The dr package

Sanford Weisberg

School of Statistics, University of Minnesota, Minneapolis, MN 55455.

October 31, 2024

Abstract

The `dr` package for R for *dimension reduction regression* was first documented in Weisberg (2002). This is a revision of that article, to correspond to version 3.0.0 of `dr` for R added to CRAN (cran.r-project.org) in Fall 2007.

Regression is the study of the dependence of a response variable y on a collection of p predictors collected in x. In *dimension reduction regression*, we seek to find a few linearly independent linear combinations $\beta_1'x, \ldots, \beta_d'x$, such that all the information about the regression is contained in these d linear combinations. If d is very small, perhaps one or two, then the regression problem can be summarized using simple graphics; for example, for $d = 1$, the plot of y versus $\beta_1'x$ contains all the regression information. When $d = 2$, a 3D plot contains all the information. Formal estimation methods can additionally be used given the lower dimensional problem based on the few linear combinations.

The primary goals of dimension reduction regression are determining the dimension d, and estimating $\beta_1, \ldots, \beta_d$, or more precisely the subspace of $\Re^p$ spanned by $\beta_1, \ldots, \beta_d$. Several methods for these goals have been suggested in the literature and some of these are implemented in `dr`. Methods implemented in the package include sliced inverse regression or `sir`, sliced average variance estimates, or `save`, principal Hessian directions, or `phd`, and inverse regression estimation, or `ire`. The first three of these methods estimate the span of the βs by examining the inverse regression problem of $x|y$, rather than the forward regression problem of $y|x$, and all estimate the βs as the solution of an eigenvalue problem. The method `ire` is new in Version 3.0.0 and estimates a basis for the subspace of interest by minimizing a quadratic objective function.

Determining d is approached as a testing problem. *Marginal dimension tests* are used for a hypothesis of the form $d = d_0$ versus an alternative $d > d_0$, without restriction on the subspaces involved other than their dimension. The first of these tests was proposed by Li (1991) in his paper on `sir`. Added to Version 3.0.0 are tests of *coordinate hypotheses*, proposed by Cook (2004), which effectively provides a test for elimination of predictors to reduce dimension, although the method is more general than this. These latter tests provide the basis for stepwise elimination of predictors. These tests are available in `dr` for methods `sir`, `save`, and `ire`.

Finally, the methods sir, save and ire are extended to allow for inclusion of categorical predictors in x.

1   Introduction

In the general regression problem, we have a response y of dimension $k \geq 1$ and p-dimensional predictor x, and the goal is to learn about how the conditional distributions $F(y|x)$ change as x varies through its sample space. In parametric regression, we specify a known functional form for these conditional distributions, apart from a few parameters to be estimated. In nonparametric regression no assumptions are made about F, but progress is really only possible if the dimensions p and k are small.

Dimension reduction regression is one intermediate possibility between the parametric and nonparametric extremes. We assume without loss of information that the conditional distributions can be indexed by d linear combinations, or for some probably unknown $p \times d$ matrix B

$$F(y|x) = F(y|B'x) \tag{1}$$

or in words, all the information in x about y is contained in the d linear combinations $B'x$. This representation always holds trivially, by setting $B = I$, the $p \times p$ identity matrix, and so the usual goal is to find the B of lowest possible dimension for which this representation holds. If (1) holds for a particular B, then it also holds for $B^* = BA$, where A is any full rank matrix, and hence the unique part of the regression summary is the subspace that is spanned by the columns of B, which we denote $\mathcal{S}(B)$. The choice of B that gives $\mathcal{S}(B)$ of smallest dimension is called the *central subspace*. Cook (1998) provides a more complete introduction to these ideas, including discussion of when the central subspace exists and is unique.

The dr package for R provides methods for this approach to regression problems. The basic idea is to estimate both a dimension d, and a set of basis vectors $\hat{\beta}_1, \ldots, \hat{\beta}_d$ that are linearly independent and span $\mathcal{S}(B)$.

For readers familiar with earlier versions of dr, here is list of the additions to Version 3.0.0. If you are new to the package, you may wish to skip to this list.

New methods Inverse regression estimates, Cook and Ni (2005) and Wen and Cook (2007) are specified in a call to dr with method="ire", as described in Section 7. This method is different from the other methods in dr in that the estimated directions minimize an objective function. Other methods compute a matrix M and then estimate a subspace by the span eigenvectors corresponding to the first few eigenvalues of M[1].

---

[1] It is possible to view the sir method as minimizing an objective function as well, although this is not the way it was originally presented; Cook (2004)

Marginal dimension tests for save Tests for the dimension hypotheses in save were presented in Shao, Cook and Weisberg (2007), and are included in dr.

Coordinate hypotheses In previous versions of dr, the only tests available were marginal dimension tests about the dimension of the central (mean) subspace, without actually specifying any characteristics of what that subspace might be. Cook (2004) suggested both marginal and conditional tests, in which specific hypotheses can be tested. For example, one could test to see if the central (mean) subspace is orthogonal to a particular predictor. Cook described both marginal and conditional coordinate tests. The marginal coordinate tests are available for sir, save and ire, with and without categorical predictors. Conditional coordinate tests are available for sir without categorical predictors, and for ire with and without categorical predictors. Conditional coordinate tests for other methods may be added to dr in a later release.

Based on the marginal coordinate tests, we could imagine a stepwise deletion of potential predictors from a pool of possible predictors. This is implemented for sir, save and ire using the standard R function drop1. Since step in R is not a generic function, a function called dr.step is included here.

Categorical predictors Chiaromonte, Cook and Li (2002) suggested how dimension reduction methods could be extended to problems with categorical predictors. This methodology is available in dr using the groups argument. For example, group = ~Sex used a one-sided formula to specify that the levels of a variable Sex provides a grouping variable. This argument is recognized for sir, Chiaromonte, Cook and Li (2002), save, Shao, Cook and Weisberg (submitted), and for ire, Wen and Cook (2007).

In addition to these major additions, several minor changes have been made as well. Many of the methods require approximation of p-values for statistics distributed as a linear combination of χ^2 random variables. The user may select to use either the Wood (1989) or Bentler and Xie (2000) approximation to a linear combination of Chi-squared random variables to approximation significance levels of many of the tests.

The standard update method in R can be used with dr. Standard shortcuts for formulas can also be used. For example, the function call dr(ais) will automatically select the first column of the data frame ais as the response and the remaining columns as predictors, using defaults for all other arguments. Similarly, dr(LBM~.,ais) will use *LBM* as the response and all other columns as predictors.

Several of the methods require approximating the response Y by a discrete version by slicing into nonoverlapping slices. The function dr.slices is used for this purpose, and has been improved to give better handling of ties. To correspond to the program *Arc*, dr.slices.arc, which is the old slicing function, is also available for use.

2 Generalities and notation

All the methods require a few underlying assumptions. The following three are most common (Cook, 2004, Section 3):

Linearity condition This assumption concerns only the the marginal distribution of the predictors x. If we write $P_{\mathcal{S}(B)}$ as the projection on the central subspace $\mathcal{S}(B)$, then we must have $\mathrm{E}(x|P_{\mathcal{S}(B)}x) = P_{\mathcal{S}(B)}x$. A stronger condition that could be checked in practice is that $\mathrm{E}(a'x|b'x)$ is linear in $b'x$ for all vectors a and b. If x is normal the condition is satisfied, but normality is much stronger than needed.

Constant covariance condition Some methods require conditions on second moments of the predictors as well that the matrix $\mathrm{Var}(x|P_{\mathcal{S}(B)}x)$ is a nonrandom matrix. This condition is also satisfied by x normal, and is satisfied approximately if x has an elliptically contoured distribution.

Coverage condition This condition generally requires that a method can recover all of the central subspace, not just part of it. Not all methods can satisfy this condition, however. Ordinary least squares regression is a dimension reduction method if the linearity condition is satisfied, but it can only recover at most one basis vector in $\mathcal{S}(B)$. Sliced inverse regression can recover only directions associated with the mean $\mathrm{E}(y|x)$ but not of higher moments. In addition, if the number of slices for sir is less than d, all directions cannot be recovered.

Suppose we have data (x_i, y_i), for $i = 1, \ldots, n$ that are independent and collected into a $n \times p$ matrix X and an n-vector Y if $k = 1$ and a $n \times k$ matrix Y if $k > 1$. A multivariate response is not implemented for phd or ire. In addition, suppose we have nonnegative weights $w_1, \ldots, w_n$ whose sum is n; if unspecified, we take all the $w_i = 1$. The weights are used to improve the elliptical symmetry of the (weighted) distribution of the x_i and thereby satisfy the linearity condition at least approximately; see Section 9.1.

All the methods described here are invariant under location and scale change, and all start by replacing X by a matrix Z such that the columns of Z have mean zero and identity covariance matrix. One way to do the centering and scaling uses the QR decomposition. Define $\bar{x} = \sum w_i x_i / n$ to be the vector of weighted column means, $W = \mathrm{diag}(w_i)$. We use the QR decomposition to find a matrix Q with orthonormal columns and an upper triangular matrix R such that

$$QR = W^{1/2}(X - 1\bar{x}') \text{ and } Z = \sqrt{n}Q \tag{2}$$

Then the matrix Z has column means zero and sample identity covariance matrix. If $\hat{B}$ has columns that estimate a basis for the central subspace for the regression of $y|z$, then $R^{-1}\hat{B}$ provides a basis for the space the regression of $y|x$.

With the exception of the ire method described in Section 7, the methods implemented in dr have the following general pattern.

4

1. The methods sir, save and phd are based on study of the inverse regression problem of $x|y$ or $z|y$ rather than the forward regression problem of $y|x$ or $y|z$. All three methods require the linearity condition, and possibly the constant covariance condition. They use the scaled and centered data Z to find a matrix $\hat{M}$ that is a consistent estimate of a population matrix M with the property that $\mathcal{S}_z(M) \subseteq \mathcal{S}_z(B)$. For sir, we estimate only vectors in the *central mean subspace*, Cook and Li (2002), which guarantees only that $\mathrm{E}(y|B'z) = \mathrm{E}(y|z)$, ignoring any information beyond the first moment; save and phd are more general. If the coverage condition is assumed, than all of $\mathcal{S}_z(B)$ is recovered.

   Assuming the dimension of $\mathcal{S}_z(B)$ is d, a basis for $\mathcal{S}(B)$ in X-coordinates is $R^{-1}\hat{B}$, which can be used to form on orthonormal basis for this space in the original p dimensional coordinate system of the predictors.

2. The tests concerning the dimension d are called *marginal dimensional tests* and are available for sir, save, ire and one version of phd; details are given in the following sections or the references therein. It is usual to do testing in a sequential fashion, first testing $d = 0$ versus $d > 0$, then testing $d \leq 1$ versus $d > 1$, and continue until a nonsignificant result is obtained.

   Also available are *coordinate tests*, in which we test to see if the central subspace is orthogonal to a prespecified subspace. These tests can be done marginally, with no assumptions about the dimension d, or for some methods conditionally, assuming d is known. This is particularly useful for deciding if a predictor, which represents a one-dimensional subspace, can be removed from a model. This allows the use of stepwise deletion of variables, as will be discussed in Section 4.2.

3. If the matrix $\hat{B}$ has columns with basis vectors for the estimated central subspace, then $(X - 1\bar{x}')\hat{B}$ gives the d linear combinations of the variables in the n-dimensional observation space; these are computed using `dr.directions`, and can be used in graphical procedures or in other fitting methods.

3 The `dr` function

The main function in the `dr` package is also called `dr`. Here are the arguments, with their defaults:

```
dr(formula, data, subset, group=NULL, na.action = na.fail,
    weights,method = "sir", chi2approx="bx",...)
```

The arguments `formula`, `data`, `subset`, `na.action`, and `weights` are the save as the arguments of the same name with the standard R function `lm`. The other arguments are unique to `dr`.

The required `formula` is a two-sided formula with the response on the left side, or a matrix if the response is multivariate, and a set of predictors on the right

Table 1: Methods implemented in `dr`. "Multivariate" is yes if the response can be multivariate. "Partial" is yes if conditioning on categorical predictors is allowed using the `group` argument. "Coord" is yes if coordinate tests are available.

Name	Reference	Multivariate	Partial	Coord
sir	Li (1991), Chiaromonte, Cook and Li (2002)	Yes	Yes	Yes
save	Cook and Weisberg (1991); Shao, Cook and Weisberg (2007)	Yes	Yes	Yes
phdy	Li (1992), Cook (1998)	No	No	No
phdres	Cook (1998)	No	No	No
phdq	Li (1992)	No	No	No
ire	Cook and Ni (2006), Wen and Cook (2007)	No	Yes	Yes

side. Although any formula that is valid for `lm` can be used, the predictors are expected to all be continuous without any nesting or crossing. A typical formula might be `y~x1+x2+x3` for a univariate response, or `cbind(y1,y2)~x1+x2+x3` for a bivariate response.

The argument `data` specifies an optional data frame for the variables. The `subset` argument specifies which rows of the data are to be used in the analysis, as in `lm`. `na.action` is the missing value action. The default `na.fail` will return from the function with no computing done. Setting `na.action=na.omit` will omit any rows with missing data and then proceed.

The `group` argument, if used, requires a one-sided formula that evaluates to a list of a small number of levels. Dimension reduction is done for each combination of the grouping variables. For example, `group=~Sex` would fit separately for each `Sex`, and `group=~Sex:AgeClass` would fit separately for each combination of `Sex` and `AgeClass`. For this latter case, both `Sex` and `AgeClass` *must* be declared as factors.

The methods generally require linearly related predictors, and this can often be achieved by transformation of predictors (see Cook and Weisberg, 1999, and the function `bctran` in the R package `alr3`), or by weighting the observations. Weights can be computed using `dr.weights`, described in Section 9.1. The argument `weights` would usually either be `NULL` of the result of calling `dr.weights`.

The `method` argument is used to select the particular method of computing dimension reduction. The default is `"sir"`, but see Table 1 for other choices. Each of these methods is described in later sections.

The "..." allows for additional arguments to be passed to particular dimension reduction methods. For example, `sir` is a slicing method and so an argument `nslices` may be used. Another useful argument is `numdir`, which is set a maximum value for d. The default is `numdir=4`. These additional arguments are described in later sections where the methods are described.

The argument `slice.info` used in earlier versions of `dr` has been removed. A new argument `slice.function` has been added. If `slice.function=dr.slices.arc`,

6

then the algorithm used by Arc for computing slices is used. Otherwise, a newer algorithm given in the function `dr.slices` is used.

An additional new argument is `chi2approx = c("bx","wood")`. The default value of `"bx"` will use the Bentler and Xie (2000) approximation to a linear combination of Chi-square random variables when needed for computing significance levels. If `chi2approx="wood"`, then the Wood (1989) method, which was previously the default in `dr`, is used.

4 SIR

4.1 Basics

Sliced inverse regression or sir was proposed by Li (1991). We assume that data consists of n rows (y_i, x_i) for $i = 1, \ldots, n$. We write z_i for the i-th row of the centered and scaled version of x_i, from (2). We suppose that the response variable y has h distinct levels or values; if it does not have h levels but y is univariate, then the range of y is sliced into h non-overlapping *slices* so that the number of observations in each slice is approximately equal. If y is multivariate, h slices are formed in a somewhat more complex way; see Section 9.2. The only difference between univariate and multivariate sir is the way that slices are formed.

Suppose that $\bar{z}_j$, for $j = 1, \ldots, h$, is the mean of the z_i for i in the jth slice. The kernel matrix for sir is

$$\hat{M} = \frac{1}{n} \sum g_j \bar{z}_j \bar{z}_j' \tag{3}$$

where g_j is the sum of the weights in slice j, and the weights are always scaled so that $\sum w_i = n$. If the dimension of the central (mean) subspace is d, then the estimated central mean subspace is the span of the d eigenvectors corresponding to the d largest eigenvalues of $\hat{M}$, translated back to x-coordinates.

The default computational method using the `dr` function is sir. Here is an example of its use.

```
> summary(s0 <- dr(LBM~log(SSF)+log(Wt)+log(Hg)+log(Ht)+log(WCC)+log(RCC)+
+    log(Hc)+log(Ferr),data=ais,slice.function=dr.slices.arc,nslices=8,
+    chi2approx="wood",numdir=4,method="sir"))

Call:
dr(formula = LBM ~ log(SSF) + log(Wt) + log(Hg) + log(Ht) + log(WCC) +
    log(RCC) + log(Hc) + log(Ferr), data = ais, slice.function = dr.slices.arc,
    nslices = 8, chi2approx = "wood", numdir = 4, method = "sir")

Method:
sir with 8 slices, n = 202.

Slice Sizes:
```

```
26 26 25 25 25 27 30 18


Estimated Basis Vectors for Central Subspace:
                Dir1       Dir2      Dir3       Dir4
log(SSF)    0.158016 -0.075965  0.15503 -0.042384
log(Wt)    -0.970701 -0.022829 -0.24362  0.258583
log(Hg)    -0.139764  0.346539  0.54270 -0.008597
log(Ht)    -0.087587 -0.331604  0.30867 -0.630151
log(WCC)    0.006682 -0.014914 -0.00581 -0.024479
log(RCC)   -0.010892  0.502020  0.71198  0.343766
log(Hc)     0.073437 -0.715120 -0.07453 -0.643773
log(Ferr) -0.003117  0.003869 -0.11969 -0.030918


                Dir1    Dir2    Dir3     Dir4
Eigenvalues  0.9380  0.2046  0.0929  0.06665
R^2(OLS|dr)  0.9987  0.9988  0.9988  0.99898


Large-sample Marginal Dimension Tests:
                Stat df    p.value
0D vs >= 1D 269.50 56 0.0000000
1D vs >= 2D  80.02 42 0.0003665
2D vs >= 3D  38.69 30 0.1327694
3D vs >= 4D  19.93 20 0.4624789
```

This output is not identical to the output for sir obtained from previous versions of `dr` because the defaults for forming slices have changed, and the labels of sections of the output have changes as well. We have set the arguments `slice.function=dr.slices.arc`, `nslices=8` and `chi2approx="wood"` to get results that agree with Table 5 in Cook (2004), apart from multiplication of some basis vectors by -1.

The `summary` output is similar for all `dr` methods. Basic information, like the fitting method, the formula, and the slice information, is printed. The vectors collected into a matrix $\hat{B}$ in the section called `Estimated Basis Vectors for Central Subspace` are computed by taking the eigenvectors in the z-scale, backtransforming to the original x-scale as previously discussed, and then normalizing so each column has length one. These vectors are an estimated basis for $\mathcal{S}(B)$, and they are orthogonal relative to the inner product determined by $\widehat{\text{Var}}(X)$, so $\hat{B}'\widehat{\text{Var}}(X)\hat{B}$ is a diagonal matrix.

Next, the corresponding eigenvalues are given, ordered largest to smallest. The line marked `R^2*(OLS|dr)` is the square of the correlation between the linear combination obtained by the ols regression of the response on the predictors and the subspace spanned by the first direction, the first two directions, and so on. In this example, the correlation with `Dir1` is almost 1, so this direction is almost the same as ols. Under the assumption of linearly related predictors, ols estimates a direction in the central subspace, so the large correlation is to be generally expected.

For sir, the marginal dimension tests have asymptotic χ^2 distributions, as suggested by Li (1991), who assumed that the marginal distribution of X was normal. Bura and Cook (2001) show that the same asymptotic distribution holds under the linearity and constant covariance conditions. The tests are customarily viewed sequentially, first testing $d = 0$ versus $d > 0$, then $d = 1$ versus $d > 1$, and continue until we get a nonsignificant result. For the example, we would tentatively conclude $d = 2$ from these tests.

4.2 Coordinate hypothesis tests

4.2.1 Marginal coordinate tests

These tests were proposed by Cook (2004). Suppose that $\mathcal{S}(B)$ represents the central subspace with basis given by the columns of B. Coordinate tests allow testing hypotheses that the space $\mathcal{S}(B)$ is orthogonal to one (or more linear combinations) of the predictors; were this so, we could achieve dimension reduction simply by dropping these predictors.

Define a *hypothesis matrix* H, and under the null hypothesis we have that all vectors in $\mathcal{S}(B)$ can be represented as linear combinations of the columns of H, or if P_H is the projection on H, the null and alternative hypotheses are $(I - P_H)B = 0$ versus $(I - P_H)B \neq 0$. The method `dr.coordinate.test` will compute the appropriate test. Cook (2004) proposed two approaches to testing this hypothesis, the *marginal* approach in which we assume nothing about the dimension d, and the *conditional* approach in which we assume that d is known.

To do a test the user must specify H, but `dr` includes automatic tools to make this easier for the most common coordinate hypotheses. For example,

```
> dr.coordinate.test(s0,hypothesis=~.-log(RCC))

      Statistic   P.value
Test  9.843353  0.1063271
```

The null hypothesis is specified by the formula given in the argument `hypothesis`. In this case, the null hypothesis is that $\mathcal{S}(B)$ can be expressed as a linear combination of all the predictors (determined by the "~.") except for $\log(RCC)$. The function recognizes that `hypothesis` is a formula and called the function `coord.hyp.basis` to generate the H matrix. Alternatively, the user can provide a matrix with p rows and p^* columns, where p is the number of predictors, and $p^* < p$ is the dimension of the hypothesis space. See the documentation for `coord.hyp.basis` for details. Cook provided several ways of estimating the p-value, depending on the assumptions made. We present the most general approximation that requires the fewest assumptions.

4.2.2 Conditional coordinate tests

These tests are of the same hypothesis as in Section 4.2.1, except we condition on a known value d for the overall dimension. The same function is used for the computation, except that the argument `d` is set to the hypothesized dimension.

```
> dr.coordinate.test(s0,hypothesis=~.-log(RCC),d=2)

       Statistic    P.value
Test    4.79865 0.01410477
```

We see that the outcome of the test can be quite different when we condition on the dimension. Conditional coordinate tests are available only for sir, ire, and ire with categorical predictors.

4.3   Backward elimination

The function `dr.coordinate.test` is used in the standard R function `drop1` to examine dropping each predictor in turn.

```
> m0 <- drop1(s0,update=TRUE)

 LBM ~ log(SSF) + log(Wt) + log(Hg) + log(Ht) + log(WCC) +
     log(RCC) + log(Hc) + log(Ferr)
             Statistic      P.value
- log(Hg)     2.527821 8.517488e-01
- log(WCC)    2.865608 8.080390e-01
- log(Ht)     6.580257 3.227738e-01
- log(Hc)     6.920327 2.902826e-01
- log(RCC)    9.843353 1.063271e-01
- log(Ferr) 12.670269 3.604327e-02
- log(SSF)  27.961216 4.802552e-05
- log(Wt)   40.410284 1.594103e-07
```

The result of using this function is another `dr` object obtained by dropping the predictor with the largest p-value. This is the basis for stepwise fitting. If `update=FALSE`, then refitting `dr` is skipped.

The function `dr.step(object,scope,d,stop)` calls `drop1` repeatedly, each time dropping the variable with the largest p-value according to the marginal coordinate test, or the conditional coordinate test (for ire or for sir or save without categorical predictors) if a non-null value of `d` is specified. The documentation for these functions gives more details[2]. The argument `scope` is a one-sided formula listing predictors that are never deleted. The argument `d` is the dimension of the central subspace for ire with or without categorical predictors, or for sir without categorical variables. The algorithm continues until the number of variables remaining is equal to the maximum dimension set by the argument `numdir` used in the call the `dr`, which is 4 by default, or until the p-value for the next variable to be removed is less than `stop`. The default for `stop` is 0. As an example, the output below will delete predictors one at a time as long as the largest p-value exceeds 0.20. The variable `log(Wt)` will never be deleted.

---

[2]The standard function `step` in R is not a generic function and so it can't be used with `dr`.

```
> s1a <- dr.step(s0,scope=~log(Wt),stop=0.20)

 LBM ~ log(SSF) + log(Wt) + log(Hg) + log(Ht) + log(WCC) +
     log(RCC) + log(Hc) + log(Ferr)
            Statistic       P.value
- log(Hg)    2.527821 8.517488e-01
- log(WCC)   2.865608 8.080390e-01
- log(Ht)    6.580257 3.227738e-01
- log(Hc)    6.920327 2.902826e-01
- log(RCC)   9.843353 1.063271e-01
- log(Ferr) 12.670269 3.604327e-02
- log(SSF)  27.961216 4.802552e-05


 LBM ~ log(SSF) + log(Wt) + log(Ht) + log(WCC) + log(RCC) +
     log(Hc) + log(Ferr)
            Statistic        P.value
- log(WCC)   2.936254 7.994406e-01
- log(Ht)    7.574226 2.362964e-01
- log(Hc)    8.629906 1.652647e-01
- log(RCC)  10.624895 8.020715e-02
- log(Ferr) 12.623898 3.709661e-02
- log(SSF)  30.531085 1.530668e-05


 LBM ~ log(SSF) + log(Wt) + log(Ht) + log(RCC) + log(Hc) +
     log(Ferr)
            Statistic       P.value
- log(Ht)    7.571690 1.504490e-01
- log(Hc)    9.011992 8.552446e-02
- log(RCC)  10.533828 4.586187e-02
- log(Ferr) 12.672829 1.847415e-02
- log(SSF)  33.662369 1.185615e-06

Stopping Criterion Met
```

4.4   Categorical predictors and partial SIR

Suppose we have a categorical predictor or grouping variable G with g levels. Chiaromonte, Cook and Li (2002) described how such a categorial predictor could be included in a dimension reduction regression problem. The basic idea is to divide the problem into g regression problems, and define the central subspace to the the union of the subspaces for the regression problems $(y|x, G = 1), (y|x, G = 2), \ldots, (y|x, G = g)$. Chiaromonte *et al.* called this *partial sir*.

   In the context of sir, we can estimate a matrix $\hat{M}_g$ of the form (3) separately for of the g levels of G, combine them, and then estimate the central subspace from the eigenvectors corresponding to larger eigenvalues of the com-

11

bined $\hat{M}$. Using the ideas in Cook (2004), Shao, Cook and Weisberg (2009) provide marginal and conditional coordinate hypothesis tests for partial sir.

Partial sir is fit in dr using by adding an argument group in the call. The value of group should be a one-sided formula. For example, if Sex was a factor with two levels and AgeClass a factor with three levels, then group=~Sex would condition on Sex, and ~Sex:AgeClass would condition on all possible non-empty combinations of Sex and AgeClass. An additional argument pool is also available on the call to dr. If pool=TRUE, then a pooled estimate of the variance of X is used in each of the g groups to compute the standardized predictors. The default is pool=FALSE, in which the variance is estimated separately in each group. Chiaromonte *et al.* used the former choice to get the marginal dimension tests, while Shao *et al.* uses the latter choice to get the coordinate hypothesis tests.

```
> summary(s1 <- update(s0, group=~Sex))

Call:
dr(formula = LBM ~ log(SSF) + log(Wt) + log(Hg) + log(Ht) + log(WCC) +
    log(RCC) + log(Hc) + log(Ferr), data = ais, group = ~Sex,
    slice.function = dr.slices.arc, nslices = 8, chi2approx = "wood",
    numdir = 4, method = "sir")


Method:
psir with 8 8 slices, n = 202.


Slice Sizes:
13 13 13 13 12 12 12 12 13 17 14 16 13 14 12 3


Estimated Basis Vectors for Central Subspace:
                Dir1      Dir2      Dir3      Dir4
log(SSF)     0.117498   0.04262  -0.09604   0.093277
log(Wt)     -0.954097  -0.16172   0.09756  -0.095106
log(Hg)     -0.107177  -0.52877  -0.90198   0.018578
log(Ht)     -0.075489   0.51072   0.31713  -0.050511
log(WCC)     0.036513   0.04380   0.12594  -0.053962
log(RCC)    -0.126238  -0.08432  -0.04154  -0.692498
log(Hc)      0.203441   0.64997  -0.18122   0.704890
log(Ferr)    0.007086  -0.01077   0.12924   0.004844


                Dir1    Dir2    Dir3    Dir4
Eigenvalues   1.8335  0.4492  0.3172  0.2030
R^2(OLS|dr)   0.9904  0.9922  0.9930  0.9952


Large-sample Marginal Dimension Tests:
              Stat   df    p.value
0D vs >= 1D  621.8  112   0.000e+00
1D vs >= 2D  251.4   91   1.110e-16
```

```
2D vs >= 3D 160.7  72 1.020e-08
3D vs >= 4D  96.6  55 4.509e-04
```

4.5   Accessor functions

Several accessor functions can be applied to `dr` objects[3] to get additional information. These functions apply for sir, save, phd, or ire.

```
dr.x(object)
dr.y(object)
dr.z(object)
dr.wts(object)
```

These functions return, respectively, the X matrix the response Y, the scaled Z matrix (but not for partial sir or partial save), and the weights.

```
dr.test(object,numdir)
```

Returns the marginal dimension tests for all methods for which these are available.

```
dr.coordinate.test(object,hypothesis,d)
```

Returns the coordinate test for the hypothesis given by the one-sided formula `hypothesis`. If d is set and the method is either sir, ire, or ire with categorical predictors, the test is conditional on the dimension of the central subspace; otherwise, the test is marginal.

```
drop1(object,d=NULL,update=FALSE)
```

Calls `dr.coordinate.test` repeatedly to fit models that drop each predictor in turn.

```
dr.basis(object,numdir=4)
dr.evalues(object)
```

This function returns the first `numdir` basis vectors for the estimated central subspace (orthogonalized using the QR decomposition if `orth=TRUE`). The function `dr.evalues` returns the eigenvalues for sir, save, and phd.

```
dr.directions(object, which=c(1,2,3,4), x=dr.x(object))
```

Let $\hat{B}$ be the columns of the estimated basis for the central subspace selected by the list `which`, and X_1 be the matrix specified by the argument `x`, centered to have mean zero in each column. Then this function returns $X_1\hat{B}$. If `x` has its default values, then $\hat{B}'X_1'\widehat{\text{Var}(X)}X_1\hat{B}$ is a diagonal matrix, so the columns of $X_1\hat{B}$ are orthogonal in the inner product determined by $\widehat{\text{Var}(X)}$.

These values are very useful in plotting.

---

[3] If you type a command like `m1 <- dr(arguments)`, then `m1` is an object.

```
plot(object,which=1:object$numdir,mark.by.y=FALSE,
     plot.method=pairs,...)
```

This is the plot method for `dr`, which extracts by default the first `numdir` direction vectors and the response, and plots them using the `plot.method`, which by default is `pairs`. Additional arguments are passed to the plotting method.

5   SAVE

The method `save`, Cook and Weisberg (1991), is more comprehensive than `sir` as it can estimate vectors that depend on both mean and variances changes between slices.

As with `sir`, we slice the range of y into h slices, but rather than compute the within-slice mean we compute within-slice covariance matrices. If C_i is the weighted within slice sample covariance matrix in slice i in z-scale, then the matrix $\hat{M}$ is given by

$$\hat{M} = \frac{1}{n} \sum g_j (I - C_j)^2$$

where g_j is the sum of the weights in slice j; if all weights are equal, then the g_j are just the number of observations in each slice. `save` looks at second moment information and may miss first-moment information, in particular it may miss linear trends.

```
> s2 <- update(s0,method="save")
> summary(s2)

Call:
dr(formula = LBM ~ log(SSF) + log(Wt) + log(Hg) + log(Ht) + log(WCC) +
    log(RCC) + log(Hc) + log(Ferr), data = ais, slice.function = dr.slices.arc,
    nslices = 8, chi2approx = "wood", numdir = 4, method = "save")

Method:
save with 8 slices, n = 202.

Slice Sizes:
26 26 25 25 25 27 30 18

Estimated Basis Vectors for Central Subspace:
               Dir1       Dir2      Dir3      Dir4
log(SSF)    0.150208   0.002503   0.01341   0.04697
log(Wt)    -0.974926   0.095672  -0.23264   0.01489
log(Hg)    -0.071376  -0.459535  -0.07929  -0.68459
log(Ht)     0.069427  -0.260067   0.79663  -0.35430
log(WCC)    0.022249   0.007472   0.03734  -0.03682
log(RCC)    0.055098  -0.318709  -0.31942   0.42383
log(Hc)    -0.116228   0.781261   0.44788   0.46787
```

```
log(Ferr)  0.001187 -0.005745  0.02777  0.05907


              Dir1    Dir2    Dir3    Dir4
Eigenvalues 0.9175 0.4866 0.4328 0.3691
R^2(OLS|dr) 0.9976 0.9978 0.9978 0.9982


Large-sample Marginal Dimension Tests:
              Stat df(Nor) p.value(Nor) p.value(Gen)
0D vs >= 1D 309.36    252      0.007916      0.05538
1D vs >= 2D 213.27    196      0.189074      0.39975
2D vs >= 3D 132.72    147      0.794519      0.82988
3D vs >= 4D  82.29    105      0.950530      0.94676
```

As demonstrated here, the `update` function can be used with `dr` objects. Shao, Cook and Weisberg (2007) have provided marginal dimension tests based on `save`. Two p-values are available: if we assume predictors are normally distributed, then the test is asymptotically Chi-squared with df given in the column `df(Nor)`, and p-value in the column `p.val(Nor)`. The column `p.val(Gen)` is based on the weaker assumption that the predictors are linearly related but not necessarily normal.

As with `sir`, marginal coordinate tests can be computed with the function `dr.coordinate.test`. The functions `drop1` and `dr.step`, described in Section 4.3 can be used as well.

```
> drop1(s1,update=FALSE)

 LBM ~ log(SSF) + log(Wt) + log(Hg) + log(Ht) + log(WCC) +
     log(RCC) + log(Hc) + log(Ferr)
                Test       P.value
- log(Ht)    14.17340 2.104712e-01
- log(RCC)   14.21030 2.084619e-01
- log(Hg)    15.69464 1.393149e-01
- log(Hc)    15.94384 1.297970e-01
- log(Ferr)  17.93088 7.174679e-02
- log(WCC)   21.40569 2.292914e-02
- log(SSF)   23.29185 1.179638e-02
- log(Wt)    71.81375 4.777401e-12
```

As with `sir` categorial predictors can be used with `save`; see Section 4.4.

```
> summary(s3 <- update(s2,group=~Sex))

Call:
dr(formula = LBM ~ log(SSF) + log(Wt) + log(Hg) + log(Ht) + log(WCC) +
    log(RCC) + log(Hc) + log(Ferr), data = ais, group = ~Sex,
    slice.function = dr.slices.arc, nslices = 8, chi2approx = "wood",
    numdir = 4, method = "save")
```

```
Method:
psave with 8 8 slices, n = 202.

Slice Sizes:
13 13 13 13 12 12 12 12 13 17 14 16 13 14 12 3

Estimated Basis Vectors for Central Subspace:
                Dir1      Dir2       Dir3      Dir4
log(SSF)   -0.0268754  0.14950 -0.002989  0.38424
log(Wt)     0.2106463 -0.95000 -0.041860 -0.27767
log(Hg)     0.0612360 -0.18296  0.526994 -0.01356
log(Ht)    -0.7695336 -0.01837  0.179239 -0.66559
log(WCC)   -0.0275814  0.01299  0.010282  0.02861
log(RCC)    0.4213156  0.19192  0.176896 -0.27583
log(Hc)    -0.4250953  0.06383 -0.810394  0.49684
log(Ferr)   0.0006939  0.01588 -0.015912 -0.09109


                Dir1   Dir2   Dir3   Dir4
Eigenvalues  0.9921 0.9251 0.8415 0.6330
R^2(OLS|dr)  0.1046 0.9915 0.9932 0.9939


Large-sample Marginal Dimension Tests:
              Stat  df p-values
0D vs >= 1D 529.7 504   0.2072
1D vs >= 2D 372.5 392   0.7528
2D vs >= 3D 274.3 294   0.7897
3D vs >= 4D 178.3 210   0.9451
```

6 Principal Hessian direction

Principal Hessian direction methods were originally proposed by Li (1992). These methods are much more complicated to use and to interpret responses than the other methods in the dr package. The phd methods are unchanged from earlier versions of dr.

The phd methods are not based on slicing, but rather on the sample Hessian matrix.

$$\hat{M} = \frac{1}{n} \sum_{i=1}^{n} w_i f_i z_i z_i' \qquad (4)$$

In this equation the z_i are from (2), the w_i are the weights. Different choices of the f_i correspond to variations of the phd method. The central subspace in z-scale is estimated by the eigenvectors corresponding to the larger eigenvalues of $\hat{M}$ in (4). Tests are based on sums of the smaller eigenvalues of $\hat{M}$.

6.1 Response based pHd

Seting $f_i = y_i$ gives response based pHd, obtained in `dr` using `method="phdy"`. This is the method proposed by Li, and is the most straightforward to use. However, Cook (1998, Chapter 12) shows that tests based on this choice for f proposed by Li are not correct, and no tests for dimension are available using this method.

6.2 Residual based pHd

Cook (1998, Chapter 12) suggests setting f_i to be the residuals from the ols regression of y on x. This method is called `method="phdres"` in `dr`. Cook shows that tests of dimension based on this choice are correct, but the hypotheses concern a subspace other than the central subspace for $y|x$; see Cook (1998, Chapter 12) for details.

Output for `phd` is again similar to `sir`, except for the tests. Here is the output for the same setup as before, but for method `phdres`:

```
> summary(s2 <- update(s0,method="phdres"))

Call:
dr(formula = LBM ~ log(SSF) + log(Wt) + log(Hg) + log(Ht) + log(WCC) +
    log(RCC) + log(Hc) + log(Ferr), data = ais, slice.function = dr.slices.arc,
    nslices = 8, chi2approx = "wood", numdir = 4, method = "phdres")

Method:
phdres, n = 202.

Estimated Basis Vectors for Central Subspace:
                Dir1      Dir2       Dir3       Dir4
log(SSF)   -0.03675  -0.23340   0.001928   0.006563
log(Wt)     0.59536   0.03252  -0.238599   0.025140
log(Hg)    -0.36061  -0.47699  -0.014747  -0.596972
log(Ht)     0.21613  -0.08133   0.959780  -0.038954
log(WCC)    0.02948  -0.07203   0.065847  -0.047897
log(RCC)   -0.29816  -0.13669  -0.123638  -0.166642
log(Hc)     0.61429   0.82846   0.044891   0.781899
log(Ferr)  -0.01761   0.01068  -0.005824  -0.001390


                Dir1     Dir2    Dir3     Dir4
Eigenvalues   2.8583  -1.4478  0.9612  -0.5621
R^2(OLS|dr)   0.8774   0.9444  0.9643   0.9891


Large-sample Marginal Dimension Tests:
               Stat df Normal theory Indep. test General theory
OD vs >= 1D 223.67 36      0.000e+00   6.386e-13      0.002078
1D vs >= 2D  69.64 28      2.091e-05          NA      0.010936
```

```
2D vs >= 3D   30.12 21      8.970e-02           NA         0.211565
3D vs >= 4D   12.70 15      6.257e-01           NA         0.253438
```

The eigenvectors and eigenvalues are as for sir and save. The test statistics
are based on the eigenvalues. The column of tests called "Normal theory" were
proposed by Li (1992) and require that the predictors are normally distributed.
These statistics are asymptotically distributed as Chi-square, with the degrees
of freedom shown.

When the method is phdres additional tests are provided. Since this method
is based on residuals, it gives tests concerning the central subspace for the
regression of the residuals on X rather than the response on X. The subspace
for this residual regression may be, but need not be, smaller than the subspace
for the original regression. For example, the column marked "Indep. test" is
essentially a test of $d = 0$ versus $d > 0$ described by Cook (1998) for the residual
regression. Should the significance level for this test be large, we might conclude
that the residual regression subspace is of dimension zero. From this we have
two possible conclusions: (1) the dimension of the response regression may be
1 if using the residuals removed a linear trend, or (2) the dimension may be 0
if the residuals did not remove a linear trend.

Similarly, if the significance level for the independence test is small, then we
can conclude that the dimension is at least 1. It could be one if the method
is picking up a nonlinear trend in the OLS direction, but it will be 2 if the
nonlinearity is in some other direction.

The independence test and the final column, also from Cook (1998), use the
same test statistic, but different distributions based on different assumptions.
Significance levels are obtained by comparing the statistic to the distribution of
a random linear combination of Chi-square statistics, each with one df. These
statistics do not require normality of the predictors.

7   Inverse regression estimation

This discussion of inverse regression estimation is more or less self contained,
and uses somewhat different notation from earlier sections.

7.1   Estimation

All methods fit by dr are *location and scale invariant*, and so for the purpose
of testing, we can use any linear transformation of the predictors we like. In
particular if X is the $n \times p$ matrix of predictors, and 1_n is an n-vector of ones,
we can compute the QR factorization,

$$(1_n, X) = (Q_0, Q) \begin{pmatrix} R_0 & R_{01} \\ 0 & R \end{pmatrix}$$

and replace X by $\sqrt{n}Q$ in all computations concerning objective functions and
tests. For quantities that are coordinate dependent, like coordinate tests, and

subspace bases, back-transformation to X-scale is done as indicated below. All formulas given in this article are in the simpler QR scale.

The data we use consists of the matrix $\sqrt{n}Q$ with rows q_i' and the response vector has elements Y_i; we have not implemented a multivariate response version. We assume that Y is discrete with h distinct values $1, \ldots, h$; if Y is continuous, we discretize it into these h distinct values by slicing and number these from 1 to h; see Cook and Ni (2005, Section 2.1). `dr` provides two functions for slicing: `dr.slices.arc` uses the same algorithm that is used in Arc, while the default `dr.slices` uses an algorithm that is better at handling ties. The former should be used for comparison to Arc, but the latter is recommended in general.

If $\delta(Y_i = y)$ is the indicator function equal to one if $Y_i = y$ and zero otherwise, then define

$$n_y \;=\; \sum_i \delta(Y_i = y) \tag{5}$$

$$f_y \;=\; n_y/n \tag{6}$$

$$\hat{\xi}_y \;=\; (1/n_y) \sum_i \delta(Y_i = y) q_i \tag{7}$$

$$\hat{\xi} \;=\; (\hat{\xi}_1, \ldots, \hat{\xi}_h) \tag{8}$$

The vector $\hat{\xi}_y$ is the mean of the q_i for which $\delta(Y_i = y)$, the "slice mean", and $\hat{\xi}$ is the $p \times h$ matrix of slice means.

Under assumptions discussed by Cook and Ni (2005, Section 2.1), the columns of $\hat{\xi}$ estimate vectors in the central subspace. Sliced inverse regression, for example, estimates the central subspace by the span the the eigenvectors of $\hat{\xi}\hat{\xi}'$ corresponding to the d largest eigenvalues. In contrast, inverse regression estimation will estimate the span of the central subspace using the columns of the $p \times d$ matrix B determined by minimizing the following objective function (see Cook and Ni, 2005, equation 2):

$$F_d(B, C) = \left(\text{vec}(\hat{\xi}\,\text{diag}(f_y)A_n) - \text{vec}(BC) \right)' V_n \left(\text{vec}(\hat{\xi}\,\text{diag}(f_y)A_n) - \text{vec}(BC) \right) \tag{9}$$

The matrix $\text{diag}(f_y)A_n$ is required to convert vectors in $\hat{\xi}$ to a fixed coordinate system; in `dr`, following Cook and Ni (2005), A_n is obtained from the R command

```
An <- qr.Q(qr(contr.helmert(h)))
```

where h is the number of slices. Thus A_n is an $h \times (h-1)$ matrix whose columns are orthogonal to each other and to 1_h. Alternatively, A_n could be defined as any $h \times (h-1)$ orthogonal matrix with orthogonal columns that spans the same space; this would simply change coordinates and C, but leave B unchanged. The matrix V_n is a positive definite symmetric inner product matrix. The optimal choice for V_n is the inverse of the asymptotic variance of $\text{vec}(\hat{\xi}\,\text{diag}(f_y)A_n)$, and an estimate of this is used in the `ire` method in `dr`.

Minimization of (9) is done over both B and C. The minimizing value of C is of no interest because it depends on the coordinates determined by A_n.

The column space of the $p \times d$ matrix B provides the estimate of the central subspace assuming it is of dimension d.

Computations use an alternating algorithm, which is likely to converge at only a linear rate, so good starting values can be very important. Assuming V_n is fixed, if B is fixed, then C can be estimated from ordinary least squares fit of $V_n^{1/2}\text{vec}(\hat{\xi}A_n)$ on $V_n^{1/2}\text{vec}(I_{h-1} \otimes B)$. Given the new value of C, Cook and Ni (2005, Section 3.3) describe a method of updating B, column by column. For starting values in this algorithm, dr sets B equal to the estimated d dimensional space obtained using sir.

The computational method will be complete if we describe the computation of V_n. Cook and Ni (2005, Section 3.3), write

$$V_n^{-1} = (A_n' \otimes I)\Gamma(A_n \otimes I)$$

A consistent sample version of $\Gamma = \text{Var}(\text{vec}(q\epsilon'))$ is required, where q' is a typical row of Q, and ϵ has typical elements $\epsilon_y = \delta(Y = y) - \text{E}(\delta(Y = y)) - q'\text{E}(q\delta(Y = y))$, the population residuals from the regression of $\delta(Y = y)$ on q. The sample version of these residuals are given by

$$\hat{\epsilon}_{yi} = \delta(Y_i = y) - f_y - f_y q_i'\hat{\psi}_y$$

where $\hat{\psi}_y$ depends on the value of the dimension d and on the estimated central subspace. We use the following procedure for setting $\hat{\psi}_y$:

1. For computations under the assumption $d = 0$, set $\hat{\psi}_y = 0$.

2. For computations under the assumption $d > 0$, initially set $\hat{\psi}_y = \hat{\xi}_y$, and obtain an estimator B.

3. If $d > 0$ and we set the argument steps to equal a positive integer, first compute step 2, and then repeat with $\hat{\psi} = P_B\hat{\xi}$. If steps is greater than one, repeat the number of times indicated by the argument; one step seems to be adequate.

At completion of the computations the columns of B provide a basis for the central subspace, assuming that it is of dimension d, in the coordinates determined by the QR factorization. We could replace B by any matrix TB where T is any $p \times p$ *restriction* matrix. Indeed, the function dr.iteration.ire minimizes the function

$$F_d(B,C,T) = \Big(\text{vec}(\hat{\xi}\text{diag}(f_y)A_n) - \text{vec}(TBC)\Big)' V_n \Big(\text{vec}(\hat{\xi}\text{diag}(f_y)A_n) - \text{vec}(TBC)\Big) \tag{10}$$

rather than (9), maximizing over B and C with T fixed.

Using this extension, dr uses the algorithm outlined in Cook and Ni (2005, Section 3.3) to replace B by another basis matrix $\hat{B}$ such that the first column of $\hat{B}$ minimizes (9) for $d = 1$, with the minimization over all vectors in the span of B, the second column of $\hat{B}$ is orthogonal to the first column such that $(\hat{b}_1, \hat{b}_2)$ minimizes (9) for $d = 2$ with $\hat{b}_1$ fixed and $\hat{b}_2$ in the span of B, and so on.

To duplicate the results for the Australian Athletes data in Cook and Ni (2005), the following can be used:

```
> (m1 <- dr(LBM~log(Ht)+log(Wt)+log(SSF)+log(RCC)+log(WCC)+log(Ferr)+
+              log(Hc)+log(Hg),data=ais,method="ire",nslices=8,numdir=4,
+              slice.function=dr.slices.arc,itmax=200,steps=1,eps=1.e-6))

 dr(formula = LBM ~ log(Ht) + log(Wt) + log(SSF) + log(RCC) +
     log(WCC) + log(Ferr) + log(Hc) + log(Hg), data = ais,
     method = "ire", nslices = 8, numdir = 4, slice.function = dr.slices.arc,
     itmax = 200, steps = 1, eps = 1e-06)
Large-sample Marginal Dimension Tests:
                   Test df       p.value iter
0D vs > 0D 1920.043970 56 0.000000e+00     0
1D vs > 1D  104.381091 42 3.176227e-07     4
2D vs > 2D   48.372868 30 1.819157e-02     5
3D vs > 3D   26.149188 20 1.609410e-01    13
4D vs > 4D    6.727698 12 8.750714e-01     6
```

The call to the `dr` function is the same as for the other dimension reduction methods, except that the argument `method="ire"`. We have specified `nslices=8` to match Cook and Ni, rather than use the default of $\max(8, p + 3)$, and have used `dr.slices.arc` to get slices, again to match Cook and Ni. The remaining three arguments are default values, `itmax` and `eps` controlling the iteration, and `steps` controlling the number of reëstimates of the covariance matrix.

The default printed output for ire is a little different from that for `save` and `sir`. Printed only are the large-sample marginal dimension tests. Cook and Ni (2005, Theorem 2) show that the appropriate statistic for these hypothesis is $nF_d(\hat{B}, \hat{C}) \sim \chi^2(p - d)(h - d - 1)$. These tests are produced whenever ire is fit.

7.2 Solution in original coordinates

In sir and save, the estimated d-dimensional basis for $\mathcal{S}(B)$ is given by the d eigenvectors corresponding to the d largest eigenvalues. In ire, we will get a different basis for each value of d, and so the subspaces need not be nested. As with sir and save, the basis is found by multiplying $R^{-1}\hat{B}$, where R is the Cholesky factor from the QR decomposition.

Two see the two dimensional solution, use the function `dr.basis`,

```
> dr.basis(m1,numdir=2)

                Dir1         Dir2
log(Ht)     0.103879230  0.529197702
log(Wt)     0.949913819 -0.022318180
log(SSF)   -0.148658473  0.098599612
log(RCC)    0.012738241 -0.470853789
log(WCC)   -0.020345785 -0.001408582
```

```
log(Ferr)   0.002164593 -0.017665131
log(Hc)    -0.144988060  0.650278923
log(Hg)     0.207776474 -0.254656236
```

For this problem, the first two vectors of the three-dimensional solution are slightly different from the two-dimensional solution:

```
> dr.basis(m1,3)

                  Dir1           Dir2          Dir3
log(Ht)     0.051162573  0.4823791902  0.510960683
log(Wt)     0.961086645 -0.0669274709 -0.312641769
log(SSF)   -0.155948990  0.0852881164  0.083787011
log(RCC)    0.050823719 -0.4333097120 -0.043731712
log(WCC)   -0.017355351 -0.0060366333 -0.003542429
log(Ferr)   0.002067821  0.0003196894  0.005114915
log(Hc)    -0.163654488  0.7123173202  0.794606795
log(Hg)     0.140368940 -0.2456972800  0.028335640
```

7.3 Coordinate hypothesis

Suppose that H is a *hypothesis matrix*; typically, H could restrict the subspace of interest to be orthogonal to one or more of the predictors, but other choices are possible as well. The *marginal predictor hypothesis* has null hypothesis $(I - P_H)S = 0$ versus $(I - P_H)S \neq 0$. Equation (12) of Cook and Ni (2005, eq. 12) provides the appropriate test statistic. (Cook and Ni have a different definition of H.) As with sir and save, both marginal and conditional tests are available

For example, a marginal predictor hypothesis test for dropping a variable and a conditional test assuming $d = 2$ are given by

```
> dr.coordinate.test(m1,~.-log(Hg))

      Test df   p.value
1 3.671465   7 0.8167445

> dr.coordinate.test(m1,~.-log(Hg),d=2)

      Test df      p.value
1 24.56324   2 4.636185e-06
```

The general syntax of the second argument is a formula that corresponds to H: Here, H is obtained from the original specification, ~. by removing log(Hg). The large p-value suggests that log(Hg) is likely to be unimportant if we assume nothing about d, but if $d = 2$ dropping log(Hg) is not supportable.

As with sir and ire, drop1 can be used to consider dropping each predictor in tern, and dr.step can be used for stepwise selection of predictors.

```
> drop1(m1,update=FALSE)
```

```
 LBM ~ log(Ht) + log(Wt) + log(SSF) + log(RCC) + log(WCC) +
      log(Ferr) + log(Hc) + log(Hg)
                    Test df      p.value
- log(WCC)    2.749903  7 9.071346e-01
- log(Hg)     3.671465  7 8.167445e-01
- log(Ht)     7.264478  7 4.018718e-01
- log(Hc)    11.420293  7 1.213078e-01
- log(RCC)   13.728866  7 5.621930e-02
- log(Ferr)  16.385955  7 2.181489e-02
- log(SSF)  242.003908  7 1.392883e-48
- log(Wt)   438.008629  7 1.667836e-90
```

7.4   Joint predictor test

One can simultaneously test $d = m$ and $(I - P_H S) = 0$ by minimizing (10) with the restriction matrix $T = P_H$. This test is not available for sir or save, and is used in the computation of the conditional coordinate test. See the documentation for dr.joint.test.

7.5   Accessor functions

See Section 4.5.

8   Partial inverse regression with a categorical predictor

Wen and Cook (2007) have extended the ire method to allow for inclusion of a categorical predictor.

```
> m2 <- dr(LBM~log(Ht)+log(Wt)+log(SSF)+log(RCC)+log(WCC)+log(Ferr)+
+            log(Hc)+log(Hg),group=~Sex,data=ais,method="ire",nslices=8,
+            numdir=4,slice.function=dr.slices.arc,itmax=200,steps=1,
+            eps=1.e-6)
```

In this case, we have conditioned on the two levels of Sex. The output is identical to that for ire, except for degrees of freedom:

```
> m2

 dr(formula = LBM ~ log(Ht) + log(Wt) + log(SSF) + log(RCC) +
     log(WCC) + log(Ferr) + log(Hc) + log(Hg), data = ais,
     group = ~Sex, method = "ire", nslices = 8, numdir = 4,
     slice.function = dr.slices.arc, itmax = 200, steps = 1,
     eps = 1e-06)
Large-sample Marginal Dimension Tests:
                  Test  df       p.value iter
```

```
0D vs > 0D 2023.12778 112 0.000000e+00    0
1D vs > 1D  223.93985  91 3.334637e-13    9
2D vs > 2D  130.55286  72 2.966408e-05   90
3D vs > 3D   75.49938  55 3.471124e-02   45
4D vs > 4D   44.25621  40 2.966903e-01  123
```

The argument `group` requires a one-sided formula, like `~Sex`, where the right-side variable can be a factor or a variate. If you have more than one grouping variable, for example `Sex` and `AgeClass`, then use `~ Sex:AgeClass`. For this form, all the variables on the right-side of the formula must be factors.

Here is the basic idea. We first replace the X matrix by the $\sqrt{n}Q$ factor from the QR decomposition. Suppose the conditioning or grouping variable G has g levels. Subdivide the data into g groups, where $Q^{(k)}$ is the part of the Q matrix for group k. $Q^{(k)}$ is not part of a QR decomposition, and so we replace each $Q^{(k)}$ by its QR decomposition. This essentially allows each group to have its own within-group covariance structure. Obtain slice means $\hat{\xi}^{(k)}$, for $k = 1, \ldots, g$. If h_k is the number of slices in the k-th group, the columns of the matrix $p \times \sum h_k$ matrix $(\hat{\xi}^{(1)}, \ldots, \hat{\xi}^{(g)})$ all estimate vectors in the central (mean) subspace. An estimate of B is obtained by minimizing the objective function

$$G_d(B, C, T) = \sum_{k=1}^{g} F_d^{(k)}(B, C, T) \tag{11}$$

where by $F_d^{(k)}(B, C, T)$ we mean (10) evaluated only for the k-th group, with $\hat{\xi} = \hat{\xi}^{(k)}, V_n = V_n^{(k)}$, and $\text{diag}(f_y) = \text{diag}(f_y^{(k)})$. Theory similar to that for ire permits test statistics similar to ire, and these are implemented in `dr`.

Output for partial ire parallels that for ire. All the functions that can be applied to ire objects can also be applied to pire objects.

9 Auxillary functions

9.1 Weights

Weights are generally used in dimension reduction methods to make the resulting weighted sample closer to a normal distribution than the original sample. Cook (1998, Section 8.4) discusses the method that is implemented here. When weights are present, they are used in centering the data and computing the covariance matrix, and they are used in computing the objective matrix M for phd. Weights may be provided by the user with the `weights` argument. If `weights=NULL`, the default, no weighting is used.

The function `dr.weights` is used to estimate weights using the algorithm described by Cook (1998, Sec. 8.4). There are several other arguments that control how the weights are computed, as described below, and on the help page for the function `dr.weights`. The algorithm works as follows:

1. For an $n \times p$ data matrix X, find estimates m and S of the mean and covariance matrix. For this purpose, in R the function `cov.rob` in the `MASS` package is used. See the documentation for `cov.rob` for a description of these additional parameters. All the defaults are sensible.

2. Compute the matrix $Z = (X - 1m')S^{-1/2}$. If the data were normally distributed N(m, S), the rows of Z would be like a sample from N($0, I$).

3. Obtain a random vector b from the N($0, \sigma^2 I$) distribution. The parameter `sigma=1` is a tuning parameter that can be set in the call to `dr`, and values near 1 or slightly smaller seem appropriate. Find the row of Z that is closest to b (the code uses Euclidean distance), and increase a counter for that row by 1.

4. The argument `nsamples` determines the number of times this last step is repeated; the default is `nsamples=10*dim(x)[1]` where X is the $n \times p$ data matrix; this number may be too small.

5. Return a vector of weights given by the value of the counter divided by `nsamples` and multiplied by n, so the sum of the weights will be n.

An example of the use of weights is:

```
> wts <- dr.weights(LBM~Ht+Wt+RCC+WCC,data=ais)
> i1 <- dr(LBM~Ht+Wt+RCC+WCC,weights=wts,method="phdres",data=ais)
```

9.2   Slices

`dr` provides two functions for producing slices, `dr.slices`, the recommended default, and `dr.slices.arc`, which is the older routine used before version 3.0.0. The newer version does a better job of handling ties. Both functions have the same calling sequence. For example,

```
> y1 <- c(1,1,1,2,3,4,5,6,7,8,8,8)
> dr.slices(y1,3)

$slice.indicator
 [1] 1 1 1 1 2 2 2 2 3 3 3 3

$nslices
[1] 3

$slice.sizes
[1] 4 4 4
```

produces a structure with three arguments, giving the indices of the observations in each slice, the number of slices and the slice sizes.

If the response has k columns, then the argument `nslices` should be a vector of k elements. If it is specified as a number rather than a vector, then that

number will give the total number of cells, approximately. For example, if $k = 2$ and `nslices=8`, the program will slice $\sqrt{8} \approx 3$ slices along each of the two response variables for a total of $3 \times 3 = 9$ two-dimensional slices.

```
> y2 <- c(1,2,3,4,1,2,3,4,1,2,3,4)
> dr.slices(cbind(y1,y2),5)

$slice.indicator
 [1] 1 2 2 2 3 4 4 4 5 6 6 6

$nslices
[1] 6

$slice.sizes
[1] 1 3 1 3 1 3
```

This produced 6 slices that are not all exactly the same size.

There are two optional arguments to the `dr` function that concern slicing. The argument `slice.function` has values either `dr.slices`, the default, or `dr.slices.arc`. The argument `nslices` to `dr` is passed to the slice function. The default for `nslices` is the minimum of 8 and $p + 3$.

9.3 Permutation tests

Cook (1998) and Yin in his unpublished 2000 Ph.D. dissertation discuss using permutation tests to get significance levels for the marginal dimension test. These are implemented in the function `dr.permutation.test`. Typical use of this function is

```
> dr.permutation.test(s0,npermute=99,numdir=4)

$summary
                 Stat p.value
0D vs >= 1D 269.500841    0.00
1D vs >= 2D  80.018879    0.00
2D vs >= 3D  38.692445    0.11
3D vs >= 4D  19.927408    0.33
4D vs >= 5D   6.464427    0.82

$npermute
[1] 99

attr(,"class")
[1] "dr.permutation.test"
```

The number of permutations defaults to 50 and the number of directions defaults to 4. Increasing either can increase the computation time required to obtain the solution. The permutation test results for the example are very similar to the

asymptotic results given earlier. The permutation tests are not implemented for ire.

10 Acknowledgements

11 References

Bentler, PM and Xie, J (2000). Corrections to test statistics in principal Hessian directions. *Statistics and Probability Letters*, 47, 381–389.

Chiaromonte, F., Cook, R. D. and Li, B. (2002). Sufficient dimension reduction in regressions with categorical predictors. *Annals of Statistics*, 30, 475–97.

Cook, RD (1998). *Regression Graphics*. New York: Wiley.

Cook, RD (2004). Testing predictor contributions in sufficient dimension reduction. *Annals of Statistics*, 32, 1062-1092.

Bura E and Cook, RD (2001). Extending sliced inverse regression: The weighted chi-squared test. *J. Amer. Statist. Assoc.* 96 996–1003.

Cook, RD, and Li, B (2002). Dimension reduction for conditional mean in regression. *Annals of Statistics*, 30, 455-474.

Li, K C. (1991). Sliced inverse regression for dimension reduction (with discussion), *Journal of the American Statistical Association*, 86, 316-342.

Li, K C. (1992). On principal Hessian directions for data visualization and dimension reduction: Another application of Stein's lemma. *Journal of the American Statistical Association*, 87, 1025–1034.

Cook, RD and Ni, L (2005). Sufficient dimension reduction via inverse regression: A minimum discrepancy approach. *Journal of the American Statistical Association*, 100, 410–428.

Shao, Y, Cook, RD and Weisberg, S (2007). Marginal Tests with Sliced Average Variance Estimation. *Biometrika*, 94, 285–296.

Shao, Yongwu, Cook, R. D. and Weisberg, S. (2009, to appear, available on line). Partial central subspace and sliced average variance estimation. *Journal of Statistical Planning and Inference*.

Weisberg, S (2002). Dimension Reduction Regression in R. *Journal of Statistical Software*, 2002, Volume 7, available from http://www.jstatsoft.org.

Wood ATA (1989) An F approximation to the distribution of a linear combination of chi-squared variables. *Communications in Statistics: Simulation and Computation*, 18, 1439-1456.

Wen, X and Cook, RD (2007). Optimal Sufficient Dimension Reduction in Regressions with Categorical Predictors, *Journal of Statistical Inference and Planning*, 137, 1961–79.