Tables, Fonts and Spacing

1 Tables

For simple and short tables there is the handy tabular environment. The syntax is

```
\begin{tabular}{column specification}
  .. & .. & .. \\
  .. & .. & .. \\
\end{tabular}
```

where `column specification` is a string containing these characters:

`l`	for a column of left aligned text,
`c`	for a column of centered text,
`r`	for a column of right aligned text,
`|`	to create a vertical bar, or
`p{width}`	for a column with wraparound text of length `width`.

The syntax for moving to the next column and starting a new line is the same syntax as when using the `\begin{matrix}..\end{matrix}` command in math mode. Placing the optional commands

```
\setlength{\tabcolsep}{12pt}
\renewcommand{\arraystretch}{2}
```

before a table redefines the spacing between columns and rows. The defaults are `6pt` and a scaling factor of `1`.

The `\hline` command creates a horizontal line in the `tabular` environment. There are more examples of tables in this document and there are many more examples of tables of varying complexity on pages 45–48 of the text.

2 Fonts

2.1 Size

The font size for the entire document is set by the `[11pt]` option in the `\documentclass` command. This document is typed in 11pt font. The font size can also be 10pt (the default option) or 12pt. Another option in the `\documentclass` command is `twocolumn`, which has the effect of producing two columns per page.

These commands can be used to change the font size:

`\Huge`	Huge
`\huge`	huge
`\LARGE`	LARGE
`\Large`	Large
`\large`	large
`\normalsize`	normalsize
`\small`	small
`\footnotesize`	footnotesize
`\scriptsize`	scriptsize
`\tiny`	tiny

The `anyfontsize` package (not needed if using XeLaTeX) has the

```
\fontsize{font size}{base line stretch}
```

command which can be used in conjunction with the `\selectfont` command to select size 

.

2.2 Face

2.2.1 Selecting the face when using pdfLaTeX

Font faces are best chosen when compiling with XeLaTeX. There are a few options for faces when compiling with pdfLaTeX, some of which carry through to the case of compiling with XeLaTeX.

Four fonts are needed in any document: Roman (the default font), Math (used for mathematics), **Sans serif** (invoked by `\textsf{..}`), and `Typewriter font` (as seen in `verb` statements). The standard options for fonts when compiling with pdfLaTeX can be found when calling these packages:

Package	Roman	Math	Sans serif	Typewriter
(none)	CM Roman	CM Roman	CM Sans	CM Typewriter
mathpazo	Palatino	Palatino		
mathptmx	Times	Times		
helvet			Helvetica	
avant			Avant Garde	
courier				Courier
chancery	Chancery			
bookman	Bookman		Avant Garde	Courier
newcent	New Century		Avant Garde	Courier
charter	Charter			
fourier	Utopia	Fourier		
euler		Euler		

An empty entry indicates that a package does not have an effect on a given font face. The last two font selections are listed separately because they are not usually found in the basic version of LaTeX, but are given by the "texlive-full" version.

To select the roman font, math font, sans serif font, and typewriter font separately, include consecutive `\usepackages` in a correct order. For example, the commands

```
\usepackage{mathpazo}
\usepackage{charter}
\usepackage{helvet}
```

uses the Palatino math font, Charter roman font, Helvetica sans serif font, and Computer Modern Typewriter font.

2.2.2 Selecting the face when using XeLaTeX

Any font installed on your computer can be used when compiling with XeLaTeX. Use commands similar to these commands before the appearance of `\begin{document}`:

```
\usepackage{mathspec}

\setmainfont{Lato-Light.ttf}
\setmathsfont(Digits,Latin,Greek)[Numbers={Lining,Proportional}]{Lato-Light.ttf}
\setsansfont{351Week4LDFComicSans.ttf}
\setmonofont{AnonymousPro-Regular.ttf}
```

These commands change the four required fonts. To change the mathematics symbols, such as Σ, $\int$, ∂, and so on, load the corresponding math package in the above table before the `mathspec` package is called. For example, to use the symbols that come from `mathpazo`, load `mathpazo` before `mathspec`.

Fonts can also be changed mid-document.

3 Margins and spacing

3.1 Margins

The margins of the document can be controlled with the `geometry` package. To set the left, top, right, and bottom margins to specific values, place a command such as

```
\usepackage[left=35mm,top=2cm,right=35mm,bottom=2cm]{geometry}
```

in the preamble. The margins used in this document are those values shown above. As another example,

```
\usepackage[landscape, margin=2in]{geometry}
```

changes all margins to be 2 inches and prints in landscape mode.

3.2 Creating Whitespace

It usually bad form to manually adjust the vertical or horizontal spacing inside the body of the document when writing an article or book, but it might be appropriate to manually force white space when creating documents such as syllabi, exams, or resumes.

The line break command `\\` has an option to add extra space; to add an extra `1cm`, use `\\[1cm]`. Alternatively, to force an extra vertical space of `1cm` between two paragraphs, the `\vspace{1cm}` command can be used. Sometimes LaTeX will think adding a vertical space using `\vspace` is a bad idea and won't cooperate; such a vertical space can be demanded with the command `\vspace*{1cm}`.

The `\vfill` command produces a length which can stretch or shrink vertically, pushing the text after the `\vfill` command as far down the page as possible. This command can be used in tandem with the `\newpage` command, which forces a new page to begin. For instance, if writing a mathematics exam, the LaTeX commands

```
\begin{problem} Evaluate \( \int \ln x \, dx \). \end{problem}
\vfill
\begin{problem} Evaluate \( \int \sin x \, dx \). \end{problem}
\vfill
\newpage
```

will produce the next page in the document (provided the `amsthm` package is loaded and `\theoremstyle{definition}` and `\newtheorem{problem}{}` both appear in the preamble). Analogous to `\vspace`, `\vspace*` and `\vfill`, there are `\hspace`, `\hspace*`, and `\hfill` commands, which produce horizontal space, a forced horizontal space, and a rubber horizontal fill.

1. Evaluate $\int \ln x \, dx$.

2. Evaluate $\int \sin x \, dx$.

3.3 Minipages

The `minipage` environment creates a page within a page, useful for side by side type:

This is left text. right

3.4 Unbreakable text

The compiler makes typesetting decisions by placing boxes around letters, parts of words, mathematics, and figures, and then appropriately arranging the boxes. Force unbreakable type by using `\mbox{unbreakable text}`. Create a frame around a box using `\fbox{.}` and an unbreakable paragraph using `\parbox{width}{.}`.

Sometimes when using commands such as `\hfill`, an empty box is needed to get spacing just right; create such an empty box with `\mbox{}`.

On a similar note, `~` is a non-breaking space character, used when a space between two words or characters should appear but those words cannot be on different lines.

3.5 Tab Stops

Most of us are familiar with tabs from our frequent use of physical typewriters. Tabs can be kept and used with the `tabbing` environment. Set tabs using `\=`, create a new line using `\\`, and move to the next tab using `\>`. For example,

The first tab appears right here, this is text after the first tab, and there is a third tab.
This is the second row, middle of second row, and the end.

Tabbing environments are treated as one box and thus cannot be split across two pages.