

# シミュレーション基礎 (5)

## 第4章

### 大きなプログラムを作る：関数

# 4.1 関数を作る

## 4.1.1 平均と分散を求める (1)

データ  $x_1, x_2, \dots, x_n$

の平均(average)と分散(variance)を計算するには

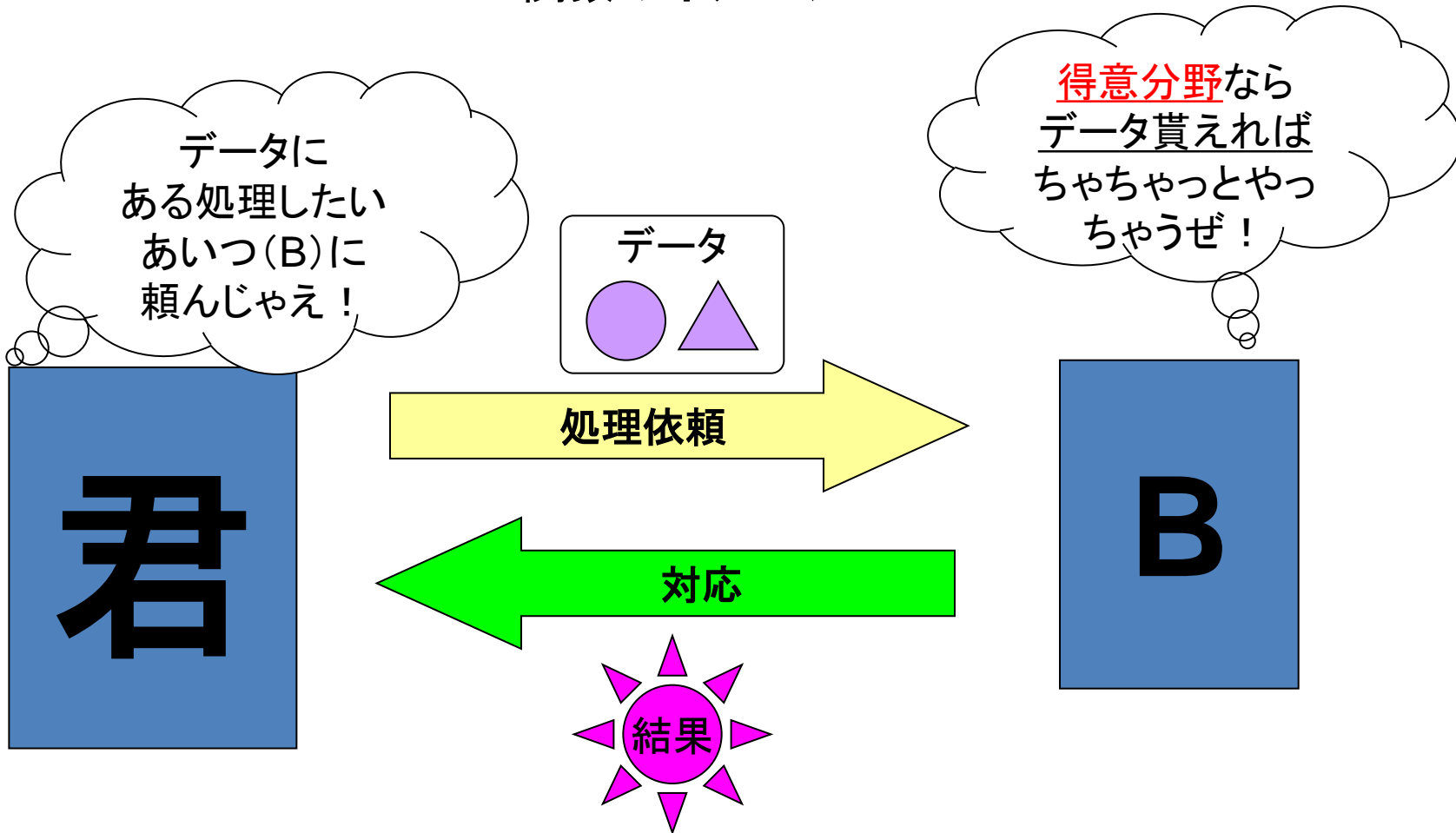
平均： 
$$\mu = \frac{1}{n} \sum_{i=1}^n x_i$$

分散： 
$$\sigma^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2 = \frac{1}{n} \sum_{i=1}^n x_i^2 - \mu^2$$

# 4.1 関数を作る

## 4.1.1 平均と分散を求める (2)

### 関数のイメージ



# 4.1 関数を作る

## 4.1.1 平均と分散を求める (3)

### 関数を使うメリット

- 使い回しがきく (ねじ、歯車…機械部品と同じ)
- 信頼性の高いプログラムの作成  
(関数はなるべく単機能に)
- チームで分担作業による作成ができる

# 4.1 関数を作る

## 4.1.1 平均と分散を求める (4)

利用する (EX4101a.mで保存)

```
clear all;  
N=10;  
X=1:N;  
Y=AvrAndVar(X)
```

関数 (AvrAndVar.mで保存)

```
function Out=AvrAndVar(X)  
N=length(X);  
Avr=sum(X)./N;  
Var=sum(X.^2)./N-Avr.^2;  
Out=[Avr Var];
```

# 4.1 関数を作る

## 4.1.1 平均と分散を求める (5)

1行目: 関数定義行  
**function**から**必ず**始まる

**AvrAndVar**は関数名  
自分で好きなニックネームをつける

```
function Out=AvrAndVar(X)
N=length(X);
Avr=sum(X)./N;
Var=sum(X.^2)./N-Avr.^2;
Out=[Avr Var];
```

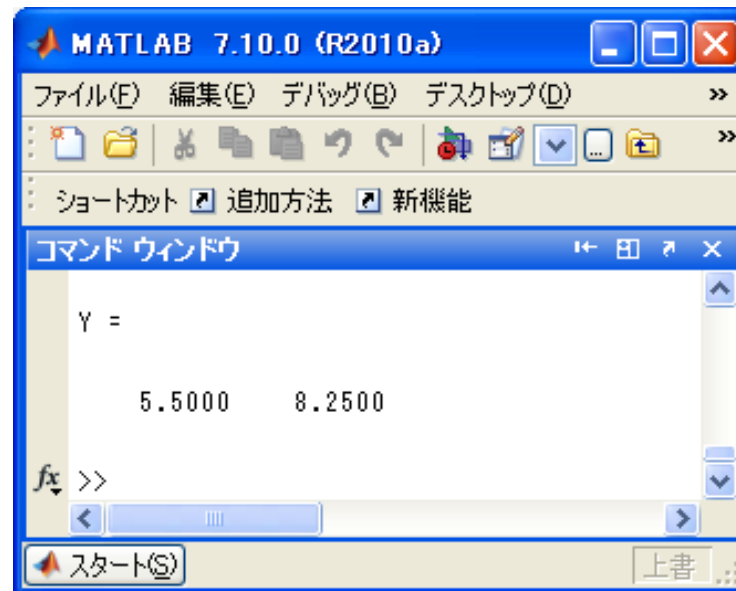
入力引数(名前は自分で)  
外部から与えられるデータを関数  
内でどう呼ぶか決める。  
複数個ある場合にはカンマ区切り  
で()内に書く

**Out**は出力引数(名前は自分で決める)。  
複数個ある場合には  
カンマで区切り、「[]と「]」で囲む

# 4.1 関数を作る

## 4.1.1 平均と分散を求める (6)

実行すると



The image shows a screenshot of the MATLAB 7.10.0 (R2010a) Command Window. The window title is "MATLAB 7.10.0 (R2010a)". The menu bar includes "ファイル(F)", "編集(E)", "デバッグ(B)", and "デスクトップ(D)". The toolbar contains various icons for file operations and execution. The Command Window itself displays the output of a function call, showing "Y =" followed by two columns of values: "5.5000" and "8.2500". The prompt "fx >>" is visible at the bottom of the Command Window. The Windows taskbar at the bottom shows the "スタート(S)" button and the "上書" (Save) button.

```
Y =  
    5.5000    8.2500  
fx >>
```

# 4.1 関数を作る

## 4.1.2 エラーに対処する (1)

データ数0の場合にエラーメッセージとreturnを使うよう改造

```
function Out=AvrAndVar3(X)
```

```
N=length(X);
```

```
if N==0
```

```
    Out=[ ]; disp( 'Error: There is no data in X' )
```

```
    return
```

```
end
```

```
Avr=sum(X)./N;
```

```
Var=sum(X.^2)./N-Avr.^2;
```

```
Out=[Avr Var];
```

returnを使うと、関数を抜け  
メインに戻り、メイン関数は  
その続きを実行する。



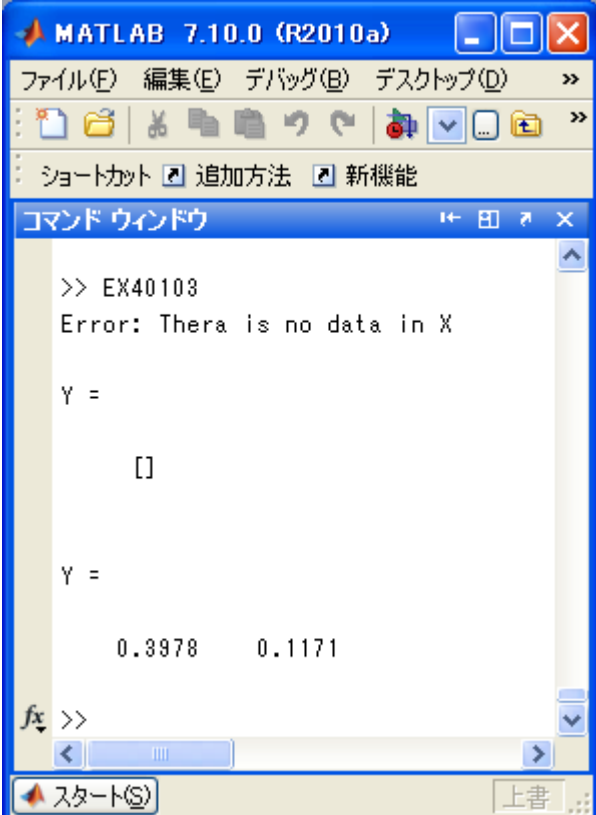
# 4.1 関数を作る

## エラーに対処する (2)

例) mainプログラム: ex4102.m

```
clear all;  
X=[];  
Y=AvrAndVar3(X)
```

エラーになるはず



The screenshot shows the MATLAB Command Window with the following text:

```
>> EX40103  
Error: There is no data in X  
  
Y =  
  
    []  
  
Y =  
  
    0.3978    0.1171
```

The window title is "MATLAB 7.10.0 (R2010a)" and the menu bar includes "ファイル(F)", "編集(E)", "デバッグ(D)", and "デスクトップ(D)". The Command Window title is "コマンド ウィンドウ". The status bar at the bottom shows "スタート(S)" and "上書".

# 4.1 関数を作る

エラーに対処する (3)

エラーメッセージと **error** を使う

```
function Out=AvrAndVar3(X)
```

```
N=length(X);
```

```
if N==0
```

```
    error( 'Error: Thera is no data in X!' );
```

```
end
```

```
Avr=sum(X)./N;
```

```
Var=sum(X.^2)./N-Avr.^2;
```

```
Out=[Avr Var];
```

**error**を使うとエラーメッセージが表示され、計算が中断される

# 4.1 関数を作る

## 4.1.3 引数は値が渡される？

メインプログラムが関数を呼び出すとき

**記憶領域と転送時間を節約するため**

原則、引数の値そのものではなく、アドレスが関数に渡される。

しかし、関数の中で変更を受ける変数については値が渡される

# 関数を作る

## 引数は値が渡される？ (2)

Test1

```
function Y=Test1(X)
```

```
Y=sum(X);
```

```
X=2.*X
```

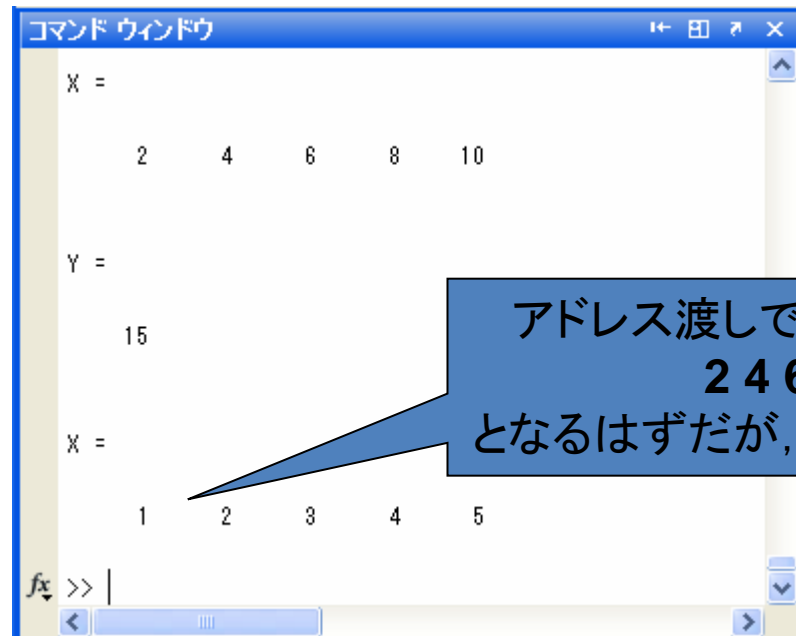
ex4103

```
clear all;
```

```
X=[1:5];
```

```
Y=Test1(X)
```

```
X
```



コマンドウィンドウ

```
X =  
    2    4    6    8   10  
  
Y =  
    15  
  
X =  
    1    2    3    4    5
```

fx >> |

アドレス渡しであればこのXは  
**2 4 6 8 10**  
となるはずだが、そうになっていない

# 4.2 引数操る

## 4.2.1 複数の出力引数

出力引数が複数にしたいときは [変数1,変数2, ...]=

**AvrAndVar4**

```
function [Avr,Var]=AvrAndVar4(X)
```

```
·  
N=length(X);  
if N==0  
    error( 'Error: There is no data in X' );  
end;  
Avr=sum(X)./N;  
Var=sum(X.^2)./N-Avr.^2;
```

関数の出力が複数ある場合には  
[ ]で括り, 各引数を「,」で区切る

呼ぶ側も [ ]を用いる

**EX4201**

```
clear all;  
N=10;  
X=1:N; %Data  
[Avr,Var]=...  
    AvrAndVar4(X)
```

```
>> [Avr,Var]=AvrAndVar4(X);  
>> Avr,Var  
  
Avr =  
  
    5.5000  
  
Var =  
  
    8.2500  
  
fx >>
```

# 4.2 引数操る

## 4.2.2 出力引数がないとき

functionに続いて関数名と入力引数だけになる

### AvrAndVar5

```
function AvrAndVar5(X)
N=length(X);
AVR=sum(X)./N;
VAR=sum(X.^2)./N-AVR.^2;
S=sprintf('Avrage=%6.3f Variance=%6.3f,AVR,VAR)
disp(S)
```

数値をフォーマット付きで文字列  
に変換する関数



### Ex4202.m

```
N=10;X=1:N;
AvrAndVar5(X);
```

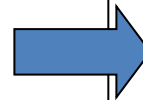
# 4.2 引数を操る

## 4.2.3 入力引数がないとき

関数名の後ろは空()

### AvrAndVar6

```
function [Avr,Var]=AvrAndVar6()  
N=10; X=1:N;  
Avr=sum(X)./N;  
Var=sum(X.^2)./N-AVR.^2;
```



### Ex4203.m

```
[Avr Var]=AvrAndVar6();  
Avr,Var
```

# 4.2 引数を操る

## 4.2.4 出力引数も入力引数もないとき

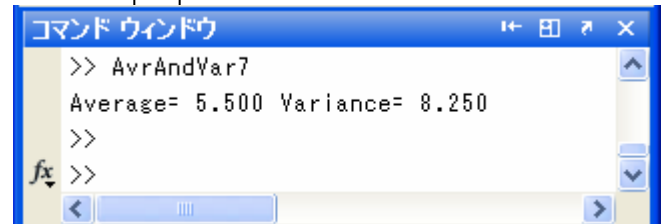
global変数: プログラム間で変数を共有する

```
AvrAndVar8  
function AvrAndVar8
```

```
global X AVR VAR  
N=length(X);  
AVR=sum(X)./N;  
VAR=sum(X.^2)./N-AVR.^2;
```

global変数の宣言  
通常**大文字**で書く習慣

```
ex204b  
clear all;  
global X VAR VAR  
N=10;X=1:N;  
AvrAndVar8;  
AVR,VAR
```



```
コマンド ウィンドウ  
>> AvrAndVar7  
Average= 5.500 Variance= 8.250  
>>  
fx >>
```



# 4.2 引数を操る

## 4.2.5 入力引数の数を可変にする

引数の数を可変にできる. これによって  
X または XおよびYの両方  
の平均と分散を計算する関数を作ることができる

nargin: 引数の数を返す関数

**AvrAndVar9**

```
function [Avr,Var]=AvrAndVar9(X,Y)
if nargin==1
    N=length(X); Z=X;
else
    N=length(X)+length(Y); Z=[X Y];
end
Avr=sum(Z)./N;
Var=sum(Z.^2)./N-Avr.^2;
```



**EX40206**

```
clear all;
N=10;X=1:N;
Y=2.*X;
[Avr Var]=...
AvrAndVar9(X);
Avr,Var
[Avr Var]=...
AvrAndVar9(X,Y);
Avr, Var
```

# 4.2 引数を操る

## 4.2.5 入力引数の数を可変にする

`varargin{ }`: 入力引数を配列として受け取る

カッコが `{ }` であることに注意

**AvrAndVar10**

```
function [Avr,Var]=AvrAndVar10(varargin)
```

```
Z=[];
```

入力変数の数を確認する

```
for l=1:length(varargin);
```

```
    Z=[Z varargin{l}];
```

入力引数の数がわからないときは" varargin" とする

```
end
```

```
N=length(Z);
```

Zに次々に新しいベクトルを追加していく

```
Avr=sum(Z)./N; Var=sum(Z.^2)./N-Avr.^2;
```

**EX40207**

```
clear all;N=10;X=1:N;
```

```
Y=2.*X;Z=3.*X;
```

```
[Avr Var]=...
```

```
AvrAndVar10(X);
```

```
Avr, Var
```

```
[Avr Var]=...
```

```
AvrAndVar10(X,Y);
```

```
Avr, Var
```

```
[Avr Var]=...
```

```
AvrAndVar10(X,Y,Z);
```

```
Avr, Var
```

# 4.2 引数を操る

## 4.2.6 メインプログラムに関数を書く

演算子@で関数を定義する

```
Average=@(X,N) sum(X)/N;
```

```
Variance=@(X,N) sum(X.^2)/N-Average(X,N)^2;
```

```
N=100; X=1:N;
```

```
Average(X,N)
```

```
Variance(X,N)
```

最後に

lookfor キーワード と help 関数名 のために

```
function Out=AvrAndVar(X)
```

```
% AvrAndVar Average and Variance of a vector X  
%Out=[Avr Var]; Avr and Var are respectively  
%the average and the variance of a given vector X.
```

```
N=length(X);
```

```
Avr=sum(X)./N;
```

```
Var=sum(X.^2)./N-Avr.^2;
```

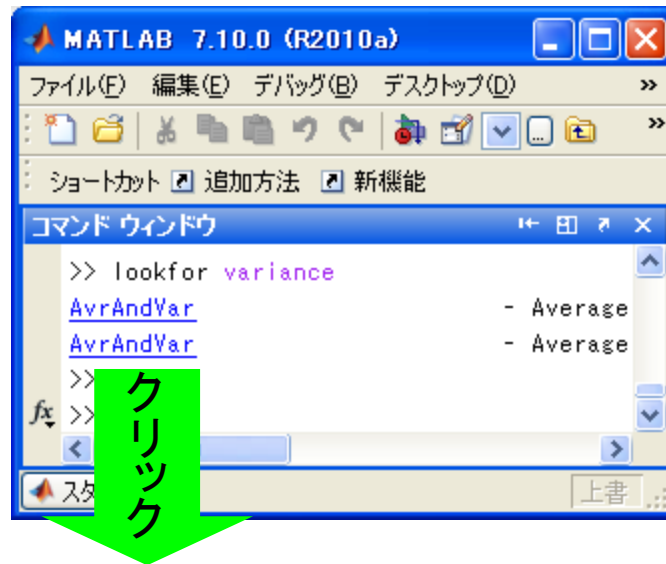
```
Out=[Avr Var];
```

関数直下のコメント行  
なにをする関数か

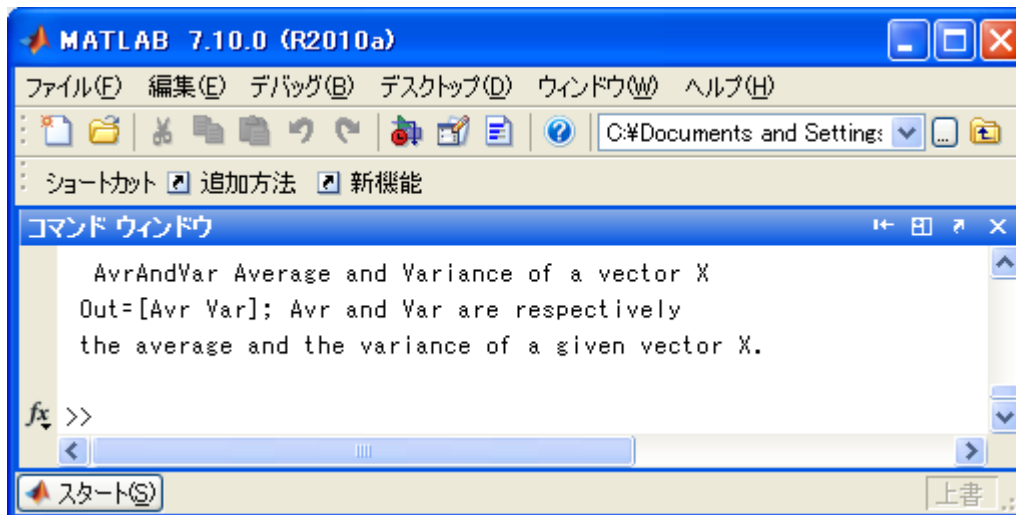
2, 3行目のコメント行  
引数の意味など

# lookfor variance

## とコマンドを打つと

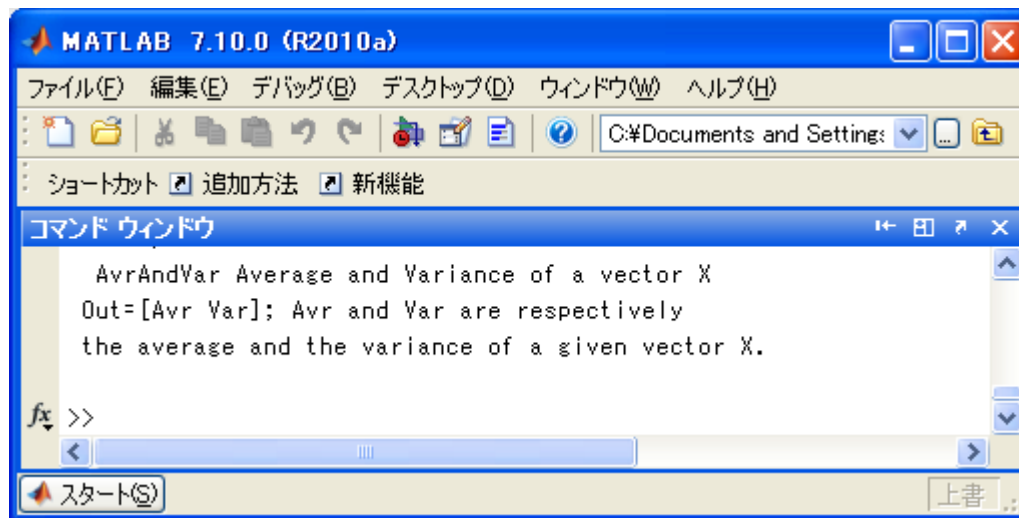


H1行に「variance」と書かれているので、検索でひっかかる



H1行とヘルプテキスト行が表示

# help AvrAndVar とコマンドを打つと



H1行とヘルプテキスト行  
が表示

# 練習

- $\text{ones}(X, Y)$  は全ての項が 1 となる行列を返す関数である。引数が  $X$  のみの場合は大きさ  $X$  の正方行列を,  $X, Y$  の場合は  $X$  行  $Y$  列の行列を返す。

同じ機能の関数  $\text{ones2}$  を自作しなさい

