Day 1, part 3: Tables and figures

Digital Skills for Research

10 May 2022

Contents

Key concepts and commands covered

Reflect on your progress and see whether you know how to:

Editors and start

1. migrate btw desktop/online editors
2. define the document margins
3. create structure (sections) and ToC
4. comment out several lines
5. typeset the title and authors
6. create columns
7. adjust the text positionally

Text formatting and formulas

1. write in several natural languages
2. deal with special char
3. use fonts, colour, size
4. use typesetting conventions for quotes, dash, non-breaking space
5. find commands for symbols
6. use mathmode and typeset formulas

1 Environments

1.1 LISTs!

Three predefined types of list environments: `\begin{itemize}, \begin{enumerate}, \begin{description}`

- simple bullets
- enumerated items (customisable with `\begin{enumerate}[a)], [i], [1.], [A], [I)], [1)]` after importing `\usepackage{enumerate}`)
- `\begin{description}` use a list of terms as items and their definitions as values against each item (see an example of description-style list below)

1.2 Examples of other default environments

```
\begin{center} ... \end{center}
```

CENTRED TEXT
Environments are used to apply formatting to blocks of text.

`\begin{quote}` for short quotes separated by blank lines and `\begin{quotation}` for several indented paragraphs. See `\begin{quote}`:

> "Don't worry about a thing,
> 'Cause every little thing gonna be all right."

1.3 Examples of imported environments

```
==
This is a demo of an environment \begin{comment}.
It requires \usepackage{comment} in the preamble.
Text between == is printed inside \begin{verbatim} environment.

\begin{comment}
This is great for multi-line comments
such as this one.
It require \usepackage{comment} in the preamble.
\end{comment}
==

\usepackage{multicols}
\begin{multicols}{3} ... \columnbreak ... \end{multicols}
```

In Section 2 we will look at some principles of arranging tables.

This is source code for a basic table:

```
\begin{tabular}{ c c c }
cell1 & cell2 & cell3 \\
cell4 & cell5 & cell6 \\
cell7 & cell8 & cell9 \\
\end{tabular}
```

This is how it compiles:

cell1	cell2	cell3
cell4	cell5	cell6
cell7	cell8	cell9

2 A special case: Tables

See Wizards and Latex tabs in your TEX editor!

2.1 Basic commands for tables formatting

A) horizontal lines with `\hline` or `\toprule, \midrule` from **booktabs** package

B) vertical separators are defined by vertical line in call parameters: `\begin{tabular}{l|c|r|}`; this is NO lines: `\begin{tabular}{ccc}`

C) & is a column separator in code

D) each row ends with \\

2.2 Table environments and wizards

There are several environments to format tables. They are presented as a description-style list mentioned above, and this is a hyperlink to a specific word in the text above; see the source file at GitHub for how to code it):

tabular basic format to stack text horizontally and vertically; is placed as-is in the text, at the position where coded; the width of each column is set with `\begin{tabular}{p{4cm}|c|p{5cm}|}`; even columns with various text placement (left, centre, right): `\begin{tabular}{l|c|r|}`

table a float object that determines best position in text; it can contain virtually anything, but often used to wrap `\begin{tabular}`

tabularx uses X columns to fit the table to the specified width (`{\begin{tabular}{\textwidth}{X|c|X}`)

longtable imported with `\usepackage{longtable}`; codes tables that span across the page boundary

This code generates Table 1.

Source code

```
\begin{table}[h!]
        \begin{center}
                \caption[Meaningless table]{This table does not make much sense; it
    is created for demonstration only}\label{tab:mytab}
                \begin{tabular}{|c|c|c|c||l|c|c|r|c|c|}
                        \hline
                        1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ \hline
                        first & second & \multicolumn{3}{|c|}{third -- fifth} &   &
    & eight &   &  \\
                        \cline{1-7} \cline{9-10}
                        1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ \hline
                        1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ \hline
                        \multirow{3}{*}{three rows}  & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9
    & 10 \\ \cline{2-10}
                        & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\
                        & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ \hline
                \end{tabular}
        \end{center}
\end{table}
```

Compiled table

Table 1. This table does not make much sense; it is created for demonstration only

1	2	3	4	5	6	7	8	9	10
first	second	third – fifth					eight		
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
three rows	2	3	4	5	6	7	8	9	10
	2	3	4	5	6	7	8	9	10
	2	3	4	5	6	7	8	9	10

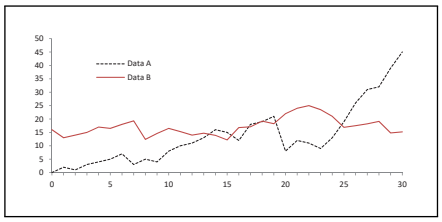
3 Graphics and drawing

3.1 Simple commands

(for raster (.png, .jpg) or vector (.eps) graphics)

Use scaling: `\includegraphics[scale=.2]{raster-vs-vectors.jpg}`

Use size (and a frame around the graphics): `\fbox{\includegraphics[width=50mm]{lines.eps}}`
Notice how you can change `\fbox{}` parameters inside the main code: `\setlength\fboxsep{10pt}`



3.2 Figure environment

```
\begin{figure}[h] % htbp - position preferences (here, top, bottom, page)
        \fbox{\includegraphics[scale=1]{cup}}
        \caption{Publisher's logo}
        \label{fig:logo}
\end{figure}
```

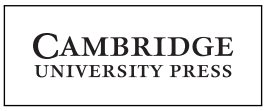


Figure 1. Publisher's logo

Use `\begin{wrapfigure}{r}{0.3333\linewidth}` to use Figures inline (wrapped with the text). For example:

Vector graphics are also known 1 as scalable vector graphics (SVG). These graphics consist of anchored dots and connected by lines and curves, similar to the connect-the-dot activities you may have done as a kid. Because these graphics are not based on pixels, they are known as resolution independent, which makes them infinitely scalable. Their lines are sharp, without any loss in quality or detail, no matter what their size. These graphics are also device-independent, which means their quality doesn't depend on the number of dots available on a printer or the number of pixels on a screen. Because they consist of lines and anchor points, the size of the files are relatively small.
You can also use an online Tables Generator.

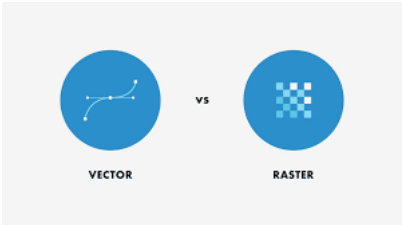


Figure 2. Two types of graphics

3.3 Drawing with TikZ

TikZ is the most complex and powerful tool to create vector graphics. The graphics are put in `\begin{tikzpicture}` environment.

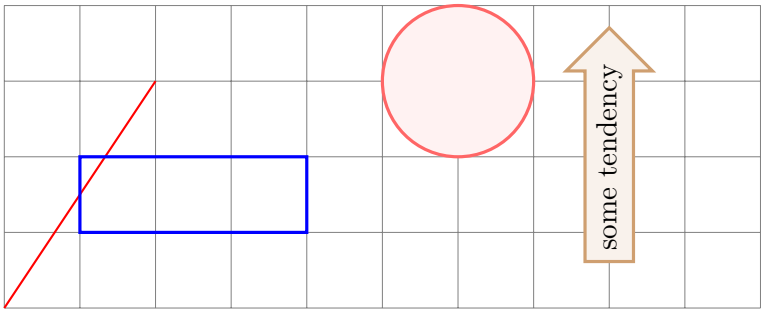
To help with coordinates, draw the support grid: `\draw[style=help lines] (-2,0) grid[step=1cm] (8,4);`

For each shape there are some drawing instructions. For example,

- for a rectangular you have to provide two points, the first one is where the "pencil" begins to draw the rectangle and the second one is the diagonally opposite corner point

```
\begin{tikzpicture}
        \draw[style=help lines] (-2,0) grid[step=1cm] (8,4);
        \draw[red, thick] (-2,0) -- (0,3);
        \draw[blue, very thick] (-1,1) rectangle (2,2);
        \filldraw[color=red!60, fill=red!5, very thick](4,3) circle (1);

       % define a style of some object, e.g. an arrow
        \tikzstyle{my arrow} = [draw=brown!75, very thick, single arrow, minimum h
        \node at (6,2) [my arrow=90] {\rotatebox{90}{some tendency}};
\end{tikzpicture}
```



NB! It might be easier to draw in Python, inc. saving as .eps

3.4 Produce automatic lists of labelled objects

`\listoftables and \listoffigures`

List of Tables

List of Figures

Task 3. Time-table your pastimes/classes and add pics/graphs

- Produce a one-page document with a table and a figure.
  - Make a timetable your pastimes/classes or make a simple table that compares vector and raster images as explained here, for example
  - no pagination, please!
  - add in-text references to both;
  - create proper captions: Chicago referencing style (as required by Benjamins Publishing house) requires "Figure captions should be placed below the figure, while table captions should be placed above the relevant table."
  - demonstrate any other efforts to learn about tables and graphics