```
Denmark      3 23.8
⋮
Switzerland  5  1.5 /
```

The question is: how should the macro \placebars be defined? Notice that the y coordinates of the bars decrease by 1 for each bar. This can be achieved by stepping a counter. It's okay to use \count0 for this purpose, because PICture making goes on inside a group. The following definitions get the ball rolling:

```
1  \def\placebars{%
2    \count0 =0
3    \Placebars}
4  \def\Placebars #1 #2 #3 {%
5    % #1=Country, #2=last digit of year,
6    % #3=rate
7    \advance \count0 by -1
8    \putrule from  0 \the\count0  to
9      #3 \the\count0
10   \put {#1 \sevenrm 7#2} [Br]
11     <-5pt,-2pt> at  0 \the\count0
12   \repeatifnecessary}
```

The line numbers aren't part of the macros; they're just for ease of reference. The \Placebars macro uses blanks to delimit its arguments, so you'll have to enter 'West Germany' as West~Germany.

\Placebars puts one labeled bar into the PICture each time its invoked. In order to run through the entire list of countries you could define \repeatifnecessary as follows:

```
13 \def\repeatifnecessary{%
14   \futurelet\next\Repeatifnecessary}
15 \def\Repeatifnecessary{%
16   \ifx /\next
17     \expandafter \finish
18   \else
19     \expandafter \Placebars
20   \fi}
21 \def\finish / {}
```

This is a little complicated, so let's take it one step at a time. The \futurelet on line 14 tells TEX to set \next equal to the character following whatever country-year-rate entry has just been processed, and to go on to expand \Repeatifnecessary. If \next is not '/', the \ifx on line 16 directs TEX to the "false" clause on line 19. The \expandafter there makes TEX expand the \fi on line 20 thereby finishing off the \ifx ... \fi structure. TEX then goes back to the \Placebars on line 19, which is now sitting directly in front of the next country-year-rate entry.

On the other hand, if \next is '/' (so the entry just processed was 'Switzerland 5 1.5'), the \ifx directs TEX to the "true" clause on line 17. The \expandafter there leads to the \else on line 18; \else ... \fi expands to ⟨null⟩, because the condition is true. TEX then goes back to the \finish on line 17, which gobbles up the '/␣' following the 'Switzerland' entry, thereby preventing these characters from showing up in your PICture. Finally TEX goes on to read whatever comes next, which in this case would be \endpicture.

The \placebars macro would be especially handy if you were going to make a lot of bar graphs like Figure 6. You should put the definitions of \placebars, ..., \finish outside a PICture, so that these macros won't vanish when the PICture ends.

What would happen if you were to inadvertently enter '\placebars /'? Things would get all screwed up, because the terminator '/' would become the first argument to \Placebars. To correct this flaw in the macros, you should change '\Placebars' on line 3 to '\repeatifnecessary'.

With a view towards generalization, you should make two more simple changes so that the definition of \Repeatifnecessary doesn't involve \Placebars. Specifically, replace '\Placebars' on line 19 by '\repeatwhat', and insert

```
\let \repeatwhat=\Placebars
```

between line 2 and (the new) line 3. The repeat structure now can be used with other macros besides \Placebars.

To sum up, the solution to the \placebars exercise has led to a useful TEXnique for defining upper level plot commands that meet specific needs.

Editor's note: One statement in this article bears repeating: "A large PICture with several curves will exceed the capacity of a standard version of TEX." It happened to this article. TUGboat is produced with a VAX/VMS TEX implementation that has had its memory increased to nearly the maximum possible for a "standard" version: $mem\_max = 65500$. To prepare this article successfully, it was necessary to (1) produce each column as a separate page, and put it together using the output driver's electronic pasteup capability; (2) insert explicit page breaks in columns containing especially taxing PICtures; (3) strip unnecessary details out of the header macros (tugbot.sty); and (4) run it by itself. Even so, the news with \tracingstats turned on was thought-provoking — only 8 words of memory untouched. Michael Wichura is using a C version of TEX without the ordinary memory restrictions.

The LaTeX User's Column

Jackie Damrau
University of New Mexico

I have received four questions since the last column. Please keep those questions or helpful hints coming. Remember, they will be answered as soon as possible via electronic mail (if possible) and then published in the next TUGboat. Until then, happy LaTeXing.

Question 1

Version 2.09 of LaTeX contains the following bug, which I have not seen reported elsewhere, possibly because it would be discovered by a LaTeXnician only when indulging in the aesthetically bizarre practice of using marginal notes and footnotes concurrently. Here it is:

If a `\marginpar` follows a `\footnote` on the same page, then the vertical positioning of the marginal will probably be incorrect. To fix it, the definition of `\@specialoutput` needs to be changed, as shown below, by moving part of the conditional statement `\ifvoid\footins...` so that it is not executed in the case when the output routine is called by `\marginpar`.

```
\def\@specialoutput{%
  \ifnum\outputpenalty > -\@Mii    \@doclearpage
  \else  \ifnum \outputpenalty <-\@Miii
      \ifnum\outputpenalty<-\@MM \deadcycles\z@\fi
      \global\setbox\@holdpg\vbox{\unvbox\@cclv}
    \else \setbox\@tempboxa\box\@cclv
      \@pagedp\dp\@holdpg \@pageht\ht\@holdpg
      \unvbox\@holdpg
% From here:  \ifvoid\footins\else
%      \advance\@pageht\skip\footins
%      \advance\@pagedp\dp\footins
%      \insert\footins{\unvbox\footins}\fi
    \@next\@currbox\@currlist{%
      \ifnum\count\@currbox >\z@
        \ifvoid\footins\else   %  :to here...
          \advance\@pageht\ht\footins
          \advance\@pageht\skip\footins
          \advance\@pagedp\dp\footins
          \insert\footins{\unvbox\footins}\fi %
        \@addtocurcol
      \else \ifvoid\footins\else %  ...and here.
          \insert\footins{\unvbox\footins}\fi %
        \@addmarginpar
      \fi}\@latexbug
    \ifnum\outputpenalty <\z@ \penalty \z@ \fi
  \fi\fi}
```

Chris Rowley
The Open University
Parsifal College
527 Finchley Road
London NW3 7BE

Answer from Leslie Lamport:

Chris Rowley has indeed found a bug in LaTeX, and his fix appears to work. I will probably incorporate it. Relay my thanks to him.

Question 2

Several of us here at the Hughes Aircraft Company Albuquerque Engineering Laboratory are trying to use TeX to produce various sorts of documents. One such application is the making of so-called vugraphs, i.e., landscape orientation charts printed on transparent plastic sheets for use with overhead projectors. The way it works out is that the title or heading of the chart needs to be centered in a field about 4 by 18.5 cm, with text then occupying a separate field about 15 by 25.5 cm. A form has been enclosed to indicate these fields. SLITeX has a lot of good features for this process, but its bias toward portrait mode and vertically centering all text as a unit results in a fair amount of ad hoc and tedious use of `\vspace`, etc. Is there any way to specify separate fields on a slide? Or, better, to describe a subpage by a single command?

We would also be interested in any advice on how to incorporate graphs into the files from which the charts are made.

Richard C. Smith
Hughes Aircraft Company
Albuquerque, New Mexico

Answer from Leslie Lamport:

What Richard Smith wants to do is easy, and there are numerous ways to do it. However, they all involve some understanding of raw TeX. First of all, the vertical centering is achieved by the `\vfil` commands in the definition of `\@makecol` in `slitex.tex`. Removing the `\vfil`'s from this definition will stop SLITeX from vertically centering the slides.

However, there's really no need to remove the vertical centering. Instead, by putting things inside `\vbox`'s of the appropriate height, he can arrange it so that the height of material on a page equals `\textheight`, so vertical centering has no effect. Exactly how he does this depends upon how he wants to enter the input.

The most elegant approach would be to modify the slide environment (by changing the definitions of `\slide` and `\endslide` in `slitex.tex`) so he would just type

```
\begin{slide}{}
The title goes here
```

```
\midslide
```
The text field goes here.
```
\end{slide}
```

This takes a little TEX hacking ability, since one has to begin a `\vbox` in the `\slide` command and end it in the `\midslide`, etc. My guess is that he will need to find a TEX hacker to do this for him, or else take the time to read the TEXbook.

Question 3

I recently sent a query to TEXhax, and Leslie Lamport was good enough to send a reply. Apparently my problems were compounded by my using an out-of-date version of `rep12.sty`. I've subsequently fetched a load of LATEX files from Peter Abbott's software-depository at Aston, and hope that this will help to avoid future problems.

To help people generally avoid such problems, it would be very useful if there was available somewhere a list of the version-numbers/dates of all TEX-related software, so that people could easily check whether they have the most recent version. On the other hand, the range of TEX-related software is now vast, so:

- whoever ran the "somewhere" would have a lot of work to do
- the information would take up a lot of space in TUGboat and/or TEXhax
- difficult questions of where to draw the line might arise. (Are macro package X and style-option Y so popular that the whole world wants to keep their copies up-to-date?)

The problem of where to draw the line is not so difficult for LATEX. Anyone using LATEX will want to keep `lfonts.tex`, `latex.tex`, `lplain.tex` and the standard style- and option-files (the ones mentioned in the book and distributed by Maria Code and Pierre MacKay) up-to-date. Running a "somewhere" that listed the current version-numbers and/or dates for this standard software would I think be manageable.

Would you consider using the **LATEX User's Column** in TUGboat to give the version-numbers/dates of the copies of `lfonts.tex`, `latex.tex`, `lplain.tex`, the standard `*.doc` and the standard `*.sty` that people should be using? I guess that it could be done in about 1.5 column-inches if a suitably small font was used, so it might be acceptable as a standard item in each TUGboat.

Obviously, the problem of keeping non-standard style and options files (e.g. as distributed by Ken Yap) up-to-date is important too, but it seems a less manageable problem and I can't see such an obvious solution.

David Rhead
BITNET:
David_Rhead@USYSTEM.CCC.NOTTINGHAM.AC.UK

Question 4

After I sent you my last mail, I sent a copy of the next-to-last paragraph to Leslie Lamport for information. He mailed back a suggestion that the information I need is in `latex.bug`.

So, it would actually be sufficient if people's attention was drawn to the date/version-number of the current `latex.bug`. If they found that they had an out-of-date `latex.bug`, they could take steps to acquire an up-to-date one and, from the up-to-date `latex.bug`, work out what other files they need.

In my situation, for example, I can get files from Aston fairly easily, and Peter Abbott at Aston gets files from SCORE, which saves other UK-people having to find them to work out how to get stuff from SCORE. I have just fetched `latex.bug_62` from Aston. It mentions stuff up to the end of '87, so it looks fairly up-to-date, but I don't actually know for certain whether or not a more recent `latex.bug` is now available somewhere.

So, would you consider using the **LATEX User's Column** in each TUGboat to state the version-number of the latest `latex.bug`? It need not take more than two sentences, e.g.,

> Details of the current versions and "dates-last-updated" of all the standard LATEX software are to be found in version 63 of `latex.bug`. If you don't already have version 63, acquire it first, so that you can work out what else you need.

David Rhead
BITNET:
David_Rhead@USYSTEM.CCC.NOTTINGHAM.AC.UK

Answer (Question 3 and 4)

I have no objections for using my column to announce the above. I would however need someone to supply me with the information. If there is a kind soul out there who would be willing to send me the information, it would be my pleasure to give the LATEX community this added help.

Editor's note: Your editor has access to this information, and will, if reminded, be happy to check the status of `latex.bug` as one of the last chores before putting TUGboat to bed. This is the current status at Score, as of June 30:

`latex.bug.79  12 May 88  22:44:10`