# ISGS LaTeX Beginner's Workshop

## Session 3 – Figures

Patrick Wolf
patrick.wolf@cs.uni-kl.de

December 09, 2016

# Table of Contents I

# 1. Recap of Session 2

– Exercise Sheet –

# Questions from last session

## Changing numbering style of captions

```
\renewcommand{\thefigure}
  {\arabic{chapter}.\arabic{figure}}
```

# Time for a break.

– 10 Minutes –

# 2. Recap and General Hints

# Automatically Generated Content

- Header: `\maketitle`
- Table of contents: `\tableofcontents`
- Bibliography: `\bibliography{<file>}`

# Automatically Generated Content

- Header: `\maketitle`
- Table of contents: `\tableofcontents`
- Bibliography: `\bibliography{<file>}`
- List of figures: `\listoffigures`
- List of tables: `\listoftables`
- Index of keywords: `\makeindex` and `\printindex`

## Index

You can define index entries using `\index{...}` commands in the text.

# Saving Compile Time

## Compiling and Recompiling...

Compilation of bigger documents (e. g. a diploma thesis) becomes quite slow

Separate the document into multiple TeX-files:

- Introduction
- Basics
- Main Part
- Results
- Outlook

# Saving Compile Time

## Compiling and Recompiling...

Compilation of bigger documents (e. g. a diploma thesis) becomes quite slow

Separate the document into multiple TeX-files:

- Introduction
- Basics
- Main Part
- Results
- Outlook

## Combination

- `\input{...}`
- `\include{...}` and `\includeonly{...}`

# Splitting the Document

## Example

```
\documentclass{article}
...
\begin{document}
  % \include{introduction}
  % \include{basics}
  \include{main_part}
  \include{results}
  % \include{outlook}
\end{document}
```

# Splitting the Document

## Example

```
\documentclass{article}
...
\begin{document}
  % \include{introduction}
  % \include{basics}
  \include{main_part}
  \include{results}
  % \include{outlook}
\end{document}
```

- Main file contains the outer structure
- Chapters contain only pure LaTeX code, no `\documentclass` etc.
- Only 2 chapters will be compiled in this example

# Input and Include I

## Input

`\input{<filename>}`

- includes the file at current position
- can be nested[1]

[1] deep hierarchies are not recommended

# Input and Include I

## Input

`\input{<filename>}`

- includes the file at current position
- can be nested[1]

## Include

`\include{<filename>}`

- performs `\clearpage`, then includes the file
- can be controlled using `\includeonly`

[1] deep hierarchies are not recommended

# Input and Include II

## Problems

- You cannot compile a chapter, you have to switch to the main file and compile it (editors like Kile or TeXnic Center take care of this)
- When "commenting out" a chapter, its labels are lost (references break and TOC needs to be rewritten)

# Include and Includeonly

## Example

```
\documentclass{article}
...
\includeonly{main_part,results}
\begin{document}
  \include{introduction}
  \include{basics}
  \include{main_part}
  \include{results}
  \include{outlook}
\end{document}
```

- Only 2 chapters will be compiled in this example
- The labels and contents of the excluded chapters are kept in the index

# Time for a break.

– 5 Minutes –

# 3. Graphics

# Introduction to Floats

## Problem

- LaTeX is a system for typesetting of text and other elements
- Text can usually be split at many positions (paragraphs, lines, words, syllables)

# Introduction to Floats

## Problem

- LaTeX is a system for typesetting of text and other elements
- Text can usually be split at many positions (paragraphs, lines, words, syllables)
- Images and tables are bigger, inseparable elements
- Placing an image in your text can cause empty space on a page

# Introduction to Floats

## Problem

- LaTeX is a system for typesetting of text and other elements
- Text can usually be split at many positions (paragraphs, lines, words, syllables)
- Images and tables are bigger, inseparable elements
- Placing an image in your text can cause empty space on a page

## Solution

Use *floating* elements:

- You define the content and some positioning parameters
- LaTeX takes care of the exact position

# Most used Types of Floats

## Figures

```
\begin{figure}
  \includegraphics{...}
  ...
\end{figure}
```

# Most used Types of Floats

## Figures

```
\begin{figure}
  \includegraphics{...}
  ...
\end{figure}
```

## Tables

```
\begin{table}
  \begin{tabular}{...}
    ...
  \end{tabular}
\end{table}
```

# Float Positioning Suggestions

## Example

```
\begin{figure}[<positioning suggestions>]
   ...
\end{figure}
```

# Float Positioning Suggestions

## Example

```
\begin{figure}[<positioning suggestions>]
  ...
\end{figure}
```

## Parameters

Combination of

- `h`: here
- `t`: top of page
- `b`: bottom of page
- `p`: on special page containing only floats

## Note:

Order of suggestions specifies the priority,
e. g. `[hbt]` (here, bottom, top) or `[tb]`.

# Float Positioning Algorithm

- as soon as possible
- not before the page of definition
- keep the order
- follow general settings
- consider local suggestions
- don't overfill pages
- before `\clearpage` (with page break) or `\FloatBarrier` (`placeins` package, no pagebreak)

# Captions and References

## Example

```
\begin{figure}[<positioning>]
   ...
   \caption{Some text here}
   \label{<label>}
\end{figure}
```

# Captions and References

## Example

```
\begin{figure}[<positioning>]
  ...
  \caption{Some text here}
  \label{<label>}
\end{figure}
```

- Caption text is displayed below images, e. g. “Figure 5: Some text here”
- Reference using `\ref{<label>}`
- Every float type (figure, table, listing) has own counters

# A Hint for Figure Placement

```
Lorem ipsum dolor sit amet, consetetur
sadipscing elitr. % <--- NO BREAK HERE
\begin{figure}
   ...
\end{figure}\\ % <--- BREAK HERE
...
```

# A Hint for Figure Placement

```
Lorem ipsum dolor sit amet, consetetur
sadipscing elitr. % <--- NO BREAK HERE
\begin{figure}
  ...
\end{figure}\\ % <--- BREAK HERE
...
```

- Important: No blank lines between text and `figure` environment!
- The paragraph break command (`\\`) signals the affiliation between text and figure
- Nicer figure positioning and paragraph breaks

# Including Images into the Document

## Figure

```
\begin{figure}[<positioning>]
  \centering
  \includegraphics[<options>]{<file>}
  \caption{...}
  \label{fig:...}
\end{figure}
```

# Including Images into the Document

## Figure

```
\begin{figure}[<positioning>]
  \centering
  \includegraphics[<options>]{<file>}
  \caption{...}
  \label{fig:...}
\end{figure}
```

Figure: Example image

# Multiple Images in a Figure (1)

## Subfloat

using `subfig` package[2]:

```
\begin{figure}
  \centering
  \subfloat[Subcaption~1] {
    \includegraphics{...}
    \label{fig:sub_a}
  }
  \subfloat[Subcaption~2] {
    \includegraphics{...}
    \label{fig:sub_b}
  }
  \caption{Main Caption}
  \label{fig:main}
\end{figure}
```

[2] Attention! The command is not named like the package

# Multiple Images in a Figure (2)

(a) Subcaption 1

(b) Subcaption 2

Figure: Main Caption

You can reference the main figure 2 and the subfigures 2a and 2b using the `\ref{...}` command.

# Options for includegraphics

## Width, Height

```
\includegraphics[width=4cm]{...}
\includegraphics[height=1in]{...}
```

- usual units like `mm`, `pt`, `in`, `em`, `ex`, `...`
- fractions of the current text width: `0.3\textwidth`
- specifying both width and height can squash the image
- specifying only width or height keeps the aspect ratio

# Options for includegraphics

## Width, Height

```
\includegraphics[width=4cm]{...}
\includegraphics[height=1in]{...}
```

- usual units like `mm`, `pt`, `in`, `em`, `ex`, `...`
- fractions of the current text width: `0.3\textwidth`
- specifying both width and height can squash the image
- specifying only width or height keeps the aspect ratio

## Angle

```
\includegraphics[angle=15]{...}
```

# Clipping Graphics

## Viewport

```
\includegraphics[viewport=lx ly ux uy,clip]{}
```

- `lx` = lower *x* coordinate, `uy` = upper *y* coordinate
- length units for raster images are pixels
- length units for documents with size information (e. g. pdf) are the usual units
- `viewport` only changes the *virtual* image border
- `clip` actually clips the image to its borders
- Order of options can change the outcome

# Clipping Graphics

## Viewport

```
\includegraphics[viewport=lx ly ux uy,clip]{}
```

- `lx` = lower *x* coordinate, `uy` = upper *y* coordinate
- length units for raster images are pixels
- length units for documents with size information (e. g. pdf) are the usual units
- `viewport` only changes the *virtual* image border
- `clip` actually clips the image to its borders
- Order of options can change the outcome

# Figures that are not Images

## Figures are versatile!

You don't have to use `\includegraphics` in the `figure` environment.

```
\begin{figure}
  \centering
  \textbf{Hello, I'm text in a figure}\\
  $\vec{c} = \vec{a} \times \vec{b}$
  \caption{Not an image}
\end{figure}
```

# Figures that are not Images

## Figures are versatile!

You don't have to use `\includegraphics` in the `figure` environment.

```
\begin{figure}
  \centering
  \textbf{Hello, I'm text in a figure}\\
  $\vec{c} = \vec{a} \times \vec{b}$
  \caption{Not an image}
\end{figure}
```

**Hello, I'm text in a figure**

$$\vec{c} = \vec{a} \times \vec{b}$$

Figure: Not an image

You can even `\input` another file within `figure`!

# Time for a break.

– 5 Minutes –

# 4. TikZ

# What is TikZ?

- Ti*k*Z = "Ti*k*Z ist *kein* Zeichenprogramm"[3] (Ti*k*Z is not a drawing program)
- "User-friendly syntax layer" for PGF (macros package for generating graphics)
- TikZ provides a special environment

```
\usepackage{tikz}
...
\begin{tikzpicture}[<options>]
  ...
\end{tikzpicture}
```

- TikZ provides also its own `\usetikzpackage` command

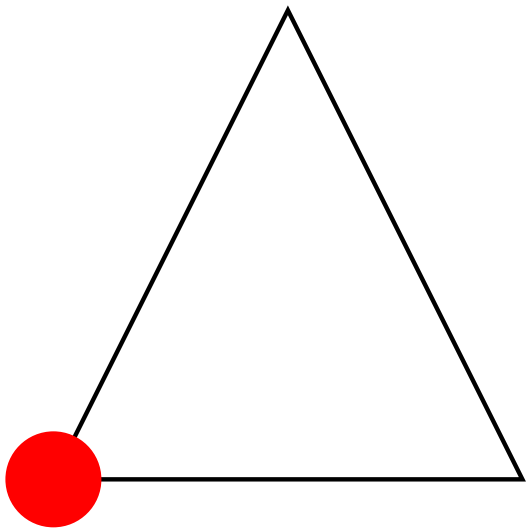
[3] Geeks love recursive acronyms

# TikZ Example

```
\begin{tikzpicture}
  \draw[thick] (0,0)--(1.5,3)--(3,0)--cycle;
  \draw[fill,red] (0,0) circle (3mm);
\end{tikzpicture}
```

# TikZ Example

```
\begin{tikzpicture}
  \draw[thick] (0,0)--(1.5,3)--(3,0)--cycle;
  \draw[fill,red] (0,0) circle (3mm);
\end{tikzpicture}
```



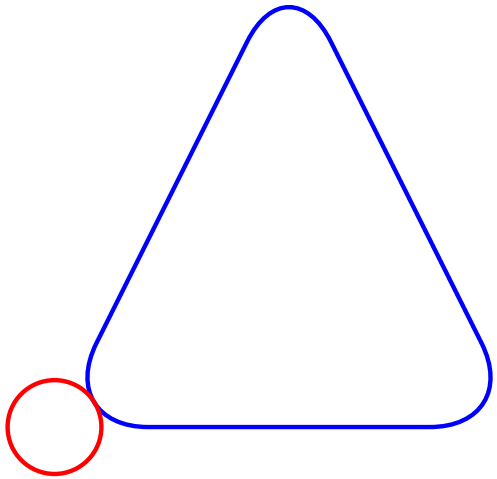
1 unit = 1 cm (unless defined otherwise)

# TikZ Example: Global Parameters

```
\begin{tikzpicture}[thick]
  \draw[blue,rounded corners=6mm]
    (0,0)--(1.5,3)--(3,0)--cycle;
  \draw[red] (0,0) circle (3mm);
\end{tikzpicture}
```

# TikZ Example: Global Parameters

```
\begin{tikzpicture}[thick]
  \draw[blue,rounded corners=6mm]
    (0,0)--(1.5,3)--(3,0)--cycle;
  \draw[red] (0,0) circle (3mm);
\end{tikzpicture}
```

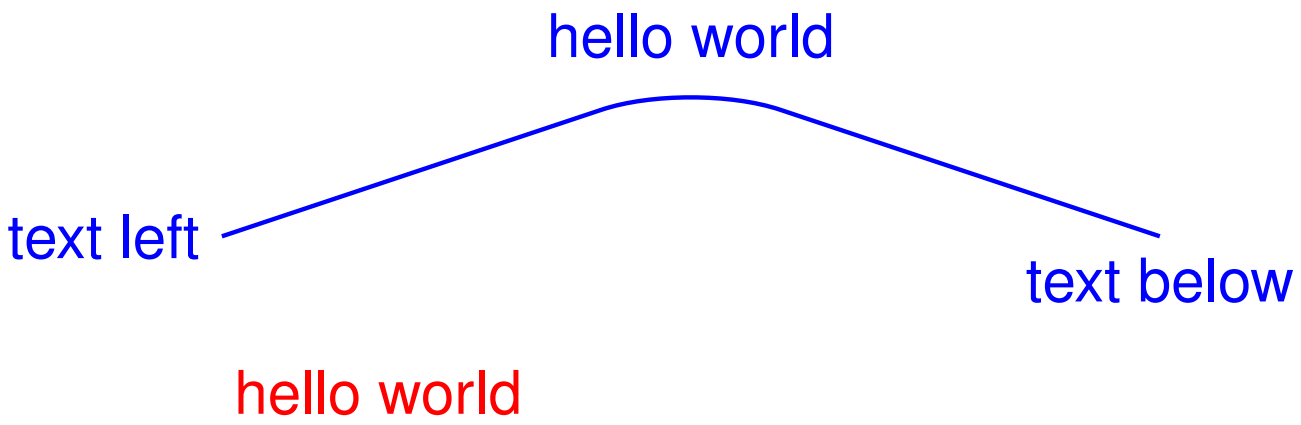


# TikZ Example: Nodes and Text

```
\begin{tikzpicture}[thick,blue]
  \draw[rounded corners=6mm]
      (0,0) node[left] {text left}
      -- (3,1) node[above] {hello world}
      -- (6,0) node[below] {text below};
  \draw[red] (1,-1) node {hello world};
\end{tikzpicture}
```

# TikZ Example: Nodes and Text

```
\begin{tikzpicture}[thick,blue]
  \draw[rounded corners=6mm]
      (0,0) node[left] {text left}
      -- (3,1) node[above] {hello world}
      -- (6,0) node[below] {text below};
  \draw[red] (1,-1) node {hello world};
\end{tikzpicture}
```

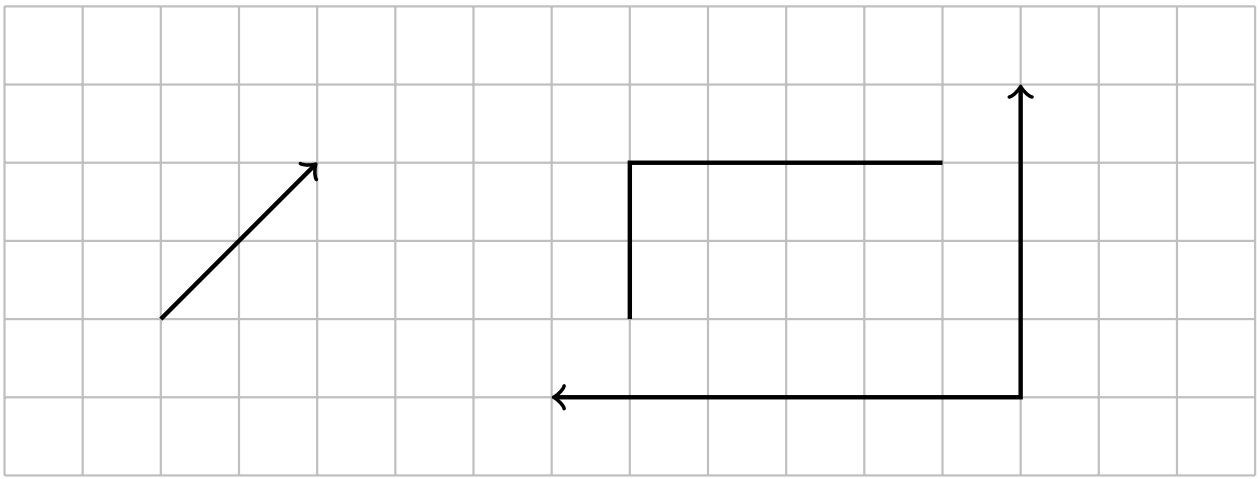


# TikZ Example: Grids and Arrows

```
\begin{tikzpicture}
  \draw[step=0.5,lightgray] (0,0) grid (8,3);
  \draw[thick,->] (1,1) -- (2,2);
  \draw[thick,<->] (3.5,0.5) -| (6.5,2.5);
  \draw[thick] (4,1) |- (6,2);
\end{tikzpicture}
```

# TikZ Example: Grids and Arrows

```
\begin{tikzpicture}
  \draw[step=0.5,lightgray] (0,0) grid (8,3);
  \draw[thick,->] (1,1) -- (2,2);
  \draw[thick,<->] (3.5,0.5) -| (6.5,2.5);
  \draw[thick] (4,1) |- (6,2);
\end{tikzpicture}
```

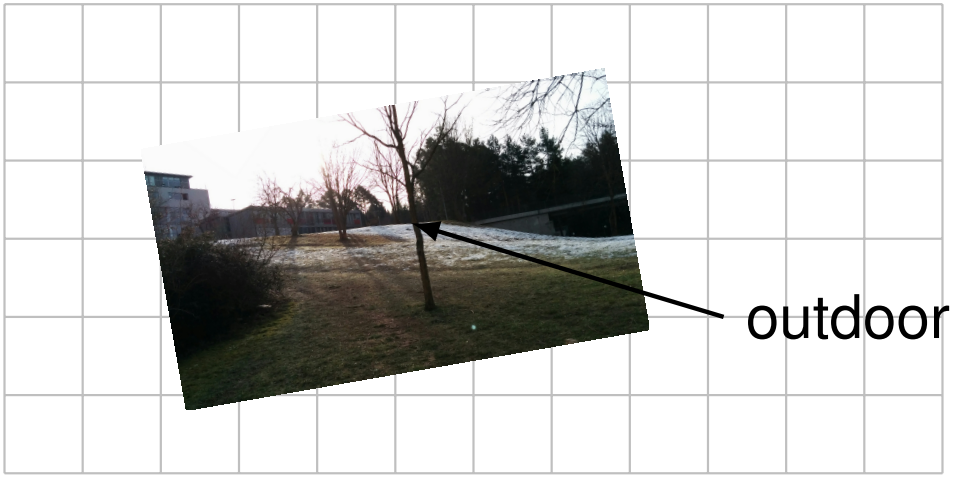


# TikZ Example: Combined with LaTeX

```
\begin{tikzpicture}
  \draw[step=0.5,lightgray] (0,0) grid (6,3);
  \draw (2.5,1.5) node
      {\includegraphics[...]{...}};
  \draw[latex-,thick] (2.6,1.6) --
      +(2,-0.6) node[right] {outdoor!};
\end{tikzpicture}
```

# TikZ Example: Combined with LaTeX

```
\begin{tikzpicture}
  \draw[step=0.5,lightgray] (0,0) grid (6,3);
  \draw (2.5,1.5) node
      {\includegraphics[...]{...}};
  \draw[latex-,thick] (2.6,1.6) --
      +(2,-0.6) node[right] {outdoor!};
\end{tikzpicture}
```

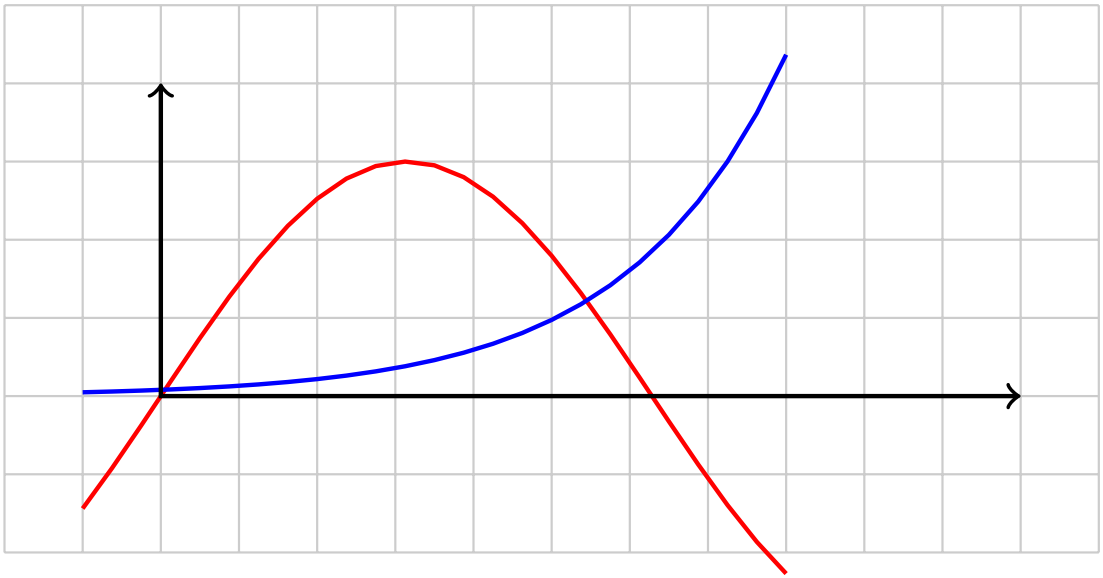


# TikZ Example: Plotting Functions

```
\begin{tikzpicture}[domain=-0.5:4]
 \draw[step=0.5,black!20](-1,-1) grid (6,2.5);
 \draw[thick,red] plot (\x, {1.5*sin(\x r)});
 \draw[thick,blue] plot (\x, {0.04*exp(\x)});
 \draw[thick,<->] (0,2)--(0,0)--(5.5,0);
\end{tikzpicture}
```

# TikZ Example: Plotting Functions

```
\begin{tikzpicture}[domain=-0.5:4]
 \draw[step=0.5,black!20](-1,-1) grid (6,2.5);
 \draw[thick,red] plot (\x, {1.5*sin(\x r)});
 \draw[thick,blue] plot (\x, {0.04*exp(\x)});
 \draw[thick,<->] (0,2)--(0,0)--(5.5,0);
\end{tikzpicture}
```

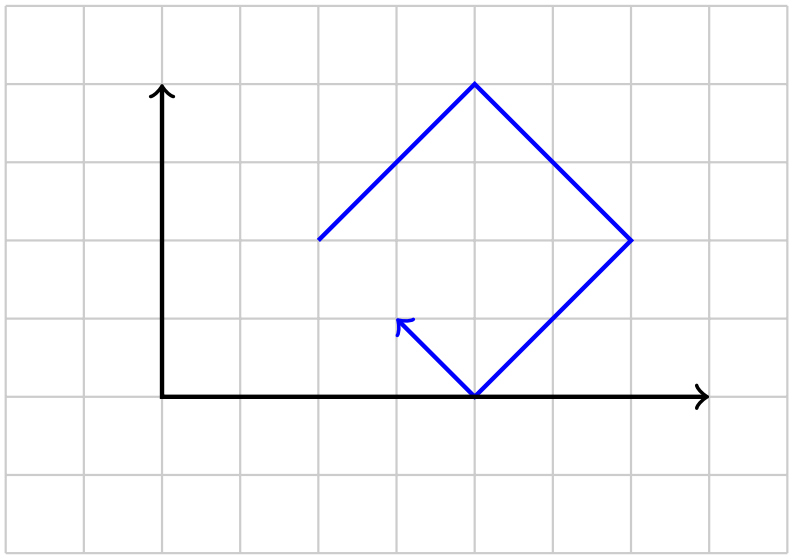


# TikZ Example: Plotting Data

```
\begin{tikzpicture}
 \draw[step=0.5,black!20](-1,-1) grid (4,2.5);
 \draw[thick,blue,->] plot file{data.txt};
 \draw[thick,<->] (0,2) |- (3.5,0);
\end{tikzpicture}
```

# TikZ Example: Plotting Data

```
\begin{tikzpicture}
 \draw[step=0.5,black!20](-1,-1) grid (4,2.5);
 \draw[thick,blue,->] plot file{data.txt};
 \draw[thick,<->] (0,2) |- (3.5,0);
\end{tikzpicture}
```



data.txt:

```
1 1
2 2
3 1
2 0
1.5 0.5
```

# 5. CSV

# CSV Tables

## CSV-Import

```
\usepackage{csvsimple}

\csvautotabular{data.csv}
```

## Example

| X | Y |
|---|---|
| 1 | 1 |
| 2 | 2 |
| 3 | 1 |
| 2 | 0 |
| 1.5 | 0.5 |

# Formatting of CSV Tables

## CSV Reader

```
\csvreader[<options>]{<file name>}
{<assignment>}{<command list>}

\csvreader[head to column names]{data.csv}{}%
{\\ \csv_entry_1 & \csv_entry_2 & ...}%
```

## Example

| X | Y |
|---|---|
| 1 | 1 |
| 2 | 2 |
| 3 | 1 |
| 2 | 0 |
| 1.5 | 0.5 |

# Formatting of CSV Tables

## CSV Style

```
\csvstyle{myStyle}
{tabular=c c, table head=X & Y \\}%

\csvreader{myStyle}{data.csv}{}{\X & \Y}
```

For more information see `csvsimple` documentation.

# Thank you for your attention!