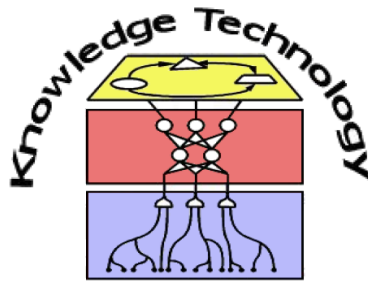


# Organization & Introduction to Matlab

Data Mining Practical Course

Doreen Jirak, Cornelius Weber, Stefan Heinrich



<http://www.informatik.uni-hamburg.de/WTM/>

# Practical Course Organization

- Meeting in room D-114, D-118 every two weeks
  - 3 hours (=4SWS)
- Procedure:
  - Tasks consist of theoretical & practical parts
  - Data and sheets available in the CommSy room
    - If not happened yet: **apply now**
    - Group of 2 students/computer
  - No corrections, no points
  - Create your own course documents
  - Discussions with tutor
    - Tutor decides whether sheet is passed (like in SE1)

# Criteria for passing the course

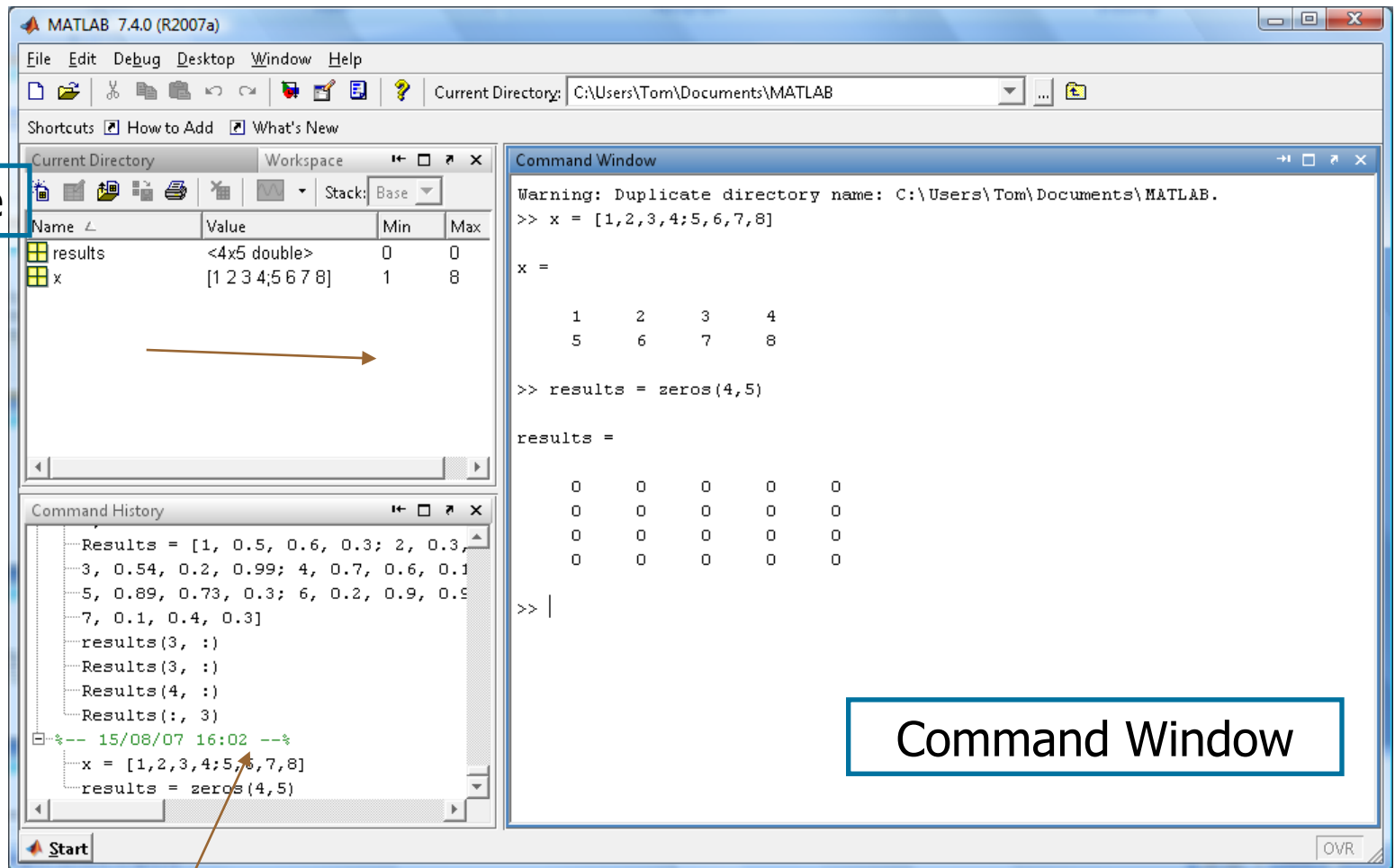
- Attendance!
  - Case of illness: Inform your tutor
- Every sheet has to be finalized in course time
- Every sheet has to be passed
- Activity in discussions and practical work
  - Every student should be prepared to explain the group work

**Questions???**

# What is Matlab

- **Matrix Laboratories**, The Mathworks Inc. software product
- High-level, interpreted language
- GUI-based, but also terminal
- Contains a lot of specialized function packages
  - Toolbox: e.g. image processing, optimization, etc.
- Needs licenses ☹
  - Solution: vpn-connection to UHH or **Octave** (free)
- Linux: just type *matlab* (without GUI: `-nodesktop`)
- Book recommendation: Wolfgang Schweizer, *Matlab Kompakt*, Oldenbourg Verlag, 4. Auflage

# Matlab GUI



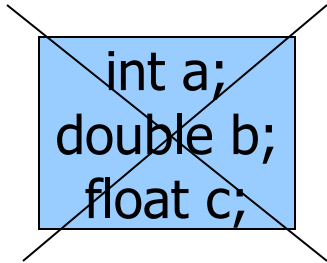
Workspace

Command Window

Command History

# Variables

- No need for types. i.e.,



```
int a;  
double b;  
float c;
```

- All variables are created with double precision unless specified and they are matrices

Example:

```
>>x=5;  
>>x1=2;
```

- After these statements, the variables are 1x1 matrices with double precision
- Also: char, string, float, logical, integer, ...

# Further Representation

- Matlab operations rely on linear algebra
  - Data can be scalar, vector or matrix
- Variable assignment: any string of upper and lower case letters along with numbers and underscores but it must begin with a letter
- Examples:
  - `a=5;`
  - `b=[1 2 3]; %row vector`
  - `c=[1;2;3]; %column vector`
  - `d=[1 2 3 4; 5 6 7 8]; %2x4 matrix`



# Data generation with : operator

- `a = 1:10`

a =

1      2      3      4      5      6      7      8      9      10

- `b = 2:-0.5:-1`

b =

2    1.5    1    0.5    0    -0.5    -1

- `c = [1:4; 5:8]`

c =

1	2	3	4
5	6	7	8

# Access to data

`c =`

1	2	3	4
5	6	7	8

`c(1, :) =`

1	2	3	4
---	---	---	---

`c(1, [1:3]) =`

1	2	3
---	---	---

`c(:, [1:3]) =`

1	2	3
5	6	7

`c(2, [2:end]) =`

6	7	8
---	---	---

# Operations

- **Simple arithmetic:**  $+$ ,  $-$ ,  $/$ ,  $*$
- **Logical Operators:**
  - Greater Than:  $>$
  - Less Than:  $<$
  - Greater Than or Equal To:  $>=$
  - Less Than or Equal To:  $<=$
  - Is Equal:  $==$
  - Not Equal To:  $\neq$
- **Boolean:**  $\&$ ,  $\sim$ ,  $|$
- **Transpose:**  $'$

# Functions

- Signature for writing your own functions

```
output=func_name(input)
```

```
[output1, output2]=fun_name(in1,in2,in3)
```

```
output=fun_name(input1, input2, varargin)
```

```
[output, varargin]=fun_name(in1, in2, varargin)
```

- File name should match function name

- Usage of built-in functions

```
result=mean(data);
```

```
[sort_out]=sort(data);
```

```
[sort_out, index]=sort(data, dim);
```

# Useful Functions today

- MATLAB built-in functions
  - **sum** – Sums the content of the variable passed
  - **prod** – Multiplies the content of the variable passed
  - **mean** – Calculates the mean of the variable passed
  - **median** – Calculates the median of the variable passed
  - **mode** – Calculates the Mode of the variable passed
  - **std** – Calculates the standard deviation of the variable passed
  - **sqrt** – Calculates the square root of the variable passed
  - **max** – Finds the maximum of the data
  - **min** – Finds the minimum of the data
  - **size** – Gives the size of the variable passed
- Special:
  - **pi**, **i** or **j** ( $\sqrt{-1}$ ), **inf**

# Plotting in Matlab

- A basic plot

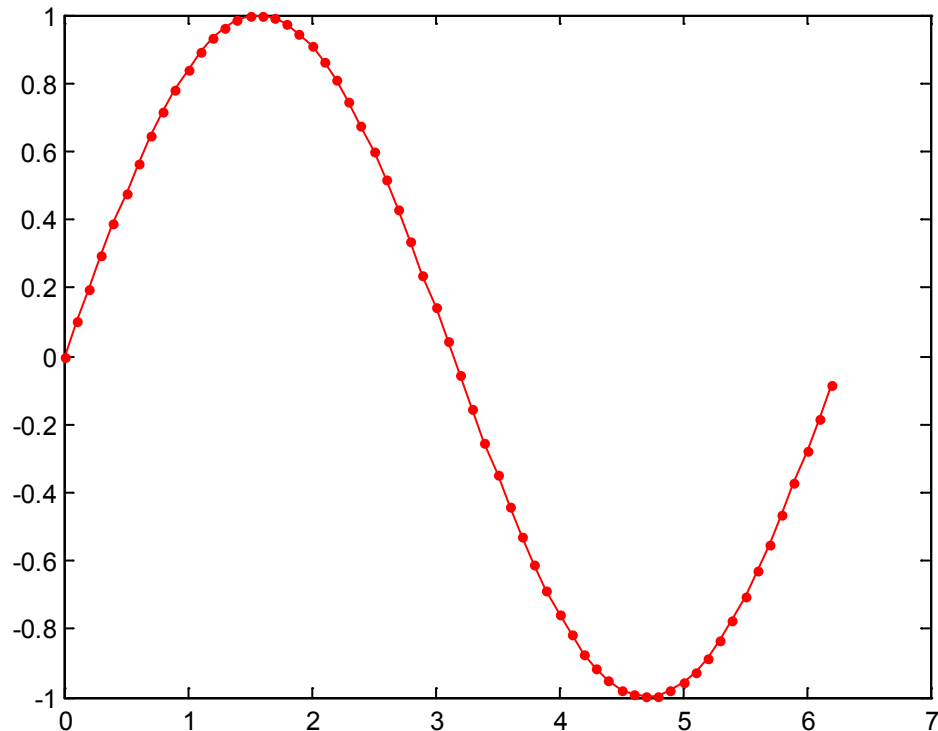
```
>> x = [0:0.1:2*pi]
```

```
>> y = sin(x)
```

```
>> plot(x, y, 'r.-')
```

Here: commands  
are typed in command  
window

But: sequence of  
Matlab commands can  
be saved as a **\*.m-file**



# Task today: Explore Matlab & Visualization

- Hints and useful command for today:
  - Data is stored in **\*.mat-files** → Download from Commsy
    - Opens data in workspace by double-click
  - Make sure your the data is in the right path
  - Matlab treats columns as individual data (e.g. data file *min\_salary*)
  - Type e.g. `help median` to get help directly on your command window
    - Else: Refer to Matlab documentation (press F1)
  - Type `help help` for further advice how to get help

# Today: Explore Matlab & Visualization (cont'd)

- Hints and useful command for today:
  - Matlab opens only one figure panel to plot your results
    - Use **hold** for multiple plots in one panel, e.g. for time-series
    - Use **clear** command again to release
  - For multiple figures use **figure(n)**
  - Use **close all** to get rid of all opened figures
  - For labeling data (etc), figure editor comprises nice features
    - You can also generate code like in sin-plot example automatically
      - Figure→File →Generate code (save as \*.m-file)
  - Delete variables in workspace: **clear**
  - Clear the command window: **clc**