Abstract

This article describes an enhancement of the \LaTeX picture–environment to draw histogram bars.

1 User interface

\texttt{\textbackslash histogram} \hspace{1em} This is a macro collection to draw histogram bars inside a \texttt{picture–environment}. Use is as follows:

\texttt{\textbackslash histogram(x_0,y_0)(x_1,y_1)...(x_n,y_n)}

The coordinate pairs specify the upper left corner of the histogram bars, i.e. this will draw a horizontal line from \((x_i, y_i)\) to \((x_{i+1}, y_i)\), then a vertical line from \((x_{i+1}, y_i)\) to \((x_{i+1}, y_{i+1})\) if \texttt{\textbackslash noverticalines} was specified, else from \((x_{i+1}, y_0)\) to \((x_{i+1}, \max(y_i, y_{i+1}))\).

Default is \texttt{\textbackslash verticalines}. \(y_0\) should be less or equal the minimum of all the \(y_i\) (i.e. other cases have not been tested).

Let’s start with an example: to get the following picture:

\begin{center}
\begin{tikzpicture}
\draw[->] (-0.5,0) -- (5.5,0) node[above] {ml};
\draw[->] (0,-0.5) -- (0,7) node[right] {n};
\draw[thick] (0,0) rectangle (5,7);
\draw[thick] (0,5) -- (1,5);
\draw[thick] (1,5) -- (1,6);
\draw[thick] (1,6) -- (2,6);
\draw[thick] (2,6) -- (2,7);
\draw[thick] (2,7) -- (3,7);
\draw[thick] (3,7) -- (3,8);
\draw[thick] (3,8) -- (4,8);
\draw[thick] (4,8) -- (4,9);
\draw[thick] (4,9) -- (5,9);
\end{tikzpicture}
\end{center}

Behandler 1

*This file has version number v1.01, last revised 1997/02/13.*
I used these \LaTeX{} commands:
\begin{verbatim}
\setlength{\unitlength}{1mm}
\begin{picture}(100,65)(-10,-15)
\thicklines
\put(0,-3){\vector(0,1){50}}
\put(-3,0){\vector(1,0){90}}
\thinlines
\put(0,0){\line(0,-1){2}}
\put(2,0){\line(0,-1){2}}
\put(20,0){\line(0,-1){2}}
\put(22,0){\line(0,-1){2}}
\put(40,0){\line(0,-1){2}}
\put(42,0){\line(0,-1){2}}
\put(60,0){\line(0,-1){2}}
\put(62,0){\line(0,-1){2}}
\put(0,-1){\makebox(2,0)[t]{\small 1}}
\put(20,-1){\makebox(2,0)[t]{\small 2}}
\put(40,-1){\makebox(2,0)[t]{\small 3}}
\put(60,-1){\makebox(2,0)[t]{\small 4}}
\put(70,-1){\makebox(0,0)[t]{ml}}
\put(0,10){\line(-1,0){2}}
\put(0,20){\line(-1,0){2}}
\put(-3,8){\makebox(0,4)[r]{5}}
\put(-3,18){\makebox(0,4)[r]{10}}
\put(-3,30){\makebox(0,4)[r]{n}}
\put(15,-10){Behandler 1}
\histogram(0,0)(0,4)(2,4)(4,4)(6,0)(8,10)(10,8)(12,6)(14,4)
(16,14)(18,8)(20,18)(22,18)(24,8)(26,0)(28,10)(30,2)
(32,12)(34,4)(36,6)(38,6)(40,18)(42,10)(44,14)(46,4)
(48,8)(50,4)(52,6)(54,4)(56,6)(58,2)(60,2)(62,0)
\end{picture}
\end{verbatim}

\section{Implementation}

Here’s how it is implemented: first we allocate three counters that are needed later on. $\texttt{\textbackslash hist@x}$ and $\texttt{\textbackslash hist@y}$ are the $x$ and $y$ coordinate of the current point, i.e. the point that serves as a start for the next box of the histogram. $\texttt{\textbackslash hist@ystart}$ holds the $y$ coordinate of the first point, i.e. $y_0$.

We need a switch to decide if the vertical lines of the histogram boxes are to be drawn from $y_i$ to $y_{i+1}$ or from $y_0$ to $\max(y_i, y_{i+1})$. Default is the latter.
The \texttt{histogram} command takes the starting point as argument and initializes the counters. \texttt{hist@x}, \texttt{hist@y} and \texttt{hist@ystart} are set to \texttt{x}_0, \texttt{y}_0 and \texttt{y}_0, respectively.

\begin{verbatim}
def\histogram(#1,#2){\hist@x #1 \hist@y #2 \hist@ystart\hist@y

\hist@next}
\end{verbatim}

Then the macro \texttt{\hist@next} is used.

\begin{verbatim}
def\hist@next\@ifnextchar ({\hist@box}{\hist@end}
\end{verbatim}

\texttt{\hist@box} does nearly all the work. The first thing to do is to set the temporary counter \texttt{@tempcnta} to \texttt{x}_i + 1 - \texttt{x}_i. Remember that \texttt{hist@x} is the \texttt{x} coordinate of the last point (i.e. \texttt{x}_i) whereas the macro’s first argument is \texttt{x}_{i+1}. So we write

\begin{verbatim}
def\hist@box(#1,#2){\@tempcnta -\hist@x
\advance\@tempcnta #1
\end{verbatim}

The next step is easy: draw the horizontal part of the histogram box. The line starts at \texttt{(x}_i, \texttt{y}_i) and has length \texttt{@tempcnta unitlength}.

\begin{verbatim}
\ifnum \@tempcnta >\z@
\put(\hist@x,\hist@y){\line(1,0){\@tempcnta}}\else
\put(\hist@x,\hist@y){\line(-1,0){-\@tempcnta}}\fi
\end{verbatim}

Now set \texttt{hist@x} to \texttt{x}_{i+1}:

\begin{verbatim}
\hist@x #1
\end{verbatim}

If \texttt{\verticalines} was set we first set \texttt{@tempcnta} to max(\texttt{y}_i, \texttt{y}_{i+1})

\begin{verbatim}
\ifhist@vert
\ifnum \hist@y >#2 \@tempcnta\hist@y
\else \@tempcnta #2 \fi
\fi
\end{verbatim}

then we set \texttt{@tempcntb} to the same value and \texttt{@tempcnta} to the length of the line to draw.

\begin{verbatim}
\@tempcntb\@tempcnta
\advance\@tempcnta -\hist@ystart
\end{verbatim}

We draw the line

\begin{verbatim}
\put(\hist@x,\@tempcntb){\line(0,-1){\@tempcnta}}\%
\end{verbatim}

which finishes this case.

\begin{verbatim}
\else
\end{verbatim}

In the other case (i.e. if \texttt{\noverticalines} was set) we have to draw a line from \texttt{y}_i to \texttt{y}_{i+1}. We set \texttt{@tempcnta} to \texttt{y}_{i+1} - \texttt{y}_i

\begin{verbatim}
\@tempcnta -\hist@y
\advance\@tempcnta #2
\end{verbatim}
and draw the line.

\ifnum \@tempcnta >\z@
  \put(\hist@x,\hist@y){\line(0,1){\@tempcnta}}\else
  \put(\hist@x,\hist@y){\line(0,-1){-\@tempcnta}}\fi

Thus endeth the drawing.

Finally we set \hist@y to $y_{i+1}$ and call \hist@next to look for the next coordinate pair.

\hist@y #2\hist@next}

There is only one thing we left out: what if there is no more open parenthesis? That's the easy part: do nothing.

\def\hist@end{}

Frank Mittelbach suggested that the $x$–coordinate should specify the midpoint of the histogram bar, not the upper left corner. However, I don’t see how this will work if the bars have different widths. What do you think about it?

Well, that’s all. Use it and enjoy.