Data, Covariance, and Correlation Matrix

Nathaniel E. Helwig

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

Updated 16-Jan-2017

Copyright

Outline of Notes

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 Organization of Data

The data matrix refers to the array of numbers

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

where x_{ij} is the j-th variable collected from the i-th item (e.g., subject).

- items/subjects are rows
- variables are columns

$\mathbf{X}$ is a data matrix of order $n \times p$ (# items by # variables).

Collection of Column Vectors

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

$$\mathbf{X} = \begin{pmatrix} | & | & & | \\ \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_p \\ | & | & & | \end{pmatrix}$$

where $\mathbf{x}_j$ is the j-th column of $\mathbf{X}$ for $j \in \{1, \ldots, p\}$.

The $n \times 1$ vector $\mathbf{x}_j$ 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:

$$\mathbf{X} = \begin{pmatrix} — & \mathbf{x}_1' & — \\ — & \mathbf{x}_2' & — \\ & \vdots & \\ — & \mathbf{x}_n' & — \end{pmatrix}$$

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

The $1 \times p$ vector $\mathbf{x}_i'$ gives the i-th item's scores for the p variables.

Calculating Variable (Column) Means

The sample mean of the *j*-th variable is given by

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

where

- $\mathbf{1}_n$ denotes an $n \times 1$ vector of ones
- $\mathbf{x}_j$ denotes the *j*-th column of $\mathbf{X}$

Calculating Item (Row) Means

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

$$\begin{aligned}\bar{x}_i &= \frac{1}{p}\sum_{j=1}^{p} x_{ij} \\ &= p^{-1}\mathbf{x}_i'\mathbf{1}_p\end{aligned}$$

where

- $\mathbf{1}_p$ denotes an $p \times 1$ vector of ones
- $\mathbf{x}_i'$ denotes the i-th row of $\mathbf{X}$

Data Frame and Matrix Classes in 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,]))
[1] 29.90727 29.98136 23.59818
> 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]))
[1]  20.09062   6.18750 230.72188
> 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
> c(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
> cbind(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 Covariation of Data

The covariance matrix refers to the symmetric array of numbers

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

where

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

Covariance Matrix from Data Matrix

We can calculate the covariance matrix such as

$$\mathbf{S} = \frac{1}{n}\mathbf{X}_c'\mathbf{X}_c$$

where $\mathbf{X}_c = \mathbf{X} - \mathbf{1}_n\bar{\mathbf{x}}' = \mathbf{C}\mathbf{X}$ with

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

Note that the centered matrix $\mathbf{X}_c$ has the form

$$\mathbf{X}_c = \begin{pmatrix} x_{11} - \bar{x}_1 & x_{12} - \bar{x}_2 & \cdots & x_{1p} - \bar{x}_p \\ x_{21} - \bar{x}_1 & x_{22} - \bar{x}_2 & \cdots & x_{2p} - \bar{x}_p \\ x_{31} - \bar{x}_1 & x_{32} - \bar{x}_2 & \cdots & x_{3p} - \bar{x}_p \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} - \bar{x}_1 & x_{n2} - \bar{x}_2 & \cdots & x_{np} - \bar{x}_p \end{pmatrix}$$

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 α such that $\mathbf{x}_j = \alpha \mathbf{1}_n$

This implies that...

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

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 $\mathbf{X}$ 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 $\mathbf{x}_j$ and $\mathbf{x}_k$ 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)

```
> 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
mpg   36.324103  -9.172379  -633.0972 -320.7321   2.1950635  -5.116685
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
mpg   36.324103  -9.172379  -633.0972 -320.7321   2.1950635  -5.116685
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
```

Covariance Matrix using cov Function (easy way)

```
# 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
mpg    36.324103  -9.172379  -633.0972 -320.7321   2.1950635  -5.116685
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
```

The Correlation Matrix

The Correlation of Data

The correlation matrix refers to the symmetric array of numbers

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

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 $\mathbf{x}_j$ and $\mathbf{x}_k$.

Correlation Matrix from Data Matrix

We can calculate the correlation matrix such as

$$\mathbf{R} = \frac{1}{n}\mathbf{X}_s'\mathbf{X}_s$$

where $\mathbf{X}_s = \mathbf{C}\mathbf{X}\mathbf{D}^{-1}$ with

- $\mathbf{C} = \mathbf{I}_n - n^{-1}\mathbf{1}_n\mathbf{1}_n'$ denoting a centering matrix
- $\mathbf{D} = \mathrm{diag}(s_1, \ldots, s_p)$ denoting a diagonal scaling matrix

Note that the standardized matrix $\mathbf{X}_s$ has the form

$$\mathbf{X}_s = \begin{pmatrix} (x_{11} - \bar{x}_1)/s_1 & (x_{12} - \bar{x}_2)/s_2 & \cdots & (x_{1p} - \bar{x}_p)/s_p \\ (x_{21} - \bar{x}_1)/s_1 & (x_{22} - \bar{x}_2)/s_2 & \cdots & (x_{2p} - \bar{x}_p)/s_p \\ (x_{31} - \bar{x}_1)/s_1 & (x_{32} - \bar{x}_2)/s_2 & \cdots & (x_{3p} - \bar{x}_p)/s_p \\ \vdots & \vdots & \ddots & \vdots \\ (x_{n1} - \bar{x}_1)/s_1 & (x_{n2} - \bar{x}_2)/s_2 & \cdots & (x_{np} - \bar{x}_p)/s_p \end{pmatrix}$$

Correlation of a Variable with Itself is One

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

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

Because $r_{jk} = 1$ whenever $j = k$, we know that

- $\mathrm{tr}(\mathbf{R}) = p$ where $\mathrm{tr}(\cdot)$ 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 $\mathbf{x}_j$ and $\mathbf{x}_k$ are linearly dependent.

Rearranging the terms, we have that

$$\frac{s_{jk}^2}{s_j^2 s_k^2} \leq 1 \quad \longleftrightarrow \quad r_{jk}^2 \leq 1$$

which implies that $|r_{jk}| \leq 1$ with equality holding if and only if $\mathbf{x}_j = \alpha \mathbf{1}_n + \beta \mathbf{x}_k$ for some scalars $\alpha \in \mathbb{R}$ and $\beta \in \mathbb{R}$.

Correlation Matrix by Hand (hard way)

```
> n <- nrow(X)
> C <- diag(n) - matrix(1/n, n, n)
> D <- diag(apply(X, 2, sd))
> Xs <- C %*% X %*% solve(D)
> R <- t(Xs) %*% Xs / (n-1)
> R[1:3,1:6]
           [,1]       [,2]       [,3]       [,4]       [,5]       [,6]
[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

# or #

> Xs <- scale(X, center=TRUE, scale=TRUE)
> R <- t(Xs) %*% Xs / (n-1)
> R[1:3,1:6]
            mpg        cyl       disp         hp       drat         wt
mpg   1.0000000 -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
```

Correlation Matrix using `cor` Function (easy way)

```
# 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   1.0000000 -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
```

Miscellaneous Topics

Two Types of Matrix Crossproducts

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

- $\mathbf{X}'\mathbf{Y}$ = "regular" crossproduct of $\mathbf{X}$ and $\mathbf{Y}$
- $\mathbf{X}\mathbf{Y}'$ = "transpose" crossproduct of $\mathbf{X}$ and $\mathbf{Y}$

Regular crossproduct is $\mathbf{X}'$ being post-multipled by $\mathbf{Y}$.

Transpose crossproduct is $\mathbf{X}$ being post-multipled by $\mathbf{Y}'$.

Simple and Efficient Crossproducts in 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

```
> 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($\cdot$) operator include:

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

Note: $\otimes$ is the Kronecker product, which is defined on the next slide.

Kronecker Product of Two Matrices

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

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

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

In R, the `kronecker` function calculates Kronecker products

```
> X <- matrix(1:4,2,2)
> Y <- matrix(5:10,2,3)
> kronecker(X, Y)
     [,1] [,2] [,3] [,4] [,5] [,6]
[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. $\mathbf{A} \otimes a = \mathbf{A}a = a\mathbf{A} = a \otimes \mathbf{A}$ for any $a \in \mathbb{R}$ and $\mathbf{A} \in \mathbb{R}^{m \times n}$
2. $(\mathbf{A} \otimes \mathbf{B})' = \mathbf{A}' \otimes \mathbf{B}'$ for any matrices $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{B} \in \mathbb{R}^{p \times q}$
3. $\mathbf{a}' \otimes \mathbf{b} = \mathbf{b}\mathbf{a}' = \mathbf{b} \otimes \mathbf{a}'$ for any vectors $\mathbf{a} \in \mathbb{R}^m$ and $\mathbf{b} \in \mathbb{R}^n$
4. $\mathrm{tr}(\mathbf{A} \otimes \mathbf{B}) = \mathrm{tr}(\mathbf{A})\mathrm{tr}(\mathbf{B})$ for any matrices $\mathbf{A} \in \mathbb{R}^{m \times m}$ and $\mathbf{B} \in \mathbb{R}^{p \times p}$
5. $(\mathbf{A} \otimes \mathbf{B})^{-1} = \mathbf{A}^{-1} \otimes \mathbf{B}^{-1}$ for any invertible matrices $\mathbf{A}$ and $\mathbf{B}$
6. $(\mathbf{A} \otimes \mathbf{B})^{\dagger} = \mathbf{A}^{\dagger} \otimes \mathbf{B}^{\dagger}$ where $(\cdot)^{\dagger}$ is Moore-Penrose pseudoinverse
7. $|\mathbf{A} \otimes \mathbf{B}| = |\mathbf{A}|^p |\mathbf{B}|^m$ for any matrices $\mathbf{A} \in \mathbb{R}^{m \times m}$ and $\mathbf{B} \in \mathbb{R}^{p \times p}$
8. $\mathrm{rank}(\mathbf{A} \otimes \mathbf{B}) = \mathrm{rank}(\mathbf{A})\mathrm{rank}(\mathbf{B})$ for any matrices $\mathbf{A}$ and $\mathbf{B}$
9. $\mathbf{A} \otimes (\mathbf{B} + \mathbf{C}) = \mathbf{A} \otimes \mathbf{B} + \mathbf{A} \otimes \mathbf{C}$ for any matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$
10. $(\mathbf{A} + \mathbf{B}) \otimes \mathbf{C} = \mathbf{A} \otimes \mathbf{C} + \mathbf{B} \otimes \mathbf{C}$ for any matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$
11. $(\mathbf{A} \otimes \mathbf{B}) \otimes \mathbf{C} = \mathbf{A} \otimes (\mathbf{B} \otimes \mathbf{C})$ for any matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$
12. $(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = (\mathbf{AC}) \otimes (\mathbf{BD})$ for any matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{D}$

Common Application of Vec and Kronecker

Suppose the rows of $\mathbf{X}$ are iid samples from some multivariate distribution with mean $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_p)'$ and covariance matrix $\boldsymbol{\Sigma}$.

- $\mathbf{x}_i \overset{\text{iid}}{\sim} (\boldsymbol{\mu}, \boldsymbol{\Sigma})$ where $\mathbf{x}_i$ is the i-th row of $\mathbf{X}$

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

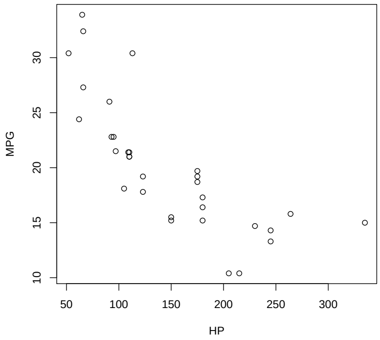
- $E(\mathbf{y}) = \mathbf{1}_n \otimes \boldsymbol{\mu}$ is the mean vector
- $V(\mathbf{y}) = \mathbf{I}_n \otimes \boldsymbol{\Sigma}$ is the covariance matrix

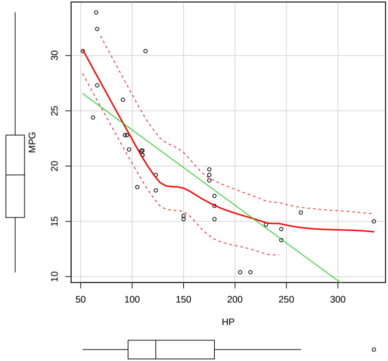
Note that the covariance matrix is block diagonal

$$\mathbf{I}_n \otimes \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{\Sigma} \end{pmatrix}$$

given that data from different subjects are assumed to be independent.

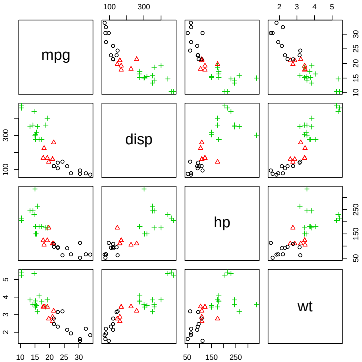
Two Versions of a Scatterplot in R

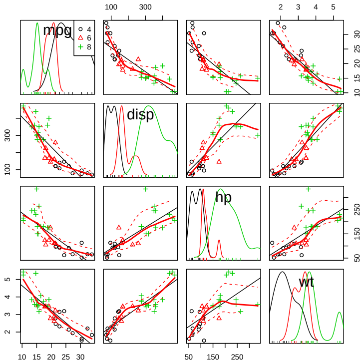





```
plot(mtcars$hp, mtcars$mpg, xlab="HP", ylab="MPG")
library(car)
scatterplot(mtcars$hp, mtcars$mpg, xlab="HP", ylab="MPG")
```

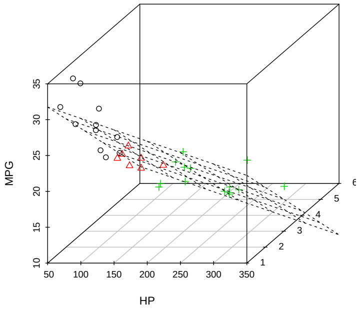
Two Versions of a Scatterplot Matrix in R

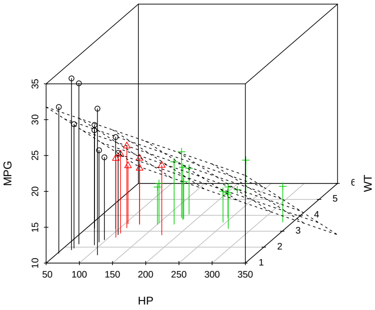





```
cylint <- as.integer(factor(mtcars$cyl))
pairs(~mpg+disp+hp+wt, data=mtcars, col=cylint, pch=cylint)
library(car)
scatterplotMatrix(~mpg+disp+hp+wt|cyl, data=mtcars)
```

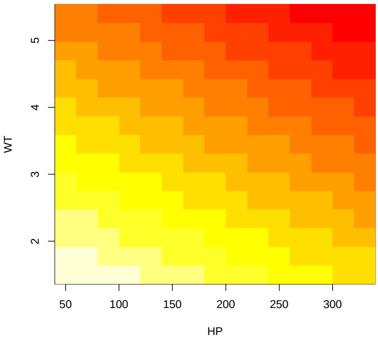
Three-Dimensional Scatterplot in R

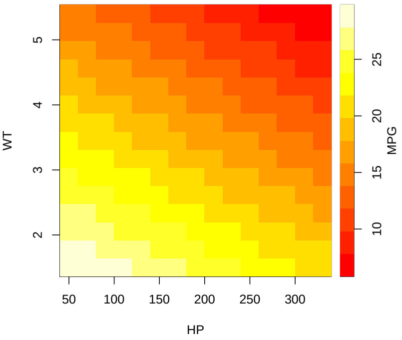





```
library(scatterplot3d)
sp3d <- scatterplot3d(mtcars$hp, mtcars$wt, mtcars$mpg,
                      type="h", color=cylint, pch=cylint,
                      xlab="HP", ylab="WT", zlab="MPG")
fitmod <- lm(mpg ~ hp + wt, data=mtcars)
sp3d$plane3d(fitmod)
```

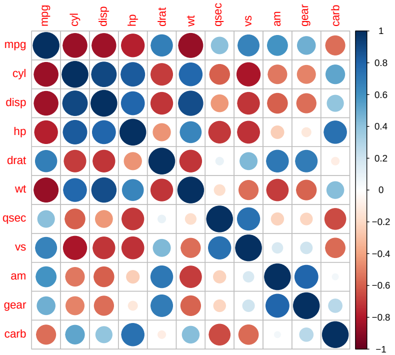
Color Image (Heat Map) Plots in R

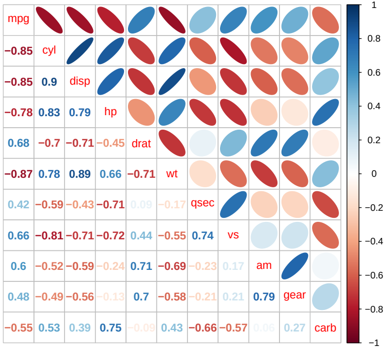





```
fitmod <- lm(mpg ~ hp + wt, data=mtcars)
hpseq <- seq(50, 330, by=20)
wtseq <- seq(1.5, 5.4, length=15)
newdata <- expand.grid(hp=hpseq, wt=wtseq)
fit <- predict(fitmod, newdata)
fitmat <- matrix(fit, 15, 15)
image(hpseq, wtseq, fitmat, xlab="HP", ylab="WT")
library(bigsplines)
imagebar(hpseq, wtseq, fitmat, xlab="HP", ylab="WT", zlab="MPG", col=heat.colors(12), ncolor=12)
```

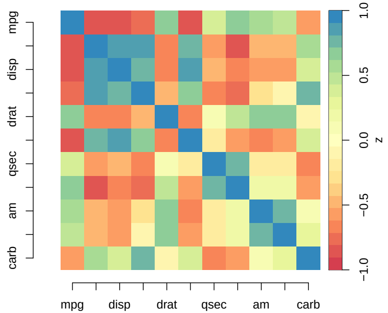
Correlation Matrix Plot in R

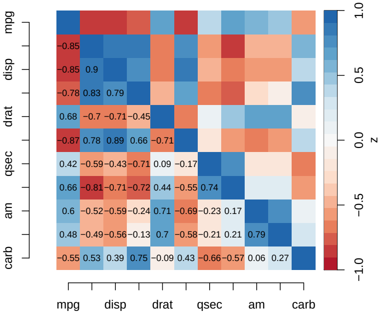





```
cmat <- cor(mtcars)
library(corrplot)
corrplot(cmat, method="circle")
corrplot.mixed(cmat, lower="number", upper="ellipse")
```

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) } }
```