

# LaTeXによる文書作成 (その2)

プログラミング演習 1#12

平成 26 年 7 月 4 日

## 1 表を作ろう

LaTeX では標準で表を作る機能を持っています。table 環境 (表を表示するエリアを設定します) 内に, tabular 環境 (表の形式を指定します) を入れてやることにより表が作成できます。下に表を作るための LaTeX ソースコードを示します。このソースコードで作った表は, 表 1 のようになります。表ではキャプション (表のタイトルなど説明書きのこと) を, 表の上に付けることが多いです。

```
\begin{table}[tbp]      %表領域開始
  \centering           %中央揃え
  \caption{組版した表} %キャプション設定
  \label{tab:table1}  %ラベル設定
  \begin{tabular}{|c|l|c|l|}\hline
    %表開始 (4 列で 1, 3 列目が中央揃え, 2, 4 列目が左揃え,
    %列間に縦線入り, 行下に直線を引く
    \multicolumn{4}{|c|}{bar}\ \ \ \hline
    %最初の行のみ, 列を 1 つにまとめる。内容は中央揃えで,
    %左右端には縦線をつける。
    item1 & foo1 & item2 & foo2 \ \ \ \hline
    %1 行目, 列の区切文字は&
    item3 & foo3 & item4 & foo4 \ \ \ \hline
    item5 & foo5 & item6 & foo6 \ \ \ \hline
  \end{tabular} %表終わり
\end{table} %表領域終わり
```

table 環境は表を入れる領域を作成するよう指示するもので, この表領域<sup>1</sup>は「フロート」と呼ばれ, LaTeX によりきれいにレイアウトされるよう自動的にその配置が決定されます。その配置を指定しているのが [bp] という記述で, これはまずページの下 (bottom) に置くことを試み, ダメなら表だけのページ (page) を作ってそこに表を置きなさい, という指示を表しています。ページの上部に置きたい場合には t (top) を指定し, 現在位置に置きたい場合には h (here) を指定します。ですから, [htbp] としてやると, 初めに文章中のその

表 1: 組版した表

bar			
item1	foo1	item2	foo2
item3	foo3	item4	foo4
item5	foo5	item6	foo6

<sup>1</sup>後述の figure 環境による図領域も同じです

位置に置こうとし、ダメならページ上部、それでもダメならページ下部、それでもダメなら別ページ、という順序で置き場所を決定していきます。

なお、常に好きな位置に配置して良いわけではなく、一般的な論文では [t] 指定、学位論文では [p] 指定を要求されることが多いです。

`\multicolumn` は、最初の `{}` にまとめる列数、次の `{}` にまとめた後の列の書式（中央揃えや左右揃え、両端に縦線を入れるか否か、など）の設定、最後の `{}` にはまとめた枠内に入れる文字や数式を記入します。各行の `\\` は、そこで表中の行が終了することを示し、`\hline` は行の左端から右端まで直線を引きなさい、という指示です。2 行目から 4 行目だけ線を引きたい場合には、`\cline{2-4}` という形で線を引く列を指定します。

本日のやってみよう#1

表を作ってみよう

表 2 のような表を作成して、先週のプログラミング演習 1 の授業で作成した  $\text{\LaTeX}$  の練習文書に埋め込むべし。表番号は自動的につくので、実際には「表 1」になるはず。

表 2: 確率と平均情報量

確率	0	0.25	0.5	0.75	1
平均情報量	0	0.81	1	0.81	0

## 2 図を取り込もう

$\text{\LaTeX}$  では標準で `picture` 環境というものがあって図を書くことができますが、思い通りの図を書くことは大変困難です<sup>2</sup>。そのため、`postscript` ファイルの一種である `eps`<sup>3</sup>形式の図を取り込むパッケージが用意されています。`eps` ファイル自体は、以前講義で紹介した `inkscape` や `gnuplot` で作成することができます。

`eps` ファイルを取り込むには、`\begin{document}` の前に、`\usepackage[dvips]{graphicx}` と書いておく必要があります。実際の図は `figure` 環境の中に記述した `\includegraphics` コマンドで取り込みます。カレントディレクトリにある `eps` ファイル `foo.eps` を取り込むには下のように記述します。

<sup>2</sup>最近では `TikZ/pgf` や `PSTricks` など使いやすいパッケージもありますが、それでもやはり GUI の方が描きやすいことが多いです。

<sup>3</sup>`eps` とは `Encapsulated Postscript` の略。

```

\documentclass[a4j]{jarticle}
...
\usepackage[dvips]{graphicx}
\begin{document}
...
\begin{figure}[tbp]
  \centering
  \includegraphics[keepaspectratio,clip,scale=.7]{foo.eps}
  \caption{foo.eps を取り込んだ例}
  \label{fig:foo}
\end{figure}
...
\end{document}

```

figure 環境では、図を取り込む領域を準備します。位置指定 [tbp] は表の時と同じです。centering で図をセンタリングします。includegraphics コマンドでは、[] の中にオプションを指定します。例では、縦横比を一定 (keepaspectratio)、余計な余白を取り除く (clip)、元の図の 70% の大きさと取り込む (scale=.7) ことを指定しています。他に使える主な指定は、幅や高さを直接指定する width や height、回転する角度を指定する angle があります。最後の {} の中には eps ファイルの名前を書きます。図の場合、キャプションは図の下に付けることになっていますから、caption と label の 2 つのコマンドは includegraphics コマンドの直後に置かれています。

gnuplot では、set term postscript eps 24 (24 は文字の大きさと 24pt の字を用いよという指定)、set output "foo.eps" としたあと、plot コマンドを実行することで、eps ファイルが作られるのは以前扱った通りです。このときは画面には図は出ないので注意して下さい。また、グラフに書きこみたいときなどは、set term svg (svg 形式で出力)、set output "foo.svg" としてグラフを生成しておく、あとから inkscape での修正が可能です。ただし、正式な論文等で捏造を疑われるような修正を行うと世界中に恥を晒すこともあり得るので気を付けましょう。

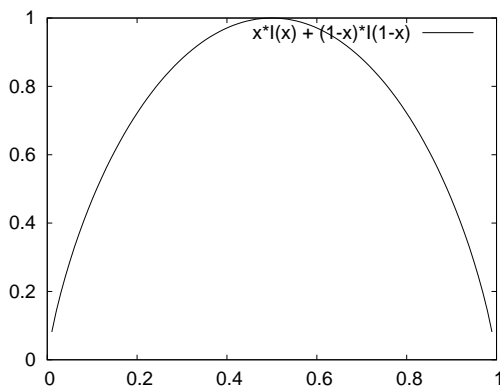


図 1: gnuplot で作ったグラフ

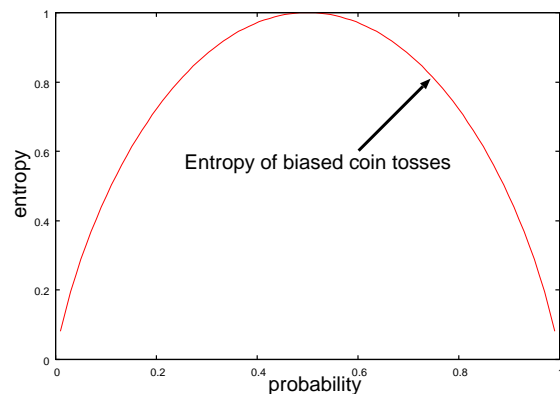


図 2: 字を書き込んだグラフ

本日のやってみよう#2

eps ファイルを取り込んでみよう

1. gnuplot を用いて

$$H(P) = -P \log_2 P - (1 - P) \log_2(1 - P)$$

のグラフを作成し, eps 形式で保存する. グラフの描画には,

```
I(p) = - log(p)/log(2)
```

```
plot [0:1][0:1] x*I(x) + (1-x)*I(1-x)
```

を実行すればよい. 図1のようになるはず. eps ファイルを画面で確認したいときは, evince コマンドが使うべし.

2. 同じグラフを gnuplot を使って svg 形式で書き出し, inkscape を使ってグラフ中に “Entropy of biased coin tosses” と記入する. 他にも, 凡例を削除したり x 軸と y 軸の説明を書き込んだりするといい感じだが, この辺はどっちでもよい. eps 形式で 1. とは異なる名前でセーブする. 出来上がりは図2の感じ.
3. 1. と 2. で作成した 2 つの eps ファイルを, 先週のプログラミング演習 1 の授業で作成した L<sup>A</sup>T<sub>E</sub>X の練習文書に取り込む.
  - 取り込んだ図表には \caption と \label を付けておくこと.
  - 今日の資料のように横に並べる必要はありません.

### 3 図や式の番号を参照しよう

図や表, 式の番号は, それぞれの環境 (figure 環境や table 環境, equation 環境) を使うと自動的に L<sup>A</sup>T<sub>E</sub>X により割り振られます. これらの番号を参照するには, \ref コマンドを使います. \ref コマンドは引数を一つとり, その引数には \label コマンドで各環境につけたラベルを記述します. また, \section などにラベルを付けることで, 章番号や節番号も \ref コマンドで参照できます.

また, 章や節にもラベルを付けることができます. \section や \subsection の直後に \label{sec:foobar} のように記述すると, その章や節の番号を \ref{sec:foobar} のように参照することができます.<sup>4</sup>

これらの相互参照機能は論文や長めのレポートを作成したりしたとき大変役に立ちますので是非使いこなせるようになっておいて下さい.

<sup>4</sup>“sec:foobar” の部分がラベルで, 好きな名前を使えばよいのですが, セクション番号なら sec: で始め, 図なら fig: で始め, 式なら eq: で始める, といったネーミングが一般的です.

### 本日のやってみよう#3

#### 相互参照を使ってみよう

1. 先週の講義で作成した  $\LaTeX$  文書の章と節すべてに `label` を付ける .

- `\section` コマンドや `\subsection` の直後に `\label{sec:foo}` (foo の部分は同じものが同一文書中出现しないよう自分で決める) のように記述する .

2. 本文の最初の章 ( `section` ) にある「第 2 章」や「第 3 章」の数字の部分を `\ref` コマンドを使って相互参照に変更する .

3. 文書中の数式 ,

$$H = \sum_{k=1}^m P(a_k) I(a_k) = - \sum_{k=1}^m P(a_k) \log_2 P(a_k)$$

と ,

$$\begin{aligned} H &= -0.60 \log_2 0.60 - 0.25 \log_2 0.25 - 0.10 \log_2 0.10 \\ &\quad - 0.05 \log_2 0.05 \\ &= 1.491 \text{ (ビット)} \end{aligned}$$

に式番号が付くように変更をする . すでに式番号が付くようになっていれば変更は不要 .

- $\LaTeX$  の数式では , `\[ ~ \]` や `$ ~ $` で書かれた数式には番号が付きません . `equation` 環境や `eqnarray` 環境で書かれた数式には番号が付きます .

4. 3. で番号が付くよう変更した数式にラベルを付ける .

- 数式環境開始直後に , `\label{eqn:bar}` の形で記述する .

5. 文中で上の 2 つの式を参照している部分 ( 第 2 章の終わりの方と , 第 3 章の右段中央付近 ) を , `\ref` コマンドを使った相互参照に変更する . 数字を直に書いてはいけない . また , 括弧は自動で付かないので自分で記述すること .

6. `platex` を実行した結果 , 参照している部分が??のようになって正しく表示されない場合は , もう一度 `platex` を実行すればよい .