Example 1: SVD to find a generalized inverse of a non-full-rank matrix

```
a <- matrix(c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0,
         0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1), 9, 4)
a.svd <- svd(a)
a.svd$d
```

```
[1] 3.464102e+00 1.732051e+00 1.732051e+00 1.922963e-16
```

```
  ds <- diag(1/a.svd$d[1:3])
  u <- a.svd$u
  v <- a.svd$v
  us <- as.matrix(u[, 1:3])
  vs <- as.matrix(v[, 1:3])
  (a.ginv <- vs %*% ds %*% t(us))
```

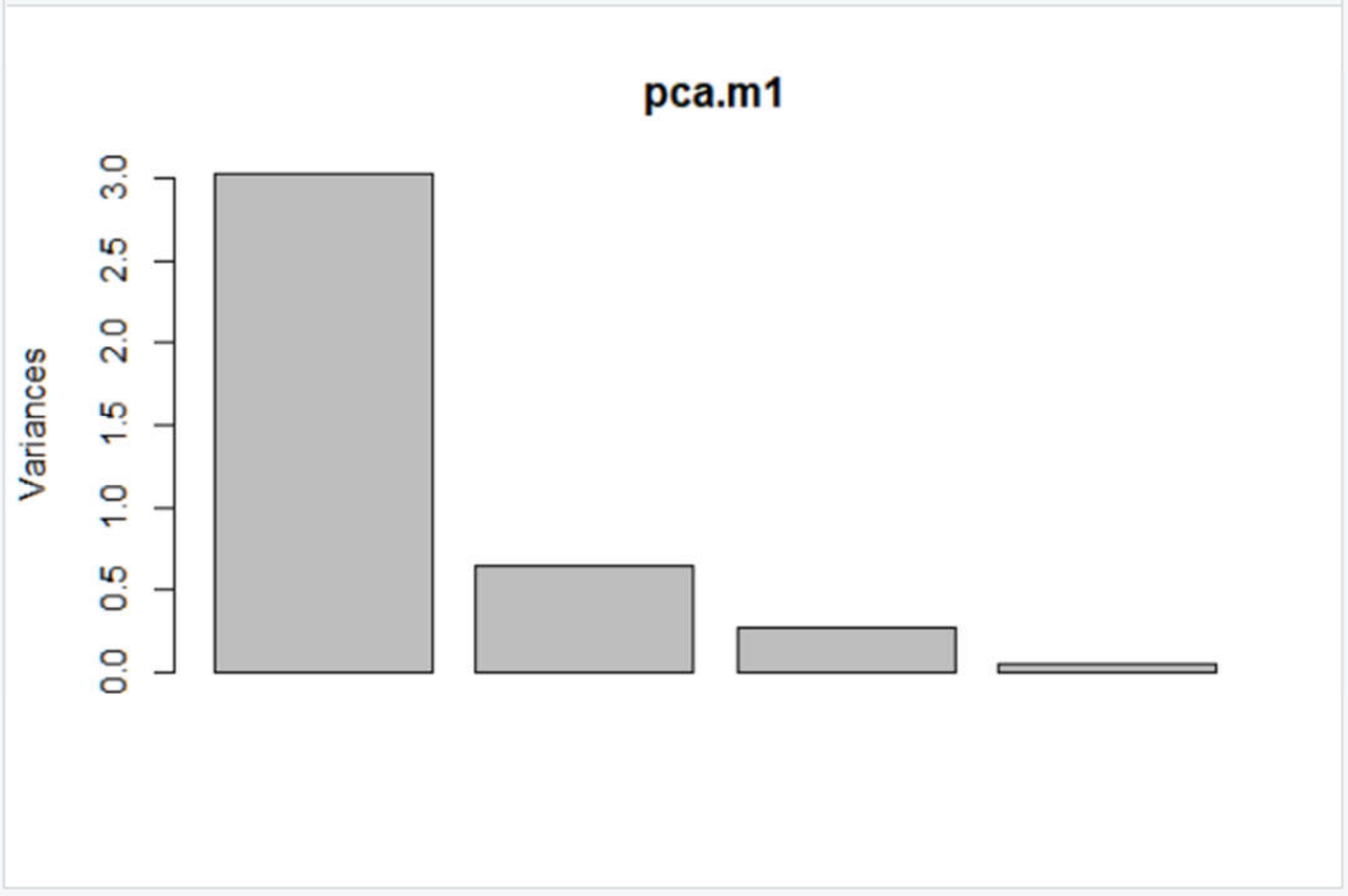
```
            [,1]        [,2]        [,3]        [,4]        [,5]        [,6]
[1,]  0.08333333  0.08333333  0.08333333  0.08333333  0.08333333  0.08333333
[2,]  0.25000000  0.25000000  0.25000000 -0.08333333 -0.08333333 -0.08333333
[3,] -0.08333333 -0.08333333 -0.08333333  0.25000000  0.25000000  0.25000000
[4,] -0.08333333 -0.08333333 -0.08333333 -0.08333333 -0.08333333 -0.08333333
            [,7]        [,8]        [,9]
[1,]  0.08333333  0.08333333  0.08333333
[2,] -0.08333333 -0.08333333 -0.08333333
[3,] -0.08333333 -0.08333333 -0.08333333
[4,]  0.25000000  0.25000000  0.25000000
```

```
  # using the function ginv defined in MASS
  ginv(a)
```

```
            [,1]        [,2]        [,3]        [,4]        [,5]        [,6]
[1,]  0.08333333  0.08333333  0.08333333  0.08333333  0.08333333  0.08333333
[2,]  0.25000000  0.25000000  0.25000000 -0.08333333 -0.08333333 -0.08333333
[3,] -0.08333333 -0.08333333 -0.08333333  0.25000000  0.25000000  0.25000000
[4,] -0.08333333 -0.08333333 -0.08333333 -0.08333333 -0.08333333 -0.08333333
            [,7]        [,8]        [,9]
[1,]  0.08333333  0.08333333  0.08333333
[2,] -0.08333333 -0.08333333 -0.08333333
[3,] -0.08333333 -0.08333333 -0.08333333
[4,]  0.25000000  0.25000000  0.25000000
```

Example 2: Principal components analysis using SVD

```
library(foreign)
auto <- read.dta("http://statistics.ats.ucla.edu/stat/data/auto.dta")
pca.m1 <- prcomp(~trunk + weight + length + headroom, data = auto,
              scale = TRUE)
screeplot(pca.m1)
```




```
# spectral decomposition: eigen values and eigen vectors
xvars <- with(auto, cbind(trunk, weight, length, headroom))
corr <- cor(xvars)
a <- eigen(corr)
(std <- sqrt(a$values))
```

```
[1] 1.7378931 0.8074981 0.5264150 0.2248592
```

```
(rotation <- a$vectors)
           [,1]       [,2]       [,3]         [,4]
[1,] -0.5067777 -0.2326998  0.8249462  0.092145980
[2,] -0.5220823  0.4535800 -0.2677106  0.670839942
[3,] -0.5361131  0.3903201 -0.1370497 -0.735833101
[4,] -0.4280061 -0.7666591 -0.4785521 -0.005704251
```

```
# svd approach
df <- nrow(xvars) - 1
zvars <- scale(xvars)
z.svd <- svd(zvars)
z.svd$d/sqrt(df)
```

```
[1] 1.7378931 0.8074981 0.5264150 0.2248592
```

```
z.svd$v
```

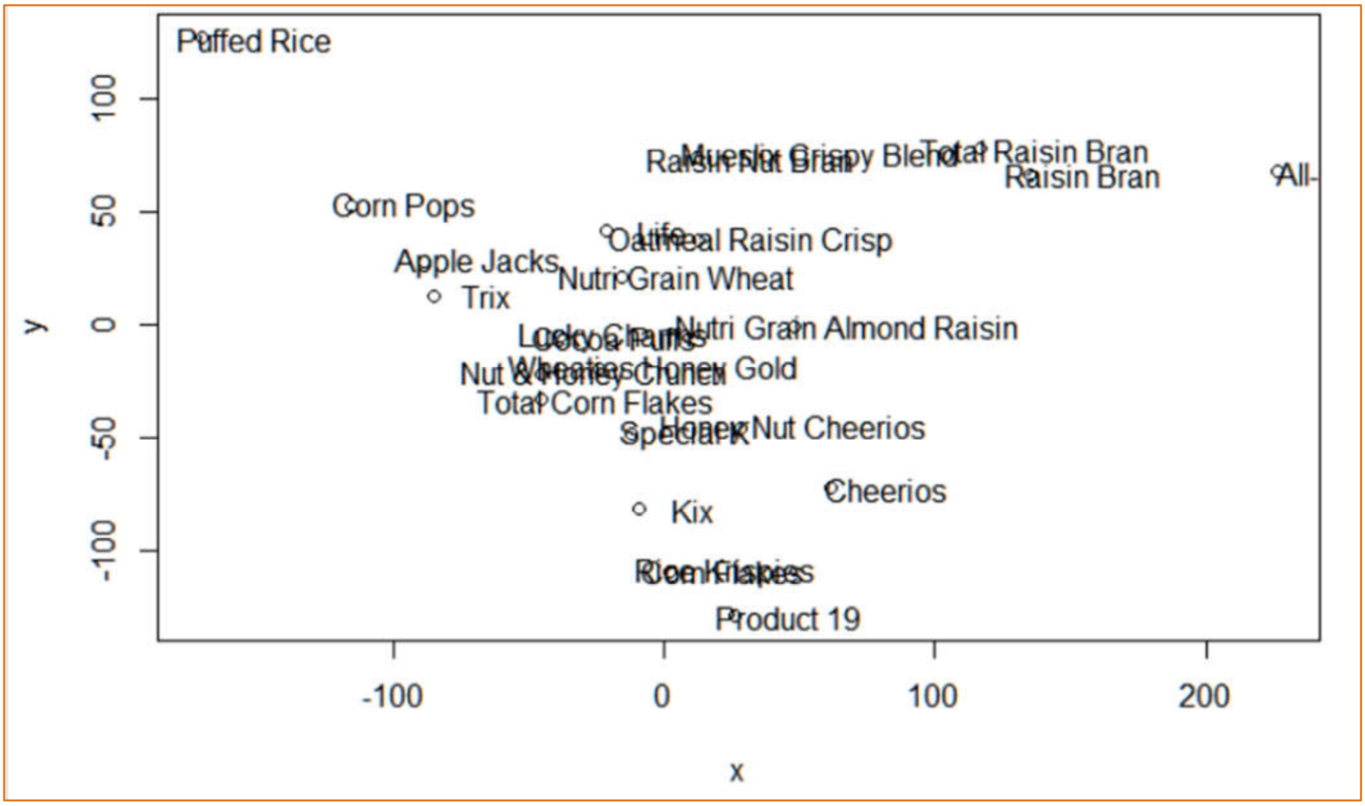
```
          [,1]       [,2]       [,3]         [,4]
[1,] 0.5067777 -0.2326998  0.8249462 -0.092145980
[2,] 0.5220823  0.4535800 -0.2677106 -0.670839942
[3,] 0.5361131  0.3903201 -0.1370497  0.735833101
[4,] 0.4280061 -0.7666591 -0.4785521  0.005704251
```

Example 3: Metric multi-dimensional scaling with SVD

```
cnut <- read.dta("http://statistics.ats.ucla.edu/stat/data/cerealnut.dta")
# centering the variables
mds.data <- as.matrix(sweep(cnut[, -1], 2, colMeans(cnut[, -1])))
dismat <- dist(mds.data)
mds.m1 <- cmdscale(dismat, k = 8, eig = TRUE)
mds.m1$eig
```

```
 [1]  1.584379e+05  1.087288e+05  1.056264e+04  3.826785e+02
 [5]  6.976171e+01  1.252082e+01  5.755998e+00  2.224324e+00
 [9]  4.513969e-12  4.508111e-12  4.121611e-12  3.188527e-12
[13]  3.150030e-12  2.297497e-12  2.091059e-12  1.246190e-12
[17]  1.131813e-12  8.794901e-13  2.967892e-13 -1.382636e-12
[21] -1.452732e-12 -1.574794e-12 -1.876268e-12 -5.916330e-12
[25] -2.520931e-11
```

```
mds.m1 <- cmdscale(dismat, k = 2, eig = TRUE)
x <- mds.m1$points[, 1]
y <- mds.m1$points[, 2]
plot(x, y)
```




```
# eigenvalues
xx <- svd(mds.data %*% t(mds.data))
xx$d
```

```
 [1] 1.584379e+05 1.087288e+05 1.056264e+04 3.826785e+02 6.976171e+01
 [6] 1.252082e+01 5.755998e+00 2.224324e+00 1.576321e-11 9.903742e-12
[11] 7.190968e-12 4.712199e-12 4.152571e-12 3.030837e-12 2.767589e-12
[16] 2.082324e-12 1.971417e-12 1.496531e-12 1.258080e-12 1.045736e-12
[21] 7.934340e-13 7.346559e-13 2.088189e-13 1.653877e-13 8.383459e-14
```

```
# coordinates
xxd <- xx$v %*% sqrt(diag(xx$d))
x1 <- xxd[, 1]
y1 <- xxd[, 2]
plot(x1, y1)
text(x1 + 20, y1, label = cnut$brand)
```