Displays for Statistics 5401

Lecture 4

September 14, 2005

Christopher Bingham, Instructor

612-625-1024, kb@umn.edu
372 Ford Hall

Class Web Page

http://www.stat.umn.edu/~kb/classes/5401

Matrix of 1's is useful

$$\mathbf{1}_m = \begin{bmatrix} 1 \\ 1 \\ 1 \\ \dots \\ 1 \end{bmatrix}$$, $m \times 1$ a column vector of m 1's

Using $\mathbf{1}_n$ you can use matrix multiplication to express a sum $\sum_{1 \le j \le n} x_j$:

Suppose $\mathbf{x} = [x_1, x_2, \dots, x_n]'$, then

$$\mathbf{1}_n'\mathbf{x} = \sum_{1 \le j \le n} 1 \times x_j = \sum_{1 \le j \le n} x_j.$$

You generate $\mathbf{1}_n$ in MacAnova by **rep(1,n)**:

```
Cmd> a <- matrix(vector(1.04,0.696, -0.651,0.13, 1.5,1.61), 2)

Cmd> a # m = 2 rows, n = 3 columns
(1,1)        1.04      -0.651         1.5
(2,1)       0.696        0.13        1.61

Cmd> sum(a) # black box column sums
(1,1)       1.736      -0.521        3.11

Cmd> ones_2 <- rep(1,2)

Cmd> ones_2' %*% a # white box column sums
(1,1)       1.736      -0.521        3.11
```

Or by a simple macro

```
Cmd> ones <- macro("rep(1,$1)"); ones(5) # same as rep(1,5)
(1)            1           1           1           1           1

Cmd> ones(2)' %*% a # exactly same as rep(2,1)' %*% a
(1,1)       1.736      -0.521        3.11
```


Rank of a matrix

Let $\mathbf{A} = [\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_n] = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m]'$
(by columns $\mathbf{A}_j$) (by rows $\mathbf{a}_i'$)
be a m by n matrix.

- Columns $\mathbf{A}_j$ are m by 1
- Rows $\mathbf{a}_i'$ are 1 by n

Also let $\mathbf{e}_k^\ell$ be column k of $\mathbf{I}_\ell$, that is

$$\mathbf{e}_k^\ell = \begin{bmatrix} 0 \\ \dots \\ 0 \\ 1 \\ 0 \\ \dots \\ 0 \end{bmatrix} \begin{matrix} 1 \\ \dots \\ k-1 \\ k \\ k+1 \\ \dots \\ \ell \end{matrix} \quad \begin{matrix} \text{length } \ell \\ \text{column vector} \end{matrix}$$

Example: $\mathbf{I}_4 = \begin{bmatrix} 1 & 0 & \underline{0} & 0 \\ 0 & 1 & \underline{0} & 0 \\ 0 & 0 & \underline{1} & 0 \\ 0 & 0 & \underline{0} & 1 \end{bmatrix}$, so $\mathbf{e}_3^4 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$

Fact: For any m by n $\mathbf{A}$,

$\mathbf{A} = \sum_{1 \le j \le n} \mathbf{A}_j (\mathbf{e}_j^n)'$, sum of *n* outer products

and

$\mathbf{A} = \sum_{1 \le j \le m} \mathbf{e}_i^m \mathbf{a}_i'$, sum of *m* outer products.

Conclusion:

- You can decompose *any* $m \times n$ matrix as a sum of outer products and you never need more than min(m,n) outer products to do it.
- Such a decomposition is not unique.

Vocabulary

The *rank* of $\mathbf{A}$ is the smallest number of non-zero outer products needed to represent $\mathbf{A}$ as a sum of outer products.

It should be clear that

- rank($\mathbf{A}$) $\le$ min(m,n)

since min(m,n) always suffices.

Particular Cases

- A column vector $\mathbf{x} \neq \mathbf{0}$ has rank 1
- A row vector $\mathbf{x}' \neq \mathbf{0}$ has rank 1
- When $\mathbf{x} \neq \mathbf{0}$ and $\mathbf{y} \neq \mathbf{0}$ are vectors, then $\mathbf{xy}'$ has rank 1 (it's a single outer product).
- rank of $\mathbf{D} = \text{diag}(d_1, d_2, .., d_p)$ = number of $d_j \neq 0$

$$\text{rank}\left(\begin{bmatrix} 3 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -2 \end{bmatrix}\right) = 2$$

- rank of $\mathbf{I}_p = p$

Vocabulary

A collection $\{\boldsymbol{\ell}_1, \boldsymbol{\ell}_2, ..., \boldsymbol{\ell}_s\}$ of non-zero vectors is *linearly independent* when, for each $\boldsymbol{\ell}_j$, it's impossible to find c_k's to express $\boldsymbol{\ell}_j$ in terms of the other $\boldsymbol{\ell}_k$'s as $\boldsymbol{\ell}_j = \sum_{k\neq \ell} c_k \boldsymbol{\ell}_k$

Facts

When $\mathbf{A} = \sum_{1\leq j\leq s} \boldsymbol{\ell}_j \mathbf{r}_j'$ has rank s, then

- $\boldsymbol{\ell}_j \neq \mathbf{0}$, $\mathbf{r}_j \neq \mathbf{0}$, j = 1, ..., s
- $\{\boldsymbol{\ell}_1, \boldsymbol{\ell}_2, ..., \boldsymbol{\ell}_s\}$ are linearly independent
- $\{\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_s\}$ are linearly independent

Conversely, when $\{\boldsymbol{\ell}_1, \boldsymbol{\ell}_2, ..., \boldsymbol{\ell}_s\}$ and $\{\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_s\}$ are linearly independent, then $\mathbf{A} = \sum_{1\leq j\leq s} \boldsymbol{\ell}_j \mathbf{r}_j'$ has rank s

An *important* consequence:
When $\{\boldsymbol{\ell}_j\}$ and $\{\mathbf{r}_j\}$ are linearly independent

$\mathbf{B} = \sum_{1\leq j\leq s} \lambda_j \boldsymbol{\ell}_j \mathbf{r}_j'$ has rank $s^* < s$ if and only if $\lambda_j \neq 0$ for exactly s^* of the λ's

Vocabulary

m × n matrix $\mathbf{A}$ has *full rank* when rank($\mathbf{A}$) = min(m,n)

Determinant of a matrix

If $\mathbf{A} = [a_{ij}]$ is a p × p (square) matrix, its *determinant* is

$$\det(\mathbf{A}) = \sum_{\{j_1 j_2 \ldots j_p\}} \pm a_{1j_1} a_{2j_2} \ldots a_{pj_p},$$

a sum of p! = p×(p-1)×...3×2×1 products. Each product has one element from each row and from each column.

Sometimes det($\mathbf{A}$) is notated $|\mathbf{A}|$.

- When p = 2, $\det(\mathbf{A}) = a_{11}a_{22} - a_{12}a_{21}$

For any p,

- $\det(\text{diag}[a_1, a_2, ..., a_p]) = a_1 \times a_2 \times ... \times a_p$
- $\det(\mathbf{I}_m) = 1 \times 1 \times ... \times 1 = 1$
- $\det(\mathbf{AB}) = \det(\mathbf{BA}) = \det(\mathbf{A}) \times \det(\mathbf{B})$, when $\mathbf{A}$ and $\mathbf{B}$ are square and the same size
- $\det(\mathbf{A}') = \det(\mathbf{A})$ (determinant of transpose = determinant of matrix)

In MacAnova use `det(a)`.

```
Cmd> a <- matrix(vector(17,3, 2,-1),2); a
(1,1)          17          2    2 by 2 matrix
(2,1)           3         -1    det = a11 a22 - a12 a21
Cmd> det(a)
(1)          -23
Cmd> a[1,1]*a[2,2] - a[1,2]*a[2,1] # a_11*a_22 - a_12*a_21
(1,1)          -23
```

Trace of a matrix

The *trace* of a square matrix is the sum of its diagonal elements

$$\text{tr}(\mathbf{A}) = \text{trace}(\mathbf{A}) \equiv \sum_{1\leq i\leq p} a_{ii}$$

- p = 2: $\text{trace}(\mathbf{A}) = a_{11} + a_{22}$
- p = 3: $\text{trace}(\mathbf{A}) = a_{11} + a_{22} + a_{33}$
- When $\mathbf{A}$ and $\mathbf{B}$ are the same size, $\text{trace}(\mathbf{A} + \mathbf{B}) = \text{trace}(\mathbf{A}) + \text{trace}(\mathbf{B})$
- When $\mathbf{A}$ is p×q and $\mathbf{B}$ q×p, $\mathbf{A}\,\mathbf{B}$ and $\mathbf{B}\,\mathbf{A}$ are defined and square and $$\text{trace}(\mathbf{A}\,\mathbf{B}) = \text{trace}(\mathbf{B}\,\mathbf{A})$$
- $\text{trace}(\mathbf{A}'\mathbf{A}) = \text{trace}(\mathbf{AA}') = \sum_i \sum_j a_{ij}^2$, the sum of squares of all the elements of $\mathbf{A}$

The last is useful if you are trying to find a matrix $\hat{\mathbf{A}}$ of some type that is close to $\mathbf{A}$ in the least squares sense, that is you want to minimize

$$\sum_i \sum_j (\hat{a}_{ij} - a_{ij})^2 = \text{trace}((\hat{\mathbf{A}} - \mathbf{A})'(\hat{\mathbf{A}} - \mathbf{A}))$$

MacAnova: Use `trace(a)`.

```
Cmd> deviations # previously defined matrix
(1,1)      -0.366        0.421        0.336      -0.764
(2,1)      -1.407        0.284        0.919       0.762
(3,1)       0.489       -0.173       -1.039       2.405

Cmd> trace(deviations' %*% deviations)
(1)       11.626

Cmd> trace(deviations %*% deviations')
(1)       11.626

Cmd> sum(vector(deviations)^2) #
(1)       11.626     Sum of squares of elements
```

`vector(deviations)` unravels the matrix `deviations` into a long vector.

```
Cmd> trace(deviations) # illegal
ERROR: argument to trace() not REAL square matrix
```

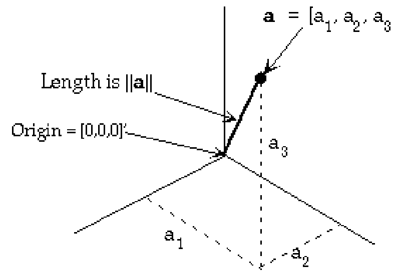
You could also use `sum(diag(a))` instead of `trace(a)`.

Some vocabulary and facts

The *length* or *norm* of a (column) vector $\mathbf{a} = [a_1\ a_2\ \ldots.\ a_p]'$:

$$||\mathbf{a}|| = \sqrt{(\mathbf{a}'\mathbf{a})} = \sqrt{\{\sum_{1 \le i \le p} a_i^2\}}$$

If $\mathbf{a}$ represents a point in p-dimensional space, $||\mathbf{a}||$ is the Euclidean distance of $\mathbf{a}$ from the origin $\mathbf{0} = [0,0,..0]'$.



p = 3

$||\mathbf{a}||$ is length of heavy line

MacAnova "Length" here is different from `length(a)` = number of elements in **a**.

Simple macro to compute $||\mathbf{a}||$

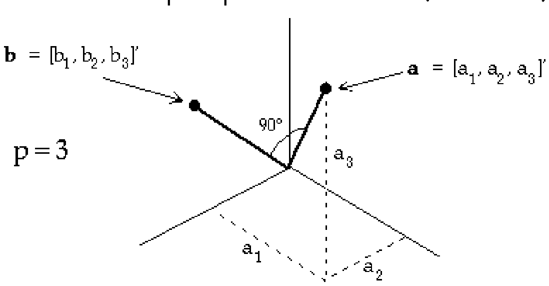
```
Cmd> norm <- macro("sqrt(sum(($1)^2))")

Cmd> z <- vector(1.1, -2.3, 4.5)

Cmd> norm(z) # compute sqrt((1.1)^2 + (-2.3)^2 + (4.5)^2)
(1)        5.172
```


Vocabulary

Vectors $\mathbf{a} \neq \mathbf{0}$ and $\mathbf{b} \neq \mathbf{0}$ with the same dimension p are *orthogonal* (*perpendicular*) when their inner product is 0:

$$\mathbf{a}'\mathbf{b} = \sum_{1 \le i \le p} a_i b_i = 0.$$

If $\mathbf{a}$ and $\mathbf{b}$ represent points in p-dimensional space, the lines from the origin to $\mathbf{a}$ and to $\mathbf{b}$ are perpendicular (at 90°).



The *angle* θ between two vectors $\mathbf{a}$ and $\mathbf{b}$ is defined by

$$\cos\theta = \mathbf{a}'\mathbf{b}/(||\mathbf{a}|| \times ||\mathbf{b}||)$$
$$= \sum_i a_i b_i / \sqrt{(\sum_i a_i^2 \sum_j b_j^2)}$$

So $\mathbf{a}'\mathbf{b} = 0 \Rightarrow \cos\theta = 0 \Rightarrow \theta = \pm 90°$

Suppose $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_r$ with all $\mathbf{u}_i \neq 0$, are *mutually orthogonal* ($\mathbf{u}_i'\mathbf{u}_j = 0,\ i \neq j$).

Then these **facts** are true:

- $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_r$ are linearly independent
- $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_r]$ (p×r) has rank r.
- $\mathbf{U}'\mathbf{U} = \text{diag}(||\mathbf{u}_1||^2, \ldots, ||\mathbf{u}_r||^2)$ is diagonal

This last is really the definition of mutual orthogonality.

Angles and correlation coefficients

The sample correlation between two variables $\{x_i\}_{1\le i\le n}$ and $\{y_i\}_{1\le i\le n}$ is

$$r_{xy} = \frac{\sum_{1\le i\le n}(x_i-\overline{x})(y_i-\overline{y})}{\sqrt{\{\sum_{1\le i\le n}(x_i-\overline{x})^2\sum_{1\le i\le n}(y_i-\overline{y})^2\}}} = \cos\theta_{xy}$$

You can interpret $-1 \le r_{xy} \le 1$ as the cosine of the angle θ_{xy} between the n-vectors of deviations from the mean.

$$\tilde{\mathbf{X}} = [x_1 - \overline{x}, \ldots, x_n - \overline{x}]'$$

and

$$\tilde{\mathbf{Y}} = [y_1 - \overline{y}, \ldots, y_n - \overline{y}]'$$

When $r_{xy} \cong 1$, $\theta_{xy} \cong 0$ so $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{Y}}$ point in almost the *same* direction.

When $r_{xy} \cong -1$, $\theta_{xy} \cong \pm 180^\circ$ so $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{Y}}$ point in almost the *opposite* direction.

Inverse of a Matrix

Vocabulary:

The *inverse* of a p by p *square* matrix $\mathbf{A}$ is the matrix $\mathbf{A}^{-1}$ (if one exists) such that

- $\mathbf{A}\mathbf{A}^{-1} = \mathbf{I}_p = \mathbf{A}^{-1}\mathbf{A}$.

There is *at most* one such matrix.

Example:

$$\mathbf{A} = \begin{bmatrix} 7 & 2 \\ 4 & 4 \end{bmatrix}, \quad \mathbf{A}^{-1} = \begin{bmatrix} 1/5 & -1/10 \\ -1/5 & 7/20 \end{bmatrix}$$

```
Cmd> a <- matrix(vector(7,4,2,4),2);a
(1,1)           7           2
(2,1)           4           4
Cmd> ainv <- matrix(vector(1/5,-1/5, -1/10,7/20),2); ainv
(1,1)         0.2        -0.1
(2,1)        -0.2        0.35
Cmd> a %*% ainv # 2 by 2 identity
(1,1)           1 -1.1102e-16      -1.1102e-16 ~ 0
(2,1)           0           1
Cmd> ainv %*% a # 2 by 2 identity
(1,1)           1           0
(2,1) -2.2204e-16           1
Cmd> solve(a) # solve() computes inverse
(1,1)         0.2        -0.1
(2,1)        -0.2        0.35
```

Note `a %*% ainv` and `ainv %*% a` aren't *exactly* $\mathbf{I}_2$ because of rounding error.

Here are two matrices with *no* inverse:

$$\mathbf{B} = \begin{bmatrix} 1 & 3 \\ 2 & 6 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

```
Cmd> b <- matrix(vector(1,3, 2,6),2); b
(1,1)           1           2
(2,1)           3           6
Cmd> solve(b) # try to find inverse
ERROR: argument to solve() is apparently singular
Cmd> c <- matrix(vector(1,0, 0,0),2); c
(1,1)           1           0
(2,1)           0           0
Cmd> solve(c) # try to find inverse
ERROR: argument to solve() is singular
```

Vocabulary

- When $\mathbf{A}^{-1}$ exists, $\mathbf{A}$ is *invertible* or *non-singular*
- $\mathbf{A}^{-1}$ does not exist $\Rightarrow$ $\mathbf{A}$ is *singular*

More "facts" when $\mathbf{A}$ is non-singular:

- $\mathbf{A}^{-1}\mathbf{A} = \mathbf{A}\mathbf{A}^{-1} = \mathbf{I}_p$ (definition of $\mathbf{A}^{-1}$)
- $(\mathbf{A}^{-1})^{-1} = \mathbf{A}$ (inverse of $\mathbf{A}^{-1}$ is $\mathbf{A}$)
- $(\mathbf{A}')^{-1} = (\mathbf{A}^{-1})'$ (transpose of inverse is inverse of transpose)
- $\mathbf{A}$ and $\mathbf{B}$ non-singular $\Rightarrow$ $(\mathbf{A}\mathbf{B})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$

Using vectors and matrices to represent data

Univariate Data (p = 1)

A *univariate* data set consists of n observations $x_1, \ldots, x_n$ on p = 1 variable.

You represent it by the column vector of length n (n×1 matrix)

$$\mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \\ \ldots \\ x_n \end{bmatrix} = [x_1, x_2, \ldots, x_n]', \quad n \text{ by } 1$$

You *could* represent the data by the *row* vector

$$[x_1, x_2, \ldots, x_n].$$

but that's not the convention we use.

We use the column vector form *only*.

The sum of the data is $\mathbf{1}_n'\mathbf{X} = \sum_{1\le i\le n} x_i$.

Multivariate Data (p > 1)

Suppose your data are n *multivariate* observations $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ (p by 1 vectors), with

$$\mathbf{x}_i = [x_{i1}, x_{i2}, \ldots, x_{ip}]'.$$

You can represent all the data by the n × p **data matrix**

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1' \\ \mathbf{x}_2' \\ \ldots \\ \mathbf{x}_n' \end{bmatrix} = [\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_p] = [x_{ij}]_{1\leq i\leq n, 1\leq j\leq p},$$

- Column vector $\mathbf{X}_j$ = all the data on *variable* j.
- Row vector $\mathbf{x}_i' = [x_{i1}, x_{i2}, \ldots, x_{ip}]$ = all the data on *case* i, expressed as the transpose of the column vector $\mathbf{x}_i$.

Sums of squares and products

Suppose $\mathbf{X} = [\mathbf{X}_1, \ldots, \mathbf{X}_p]$ is n by p, then

- The *diagonal elements of*
  $$\mathbf{X}'\mathbf{X} = [\mathbf{X}_j'\mathbf{X}_k]_{1\leq j\leq p, 1\leq k\leq p}$$
  are *sums of squares* $\sum_{1\leq i\leq n} x_{ij}^2 = \mathbf{X}_j'\mathbf{X}_j$
- The *off diagonal* elements of $\mathbf{X}'\mathbf{X}$ are *sums of products* $\sum_{1\leq i\leq n} x_{ij}x_{ik} = \mathbf{X}_j'\mathbf{X}_k$, $j \neq k$

$\mathbf{X}'\mathbf{X}$ is also a *sum of outer products* $\mathbf{x}_i\mathbf{x}_i'$

$$\mathbf{X}'\mathbf{X} = \sum_{1\leq i\leq n} \mathbf{x}_i\mathbf{x}_i', \ \mathbf{x}_i' \text{ a row of } \mathbf{X}.$$

When the data are univariate, $\mathbf{X}$ is n by 1 and $\mathbf{X}'\mathbf{X} = \sum_{1\leq i\leq n} x_i^2$.

Important mnemonic

A square x_i^2 in a univariate formula often becomes an outer product $\mathbf{x}_i\mathbf{x}_i'$ in a related multivariate formula.

$$\sum_{1\leq i\leq n} x_i^2 \Rightarrow \sum_{1\leq i\leq n} \mathbf{x}_i\mathbf{x}_i'$$

MacAnova

```
Cmd> x <- matrix(vector(2.4,12.3,10.6,15.1,-1.3, \
        22.9,15.7,15.7,17.2,22.5,  44,32.7,35.2,33.5,26.7), 5)

Cmd> x # n = 5; p = 3
(1,1)         2.4         22.9          44
(2,1)        12.3         15.7        32.7
(3,1)        10.6         15.7        35.2         x
(4,1)        15.1         17.2        33.5
(5,1)        -1.3         22.5        26.7

Cmd> xx <- x' %*% x; xx # p by p  (3 by 3)
(1,1)      499.11      644.96      1352.1
(2,1)      644.96      1819.5      3250.6         x'x
(3,1)      1352.1      3250.6      6079.5

Cmd> sum(x[,1]*x[,3]) # summed products of cols 1 and 3
(1,1)      1352.1           xx[1,3]
```

Use loop to compute $\mathbf{X}'\mathbf{X}$ as a sum of outer products $\sum_{1\leq i\leq n} \mathbf{x}_i\mathbf{x}_i'$:

```
Cmd> xx1 <- dmat(3,0) # 3 by 3 starting matrix of 0's

Cmd> n <- nrows(x) # sample size

Cmd> for(i,1,n){ # X'X as sum of outer products
        @xi <- vector(x[i,]) # column i of x
        xx1 <- xx1 + outer(@xi,@xi)
    ;;} # ";;" prevents extraneous output

Cmd> xx1 # same as xx
(1,1)      499.11      644.96      1352.1
(2,1)      644.96      1819.5      3250.6
(3,1)      1352.1      3250.6      6079.5
```

MacAnova

The ";;" before the final "}" prevents printing each time through the loop. `@xi` is a temporary variable that is automatically deleted after the loop.

Statistical application of matrix formulas

Suppose $\mathbf{X} = \begin{bmatrix} \mathbf{x}_1' \\ \mathbf{x}_2' \\ \ldots \\ \mathbf{x}_n' \end{bmatrix} = [\mathbf{X}_1\ \mathbf{X}_2\ \ldots\ \mathbf{X}_p]$ is a *data matrix* containing n cases of p variables.

The *sample mean vector* is

$$\overline{\mathbf{x}} = (1/n)\sum_{1\leq i\leq n} \mathbf{x}_i = \begin{bmatrix} \overline{x_1} \\ \overline{x_2} \\ \ldots \\ \overline{x_p} \end{bmatrix},$$

where

$$\overline{x_j} = (1/n)\sum_{1\leq i\leq n} x_{ij} = (1/n)\mathbf{1}_n'\mathbf{X}_j \quad j = 1, \ldots, p,$$

is a *univariate* average.

The *row vector* $\overline{\mathbf{x}}' = (1/n)\mathbf{1}_n'\mathbf{X}$

```
Cmd> xbar <- rep(1,n)' %*% x / n
```

This gives the same result as `sum(x)/n`

```
Cmd> equal(xbar,sum(x)/n)
(1) T
```


The **sample variance** (or **covariance** or **variance/covariance**) **matrix** is

$$\mathbf{S} = [s_{ij}] \equiv (n-1)^{-1} \sum_{1\leq i\leq n}(\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})'$$

Compare this with the univariate sample variance $s^2 = (n-1)^{-1} \sum_{1\leq i\leq n}(x_i - \overline{x})^2$.

If $\tilde{\mathbf{X}} = [\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \ldots, \tilde{\mathbf{x}}_n]'$, $\tilde{\mathbf{x}}_i = \mathbf{x}_i - \overline{\mathbf{x}}$, the matrix of deviation of the observations from their sample mean

$$\mathbf{S} = (n-1)^{-1}\tilde{\mathbf{X}}'\tilde{\mathbf{X}} = (n-1)^{-1} \sum_{1\leq i\leq n}\tilde{\mathbf{x}}_i\tilde{\mathbf{x}}_i'.$$

Note: This differs from a similar definition with a divisor of n.

$$\mathbf{S}_n = \sum(\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})'/n$$

As estimators of the population variance matrix $\mathbf{\Sigma}$ (not yet defined), $\mathbf{S}$ is unbiased and $\mathbf{S}_n$ is biased.

- On the *diagonal*, $s_{jj} = (n-1)^{-1}\sum(x_{ij} - \overline{x}_j)^2$ are the usual *sample variances* s_j^2. $\sqrt{s_{ii}}$ = sample *standard deviation*
- The *off-diagonal* elements

  $$s_{jk} = (n-1)^{-1}\sum(x_{ij} - \overline{x}_j)(x_{ik} - \overline{x}_k)$$

  are the *sample covariances*.

The divisor n-1 is the *degrees of freedom*. The n deviations $\mathbf{x}_i - \overline{\mathbf{x}}$ from the mean satisfy one linear equality, namely, $\sum_{1\leq i\leq n}(\mathbf{x}_i - \overline{\mathbf{x}}) = 0$

Important observation:

You can get the multivariate ($p > 1$) formula

$$\mathbf{S} = (n-1)^{-1} \sum_{1\leq i\leq n}(\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})'$$

the univariate ($p = 1$) formula

$$s^2 = (n-1)^{-1}\sum_{1\leq i\leq n}(x_i - \overline{x})^2$$

Replace $(x_i - \overline{x})^2$ by $(\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})'$.