TECHNISCHE UNIVERSITÄT DRESDEN

Introduction to Matlab

Matrices, Random Numbers, Plotting

Pouyan R. Fard
Prof. Dr. Stefan J. Kiebel

Dresden, 27.04.2017

DRESDEN concept
Exzellenz aus Wissenschaft und Kultur

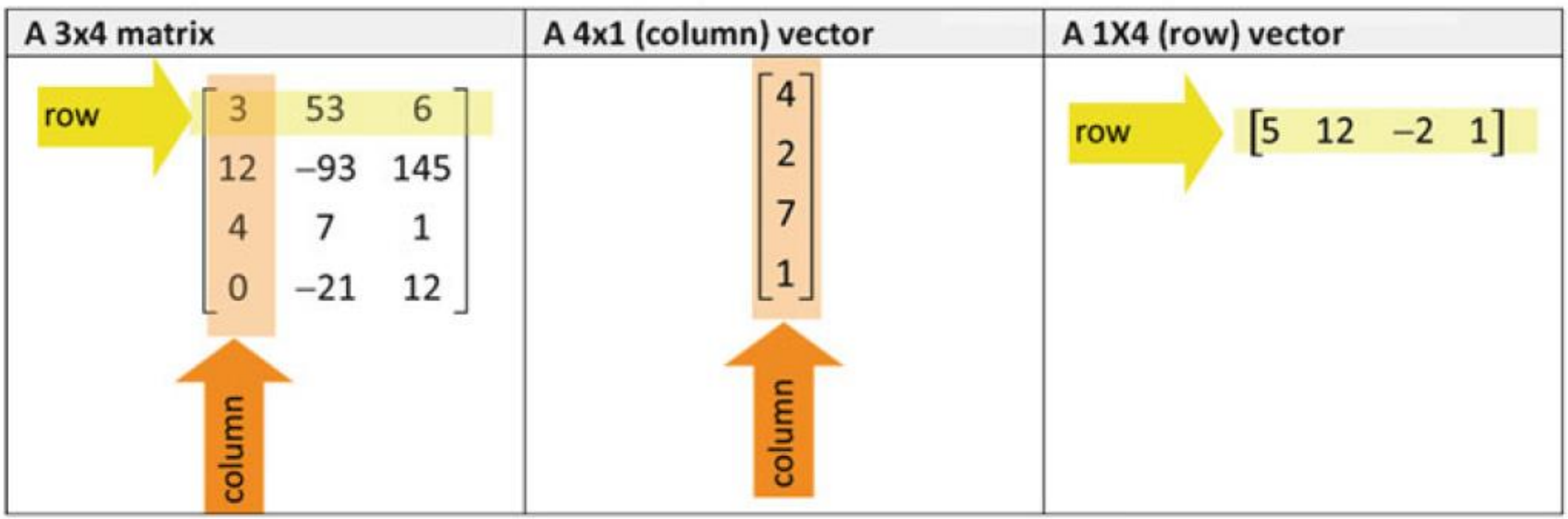
TECHNISCHE UNIVERSITÄT DRESDEN

Seminar overview

Date	Topics	Projects
06.04.	Intro, basic operations, First Exercise (Morse Code)	
13.04.	First Exercise Continued(Morse Code)	
27.04.	Second Exercise (Game of Life)	
04.05.	Second Exercise Continued (Game of Life)	
05.05.	Third Exercise (Drift-Diffusion Model)	Project Distribution
11.05.	Third Exercise Continued (Drift-Diffusion Model)	
30.05.	-	Project Deadline (6 PM)

Matrices and Vectors

- Thinking in a matrix way



Matrices and Vectors

- Initializing vectors and matrices in MATLAB

```
>> a=[3,5,7,8]                    <ENTER>
a =
      5    12    -2    1
>> b=[4;2;7;1]                    <ENTER>
b =
      4
      2
      7
      1
>> c=[3, 53,6;12,-93,145;4,7,1;0,-21,12]        <ENTER>
c =
      3   53       6
     12   -93     145
      4   7        1
      0   -21      12
```

Matrices and Vectors

- Size of matrices

```
>> size(c)
ans =
      4    3
```

```
>> length(c)
ans=
      4
```

- Usage of spaces, commas, and semi-colons

```
>> x=[ 1 2 3; 2 5 7]
x =
       1      2      3
       2      5      7
```

- Dimensions must be consistent.

```
>> x = [2 3; 2 5 7];
??? Error using ==> vertcat
CAT arguments dimensions are not consistent.
```

Matrices and Vectors

Mathematical representation	MATLAB (type after the prompt >> followed by Enter)	Dimension
$M = \begin{bmatrix} 3 & 12 & \pi \end{bmatrix}$	`M=[3,12,pi];`	1×3 Row vector

$N = \begin{bmatrix} 3 & 12 & \pi \\ 8 & 9 & 10 \end{bmatrix}$	`N=[3,12,pi; 8,9,10];`
	Or equivalently, if you have already inserted M:
	`N=[M; 8,9,10];`
		2×3 Matrix

$P = \begin{bmatrix} 4 \\ 2 \\ -1 \end{bmatrix}$	P=[4;2;–1]	3×1 Column vector
$Q = \begin{bmatrix} 4 & -4 \\ 2 & -2 \\ -1 & 1 \end{bmatrix}$	Q=[4, –4;2, –2;–1, 1]; Or equivalently, if you have already inserted P: Q=[P;–P];	3×2 Matrix

Initializing Matrices

- Initializing an empty matrix

```
>> y = [ ];                    <ENTER>
>> whos y                      <ENTER>
Name        Size        Bytes        Class        Attributes

y           0x0         0                         double
```

- Initializing an identity matrix with size n — **eye**(*n*)
- Initializing a matrix whose all elements are 1 — **ones**(*n*,*m*)
    - n= number of rows , m= number of columns
- Initializing a matrix whose all elements are 0 — **zeros**(*n*,*m*)
    - n= number of rows , m= number of columns
- Initializing a matrix of random integers — ***magic(n)***
    - n= number of rows and columns

TECHNISCHE UNIVERSITÄT DRESDEN

Indexing

Start:Step:Stop

Type the following commands:

To TYPE after prompt >> followed by Enter	MATLAB answer	Meaning of the operation
2:5:25	`ans =` `2 7 12 17 22`	Generate a vector going from 2 to 25 incremented by 5. Note that 22+5=27, which is greater than 25. MATLAB will generate numbers until it reaches or exceeds the Stop value (i.e., 25)
`i:j`	`ans =` `2 3 4`	Generate a vector going from 2 to 4. Here the step value is not specified, and MATLAB uses the default value 1
`10:-3:-5`	`ans =` `10 7 4 1 -2 -5`	Generate a vector going from 10 to –5, increasing the first value by –5. This is equivalent to generating a vector of decreasing values

Indexing

- Accessing single elements in matrices

```
>> Q(3,2)                    <ENTER>
ans =
     1
```

```
>> Q([1,3],2)                <ENTER>
ans =
    -4
     1
```

- Accessing multiple elements in matrices

```
>> x=[1 2 3; 4 5 6; 7 8 9; 10 11 12; 13 14 15]     <ENTER>
```

```
>> i=2; j=4;                 <ENTER>
>> x(i:j,2)                  <ENTER>
ans =
     5
     8
    11
```

Indexing

- Accessing multiple elements in matrices

```
>> x(3,1:3)                    <ENTER>
ans =
     7     8     9
>> x(3,:)                      <ENTER>
ans =
     7     8     9
```

- Deleting rows or columns of a matrix

```
>> x(:,2) = [ ]                <ENTER>
x =
     1     3
     4     6
     7     9
    10    12
    13    15
```

TECHNISCHE UNIVERSITÄT DRESDEN

Indexing

- Deleting rows or columns of a matrix

```
>> x([3,4],:) =[ ]            <ENTER>
x =
     1     3
     4     6
    13    15
```

- Deleting one single entry in a matrix is not possible!

```
>> x(1,2)=[ ]
??? Subscripted assignment dimension mismatch.
```

- Deleting one single entry is possible in vector.

Matrix Operations

Operation	Definition	Math example	Matlab example

Addition (subtraction)	The result of **A**+**B** or (**A**−**B**) is calculated entrywise, i.e., the element $B_{i,j}$ is added to (subtracted from) the element in $A_{i,j}$	$A=\begin{bmatrix}1 & 5\\2 & 3\end{bmatrix}, B=\begin{bmatrix}2 & 3\\4 & 1\end{bmatrix}$
		$A+B=\begin{bmatrix}3 & 8\\6 & 4\end{bmatrix}$
		$A-B=\begin{bmatrix}-1 & 2\\-2 & 2\end{bmatrix}$
			`>>A=[1,5;2,3];`
			`>>B=[2,3;4,1];`
			`>>A+B`
			`ans =`
			`3 8`
			`6 4`
			`>> A-B`
			`ans =`
			`-1 2`
			`-2 2`

Scalar multiplication	The multiplication of a scalar (= number) *s* by a matrix **C** is obtained by multiplying every entry of **C** by *s*	$C=\begin{bmatrix}3 & 2\\4 & 1\end{bmatrix}, s=4$
		$s\cdot C=\begin{bmatrix}12 & 8\\16 & 4\end{bmatrix}$
			`>>C=[3,2;4,1];`
			`>>s=4;`
			`>>s*C`
			`ans =`
			`12 8`
			`16 4`

Transposition	The transpose of an m×n matrix **D** is an n×m matrix denoted by $\mathbf{D}^T$ obtained by turning rows into columns and columns into rows	$D=\begin{bmatrix}3 & 12 & 2\\8 & 9 & 10\end{bmatrix}$
		$D^T=\begin{bmatrix}3 & 8\\12 & 9\\2 & 10\end{bmatrix}$
			`>> D=[2,12,2;8,9,10];`
			`>> D'`
			`ans =`
			`3 8`
			`12 9`
			`2 10`

TECHNISCHE UNIVERSITÄT DRESDEN

Matrix Operations

- Element-wise addition with a single-element matrix

```
>> p = [1  2;  3  4]
p =
   1   2
   3   4
>> p = p + 2
p =
   3   4
   5   6
```

- When dimensions don’t agree:

```
>> r = [2  1; 1  1; 1  1]
r =
   2   1
   1   1
   1   1
>> n = p + r
??? Error using ==> plus
Matrix dimensions must agree.
```

TECHNISCHE UNIVERSITÄT DRESDEN

Matrix Operations

- Element-wise operations

Description	MATLAB operator	Example

Element-by-element Multiplication	`.*`	`>> A.*B`
		`ans=`
		`2 15`
		`8 3`

Element-by-element Right division	`./`	`>> A./B`
		`ans =`
		`0.5000 1.6667`
		`0.5000 3.0000`

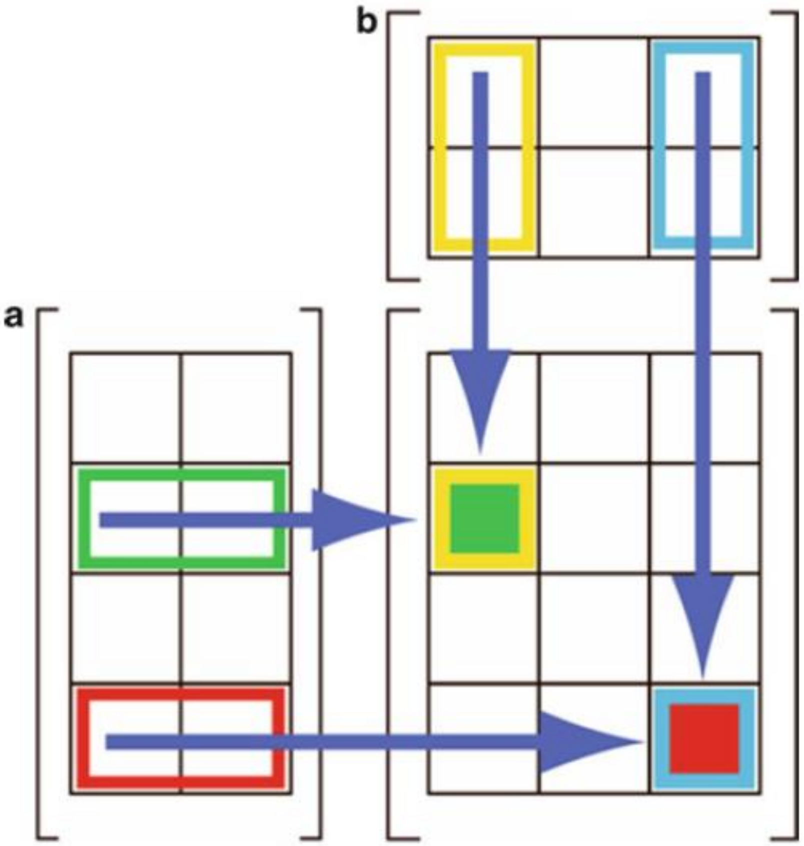
Element-by-element Left division	`.\`	`>> A.\B`
		`ans =`
		`2.0000 0.6000`
		`2.0000 0.3333`

Element-by-element Exponentiation	`.^`	`>> A.^B`
		`ans =`
		`1 125`
		`16 3`

Matrix Multiplication

- A is a $m \times n$ matrix
- B is a $n \times p$ matrix

- $(AB)_{i,j} = \sum_{r=1}^{n} A_{i,r} B_{r,j}$



Matrix Multiplication

```
>> D*C
??? Error using ==> mtimes
Inner matrix dimensions must agree.

>> C*D
ans =
    22    54    26
    16    57    18
```

TECHNISCHE UNIVERSITÄT DRESDEN

Matrix Multiplication

- **Example 1:** Application in linear algebra

$$
\begin{aligned}
x + y + 2z &= 9 \\
2x + 4y - 3z &= 1 \\
3x + 6y - 5z &= 0
\end{aligned}
$$

```
>> A = [1  1  2  9;  2  4  -3  1;  3  6  -5  0]
A =
   1   1    2   9
   2   4   -3   1
   3   6   -5   0
>> rref(A)
ans =
   1   0   0   1
   0   1   0   2
   0   0   1   3
```

Matrix Multiplication

Example 2: Suppose you have five different products in your shop being sold with five different prices, and you sell them in five different quantities. How you can compute your revenue using matrix calculations?

```
>> Prices = [10   20   30   40   50];
>> Sales = [50;   30;   20;   10;   1];
>> Revenue = Prices*Sales
Revenue =
   2150
```

Example: General Linear Model

General Linear Models (GLMs) are widely used to localize brain activity in functional imaging. A standard GLM can be written as:

$$\boldsymbol{Y} = \boldsymbol{X\beta} + \boldsymbol{\varepsilon}$$

Where:

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} = \begin{bmatrix} 1 & X_{11} & \cdots & X_{1p} \\ 1 & X_{21} & & X_{2p} \\ & \vdots & \ddots & \vdots \\ 1 & X_{n1} & \cdots & X_{np} \end{bmatrix} \times \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

BOLD Data

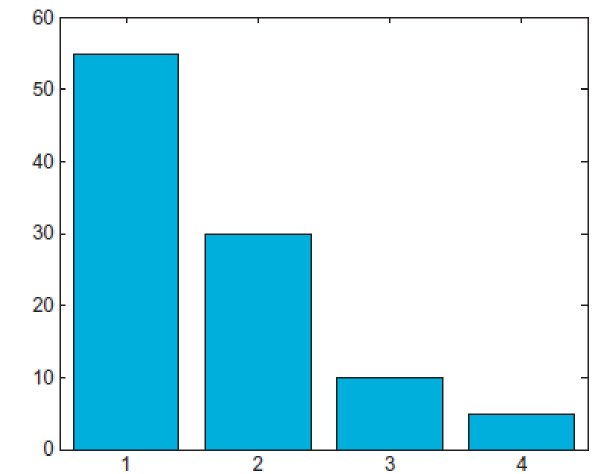
Design Matrix

Regression Coefficients

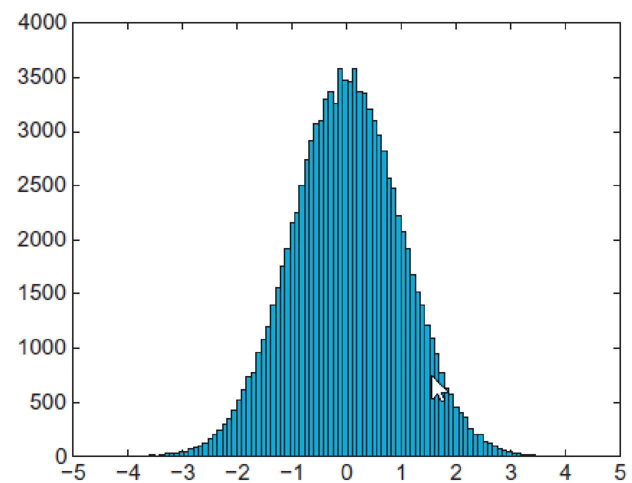
Noise

Bar Plots / Histograms

```
>> results = [55   30   10   5]
results =
   55   30   10   5
>> bar(results)
```




```
>> suspicious = randn(100000,1);
>> figure
>> hist(suspicious, 100)
```



Random Numbers

- Uniform Distribution

```
>> rand(2)

ans =

    0.1334    0.8875
    0.2043    0.3274
```

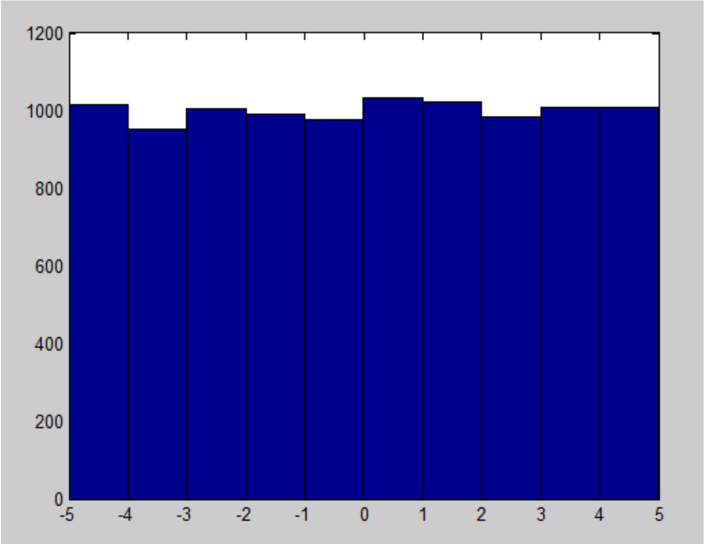
```
>> r = -5 + (5+5)*rand(1,6)

r =

    1.6665    1.0341   -1.2983    0.3298   -1.8825   -3.1169
```

```
>> r = -5 + (5+5)*rand(1,10000);
>> hist(r)
```

```
>> X=rand(size(r))

X =

    0.8594    0.4519    0.5600    0.6888    0.7874    0.7995
```

- Pseudo-random uniformly distributed integers

```
>> r1 = randi(10,1,5)

r1 =

    10     8     8     2     9
```



Random Numbers

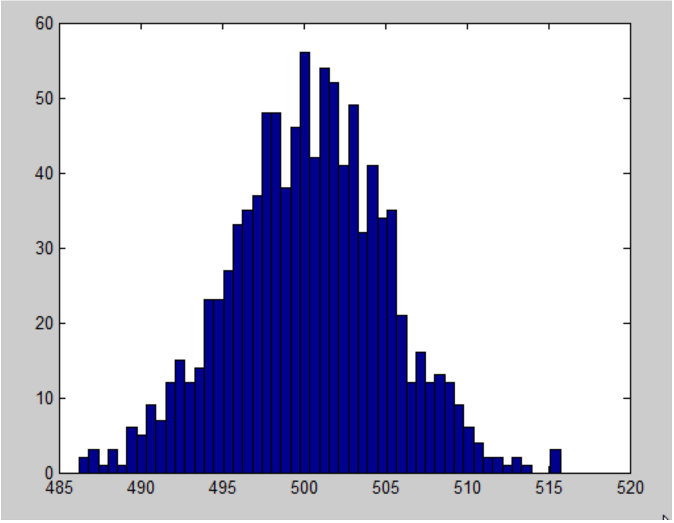
- Normal Distribution

```
>> y=normrnd(5,10,[1,10])

y =

    3.1124    7.1382    3.4948    3.4693   11.4685    3.2389   -0.4196   -0.0679  -15.7383    9.8145
```

```
>> y=normrnd(500,5,[1,1000]);
hist(y,50)
```

```
>> mean(y)

ans =

  500.2999
```

```
>> std(y)

ans =

    4.7836
```



Random Numbers

```
>> x1 = random('Normal',0,1,2,4)

x1 =

   -1.2482   -0.7538    0.9198    0.5396
    0.1622   -0.4963   -0.9311   -0.1406
```

```
>> x2 = random('Poisson',1:6,1,6)

x2 =

     2     1     1     3     6     8
```

- See MATLAB help for my for details.

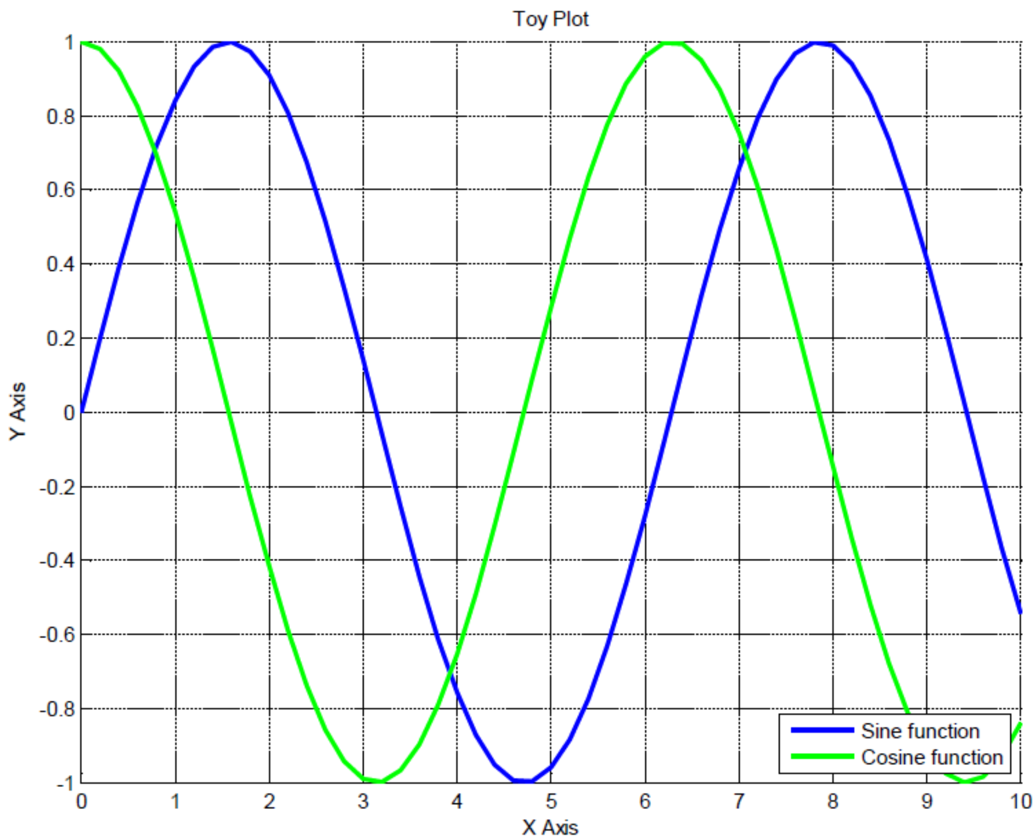
Random Numbers

```
>> x1 = random('Normal',0,1,2,4)

x1 =

   -1.2482   -0.7538    0.9198    0.5396
    0.1622   -0.4963   -0.9311   -0.1406
```

```
>> x2 = random('Poisson',1:6,1,6)

x2 =

     2     1     1     3     6     8
```

- See MATLAB help for my for details.

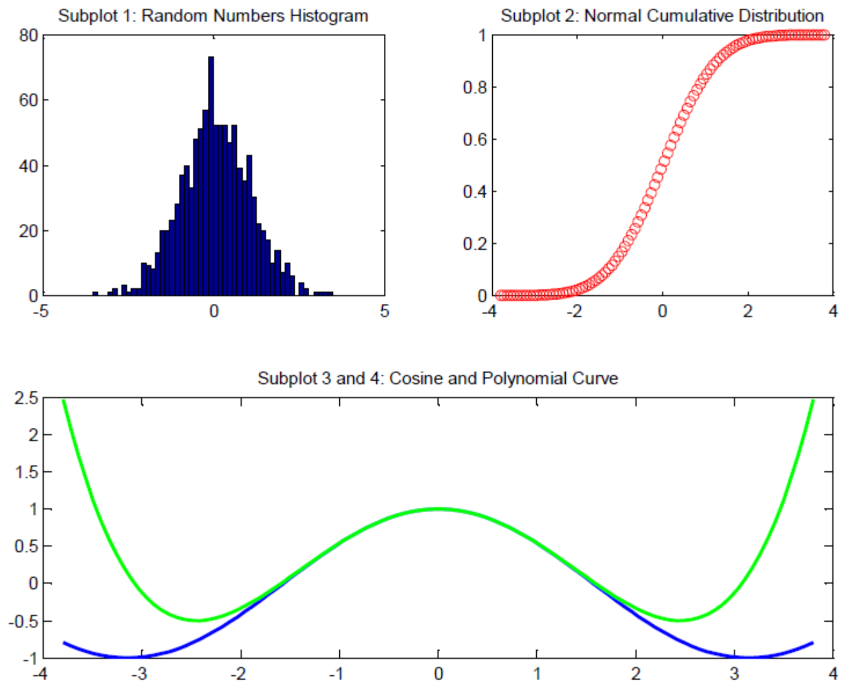
Basics of Plotting




```
>> x=0:0.2:10;
y1=sin(x);
y2=cos(x);
plot(x,y1,'b','LineWidth',2);
hold on;
plot(x,y2,'g','LineWidth',2);
xlabel('X Axis');
ylabel('Y Axis');
title('Toy Plot');
legend('Sine function','Cosine function','Location','Southeast');
grid on
box off
```

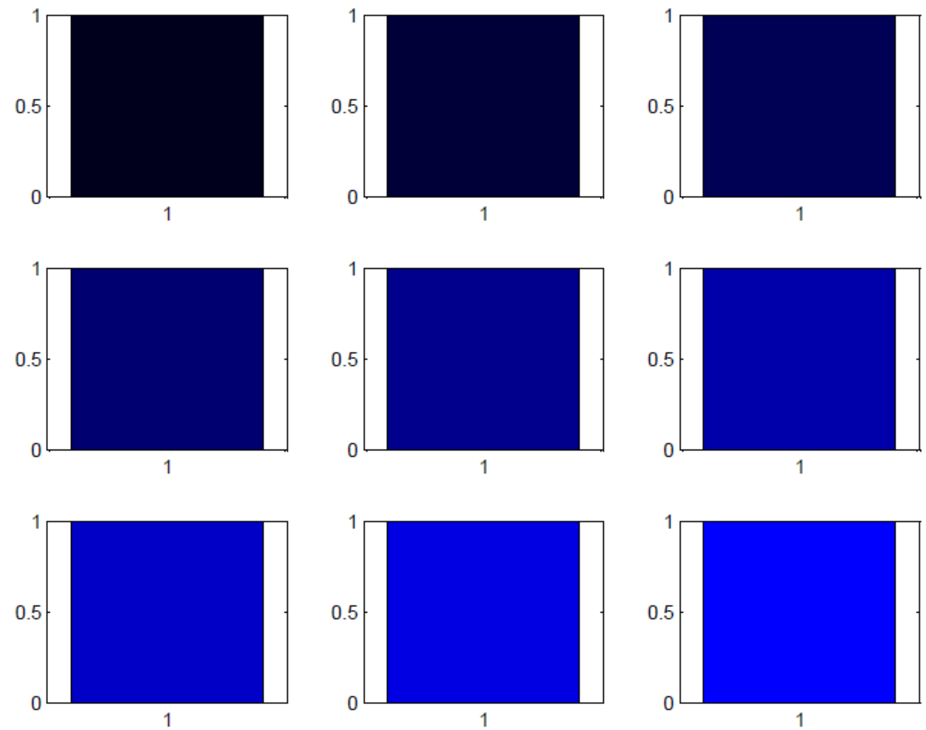
TECHNISCHE UNIVERSITÄT DRESDEN

Subplots




```
x = linspace(-3.8,3.8);
figure
subplot(2,2,1);
hist(normrnd(0,1,[1,1000]),50);
title('Subplot 1: Random Numbers Histogram','LineWidth',2)
xlim([-5 5])

subplot(2,2,2);
plot(x,normcdf(x,0,1),'ro');
title('Subplot 2: Normal Cumulative Distribution','LineWidth',1.5)

y_cos = cos(x);
y_poly = 1 - x.^2./2 + x.^4./24;
subplot(2,2,[3,4]);
plot(x,y_cos,'b',x,y_poly,'g','LineWidth',2);
title('Subplot 3 and 4: Cosine and Polynomial Curve')
```

Subplots




```
>> figure %Open a new figure
for ii=1:9 %Start loop, have counter ii run from 1 to 9
subplot(3,3,ii) %Draw into the subplot ii, arranged in 3 rows, 3 columns
h=bar(1,1); %This is just going to fill the plot with a uniform color
set(h,'FaceColor',[0 0 ii/9]); %Draw each in a slightly different color
end %End loop
```

TECHNISCHE UNIVERSITÄT DRESDEN

References

- **MATLAB for Psychologists (2012),** Borgo, M., Soranzo, A., Grassi, M., Springer-Verlag, 2012, ISBN. 978-1-4614-2196-2.
  - Chapter 1. Basic Operations, pp. 1-23.
  - Chapter 2. Data Handling, pp. 25-46

- **MATLAB for Neuroscientists, 2nd Ed: An Introduction to Scientific Computing (2014),** *Wallisch, P., Lusignan, M.E., Benayoun, M.D., Baker, T.I., Dickey, A.S. and Hatsopoulos, N.G.,* Academic Press, ISBN. 978-0123838360.
  - Chapter 2-3. pp. 7-114.

- **MATLAB help:**
  - http://www.mathworks.com/help/matlab/random-number-generation.html
  - http://www.mathworks.com/help/stats/random.html
  - http://www.mathworks.com/help/matlab/ref/subplot.html