Data, Covariance, and Correlation Matrix

Nathaniel E. Helwig

Assistant Professor of Psychology and Statistics
University of Minnesota (Twin Cities)

Updated 16-Jan-2017
Copyright © 2017 by Nathaniel E. Helwig
1) The Data Matrix
   - Definition
   - Properties
   - R code

2) The Covariance Matrix
   - Definition
   - Properties
   - R code

3) The Correlation Matrix
   - Definition
   - Properties
   - R code

4) Miscellaneous Topics
   - Crossproduct calculations
   - Vec and Kronecker
   - Visualizing data
The Data Matrix
The data matrix refers to the array of numbers

\[
X = \begin{pmatrix}
    x_{11} & x_{12} & \cdots & x_{1p} \\
    x_{21} & x_{22} & \cdots & x_{2p} \\
    x_{31} & x_{32} & \cdots & x_{3p} \\
    \vdots & \vdots & \ddots & \vdots \\
    x_{n1} & x_{n2} & \cdots & x_{np}
\end{pmatrix}
\]

where \( x_{ij} \) is the \( j \)-th variable collected from the \( i \)-th item (e.g., subject).
- items/subjects are rows
- variables are columns

\( X \) is a data matrix of order \( n \times p \) (# items by # variables).
We can view a data matrix as a collection of column vectors:

\[ X = \begin{pmatrix} x_1 & x_2 & \cdots & x_p \end{pmatrix} \]

where \( x_j \) is the \( j \)-th column of \( X \) for \( j \in \{1, \ldots, p\} \).

The \( n \times 1 \) vector \( x_j \) gives the \( j \)-th variable’s scores for the \( n \) items.
Collection of Row Vectors

We can view a data matrix as a collection of row vectors:

\[
X = \begin{pmatrix}
  x'_1 \\
  x'_2 \\
  \vdots \\
  x'_n
\end{pmatrix}
\]

where \(x'_i\) is the \(i\)-th row of \(X\) for \(i \in \{1, \ldots, n\}\).

The \(1 \times p\) vector \(x'_i\) gives the \(i\)-th item’s scores for the \(p\) variables.
The sample mean of the $j$-th variable is given by

$$
\bar{x}_j = \frac{1}{n} \sum_{i=1}^{n} x_{ij}
$$

$$
= n^{-1} 1'_n x_j
$$

where

- $1_n$ denotes an $n \times 1$ vector of ones
- $x_j$ denotes the $j$-th column of $X$
Calculating Item (Row) Means

The sample mean of the $i$-th item is given by

$$
\bar{x}_i = \frac{1}{p} \sum_{j=1}^{p} x_{ij}
$$

$$
= p^{-1} x'_i 1_p
$$

where

- $1_p$ denotes an $p \times 1$ vector of ones
- $x'_i$ denotes the $i$-th row of $X$
Data Frame and Matrix Classes in R

```r
> data(mtcars)
> class(mtcars)
[1] "data.frame"
> dim(mtcars)
[1] 32 11
> head(mtcars)

mpg  cyl  disp  hp  drat   wt  qsec  vs  am  gear  carb
Mazda RX4       21.0   6  160  110  3.90  2.620  16.46  0  1     4  4
Mazda RX4 Wag   21.0   6  160  110  3.90  2.875  17.02  0  1     4  4
Datsun 710      22.8   4  108   93  3.85  2.320  18.61  1  1     4  1
Hornet 4 Drive  21.4   6  258  110  3.08  3.215  19.44  1  0     3  1
Hornet Sportabout 18.7  8  360  175  3.15  3.440  17.02  0  0     3  2
Valiant         18.1   6  225  105  2.76  3.460  20.22  1  0     3  1
> X <- as.matrix(mtcars)
> class(X)
[1] "matrix"
```
Row and Column Means

> # get row means (3 ways)
> rowMeans(X)[1:3]
  Mazda RX4 Mazda RX4 Wag   Datsun 710
     29.90727     29.98136     23.59818
> c(mean(X[,1]), mean(X[,2]), mean(X[,3]))
> apply(X,1,mean)[1:3]
  Mazda RX4 Mazda RX4 Wag   Datsun 710
     29.90727     29.98136     23.59818

> # get column means (3 ways)
> colMeans(X)[1:3]
          mpg      cyl     disp
     20.09062  6.18750 230.72188
> c(mean(X[,1]), mean(X[,2]), mean(X[,3]))
> apply(X,2,mean)[1:3]
          mpg      cyl     disp
     20.09062  6.18750 230.72188
Other Row and Column Functions

> # get column medians
> apply(X,2,median)[1:3]
   mpg  cyl  disp
   19.2  6.0 196.3
> median(X[,1]), median(X[,2]), median(X[,3]))
[1] 19.2 6.0 196.3

> # get column ranges
> apply(X,2,range)[,1:3]
   mpg  cyl  disp
[1,] 10.4  4  71.1
[2,] 33.9  8 472.0
> range(X[,1]), range(X[,2]), range(X[,3]))
   [,1]  [,2]  [,3]
[1,] 10.4  4  71.1
[2,] 33.9  8 472.0
The Covariance Matrix
The Covariance Matrix

**Definition**

The covariance matrix refers to the symmetric array of numbers

\[
S = \begin{pmatrix}
    s_1^2 & s_{12} & s_{13} & \cdots & s_{1p} \\
    s_{21} & s_2^2 & s_{23} & \cdots & s_{2p} \\
    s_{31} & s_{32} & s_3^2 & \cdots & s_{3p} \\
    \vdots & \vdots & \vdots & \ddots & \vdots \\
    s_{p1} & s_{p2} & s_{p3} & \cdots & s_p^2
\end{pmatrix}
\]

where

- \( s_j^2 = \frac{1}{n} \sum_{i=1}^{n} (x_{ij} - \bar{x}_j)^2 \) is the variance of the \( j \)-th variable
- \( s_{jk} = \frac{1}{n} \sum_{i=1}^{n} (x_{ij} - \bar{x}_j)(x_{ik} - \bar{x}_k) \) is the covariance between the \( j \)-th and \( k \)-th variables
- \( \bar{x}_j = \frac{1}{n} \sum_{i=1}^{n} x_{ij} \) is the mean of the \( j \)-th variable
Covariance Matrix from Data Matrix

We can calculate the covariance matrix such as

\[
S = \frac{1}{n} X_c' X_c
\]

where \( X_c = X - 1_n \bar{x}' = CX \) with

- \( \bar{x}' = (\bar{x}_1, \ldots, \bar{x}_p) \) denoting the vector of variable means
- \( C = I_n - n^{-1}1_n1_n' \) denoting a centering matrix

Note that the centered matrix \( X_c \) has the form

\[
X_c = \begin{pmatrix}
    x_{11} - \bar{x}_1 & x_{12} - \bar{x}_2 & \cdots & x_{1p} - \bar{x}_p \\
    x_{21} - \bar{x}_1 & x_{22} - \bar{x}_2 & \cdots & x_{2p} - \bar{x}_p \\
    x_{31} - \bar{x}_1 & x_{32} - \bar{x}_2 & \cdots & x_{3p} - \bar{x}_p \\
    \vdots & \vdots & \ddots & \vdots \\
    x_{n1} - \bar{x}_1 & x_{n2} - \bar{x}_2 & \cdots & x_{np} - \bar{x}_p
\end{pmatrix}
\]
Variances are Nonnegative

Variances are sums-of-squares, which implies that $s_j^2 \geq 0 \ \forall j$.

- $s_j^2 > 0$ as long as there does not exist an $\alpha$ such that $x_j = \alpha 1_n$

This implies that...

- $\text{tr}(S) \geq 0$ where $\text{tr}(\cdot)$ denotes the matrix trace function
- $\sum_{j=1}^{p} \lambda_j \geq 0$ where $(\lambda_1, \ldots, \lambda_p)$ are the eigenvalues of $S$

If $n < p$, then $\lambda_j = 0$ for at least one $j \in \{1, \ldots, p\}$. If $n \geq p$ and the $p$ columns of $X$ are linearly independent, then $\lambda_j > 0$ for all $j \in \{1, \ldots, p\}$.
The Cauchy-Schwarz Inequality

From the Cauchy-Schwarz inequality we have that

\[ s_{jk}^2 \leq s_j^2 s_k^2 \]

with the equality holding if and only if \( x_j \) and \( x_k \) are linearly dependent.

We could also write the Cauchy-Schwarz inequality as

\[ |s_{jk}| \leq s_j s_k \]

where \( s_j \) and \( s_k \) denote the standard deviations of the variables.
Covariance Matrix by Hand (hard way)

```r
> n <- nrow(X)
> C <- diag(n) - matrix(1/n, n, n)
> Xc <- C %*% X
> S <- t(Xc) %*% Xc / (n-1)
> S[1:3,1:6]

mpg     cyl    disp     hp    drat    wt
cyl     -9.172379  3.189516  199.6603  101.9315 -0.6683669  1.367371
disp   -633.097208 199.660282 15360.7998  6721.1587 -47.0640192 107.684204

# or #

> Xc <- scale(X, center=TRUE, scale=FALSE)
> S <- t(Xc) %*% Xc / (n-1)
> S[1:3,1:6]

mpg     cyl    disp     hp    drat    wt
cyl     -9.172379  3.189516  199.6603  101.9315 -0.6683669  1.367371
disp   -633.097208 199.660282 15360.7998  6721.1587 -47.0640192 107.684204
```

Nathaniel E. Helwig (U of Minnesota)
Covariance Matrix using `cov` Function (easy way)

```r
# calculate covariance matrix
> S <- cov(X)
> dim(S)
[1] 11 11

# check variance
> S[1,1]
[1] 36.3241
> var(X[,1])
[1] 36.3241
> sum((X[,1]-mean(X[,1]))^2) / (n-1)
[1] 36.3241

# check covariance
> S[1:3,1:6]

mpg  cyl  disp  hp  drat  wt
cyl -9.172379  3.189516  199.6602  101.9315 -0.6683669  1.367371
disp -633.097208 199.660282 15360.7998  6721.1587 -47.0640192 107.684204
```

Nathaniel E. Helwig (U of Minnesota) Data, Covariance, and Correlation Matrix Updated 16-Jan-2017 : Slide 19
The Correlation Matrix

Definition

The Correlation of Data

The **correlation matrix** refers to the symmetric array of numbers

$$R = \begin{pmatrix}
1 & r_{12} & r_{13} & \cdots & r_{1p} \\
 r_{21} & 1 & r_{23} & \cdots & r_{2p} \\
 r_{31} & r_{32} & 1 & \cdots & r_{3p} \\
 \vdots & \vdots & \vdots & \ddots & \vdots \\
 r_{p1} & r_{p2} & r_{p3} & \cdots & 1
\end{pmatrix}$$

where

$$r_{jk} = \frac{s_{jk}}{s_j s_k} = \frac{\sum_{i=1}^{n} (x_{ij} - \bar{x}_j)(x_{ik} - \bar{x}_k)}{\sqrt{\sum_{i=1}^{n} (x_{ij} - \bar{x}_j)^2} \sqrt{\sum_{i=1}^{n} (x_{ik} - \bar{x}_k)^2}}$$

is the Pearson correlation coefficient between variables $x_j$ and $x_k$. 

Correlation Matrix from Data Matrix

We can calculate the correlation matrix such as

\[ R = \frac{1}{n} X_s' X_s \]

where \( X_s = CXD^{-1} \) with

- \( C = I_n - n^{-1}1_n1_n' \) denoting a centering matrix
- \( D = \text{diag}(s_1, \ldots, s_p) \) denoting a diagonal scaling matrix

Note that the standardized matrix \( X_s \) has the form

\[
X_s = \begin{pmatrix}
\frac{(x_{11} - \bar{x}_1)}{s_1} & \frac{(x_{12} - \bar{x}_2)}{s_2} & \cdots & \frac{(x_{1p} - \bar{x}_p)}{s_p} \\
\frac{(x_{21} - \bar{x}_1)}{s_1} & \frac{(x_{22} - \bar{x}_2)}{s_2} & \cdots & \frac{(x_{2p} - \bar{x}_p)}{s_p} \\
\frac{(x_{31} - \bar{x}_1)}{s_1} & \frac{(x_{32} - \bar{x}_2)}{s_2} & \cdots & \frac{(x_{3p} - \bar{x}_p)}{s_p} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{(x_{n1} - \bar{x}_1)}{s_1} & \frac{(x_{n2} - \bar{x}_2)}{s_2} & \cdots & \frac{(x_{np} - \bar{x}_p)}{s_p}
\end{pmatrix}
\]
Correlation of a Variable with Itself is One

Assuming that \( s_j^2 > 0 \) for all \( j \in \{1, \ldots, p\} \), we have that

\[
\text{Cor}(\mathbf{x}_j, \mathbf{x}_k) = \frac{\sum_{i=1}^{n} (x_{ij} - \bar{x}_j)(x_{ik} - \bar{x}_k)}{\sqrt{\sum_{i=1}^{n} (x_{ij} - \bar{x}_j)^2} \sqrt{\sum_{i=1}^{n} (x_{ik} - \bar{x}_k)^2}} = \begin{cases} 
1 & \text{if } j = k \\
r_{jk} & \text{if } j \neq k 
\end{cases}
\]

Because \( r_{jk} = 1 \) whenever \( j = k \), we know that

- \( \text{tr}(\mathbf{R}) = p \) where \( \text{tr}(\cdot) \) denotes the matrix trace function
- \( \sum_{j=1}^{p} \lambda_j = p \) where \((\lambda_1, \ldots, \lambda_p)\) are the eigenvalues of \( \mathbf{R} \)

We also know that the eigenvalues satisfy

- \( \lambda_j = 0 \) for at least one \( j \in \{1, \ldots, p\} \) if \( n < p \)
- \( \lambda_j > 0 \ \forall j \) if columns of \( \mathbf{X} \) are linearly independent
The Cauchy-Schwarz Inequality (revisited)

Reminder: the Cauchy-Schwarz inequality implies that

\[ s_{jk}^2 \leq s_j^2 s_k^2 \]

with the equality holding if and only if \( x_j \) and \( x_k \) are linearly dependent.

Rearranging the terms, we have that

\[ \frac{s_{jk}^2}{s_j^2 s_k^2} \leq 1 \quad \leftrightarrow \quad r_{jk}^2 \leq 1 \]

which implies that \( |r_{jk}| \leq 1 \) with equality holding if and only if \( x_j = \alpha 1_n + \beta x_k \) for some scalars \( \alpha \in \mathbb{R} \) and \( \beta \in \mathbb{R} \).
Correlation Matrix by Hand (hard way)

\[
\begin{align*}
\text{Correlation Matrix} & \\
> n <- \text{nrow}(X) \\
> C <- \text{diag}(n) - \text{matrix}(1/n, n, n) \\
> D <- \text{diag}(\text{apply}(X, 2, \text{sd})) \\
> Xs <- C \times X \times \text{solve}(D) \\
> R <- t(Xs) \times Xs / (n-1) \\
> R[1:3, 1:6]
\end{align*}
\]

\[
\begin{array}{cccccc}
[1,] & 1.0000000 & -0.8521620 & -0.8475514 & -0.7761684 & 0.6811719 & -0.8676594 \\
[2,] & -0.8521620 & 1.0000000 & 0.9020329 & 0.8324475 & -0.6999381 & 0.7824958 \\
[3,] & -0.8475514 & 0.9020329 & 1.0000000 & 0.7909486 & -0.7102139 & 0.8879799 \\
\end{array}
\]

# or #

\[
\begin{align*}
\text{Correlation Matrix} & \\
> Xs <- \text{scale}(X, \text{center}=TRUE, \text{scale}=TRUE) \\
> R <- t(Xs) \times Xs / (n-1) \\
> R[1:3, 1:6]
\end{align*}
\]

\[
\begin{array}{cccccc}
\text{mpg} & \text{cyl} & \text{disp} & \text{hp} & \text{drat} & \text{wt} \\
\text{mpg} & 1.0000000 & -0.8521620 & -0.8475514 & -0.7761684 & 0.6811719 & -0.8676594 \\
\text{cyl} & -0.8521620 & 1.0000000 & 0.9020329 & 0.8324475 & -0.6999381 & 0.7824958 \\
\text{disp} & -0.8475514 & 0.9020329 & 1.0000000 & 0.7909486 & -0.7102139 & 0.8879799 \\
\end{array}
\]
Correlation Matrix using `cor` Function (easy way)

```r
# calculate correlation matrix
> R <- cor(X)
> dim(R)
[1] 11 11

# check correlation of mpg and cyl
> R[1,2]
[1] -0.852162
> cor(X[,1],X[,2])
[1] -0.852162
> cov(X[,1],X[,2]) / (sd(X[,1]) * sd(X[,2]))
[1] -0.852162

# check correlations
> R[1:3,1:6]
   mpg  cyl  disp  hp  drat  wt
mpg 1.000000 -0.8521620 -0.8475514 -0.7761684 0.6811719 -0.8676594
cyl -0.8521620 1.0000000  0.9020329  0.8324475 -0.6999381  0.7824958
disp -0.8475514  0.9020329  1.0000000  0.7909486 -0.7102139  0.8879799
```

Nathaniel E. Helwig (U of Minnesota)
Two Types of Matrix Crossproducts

We often need to calculate one of two different types of crossproducts:

- \( X'Y \) = “regular” crossproduct of \( X \) and \( Y \)
- \( XY' \) = “transpose” crossproduct of \( X \) and \( Y \)

Regular crossproduct is \( X' \) being post-multiplied by \( Y \).

Transpose crossproduct is \( X \) being post-multiplied by \( Y' \).
Simple and Efficient Crossproducts in R

```r
> X <- matrix(rnorm(2*3), 2, 3)
> Y <- matrix(rnorm(2*3), 2, 3)
> t(X) %*% Y
             [,1]         [,2]         [,3]
[1,]  0.1342302 -1.8181837 -1.107821
[2,]  1.1014703 -0.6619466 -1.356606
[3,]  0.8760823 -1.0077151 -1.340044
> crossprod(X, Y)
             [,1]         [,2]         [,3]
[1,]  0.1342302 -1.8181837 -1.107821
[2,]  1.1014703 -0.6619466 -1.356606
[3,]  0.8760823 -1.0077151 -1.340044
> X %*% t(Y)
             [,1]         [,2]
[1,]  0.8364239  3.227566
[2,] -1.3899946 -2.704184
> tcrossprod(X, Y)
             [,1]         [,2]
[1,]  0.8364239  3.227566
[2,] -1.3899946 -2.704184
```
Turning a Matrix into a Vector

The vectorization (vec) operator turns a matrix into a vector:

$$\text{vec}(\mathbf{X}) = (x_{11}, x_{21}, \ldots, x_{n1}, x_{12}, \ldots, x_{n2}, \ldots, x_{1p}, \ldots, x_{np})'$$

where the vectorization is done column-by-column.

In R, we just use the combine function `c` to vectorize a matrix

```r
> X <- matrix(1:6, 2, 3)
> X
     [,1] [,2] [,3]
[1,]   1   3   5
[2,]   2   4   6
> c(X)
[1] 1 2 3 4 5 6
> c(t(X))
[1] 1 3 5 2 4 6
```
vec Operator Properties

Some useful properties of the vec(·) operator include:

- \( \text{vec}(a') = \text{vec}(a) = a \) for any vector \( a \in \mathbb{R}^m \)
- \( \text{vec}(ab') = b \otimes a \) for any vectors \( a \in \mathbb{R}^m \) and \( b \in \mathbb{R}^n \)
- \( \text{vec}(A)'\text{vec}(B) = \text{tr}(A'B) \) for any matrices \( A, B \in \mathbb{R}^{m \times n} \)
- \( \text{vec}(ABC) = (C' \otimes A)\text{vec}(B) \) if the product \( ABC \) exists

Note: \( \otimes \) is the Kronecker product, which is defined on the next slide.
## Kronecker Product of Two Matrices

Given $X = \{x_{ij}\}_{n \times p}$ and $Y = \{y_{ij}\}_{m \times q}$, the Kronecker product is

$$
X \otimes Y = \begin{pmatrix}
    x_{11}Y & x_{12}Y & \cdots & x_{1p}Y \\
x_{21}Y & x_{22}Y & \cdots & x_{2p}Y \\
    \vdots & \vdots & \ddots & \vdots \\
x_{n1}Y & x_{n2}Y & \cdots & x_{np}Y
\end{pmatrix}
$$

which is a matrix of order $mn \times pq$.

In R, the `kronecker` function calculates Kronecker products:

```r
> X <- matrix(1:4, 2, 2)
> Y <- matrix(5:10, 2, 3)
> kronecker(X, Y)
[1,]   5   7   9  15  21  27
[2,]   6   8  10  18  24  30
[3,]  10  14  18  20  28  36
[4,]  12  16  20  24  32  40
```
Kronecker Product Properties

Some useful properties of the Kronecker product include:

1. $A \otimes a = Aa = aA = a \otimes A$ for any $a \in \mathbb{R}$ and $A \in \mathbb{R}^{m \times n}$

2. $(A \otimes B)' = A' \otimes B'$ for any matrices $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{p \times q}$

3. $a' \otimes b = ba' = b \otimes a'$ for any vectors $a \in \mathbb{R}^m$ and $b \in \mathbb{R}^n$

4. $\text{tr}(A \otimes B) = \text{tr}(A)\text{tr}(B)$ for any matrices $A \in \mathbb{R}^{m \times m}$ and $B \in \mathbb{R}^{p \times p}$

5. $(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$ for any invertible matrices $A$ and $B$

6. $(A \otimes B)^\dagger = A^\dagger \otimes B^\dagger$ where $()^\dagger$ is Moore-Penrose pseudoinverse

7. $|A \otimes B| = |A|^p |B|^m$ for any matrices $A \in \mathbb{R}^{m \times m}$ and $B \in \mathbb{R}^{p \times p}$

8. $\text{rank}(A \otimes B) = \text{rank}(A)\text{rank}(B)$ for any matrices $A$ and $B$

9. $A \otimes (B + C) = A \otimes B + A \otimes C$ for any matrices $A$, $B$, and $C$

10. $(A + B) \otimes C = A \otimes C + B \otimes C$ for any matrices $A$, $B$, and $C$

11. $(A \otimes B) \otimes C = A \otimes (B \otimes C)$ for any matrices $A$, $B$, and $C$

12. $(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$ for any matrices $A$, $B$, $C$, and $D$
Suppose the rows of $X$ are iid samples from some multivariate distribution with mean $\mu = (\mu_1, \ldots, \mu_p)'$ and covariance matrix $\Sigma$.

- $x_i \overset{iid}{\sim} (\mu, \Sigma)$ where $x_i$ is the $i$-th row of $X$

If we let $y = \text{vec}(X')$, then the expectation and covariance are

- $E(y) = I_n \otimes \mu$ is the mean vector
- $V(y) = I_n \otimes \Sigma$ is the covariance matrix

Note that the covariance matrix is block diagonal

$$I_n \otimes \Sigma = \begin{pmatrix} \Sigma & 0 & \cdots & 0 \\ 0 & \Sigma & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \Sigma \end{pmatrix}$$

given that data from different subjects are assumed to be independent.
Two Versions of a Scatterplot in R

```r
plot(mtcars$hp, mtcars$mpg, xlab="HP", ylab="MPG")
library(car)
scatterplot(mtcars$hp, mtcars$mpg, xlab="HP", ylab="MPG")
```
Two Versions of a Scatterplot Matrix in R

cylint <- as.integer(factor(mtcars$cyl))
pairs(~mpg+disp+hp+wt, data=mtcars, col=cylint, pch=cylint)
library(car)
scatterplotMatrix(~mpg+disp+hp+wt|cyl, data=mtcars)
library(scatterplot3d)
sp3d <- scatterplot3d(mtcars$hp, mtcars$wt, mtcars$mpg,
  type="h", color=cylint, pch=cylint,
  xlab="HP", ylab="WT", zlab="MPG")
fitmod <- lm(mpg ~ hp + wt, data=mtcars)
sp3d$plane3d(fitmod)
fitmod <- lm(mpg ~ hp + wt, data=mtcars)
hpseq <- seq(50, 330, by=20)
wtseq <- seq(1.5, 5.4, length=15)
newdata <- expand.grid(hp=hpseq, wt=wtseq)
fit <- predict(fitmod, newdata)
fitmat <- matrix(fit, 15, 15)
image(hpseq, wtseq, fitmat, xlab="HP", ylab="WT")
library(bigsplines)
imagebar(hpseq, wtseq, fitmat, xlab="HP", ylab="WT", zlab="MPG", col=heat.colors(12), ncolor=12)
Correlation Matrix Plot in R

```r
# Load the mtcars dataset
data(mtcars)

cmat <- cor(mtcars)
library(corrplot)
corrplot(cmat, method="circle")
corrplot.mixed(cmat, lower="number", upper="ellipse")
```
Correlation Matrix Color Image (Heat Map) in R

cmat <- cor(mtcars)
p <- nrow(cmat)
library(RColorBrewer)
imagebar(1:p, 1:p, cmat[,p:1], axes=F, zlim=c(-1,1), xlab="", ylab="", col=brewer.pal(7, "RdBu"))
axis(1, 1:p, labels=rownames(cmat))
axis(2, p:1, labels=colnames(cmat))
for(k in 1:p) { for(j in 1:k) { if(j < k) text(j, p+1-k, labels=round(cmat[j,k],2), cex=0.75) } }

Nathaniel E. Helwig (U of Minnesota)