

于是我们对 O2UP 做对数变换，并重新做散点图矩阵

```r
scatterplotMatrix(~BOD+TKN+TS+TVS+COD+O2UP.log,data=data,
                  smoother=FALSE,reg.line=FALSE,
                  id.n=n,id.cex=0.7)
```



我们看到第 17 个点的 TVS 值很不寻常，在下面的分析中，我们先将第 17 个案例去掉，最后再单独考虑它的影响。

我们下面考虑是否对 predictors 做变换。

```r
summary(b1 <- powerTransform(cbind(BOD, TKN, TS, TVS, COD) ~ 1,
                             data=data, subset=-17))
```

```
## bcPower Transformations to Multinormality
##
##     Est.Power Std.Err. Wald Lower Bound Wald Upper Bound
## BOD    0.6749   0.2469           0.1909           1.1589
## TKN   -0.5903   1.0466          -2.6416           1.4610
## TS     0.0668   0.4764          -0.8669           1.0005
## TVS    2.3332   3.7079          -4.9342           9.6006
## COD    0.2722   0.5866          -0.8776           1.4219
##
```

```
## Likelihood ratio tests about transformation parameters
##                                    LRT df       pval
## LR test, lambda = (0 0 0 0 0) 11.12255  5 0.04900354
## LR test, lambda = (1 1 1 1 1) 10.87991  5 0.05381375
```

并没有很显著地拒绝 “不用做变换” 地原假设。下面我们用未作变换地 predictors 继续分析。

```
knitr::kable(
  summary(lm.tot<-lm(reformulate(predictors,response =
                                   "O2UP.log"),data=data,subset = -17))$coef)
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-5.0738521	3.1110522	-1.6309119	0.1268859
BOD	-0.0001643	0.0005413	-0.3035464	0.7662770
TKN	0.0015728	0.0012969	1.2127947	0.2467873
TS	0.0000320	0.0001250	0.2562370	0.8017779
TVS	0.0521049	0.0471796	1.1043963	0.2894437
COD	0.0001379	0.0000742	1.8592693	0.0857668

没有一个 predictor 是显著的，这提示我们需要进行变量选择。

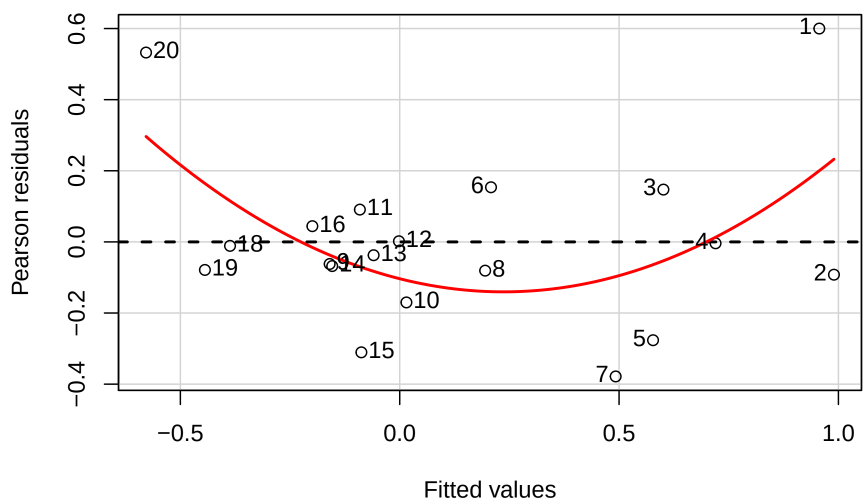
在 8.7.1 中，最小的 C_p 统计量（高斯模型中等价于最小的 AIC）的模型为 `O2UP.log ~ TS + COD`，去掉第 17 个点后再计算所有子集模型的 C_p，仍选出该子集。

```
knitr::kable(
  summary(lm.best<-lm(O2UP.log~TS+COD,data=data,subset=-17))$coef)
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-1.3555001	0.2054802	-6.596742	0.0000061
TS	0.0001492	0.0000563	2.649018	0.0175042
COD	0.0001397	0.0000548	2.549618	0.0214198

TS 与 COD 在 0.05 水平下显著。另外查看散点图我们发现，被删除的第 17 个数据点的 TS，COD 值并无异常，它对拟合结果并不会造成多大影响。

```
residualPlot(lm.best,id.n=length(lm.best$residuals))
```



除了第 1 个和第 20 个点，其余点的残差是正常的。

8.9

$$\mathrm{Var}(\hat{\beta}^*|X) = \sigma^2(\mathcal{X}'\mathcal{X})^{-1} \tag{3}$$

其中，

$$(\mathcal{X}'\mathcal{X})^{-1} = \begin{pmatrix} SX_1X_1 & SX_1X_2 \\ SX_2X_1 & SX_2X_2 \end{pmatrix}^{-1} \tag{4}$$

$$= \frac{1}{SX_1X_1 \cdot SX_2X_2 - (SX_1X_2)^2} \begin{pmatrix} SX_2X_2 & -SX_1X_2 \\ -SX_2X_1 & SX_1X_1 \end{pmatrix} \tag{5}$$

利用

$$R_{12}^2 = \frac{(SX_1X_2)^2}{SX_1X_1SX_2X_2} \tag{6}$$

$$\begin{aligned}
\mathrm{Var}(\hat{\beta}_1) &= \sigma^2 \frac{1}{SX_1X_1 \cdot SX_2X_2 - (SX_1X_2)^2} SX_2X_2 && (7)\\
&= \sigma^2 \frac{1}{SX_1X_1 \cdot SX_2X_2(1-R_{12}^2)} SX_2X_2 && (8)\\
&= \frac{\sigma^2}{1-R_{12}^2} \frac{1}{SX_1X_1} && (9)
\end{aligned}$$

8.10

为了解释的方便，不妨将 X_j 放到最后一列，记为 X_p。记 $\boldsymbol{A} = \boldsymbol{X}'\boldsymbol{X}$，为 $(p-1)\times(p-1)$ 对称阵。则

$$\mathrm{Var}(\hat{\beta}_p) = \sigma^2 [\boldsymbol{A}^{-1}]_{pp} \tag{10}$$

利用习题 2.7.7 （扫描算法）的结果（将那里的 Y 取为 X_p），当从第 0 个支点扫描到第 $p-1$ 个支点时，

$$\mathrm{Sweep}\boldsymbol{A}[0,1,2,...,p-1] = \begin{pmatrix} (\boldsymbol{X}'\boldsymbol{X})^{-1} & \hat{\boldsymbol{\beta}}_p \\ -\hat{\boldsymbol{\beta}}_p' & RSS_p \end{pmatrix} \tag{11}$$

其中 RSS_p 是第 p 个变量 X_p 对其余 $(p-1+1)$ 个变量（$\{\mathbf{1}, X_1, .., X_{p-1}\}$）回归的残差平方和。

利用 2.7.6 的结论，再扫描第 p 个支点后，得到矩阵 $\boldsymbol{A}^{-1}$，而扫描算法对第 k 个支点的变换：$b_{kk} = \frac{1}{a_{kk}}$ 可知

$$\begin{aligned}
[\boldsymbol{A}^{-1}]_{pp} &= [\mathrm{Sweep}\boldsymbol{A}[0,1,2,...,p-1,p]]_{pp} && (12)\\
&= 1/[\mathrm{Sweep}\boldsymbol{A}[0,1,2,...,p-1]]_{pp} && (13)\\
&= \frac{1}{RSS_p} && (14)
\end{aligned}$$

所以

$$\begin{aligned}
\mathrm{Var}(\hat{\beta}_p) &= \frac{\sigma^2}{RSS_p} && (15)\\
&= \frac{\sigma^2}{SX_pX_p(1-R_p^2)} && (16)
\end{aligned}$$