# Chapter 8
# Case Study in Data Reduction

Recall that the aim of data reduction is to reduce (without using the outcome) the number of parameters needed in the outcome model. The following case study illustrates these techniques:

1. redundancy analysis;
2. variable clustering;
3. data reduction using principal component analysis (PCA), sparse PCA, and pretransformations;
4. restricted cubic spline fitting using ordinary least squares, in the context of scaling; and
5. scaling/variable transformations using canonical variates and nonparametric additive regression.

## 8.1 Data

Consider the 506-patient prostate cancer dataset from Byar and Green.[87] The data are listed in [28, Table 46] and are available in ASCII form from `StatLib` (`lib.stat.cmu.edu`) in the `Datasets` area from this book's Web page. These data were from a randomized trial comparing four treatments for stage 3 and 4 prostate cancer, with almost equal numbers of patients on placebo and each of three doses of estrogen. Four patients had missing values on all of the following variables: `wt, pf, hx, sbp, dbp, ekg, hg, bm`; two of these patients were also missing `sz`. These patients are excluded from consideration. The ultimate goal of an analysis of the dataset might be to discover patterns in survival or to do an analysis of covariance to assess the effect of treatment while adjusting for patient heterogeneity. See Chapter 21 for such analyses. The data reductions developed here are general and can be used for a variety of dependent variables.

F.E. Harrell, Jr., *Regression Modeling Strategies*, Springer Series in Statistics, DOI 10.1007/978-3-319-19425-7_8

The variable names, labels, and a summary of the data are printed below.

```
require(Hmisc)
```

```
getHdata(prostate)  # Download and make prostate accessible
# Convert an old date format to R format
prostate$sdate ← as.Date(prostate$sdate)
d ← describe(prostate[2:17])
latex(d, file='')
```

**prostate[2:17]**
**16 Variables 502 Observations**

---

**stage : Stage**

```
    n missing unique Info  Mean
  502       0      2 0.73 3.424

  3 (289, 58%), 4 (213, 42%)
```

---

**rx**

```
    n missing unique
  502       0      4

  placebo (127, 25%), 0.2 mg estrogen (124, 25%)
  1.0 mg estrogen (126, 25%), 5.0 mg estrogen (125, 25%)
```

---

**dtime : Months of Follow-up**

```
    n missing unique Info  Mean  .05  .10   .25   .50   .75   .90   .95
  502       0     76    1 36.13 1.05 5.00 14.25 34.00 57.75 67.00 71.00

  lowest :  0  1  2  3  4, highest: 72 73 74 75 76
```

---

**status**

```
    n missing unique
  502       0     10

  alive (148, 29%), dead - prostatic ca (130, 26%)
  dead - heart or vascular (96, 19%), dead - cerebrovascular (31, 6%)
  dead - pulmonary embolus (14, 3%), dead - other ca (25, 5%)
  dead - respiratory disease (16, 3%)
  dead - other specific non-ca (28, 6%), dead - unspecified non-ca (7, 1%)
  dead - unknown cause (7, 1%)
```

---

**age : Age in Years**

```
    n missing unique Info  Mean .05 .10 .25 .50 .75 .90 .95
  501       1     41    1 71.46  56  60  70  73  76  78  80

  lowest : 48 49 50 51 52, highest: 84 85 87 88 89
```

---

**wt : Weight Index = wt(kg)-ht(cm)+200**

```
    n missing unique Info  Mean   .05   .10   .25   .50    .75    .90    .95
  500       2     67    1 99.03 77.95 82.90 90.00 98.00 107.00 116.00 123.00

  lowest :  69  71  72  73  74, highest: 136 142 145 150 152
```

---

**pf**

```
     n missing unique
   502       0      4

   normal activity (450, 90%), in bed < 50% daytime (37, 7%)
   in bed > 50% daytime (13, 3%), confined to bed (2, 0%)
```

---

**hx : History of Cardiovascular Disease**

```
     n missing unique Info Sum  Mean
   502       0      2 0.73  213 0.4243
```

---

**sbp : Systolic Blood Pressure/10**

```
     n missing unique Info Mean .05 .10 .25 .50 .75 .90 .95
   502       0     18 0.98 14.35  11  12  13  14  16  17  18

             8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 30
   Frequency 1 3 14 27 65 74 98 74 72 34 17 12  3  2  3  1  1  1
   %         0 1  3  5 13 15 20 15 14  7  3  2  1  0  1  0  0  0
```

---

**dbp : Diastolic Blood Pressure/10**

```
     n missing unique Info Mean .05 .10 .25 .50 .75 .90 .95
   502       0     12 0.95 8.149   6   6   7   8   9  10  10

             4 5  6   7   8  9 10 11 12 13 14 18
   Frequency 4 5 43 107 165 94 66  9  5  2  1  1
   %         1 1  9  21  33 19 13  2  1  0  0  0
```

---

**ekg**

```
     n missing unique
   494       8      7

   normal (168, 34%), benign (23, 5%)
   rhythmic disturb & electrolyte ch (51, 10%)
   heart block or conduction def (26, 5%), heart strain (150, 30%)
   old MI (75, 15%), recent MI (1, 0%)
```

---

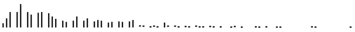
**hg : Serum Hemoglobin (g/100ml)**

```
     n missing unique Info Mean  .05  .10  .25  .50  .75  .90  .95
   502       0     91    1 13.45 10.2 10.7 12.3 13.7 14.7 15.8 16.4

   lowest :  5.899  7.000  7.199  7.800  8.199
   highest: 17.297 17.500 17.598 18.199 21.199
```

---

**sz**: Size of Primary Tumor (cm$^2$)



```
     n missing unique Info Mean .05 .10 .25  .50  .75  .90  .95
   497       5     55    1 14.63 2.0 3.0 5.0 11.0 21.0 32.0 39.2

   lowest :  0  1  2  3  4, highest: 54 55 61 62 69
```

---

**sg : Combined Index of Stage and Hist. Grade**

```
     n missing unique Info Mean .05 .10 .25 .50 .75 .90 .95
   491      11     11 0.96 10.31   8   8   9  10  11  13  13

             5 6 7  8   9 10  11 12 13 14 15
   Frequency 3 8 7 67 137 33 114 26 75  5 16
   %         1 2 1 14  28  7  23  5 15  1  3
```

---

```
ap : Serum Prostatic Acid Phosphatase
      n missing unique Info Mean   .05   .10   .25   .50   .75    .90    .95
    502       0    128    1 12.18 0.300 0.300 0.500 0.700 2.975 21.689 38.470

   lowest :   0.09999   0.19998   0.29999   0.39996   0.50000
   highest: 316.00000 353.50000 367.00000 596.00000 999.87500
```

```
bm : Bone Metastases
      n missing unique Info Sum  Mean
    502       0      2 0.41   82 0.1633
```

`stage` is defined by `ap` as well as X-ray results. Of the patients in stage 3, 0.92 have `ap` $\leq 0.8$. Of those in stage 4, 0.93 have `ap` $> 0.8$. Since `stage` can be predicted almost certainly from `ap`, we do not consider `stage` in some of the analyses.

## 8.2 How Many Parameters Can Be Estimated?

There are 354 deaths among the 502 patients. If predicting survival time were of major interest, we could develop a reliable model if no more than about $354/15 = 24$ parameters were *examined* against $Y$ in unpenalized modeling. Suppose that a full model with no interactions is fitted and that linearity is not assumed for any continuous predictors. Assuming `age` is almost linear, we could fit a restricted cubic spline function with three knots. For the other continuous variables, let us use five knots. For categorical predictors, the maximum number of degrees of freedom needed would be one fewer than the number of categories. For `pf` we could lump the last two categories since the last category has only 2 patients. Likewise, we could combine the last two levels of `ekg`. Table 8.1 lists the candidate predictors with the maximum number of parameters we consider for each.

**Table 8.1** Degrees of freedom needed for predictors

| Predictor: | rx | age | wt | pf | hx | sbp | dbp | ekg | hg | sz | sg | ap | bm |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| # Parameters: | 3 | 2 | 4 | 2 | 1 | 4 | 4 | 5 | 4 | 4 | 4 | 4 | 1 |

## 8.3 Redundancy Analysis

As described in Section 4.7.1, it is occasionally useful to do a rigorous redundancy analysis on a set of potential predictors. Let us run the algorithm discussed there, on the set of predictors we are considering. We will use a low threshold (0.3) for $R^2$ for demonstration purposes.

```
# Allow only 1 d.f. for three of the predictors
prostate ←
  transform(prostate,
            ekg.norm = 1*(ekg %in% c("normal","benign")),
            rxn = as.numeric(rx),
            pfn = as.numeric(pf))
# Force pfn, rxn to be linear because of difficulty of placing
# knots with so many ties in the data
# Note: all incomplete cases are deleted (inefficient)
redun(~ stage + I(rxn) + age + wt + I(pfn) + hx +
      sbp + dbp + ekg.norm + hg + sz + sg + ap + bm,
      r2=.3, type='adjusted', data=prostate)
```

```

Redundancy Analysis

redun(formula = ~stage + I(rxn) + age + wt + I(pfn) + hx +
    sbp + dbp + ekg.norm + hg + sz + sg + ap + bm,
    data = prostate, r2 = 0.3, type = "adjusted")

n: 483   p: 14   nk: 3

Number of NAs:   19
Frequencies of Missing Values Due to Each Variable
   stage   I(rxn)      age       wt   I(pfn)       hx      sbp
dbp
       0        0        1        2        0        0        0
0
ekg.norm       hg       sz       sg       ap       bm
       0        0        5       11        0        0


Transformation of target variables forced to be linear

R^2 cutoff: 0.3  Type: adjusted

R^2 with which each variable can be predicted from all other
    variables:

   stage   I(rxn)      age       wt   I(pfn)       hx      sbp
dbp
   0.658    0.000    0.073    0.111    0.156    0.062    0.452
0.417
ekg.norm       hg       sz       sg       ap       bm
   0.055    0.146    0.192    0.540    0.147    0.391

Rendundant variables:

stage sbp bm sg

Predicted from variables:

I(rxn) age wt I(pfn) hx dbp ekg.norm hg sz ap
```

```
 Variable Deleted   $R^2$ $R^2$ after later deletions
1           stage 0.658         0.658 0.646 0.494
2             sbp 0.452               0.453 0.455
3              bm 0.374                     0.367
4              sg 0.342
```

By any reasonable criterion on $R^2$, none of the predictors is redundant. `stage` can be predicted with an $R^2 = 0.658$ from the other 13 variables, but only with $R^2 = 0.493$ after deletion of 3 variables later declared to be “redundant.”

## 8.4 Variable Clustering

From Table 8.1, the total number of parameters is 42, so some data reduction should be considered. We resist the temptation to take the “easy way out” using stepwise variable selection so that we can achieve a more stable modeling 1 process and obtain unbiased standard errors. Before using a variable clustering procedure, note that `ap` is extremely skewed. To handle skewness, we use Spearman rank correlations for continuous variables (later we transform each variable using `transcan`, which will allow ordinary correlation coefficients to be used). After classifying `ekg` as “normal/benign” versus everything else, the Spearman correlations are plotted below.

```
x ← with(prostate,
          cbind(stage, rx, age, wt, pf, hx, sbp, dbp,
                ekg.norm, hg, sz, sg, ap, bm))
# If no missing data, could use cor(apply(x, 2, rank))
r ← rcorr(x, type="spearman")$r    # rcorr in Hmisc
maxabsr ← max(abs(r[row(r) != col(r)]))
```

```
p ← nrow(r)
plot(c(-.35,p+.5),c(.5,p+.25), type='n', axes=FALSE,
     xlab='',ylab='')    # Figure 8.1
v ← dimnames(r)[[1]]
text(rep(.5,p), 1:p, v, adj=1)
for(i in 1:(p-1)) {
  for(j in (i+1):p) {
    lines(c(i,i),c(j,j+r[i,j]/maxabsr/2),
          lwd=3, lend='butt')
    lines(c(i-.2,i+.2),c(j,j), lwd=1, col=gray(.7))
  }
  text(i, i, v[i], srt=-45, adj=0)
}
```

We perform a hierarchical cluster analysis based on a similarity matrix that contains pairwise Hoeffding $D$ statistics.[295] $D$ will detect nonmonotonic associations.

```
vc ← varclus(∼ stage + rxn + age + wt + pfn + hx +
               sbp + dbp + ekg.norm + hg + sz + sg + ap + bm,
               sim='hoeffding', data=prostate)
plot(vc)    # Figure 8.2
```

We combine `sbp` and `dbp`, and tentatively combine `ap, sg, sz`, and `bm`.

## 8.5 Transformation and Single Imputation Using `transcan`

Now we turn to the scoring of the predictors to potentially reduce the number of regression parameters that are needed later by doing away with the need for nonlinear terms and multiple dummy variables. The R `Hmisc` package `transcan` function defaults to using a maximum generalized variance method[368] that incorporates canonical variates to optimally transform both sides of a multiple regression model. Each predictor is treated in turn as a variable being predicted, and all variables are expanded into restricted cubic splines (for continuous variables) or dummy variables (for categorical ones).

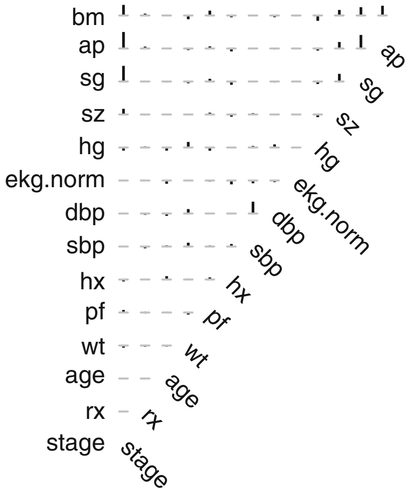


**Fig. 8.1** Matrix of Spearman $\rho$ rank correlation coefficients between predictors. Horizontal gray scale lines correspond to $\rho = 0$. The tallest bar corresponds to $|\rho| = 0.78$.

```
# Combine 2 levels of ekg (one had freq. 1)
levels(prostate$ekg)[levels(prostate$ekg) %in%
                     c('old MI', 'recent MI')] ← 'MI'

prostate$pf.coded ← as.integer(prostate$pf)
```

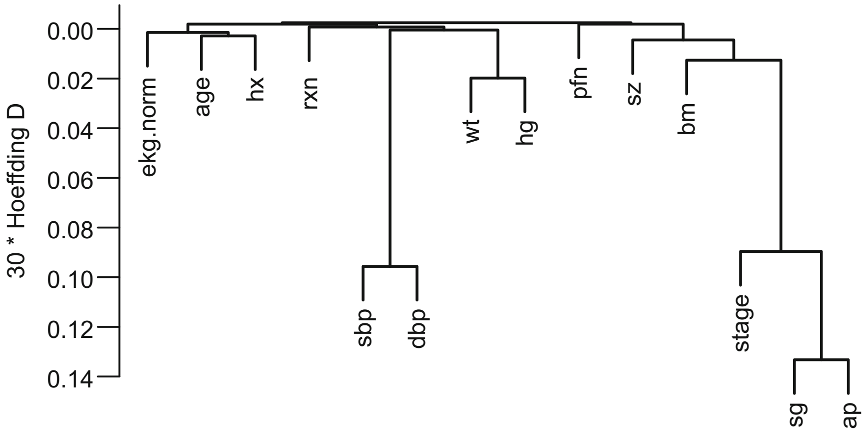


**Fig. 8.2** Hierarchical clustering using Hoeffding's $D$ as a similarity measure. Dummy variables were used for the categorical variable `ekg`. Some of the dummy variables cluster together since they are by definition negatively correlated.

```
# make a numeric version; combine last 2 levels of original
levels(prostate$pf) ← levels(prostate$pf)[c(1,2,3,3)]

ptrans ←
  transcan(~ sz + sg + ap + sbp + dbp +
           age + wt + hg + ekg + pf + bm + hx, imputed=TRUE,
           transformed=TRUE, trantab=TRUE, pl=FALSE,
           show.na=TRUE, data=prostate, frac=.1, pr=FALSE)
summary(ptrans, digits=4)
```

```
transcan(x = ~sz + sg + ap + sbp + dbp + age + wt + hg + ekg +
    pf + bm + hx, imputed = TRUE, trantab = TRUE, transformed = TRUE,
    pr = FALSE, pl = FALSE, show.na = TRUE, data = prostate,
    frac = 0.1)

Iterations: 8

R^2 achieved in predicting each variable:

   sz    sg    ap   sbp   dbp   age    wt    hg   ekg    pf    bm    hx
0.207 0.556 0.573 0.498 0.485 0.095 0.122 0.158 0.092 0.113 0.349 0.108

Adjusted R^2:

   sz    sg    ap   sbp   dbp   age    wt    hg   ekg    pf    bm    hx
0.180 0.541 0.559 0.481 0.468 0.065 0.093 0.129 0.059 0.086 0.331 0.083

Coefficients of canonical variates for predicting each (row) variable

       sz    sg    ap   sbp   dbp   age    wt    hg   ekg    pf    bm
sz           0.66  0.20  0.33  0.33 -0.01 -0.01  0.11  0.11  0.03 -0.36
sg   0.23        0.84  0.08  0.07 -0.02  0.01 -0.01 -0.07  0.02 -0.20
ap   0.07  0.80       -0.11 -0.05  0.03 -0.02  0.01  0.01  0.00 -0.83
sbp  0.13  0.10 -0.14       -0.94  0.14 -0.09  0.03  0.10  0.10 -0.03
dbp  0.13  0.09 -0.06 -0.98        0.14  0.07  0.05  0.03  0.04  0.03
age -0.02 -0.06  0.18  0.58  0.57        0.14  0.46  0.43 -0.03  1.05
wt  -0.02  0.06 -0.08 -0.31  0.23  0.12        0.51 -0.06  0.21 -1.09
hg   0.13 -0.02  0.03  0.09  0.15  0.33  0.43       -0.02  0.24 -1.53
ekg  0.20 -0.38  0.10  0.42  0.12  0.41 -0.04 -0.04        0.15 -0.42
pf   0.04  0.08  0.02  0.36  0.14 -0.03  0.22  0.29  0.13       -1.75
bm  -0.02 -0.03 -0.13  0.00  0.00  0.03 -0.04 -0.06 -0.01 -0.06
```

```
hx    0.04  0.05 -0.01 -0.04  0.00 -0.06  0.02 -0.01 -0.09 -0.04 -0.05
       hx
sz    0.34
sg    0.14
ap   -0.03
sbp  -0.14
dbp  -0.01
age  -0.76
wt    0.27
hg   -0.12
ekg  -1.23
pf   -0.46
bm   -0.02
hx

Summary of imputed values

sz
       n missing  unique    Info    Mean
       5       0       4    0.95   12.86

6 (2, 40%), 7.416 (1, 20%), 20.18 (1, 20%), 24.69 (1, 20%)
sg
       n missing  unique    Info    Mean     .05     .10     .25     .50
      11       0      10       1    10.1   6.900   7.289   7.697  10.270
     .75     .90     .95
  10.560  15.000  15.000

           6.511 7.289 7.394 8 10.25 10.27 10.32 10.39 10.73 15
Frequency      1     1     1 1     1     1     1     1     1  2
%              9     9     9 9     9     9     9     9     9 18
age
       n missing  unique    Info    Mean
       1       0       1       0   71.65
wt
       n missing  unique    Info    Mean
       2       0       2       1   97.77

91.24 (1, 50%), 104.3 (1, 50%)
ekg
       n missing  unique    Info    Mean
       8       0       4     0.9   2.625

1 (3, 38%), 3 (3, 38%), 4 (1, 12%), 5 (1, 12%)

Starting estimates for imputed values:

  sz   sg  ap  sbp dbp  age   wt   hg ekg  pf  bm  hx
11.0 10.0 0.7 14.0 8.0 73.0 98.0 13.7 1.0 1.0 0.0 0.0
```

```
ggplot(ptrans, scale=TRUE) +
  theme(axis.text.x=element_text(size=6))   # Figure 8.3
```

The plotted output is shown in Figure 8.3. Note that at face value the transformation of ap was derived in a circular manner, since the combined index of stage and histologic grade, sg, uses in its stage component a cutoff on ap. However, if sg is omitted from consideration, the resulting transformation for ap does not change appreciably. Note that bm and hx are represented as binary variables, so their coefficients in the table of canonical variable coefficients are on a different scale. For the variables that were actually transformed, the coefficients are for standardized transformed variables (mean 0, variance 1). From examining the $R^2$s, age, wt, ekg, pf, and hx are not strongly related to other variables. Imputations for age, wt, ekg are thus relying more on the median or modal values from the marginal distributions. From the coefficients of first (standardized) canonical variates, sbp is predicted almost solely from dbp; bm is predicted mainly from ap, hg, and pf. 2

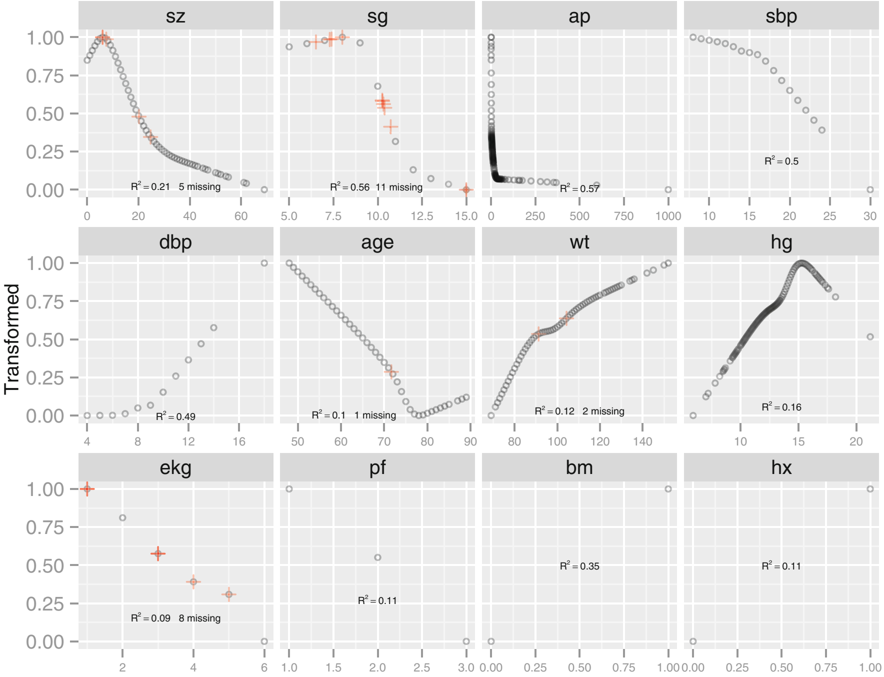


**Fig. 8.3** Simultaneous transformation and single imputation of all candidate predictors using `transcan`. Imputed values are shown as red plus signs. Transformed values are arbitrarily scaled to $[0, 1]$.

## 8.6 Data Reduction Using Principal Components

The first PC, $\text{PC}_1$, is the linear combination of standardized variables having maximum variance. $\text{PC}_2$ is the linear combination of predictors having the second largest variance such that $\text{PC}_2$ is orthogonal to (uncorrelated with) $\text{PC}_1$. If there are $p$ raw variables, the first $k$ PCs, where $k < p$, will explain only part of the variation in the whole system of $p$ variables unless one or more of the original variables is exactly a linear combination of the remaining variables. Note that it is common to scale and center variables to have mean zero and variance 1 before computing PCs.

The response variable (here, time until death due to any cause) is not examined during data reduction, so that if PCs are selected by variance explained in the $X$-space and not by variation explained in $Y$, one needn't correct for model uncertainty or multiple comparisons.

PCA results in data reduction when the analyst uses only a subset of the $p$ possible PCs in predicting $Y$. This is called *incomplete principal component regression*. When one sequentially enters PCs into a predictive model in a strict pre-specified order (i.e., by descending amounts of variance explained for the system of $p$ variables), model uncertainty requiring bootstrap adjustment is minimized. In contrast, model uncertainty associated with stepwise regression (driven by associations with $Y$) is massive.

For the prostate dataset, consider PCs on raw candidate predictors, expanding polytomous factors using dummy variables. The R function `princomp` is used, after singly imputing missing raw values using `transcan`'s optimal additive nonlinear models. In this series of analyses we ignore the treatment variable, `rx`.

```
# Impute all missing values in all variables given to transcan
imputed ← impute(ptrans, data=prostate, list.out=TRUE)
```

```
Imputed missing values with the following frequencies
 and stored them in variables with their original names:

 sz  sg age  wt ekg
  5  11   1   2   8
```

```
imputed ← as.data.frame(imputed)

# Compute principal components on imputed data.
# Create a design matrix from ekg categories
Ekg ← model.matrix(~ ekg, data=imputed)[, -1]
# Use correlation matrix
pfn ← prostate$pfn
prin.raw ← princomp(~ sz + sg + ap + sbp + dbp + age +
                     wt + hg + Ekg + pfn + bm + hx,
                     cor=TRUE, data=imputed)

plot(prin.raw, type='lines', main='', ylim=c(0,3)) #Figure 8.4
# Add cumulative fraction of variance explained
addscree ← function(x, npcs=min(10, length(x$sdev)),
                     plotv=FALSE,
                     col=1, offset=.8, adj=0, pr=FALSE) {
  vars ← x$sdev^2
  cumv ← cumsum(vars)/sum(vars)
  if(pr) print(cumv)
  text(1:npcs, vars[1:npcs] + offset*par('cxy')[2],
       as.character(round(cumv[1:npcs], 2)),
       srt=45, adj=adj, cex=.65, xpd=NA, col=col)
  if(plotv) lines(1:npcs, vars[1:npcs], type='b', col=col)
}
addscree(prin.raw)
prin.trans ← princomp(ptrans$transformed, cor=TRUE)
addscree(prin.trans, npcs=10, plotv=TRUE, col='red',
         offset=-.8, adj=1)
```

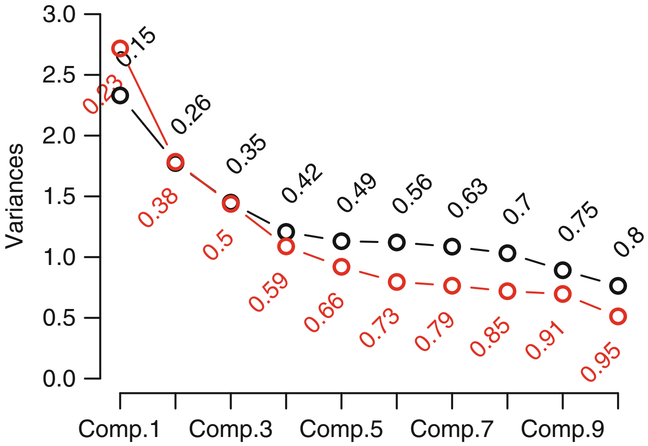


**Fig. 8.4** Variance of the system of raw predictors (black) explained by individual principal components (lines) along with cumulative proportion of variance explained (text), and variance explained by components computed on `transcan`-transformed variables (red)

The resulting plot shown in Figure 8.4 is called a "scree" plot [325, pp. 96–99, 104, 106]. It shows the variation explained by the first $k$ principal components as $k$ increases all the way to 16 parameters (no data reduction). It requires 10 of the 16 possible components to explain $> 0.8$ of the variance, and the first 5 components explain 0.49 of the variance of the system. Two of the 16 dimensions are almost totally redundant.

After repeating this process when transforming all predictors via `transcan`, we have only 12 degrees of freedom for the 12 predictors. The variance explained is depicted in Figure 8.4 in red. It requires at least 9 of the 12 possible components to explain $\geq 0.9$ of the variance, and the first 5 components explain 0.66 of the variance as opposed to 0.49 for untransformed variables.

Let us see how the PCs "explain" the times until death using the Cox regression[132] function from `rms`, `cph`, described in Chapter 20. In what follows we vary the number of components used in the Cox models from 1 to all 16, computing the AIC for each model. AIC is related to model log likelihood penalized for number of parameters estimated, and lower is better. For reference, the AIC of the model using all of the original predictors, and the AIC of a full additive spline model are shown as horizontal lines.

```
require(rms)
```

```
S ← with(prostate, Surv(dtime, status != "alive"))
# two-column response var.

pcs ← prin.raw$scores          # pick off all PCs
aic ← numeric(16)
for(i in 1:16) {
  ps ← pcs[,1:i]
  aic[i] ← AIC(cph(S ~ ps))
}   # Figure 8.5
plot(1:16, aic, xlab='Number of Components Used',
     ylab='AIC', type='l', ylim=c(3950,4000))
f ← cph(S ~ sz + sg + log(ap) + sbp + dbp + age + wt + hg +
          ekg + pf + bm + hx, data=imputed)
abline(h=AIC(f), col='blue')
f ← cph(S ~ rcs(sz,5) + rcs(sg,5) + rcs(log(ap),5) +
          rcs(sbp,5) + rcs(dbp,5) + rcs(age,3) + rcs(wt,5) +
          rcs(hg,5) + ekg + pf + bm + hx,
          tol=1e-14, data=imputed)
```

```
abline(h=AIC(f), col='blue', lty=2)
```

For the money, the first 5 components adequately summarizes all variables, if linearly transformed, and the full linear model is no better than this. The model allowing all continuous predictors to be nonlinear is not worth its added degrees of freedom.

Next check the performance of a model derived from cluster scores of transformed variables.

```
# Compute PC1 on a subset of transcan-transformed predictors
pco ← function(v) {
  f ← princomp(ptrans$transformed[,v], cor=TRUE)
  vars ← f$sdev^2
  cat('Fraction of variance explained by PC1:',
      round(vars[1]/sum(vars),2), '\n')
  f$scores[,1]
}
tumor   ← pco(c('sz','sg','ap','bm'))
```

```
Fraction of variance explained by PC1: 0.59
```

```
bp      ← pco(c('sbp','dbp'))
```

```
Fraction of variance explained by PC1: 0.84
```

```
cardiac ← pco(c('hx','ekg'))
```

```
Fraction of variance explained by PC1: 0.61
```

```
# Get transformed individual variables that are not clustered
other   ← ptrans$transformed[,c('hg','age','pf','wt')]
f ← cph(S ~ tumor + bp + cardiac + other)  # other is matrix
AIC(f)
```

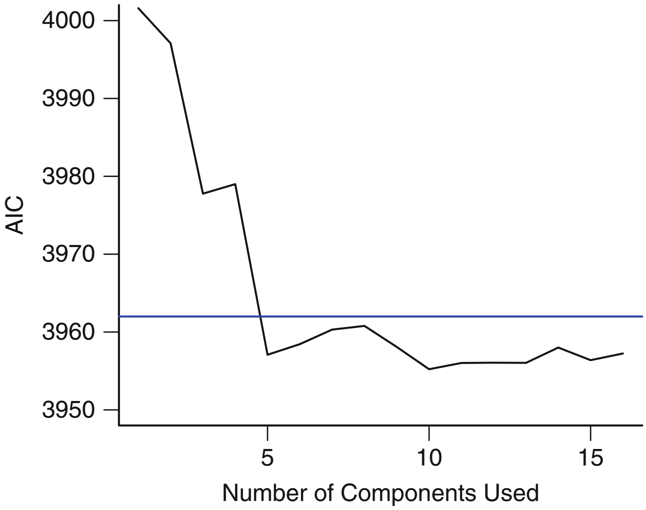


**Fig. 8.5** AIC of Cox models fitted with progressively more principal components. The solid blue line depicts the AIC of the model with all original covariates. The dotted blue line is positioned at the AIC of the full spline model.

```
[1] 3954.393
```

```
print(f, latex=TRUE, long=FALSE, title='')
```

| | | Model Tests | | Discrimination Indexes | |
|---|---|---|---|---|---|
| Obs | 502 | LR $\chi^2$ | 81.11 | $R^2$ | 0.149 |
| Events | 354 | d.f. | 7 | $D_{xy}$ | 0.286 |
| Center | 0 | $\Pr(>\chi^2)$ | 0.0000 | $g$ | 0.562 |
| | | Score $\chi^2$ | 86.81 | $g_r$ | 1.755 |
| | | $\Pr(>\chi^2)$ | 0.0000 | | |

| | Coef | S.E. | Wald $Z$ | $\Pr(>\lvert Z\rvert)$ |
|---|---|---|---|---|
| tumor | -0.1723 | 0.0367 | -4.69 | $<$ 0.0001 |
| bp | -0.0251 | 0.0424 | -0.59 | 0.5528 |
| cardiac | -0.2513 | 0.0516 | -4.87 | $<$ 0.0001 |
| hg | -0.1407 | 0.0554 | -2.54 | 0.0111 |
| age | -0.1034 | 0.0579 | -1.79 | 0.0739 |
| pf | -0.0933 | 0.0487 | -1.92 | 0.0551 |
| wt | -0.0910 | 0.0555 | -1.64 | 0.1012 |

The `tumor` and `cardiac` clusters seem to dominate prediction of mortality, and the AIC of the model built from cluster scores of transformed variables compares favorably with other models (Figure 8.5).

### *8.6.1 Sparse Principal Components*

A disadvantage of principal components is that every predictor receives a nonzero weight for every component, so many coefficients are involved even through the effective degrees of freedom with respect to the response model are reduced. *Sparse principal components*[672] uses a penalty function to reduce the magnitude of the loadings variables receive in the components. If an L1 penalty is used (as with the *lasso*), some loadings are shrunk to zero, resulting in some simplicity. Sparse principal components combines some elements of variable clustering, scoring of variables within clusters, and redundancy analysis.

Filzmoser, Fritz, and Kalcher[191] have written a nice R package `pcaPP` for doing sparse PC analysis.[a] The following example uses the prostate data again. To allow for nonlinear transformations and to score the `ekg` variable in the prostate dataset down to a scalar, we use the `transcan`-transformed predictors as inputs.

```
require(pcaPP)
```

```
s ← sPCAgrid(ptrans$transformed, k=10, method='sd',
              center=mean, scale=sd, scores=TRUE,
              maxiter=10)
plot(s, type='lines', main='', ylim=c(0,3))   # Figure 8.6
addscree(s)
s$loadings   # These loadings are on the orig. transcan scale
```

```
Loadings:
    Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8 Comp.9 Comp.10
sz   0.248                                                   0.950
sg   0.620                                                          0.522
ap   0.634                                                  -0.305
sbp        -0.707
dbp         0.707
age                       1.000
wt                                              1.000
hg                                       1.000
ekg                                                    1.000
pf                 1.000
bm  -0.391                                                          0.852
hx                               1.000

               Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8
SS loadings     1.000  1.000  1.000  1.000  1.000  1.000  1.000  1.000
Proportion Var  0.083  0.083  0.083  0.083  0.083  0.083  0.083  0.083
Cumulative Var  0.083  0.167  0.250  0.333  0.417  0.500  0.583  0.667
               Comp.9 Comp.10
SS loadings     1.000   1.000
Proportion Var  0.083   0.083
Cumulative Var  0.750   0.833
```

Only nonzero loadings are shown. The first sparse PC is the `tumor` cluster used above, and the second is the blood pressure cluster. Let us see how well incomplete sparse principal component regression predicts time until death.

[a] The `spca` package is a new sparse PC package that should also be considered.

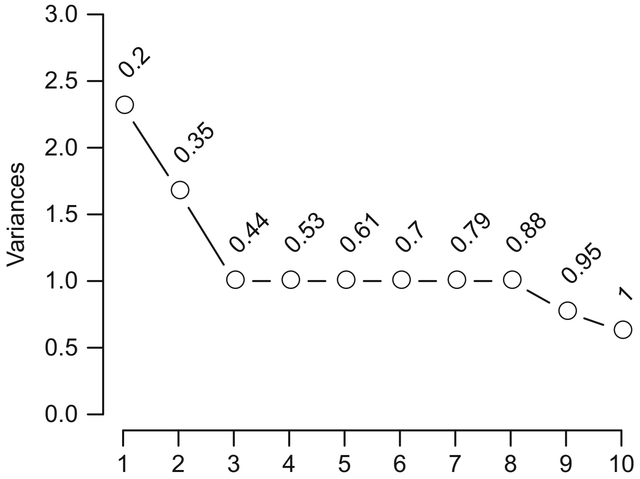


**Fig. 8.6** Variance explained by individual sparse principal components (lines) along with cumulative proportion of variance explained (text)

```
pcs ← s$scores           # pick off sparse PCs
aic ← numeric(10)
for(i in 1:10) {
  ps ← pcs[,1:i]
  aic[i] ← AIC(cph(S ∼ ps))
}   # Figure 8.7
plot(1:10, aic, xlab='Number of Components Used',
     ylab='AIC', type='l',  ylim=c(3950,4000))
```

More components are required to optimize AIC than were seen in Figure 8.5, but a model built from 6–8 sparse PCs performed as well as the other models.

## 8.7 Transformation Using Nonparametric Smoothers

The ACE nonparametric additive regression method of Breiman and Friedman[68] transforms both the left-hand-side variable and all the right-hand-side variables so as to optimize $R^2$. ACE can be used to transform the predictors using the R `ace` function in the `acepack` package, called by the `transace` function in the `Hmisc` package. `transace` does not impute data but merely does casewise deletion of missing values. Here `transace` is run after single imputation by `transcan`. `binary` is used to tell `transace` which variables not to try to predict (because they need no transformation). Several predictors are restricted to be monotonically transformed.

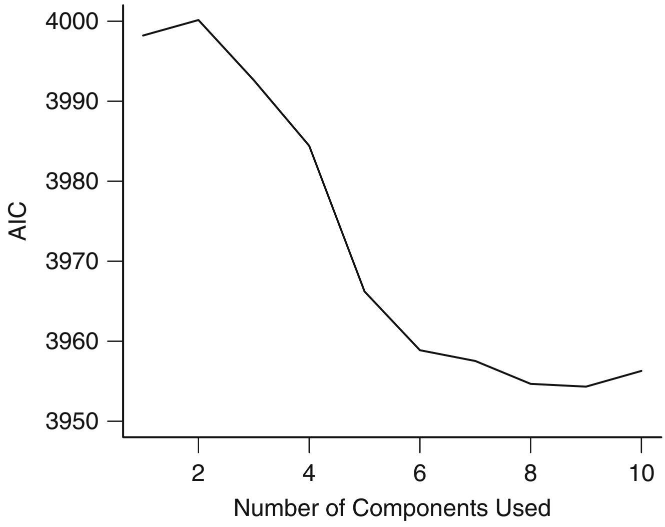


**Fig. 8.7** Performance of sparse principal components in Cox models

```
x ← with(imputed,
         cbind(sz, sg, ap, sbp, dbp, age, wt, hg, ekg, pf,
               bm, hx))
monotonic ← c("sz","sg","ap","sbp","dbp","age","pf")
transace(x, monotonic,   # Figure 8.8
        categorical="ekg", binary=c("bm","hx"))
```

```
R^2 achieved in predicting each variable:

       sz        sg        ap       sbp       dbp       age        wt
0.2265824 0.5762743 0.5717747 0.4823852 0.4580924 0.1514527 0.1732244
       hg       ekg        pf        bm        hx
0.2001008 0.1110709 0.1778705        NA        NA
```

Except for `ekg`, `age`, and for arbitrary sign reversals, the transformations in Figure 8.8 determined using `transace` were similar to those in Figure 8.3. The `transcan` transformation for `ekg` makes more sense.

## 8.8 Further Reading

1 Sauerbrei and Schumacher[541] used the bootstrap to demonstrate the variability of a standard variable selection procedure for the prostate cancer dataset.

2 Schemper and Heinze[551] used logistic models to impute dichotomizations of the predictors for this dataset.

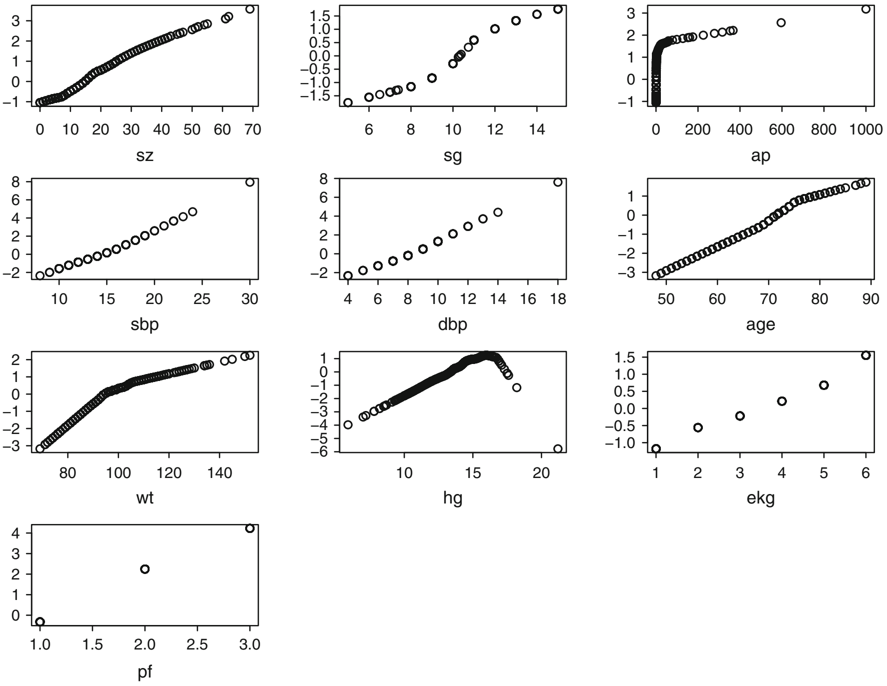


**Fig. 8.8** Simultaneous transformation of all variables using ACE.

## 8.9 Problems

The Mayo Clinic conducted a randomized trial in primary biliary cirrhosis (PBC) of the liver between January 1974 and May 1984, to compare D-penicillamine with placebo. The drug was found to be ineffective [197, p. 2], and the trial was done before liver transplantation was common, so this trial constitutes a natural history study for PBC. Followup continued through July, 1986. For the 19 patients that did undergo transplant, followup time was censored (`status=0`) at the day of transplant. 312 patients were randomized, and another 106 patients were entered into a registry. The nonrandomized patients have most of their laboratory values missing, except for bilirubin, albumin, and prothrombin time. 28 randomized patients had both serum cholesterol and triglycerides missing. The data, which consist of clinical, biochemical, serologic, and histologic information, are listed in [197, pp. 359–375]. The PBC data are discussed and analyzed in [197, pp. 2–7, 102–104, 153–162], [158], [7] (a tree-based analysis which on its p. 480 mentions some possible lack of fit of the earlier analyses), and [361]. The data are stored in the datasets web site so may be accessed using the `Hmisc` `getHdata` function with argument `pbc`. Use only the data on randomized patients for all analyses. For Problems 1–6, ignore followup time, status, and drug.

1. Do an initial variable clustering based on ranks, using pairwise deletion of missing data. Comment on the potential for one-dimensional summaries of subsets of variables being adequate summaries of prognostic information.
2. `cholesterol, triglycerides, platelets`, and `copper` are missing on some patients. Impute them using a method you recommend. Use some or all of the remaining predictors and possibly the outcome. Provide a correlation coefficient describing the usefulness of each imputation model. Provide the actual imputed values, specifying observation numbers. For all later analyses, use imputed values for missing values.
3. Perform a scaling/transformation analysis to better measure how the predictors interrelate and to possibly pretransform some of them. Use `transcan` or ACE. Repeat the variable clustering using the transformed scores and Pearson correlation or using an oblique rotation principal component analysis. Determine if the correlation structure (or variance explained by the first principal component) indicates whether it is possible to summarize multiple variables into single scores.
4. Do a principal component analysis of all transformed variables simultaneously. Make a graph of the number of components versus the cumulative proportion of explained variation. Repeat this for laboratory variables alone.
5. Repeat the overall PCA using sparse principal components. Pay attention to how best to solve for sparse components, e.g., consider the `lambda` parameter in `sPCAgrid`.
6. How well can variables (lab and otherwise) that are routinely collected (on nonrandomized patients) capture the information (variation) of the variables that are often missing? It would be helpful to explore the strength of interrelationships by

   a. correlating two $\mathrm{PC}_1$s obtained from untransformed variables,
   b. correlating two $\mathrm{PC}_1$s obtained from transformed variables,
   c. correlating the best linear combination of one set of variables with the best linear combination of the other set, and
   d. doing the same on transformed variables.

   For this problem consider only complete cases, and transform the 5 nonnumeric categorical predictors to binary 0–1 variables.
7. Consider the patients having complete data who were randomized to placebo. Consider only models that are linear in all the covariates.

   a. Fit a survival model to predict time of death using the following covariates: `bili, albumin, stage, protime, age, alk.phos, sgot, chol, trig, platelet, copper`.
   b. Perform an ordinary principal component analysis. Fit the survival model using only the first 3 PCs. Compare the likelihood ratio $\chi^2$ and AIC with that of the model using the original variables.

c. Considering the PCs are fixed, use the bootstrap to estimate the 0.95 confidence interval of the inter-quartile-range age effect on the original scale, and the same type of confidence interval for the coefficient of $\mathrm{PC}_1$.
d. Now accounting for uncertainty in the PCs, compute the same two confidence intervals. Compare and interpret the two sets. Take into account the fact that PCs are not unique to within a sign change.

R programming hints for this exercise are found on the course web site.