Stat 505 Matrix Operations With R

Jim Robison-Cox

- In R, a matrix is stored as one long vector with a dimension attribute. Think of column 1 stacked atop column 2 atop column 3, ...

- Build matrices with `matrix` or `array` or `cbind` or `rbind`

```
matrix(0, 3, 3)
array(0, c(3,3))
diag(3) * 0
rbind( rep(1,3), rep(2,3), rep(3,3))
cbind( rep(1,3), rep(2,3), rep(3,3))
```

- Matrix multiply works for matrices of conformable size. If one side of `%*%` is a vector without dimension, R tries to make it a $1 \times n$ or $n \times 1$ conformable vector.

```
c(1,2,3) %*% diag(3)
x <- c(3,5,7)
X <- cbind(1, x, (x-5)^2)
X %*% c(1, 2, .45)
```

  If you just use `*`, R does element-wise multiplication.

  Vectors $\boldsymbol{x}$ and $\boldsymbol{z}$ are 'orthogonal' if they are the same length and $\boldsymbol{x}^\top \boldsymbol{z} = 0$

- Rank of a matrix is the number of linearly independent columns. **Rank cannot be increased through multiplication!** (It can be decreased, though.)

- If a matrix is nonsingular, then solve will invert it.

```
solve(diag(1:3) + 4)
```

- Generalized inverse of a matrix (need not be square). $\boldsymbol{G}$ is a generalized inverse of $\boldsymbol{A}$ if $\boldsymbol{AGA} = \boldsymbol{A}$ If you find that G is a generalized inverse of $\boldsymbol{A}$, then you can generate all generalized inverses by letting $\boldsymbol{Z}$ vary in $\boldsymbol{G} + (\boldsymbol{I} - \boldsymbol{GA})\boldsymbol{Z}$. The Moore-Penrose inverse, $\boldsymbol{A}^+$, is a special and unique generalized inverse satisfying:

  1. $\boldsymbol{AA}^+\boldsymbol{A} = \boldsymbol{A}$
  2. $\boldsymbol{A}^+\boldsymbol{AA}^+ = \boldsymbol{A}^+$
  3. $\boldsymbol{AA}^+$ is symmetric
  4. $\boldsymbol{A}^+\boldsymbol{A}$ is symmetric

  The Moore-Penrose inverse of a matrix is available in the MASS package.

```
(A <- cbind( diag(c( 1,2,3)), c(4,8,12)))
A %*% MASS::ginv(A) %*% A
zapsmall(A %*% MASS::ginv(A) %*% A)
```

  Computers have trouble representing numbers close to 0 or ∞. The zapsmall function converts values less than some tolerance to zeroes.

- In linear models we often need the cross products: $\boldsymbol{X}^\top\boldsymbol{X}$ and $\boldsymbol{X}^\top\boldsymbol{y}$. There is a special R function to compute those quickly.

```
crossprod(X)
y <-  c(9, 12, 17)
crossprod(X, y)
```

- Random vectors: The expectation, or mean of a RV is made up of the means of each component.

$$E(\boldsymbol{y}) = E\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_n \end{pmatrix}$$

  The expectation of a linear combination is: $E(\boldsymbol{a}^\top \boldsymbol{y}) = \boldsymbol{a}^\top \boldsymbol{\mu}_y$

  To look at the second moment, dimensions get squared, the variance-covariance matrix is n by n.

$$\mathrm{Var}(\boldsymbol{y}) = E[(\boldsymbol{y} - \boldsymbol{\mu}_y)(\boldsymbol{y} - \boldsymbol{\mu}_y)^\top] = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{12} & \sigma_2^2 & \cdots & \sigma_{2n} \\ \vdots & \sigma_{ij} & \ddots & \vdots \\ \sigma_{n1} & \sigma_{n2} & \cdots & \sigma_n^2 \end{pmatrix}$$

  where the diagonal shows variances for each random variable and the off-diagonals are covariances. The variance of a linear combination is: $\mathrm{Var}(\boldsymbol{a}^\top \boldsymbol{y}) = \boldsymbol{a}^\top \mathrm{Var}(y) \boldsymbol{a}$

- Variance-covariance matrices can be factored

$$\boldsymbol{V} = \boldsymbol{DRD}$$

  where $\boldsymbol{D}$ is a diagonal matrix of standard deviations and $\boldsymbol{R}$ is the correlation matrix with ones on the diagonal.

- If $\boldsymbol{z}$ is a vector of 6 independent draws from a $N(0, \sigma^2)$ distribution, what are the mean and variance of $\begin{bmatrix} 1 \\ 3 \\ 8 \\ 2 \end{bmatrix} + \boldsymbol{\Sigma z}$ for a 4 by 6 positive definite matrix $\boldsymbol{\Sigma}$? (It will be normally distributed because it's a combination of normals.)

- **Ordinary Least Squares**
  $\boldsymbol{y} = \boldsymbol{X\beta} + \boldsymbol{\epsilon}$, $\boldsymbol{\epsilon} \sim (\boldsymbol{0}, \sigma^2 \boldsymbol{I})$. Suppose we need to find $\boldsymbol{\beta}$ to minimize the sum of squared errors.

$$Q(\boldsymbol{\beta}) = \sum_{i=1}^{n} (y_i - \boldsymbol{x}_i{}^\top \boldsymbol{\beta})^2 = (\boldsymbol{y} - \boldsymbol{X\beta})^\top (\boldsymbol{y} - \boldsymbol{X\beta}) = \boldsymbol{y}^\top \boldsymbol{y} - 2\boldsymbol{y}^\top \boldsymbol{X\beta} + \boldsymbol{\beta}^\top \boldsymbol{X}^\top \boldsymbol{X\beta}$$

  Use these facts about matrix derivatives:

$$\frac{d\boldsymbol{Ax}}{d\boldsymbol{x}^\top} = \boldsymbol{A} \quad \frac{d\boldsymbol{x}^\top \boldsymbol{Ax}}{d\boldsymbol{x}^\top} = 2\boldsymbol{Ax}$$

  We need to take a derivative wrt $\boldsymbol{\beta}$ and set it to 0.

$$-2\boldsymbol{X}^\top \boldsymbol{y} + 2\boldsymbol{X}^\top \boldsymbol{X\beta} = 0 \quad \text{or } \boldsymbol{X}^\top \boldsymbol{X\beta} = \boldsymbol{X}^\top \boldsymbol{y}$$

  These are called the "normal" equations. There could no solution, a unique solution, or many solutions.

  [Aside: the second derivative test works in higher dimensions as well. The second derivative is $2\boldsymbol{X}^\top \boldsymbol{X}$ which is positive definite, so the solution(s) are minima, not maxima.]

- **Generalized Least Squares**
  $\boldsymbol{y} = \boldsymbol{X\beta} + \boldsymbol{\epsilon}$, $\boldsymbol{\epsilon} \sim (\boldsymbol{0}, \sigma^2\boldsymbol{V})$. Now the function to minimize takes variance into account:
  $$Q(\boldsymbol{\beta}) = (\boldsymbol{y} - \boldsymbol{X\beta})^\top \boldsymbol{V}^{-1}(\boldsymbol{y} - \boldsymbol{X\beta}) = \boldsymbol{y}^\top \boldsymbol{V}^{-1}\boldsymbol{y} - 2\boldsymbol{y}^\top \boldsymbol{V}^{-1}\boldsymbol{X\beta} + \boldsymbol{\beta}^\top \boldsymbol{X}^\top \boldsymbol{V}^{-1}\boldsymbol{X\beta}$$
  and GLS solutions must solve
  $$\boldsymbol{X}^\top \boldsymbol{V}^{-1}\boldsymbol{X\beta} = \boldsymbol{X}^\top \boldsymbol{V}^{-1}\boldsymbol{y}$$

- Solving a system of equations: If $\boldsymbol{X}$ is of full column rank (as in regression) the solutions are
  $$\widehat{\boldsymbol{\beta}} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1}\boldsymbol{X}^\top \boldsymbol{y} \quad \text{or } \widehat{\boldsymbol{\beta}} = (\boldsymbol{X}^\top \boldsymbol{V}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^\top \boldsymbol{V}^{-1}\boldsymbol{y}$$
  DO NOT use matrix inversion to find the solution – it's too inaccurate.
  Instead use `solve(crossprod(X), crossprod(X, y))` which uses the QR decomposition $\boldsymbol{X}^\top \boldsymbol{X} = \boldsymbol{Q} \times \boldsymbol{R}$ where $\boldsymbol{QQ}^\top = \boldsymbol{I}$ and $\boldsymbol{R}$ is upper triangular.

```
(beta.hat <- solve(crossprod(X), crossprod(X, y)))
## check that QR is really crossprod(X)
qrdecomp <- qr(crossprod(X))
crossprod(X) -  qr.Q(qrdecomp) %*% qr.R(qrdecomp)
```

- When $\boldsymbol{X}$ is not of full column rank, there is not unique a solution to the normal equations. We can still use a generalized inverse.
  $$\tilde{\boldsymbol{\beta}} = (\boldsymbol{X}^\top \boldsymbol{V}^{-1}\boldsymbol{X})^g \boldsymbol{X}^\top \boldsymbol{V}^{-1}\boldsymbol{y}$$
  It does not matter which generalized inverse we use. All give the same solution for any estimable $\boldsymbol{\lambda}^\top \boldsymbol{\beta}$.
  Estimable: $\boldsymbol{\lambda}^\top \boldsymbol{\beta}$ is estimable if there exists some $\boldsymbol{b}^\top \boldsymbol{y}$ with $E(\boldsymbol{b}^\top \boldsymbol{y}) = \boldsymbol{\lambda}^\top \boldsymbol{\beta}\ \forall \boldsymbol{\beta}$. This condition implies that $\boldsymbol{\lambda}^\top$ is in the row space of $\boldsymbol{X}$, that is, $\boldsymbol{\lambda}^\top$ is a linear combination of rows of $\boldsymbol{X}$.

- The QR decomposition works for symmetric matrices, but the singular value decomposition (SVD) is more general: $\boldsymbol{A} = \boldsymbol{UDV}^\top$ with $\boldsymbol{UU}^\top = \boldsymbol{I}$ and $\boldsymbol{VV}^\top = \boldsymbol{I}$. The Cholesky decomposition (for symmetric matrices) is $\boldsymbol{A} = \boldsymbol{L}^\top \boldsymbol{L}$

```
svd.decomp <- svd(crossprod(X))
svd.decomp$u %*% diag(svd.decomp$d) %*% t(svd.decomp$v)
crossprod(chol(crossprod(X)))
```

  If $\boldsymbol{A}$ is symmetric (and therefore square), then $\boldsymbol{U} = \boldsymbol{V}$. The rank of the matrix is the number of non-zeroes on the diagonal of D. Finally, there is the eigen vector – eigen value or spectral decomposition, $\boldsymbol{A} = \boldsymbol{QGQ}^{-1}$

```
eigen(crossprod(X))
```

- Projections:
  The column space of $\boldsymbol{X}^\top \boldsymbol{X}$ (which is the column space of $\boldsymbol{X}^\top$) is the set of vectors generated by taking all possible linear combinations of scalars times columns of $\boldsymbol{X}^\top \boldsymbol{X}$.
  $\boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{V}^{-1}\boldsymbol{X})^g \boldsymbol{X}^\top \boldsymbol{V}^{-1}$ projects any vector into the column space of $\boldsymbol{X}$. In particular, when $\boldsymbol{V} = \boldsymbol{I}$, $\boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{X})^g \boldsymbol{X}^\top$ is the perpendicular projection operator (ppo), taking the shortest path to the column space of $\boldsymbol{X}$. Fitted values, $\widehat{\boldsymbol{y}} = \boldsymbol{X}\widehat{\boldsymbol{\beta}} = \boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{V}^{-1}\boldsymbol{X})^g \boldsymbol{X}^\top \boldsymbol{V}^{-1}\boldsymbol{y}$, are the projection of $\boldsymbol{y}$ into the column space of $\boldsymbol{X}$. When errors are iid, $\widehat{\boldsymbol{y}} = \boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{X})^g \boldsymbol{X}^\top \boldsymbol{y} = \boldsymbol{Hy}$ so $\boldsymbol{H} = \boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{X})^g \boldsymbol{X}^\top$ is called the 'hat' matrix. In effect, it puts a hat on $\boldsymbol{y}$. $\boldsymbol{H}$ projects $\boldsymbol{y}$ into the column space of $\boldsymbol{X}$. $\boldsymbol{I} - \boldsymbol{H}$ projects into the null space of $\boldsymbol{X}$, giving us the residuals $\boldsymbol{e} = (\boldsymbol{I} - \boldsymbol{H})\boldsymbol{y}$. Every vector in the null space of $\boldsymbol{X}$ is orthogonal to anything in the column space of $\boldsymbol{X}$, so residuals are orthogonal to fits and to any column of $\boldsymbol{X}$.