

PSTricks

pst-bar

Balken- und Säulendiagramme mit pstricks

v.0.92

12. März 2009

Documentation by
Alan Ristow
Herbert Voß
Juliane Horn (Übersetzung)

Package author(s):
Alan Ristow

Mithilfe von `pst-bar` kann man `pstricks` verwenden, um Balkendiagramme direkt aus einer Daten-Datei zu erzeugen. Dieses Paket befindet sich allerdings noch im Beta-Stadium – die üblichen Warnungen bezüglich Beta-Software treffen daher zu. Es wird zusätzliche Merkmale und verbesserte Dokumentationen (den Inhalt sowie die Ausstattung betreffend) geben, wenn sich der Code stabilisiert.

1 Einführung

Aufgrund der Leistung und Flexibilität von `pstricks` [1] ist es möglich, mit `pst-bar` Balkendiagramme zu erstellen, deren Einzelheiten in einer mit Komma abgetrennten Datei gespeichert sind. Es gibt mehrere Lösungen, um Säulendiagramme in $\text{T}_\text{E}\text{X}$ und $\text{L}^{\text{A}}\text{T}_\text{E}\text{X}$ -Dokumenten zu „zeichnen“. Die offensichtliche Lösung ist, ein externes Programm wie `gnuplot` oder `Matlab` zu verwenden, das Balkendiagramme in einem Format speichert, welches direkt von $\text{T}_\text{E}\text{X}$ oder $\text{L}^{\text{A}}\text{T}_\text{E}\text{X}$ gelesen werden kann. Diese Methode hat den Nachteil, dass zusätzlicher Aufwand nötig ist, um den Diagrammen ein einheitliches Erscheinungsbild zu geben, welches zu anderen Grafiken im Dokument passt. In vielen Fällen kann der Diagrammtext von $\text{L}^{\text{A}}\text{T}_\text{E}\text{X}$ mithilfe der Pakete `psfrag` oder `psfragx` eingefügt werden (das `Matlab`-Programm `LaPrint` kann diesen Prozess bei `Matlab`-Diagrammen beschleunigen, da es speziell für diesen Zweck entwickelt wurde [2]). Jedoch schreiben einige Programme die Textelemente in EPS (Encapsulated Postscript)-Dateien, sodass die Optionen von `psfrag` nicht funktionieren.

Balkendiagramme können aber auch mit den eingebauten Zeichenwerkzeugen in $\text{T}_\text{E}\text{X}$ und $\text{L}^{\text{A}}\text{T}_\text{E}\text{X}$ gezeichnet werden. Dafür wird das `histogr` Paket [3] verwendet, welches neben Histogrammen auch sehr einfache Balkendiagramme erstellen kann. Es scheint eine Annäherung an das `bar` Paket [4] für $\text{L}^{\text{A}}\text{T}_\text{E}\text{X}$ 2.09 zu sein, da es eine neue Umgebung für das Zeichnen von Säulendiagrammen definiert. Allerdings sind die eingebauten Zeichenwerkzeuge begrenzt und die Arbeit am `bar` Paket wurde wohl nach 1994 beendet.

Es wurden technisch verfeinerte Zeichenwerkzeuge für $\text{T}_\text{E}\text{X}$ und $\text{L}^{\text{A}}\text{T}_\text{E}\text{X}$ entwickelt; die bekanntesten sind `eepic`, `texdraw` und `latexdraw`, `LaTeXPiX`, `MetaPost`, `mfpic`, und `pstricks`. Mit jedem dieser Mittel können Balkendiagramme gesetzt werden; jedoch besitzt keines eine eingebaute Unterstützung für diese, sodass die Diagramme mit der „Brachialgewalt“-Methode angefertigt werden müssen, das bedeutet ein manuelles Einzeichnen von Kästen, welche die Balken im Diagramm darstellen. Je nach Werkzeug kann es sein, dass auch die Achsen manuell eingezeichnet werden müssen. Das `pst-bar` Paket soll den Prozess des Zeichnens von Balken- und Säulendiagrammen unter Verwendung von `pstricks` automatisieren.

Derzeit gibt es zwei Möglichkeiten für die Erstellung von Balkendiagrammen mit `pstricks`. Die Erste ist die oben genannte Brachialgewalt-Methode, bei der eine Reihe von `\psframes` mit passenden Höhen, Breiten und Farben gezeichnet werden, um zum gewünschten Ergebnis zu gelangen [5]. Zwar ist diese Möglichkeit vortrefflich und bietet den größten Spielraum, allerdings ist sie mühsam und arbeitsintensiv. Die zweite Option ist das `bardiag` Paket; es automatisiert im Wesentlichen [6] die Vorgehensweise mit brachialer Gewalt im. Dieser Ansatz ist wirkungsvoll, jedoch benötigt `bardiag` zahlreiche externe Pakete. Weiterhin verwendet es $\text{L}^{\text{A}}\text{T}_\text{E}\text{X}$, um mathematische Befehle durchzuführen und es ist nicht kompatibel mit dem normalen $\text{T}_\text{E}\text{X}$.

Durch `pst-bar` werden diese Probleme vermieden. Es beruft sich ausschließlich auf Pakete der `pstricks`-Familie, stellt umfassende Anpassungsfunktionen zur Verfügung, baut auf Postscript für fast alle mathematischen Operationen und sollte mit dem einfachen $\text{T}_\text{E}\text{X}$ kompatibel sein (bisher ungetestet). Damit Balkendiagramme angefertigt und beschriftet werden können, benutzt `pst-bar` Befehle, um die Angaben der durch Komma abgegrenzten Datei zu lesen. Vielfache Datenreihen können dabei entweder gesammelt oder gestapelt grafisch dargestellt werden; die Balken können

horizontal oder vertikal eingezeichnet werden. Eine zusätzliche Option ermöglicht es, Blockdiagramme zu erzeugen, welche die Unterschiede zwischen den nachfolgenden Reihen von Daten in der Datendatei zeigen (siehe auch Kapitel 3.3).

Zu diesem Zeitpunkt sollte `pst-bar` noch als Beta-Software betrachtet werden. Es wird nicht garantiert, dass die Änderungen am Code, welche zwischen dieser Ausgabe und der ersten dauerhaften Ausgabe auftreten, auch rückwärts kompatibel sind.

2 Erstellen von Balken- und Säulendiagrammen

Das Zeichnen von Balkendiagrammen mit `pst-bar` erfolgt in drei Schritten: (i) Lesen der Datendatei, (ii) Zeichnen der Balken und (iii) Einzeichnen der Achsen. Die ersten beiden Schritte erfolgen direkt durch `pst-bar`. Für den dritten ist das `pst-plot` Paket notwendig, das fester Bestandteil von `pstricks` ist.

Die folgenden Abschnitte befassen sich mit dem Format der Datendatei, mit dem Laden von Daten aus der Datei und mit dem Erstellen eines Balkendiagramms mithilfe dieser Daten.

2.1 Format der Datendatei

Die Datei mit den Daten muss durch Komma begrenzt sein und jede Reihe muss die gleiche Anzahl an Einträgen enthalten. Eine Kopfzeile ist erlaubt, aber kein Eintrag in dieser Zeile darf Kommas oder `\par` Befehle beinhalten (gilt ebenso für `\\`). Eine der Dateien, welche für diese Dokumentation benutzt werden, sieht so aus:

```
Set 1, Set 2, Set 3
1, 2, 3
1, 2, 3
```

In diesem Fall sind `Set 1`, `Set 2`, und `Set 3` die Kopfzeilen. Die nachfolgenden Reihen, später als *data rows* gekennzeichnet, umfassen die Daten, die mit den Überschriften verknüpft werden. Die Datei wird in einem säulenähnlichen Stil angeordnet, sodass man mit `Set 1` die Daten 1 und 1 verbindet, mit `Set 2` die Daten 2 und 2 und so weiter. Zuerst werden die Daten grafisch dargestellt und anschließend mit dem Text der Kopfzeile beschriftet.

Achtung! Es gibt zwei sehr wichtige Bedingungen für die Daten innerhalb der Datei:

- Alle Daten außerhalb der Kopfzeile müssen numerisch sein. Falls keine Kopfzeile vorhanden ist, müssen alle Daten der Datei numerisch sein.
- Jede Zeile muss dieselbe Anzahl von Einträgen haben.

Werden diese Bedingungen nicht eingehalten, können Postscript-Fehler oder (seltener) falsch beschriftete oder gezeichnete Balkendiagramme die Folge sein.

Wie diese Dateien grafisch dargestellt werden, hängt von der Art des gewünschten Diagramms ab. Siehe dazu Kapitel 3.1 – 3.3 und die Beispiele in Kapitel 4.

2.2 Lesen der Datendatei

Der Befehl `\readpsbardata` liest die Datei:

```
\readpsbardata[<options>]{<data>}{<filename>}
```

Geben Sie für die Daten einen Makro-Namen an, den Sie noch nicht benutzen.

Die verfügbaren Optionen sind:

Parameter	Definition
header	true, wenn die erste Zeile der Datei eine Kopfzeile ist false, wenn true nicht zutrifft Standard: true

Setzt man header auf false, so geht pst-bar davon aus, dass kein header (Kopfzeile) existiert und die erste Zeile wird als Daten-Zeile angesehen.

2.3 Einzeichnen der Balken

Der Befehl `\psbarchart` zeichnet die Balken für das Diagramm.

```
\psbarchart[<options>]{<data>}
```

Für <data> verwenden Sie bitte den Makro-Namen, den Sie bei `\readpsbardata` ausgewählt hatten. Die Säulen werden in Spalten eingeteilt, die eine `\psxunit` breit sind. Die Anzahl der Säulen in einer Spalte entspricht dabei der Zahl der Datenreihen in einer Datei. Die Unterseite jeder Spalte wird mit den Überschriften (header) der Datei beschriftet, wenn man bei `\readpsbardata` für header true angegeben hat. Allerdings wird nur die Balken- und Säulenbeschriftung erstellt – der Rahmen und die Achsen müssen extra bestimmt werden.

Die verfügbaren Optionen sind:

Parameter	Definition
chartstyle	cluster erzeugt eine Gruppe von Balken stack sammelt die Gruppen in einer Einzelspalte block erstellt einen schwebenden Balken (Minimum ist nicht 0) Standard: cluster
barstyle	Name(n) des Balken-Stils für jeden Balken Standard: {black,darkgray,gray,lightgray,white,red,green,blue}
barcolsep	beeinflusst den Zwischenraum zwischen den Spalten Standard: 0.4
barsep	beeinflusst den Zwischenraum zwischen den Balken (cluster und block Diagramme) Standard: 0.0
barlabelrot	Drehwinkel der Spaltenbeschriftung Standard: 0
orientation	vertical für vertikal gezeichnete Balken horizontal für horizontal gezeichnete Balken Standard: vertical

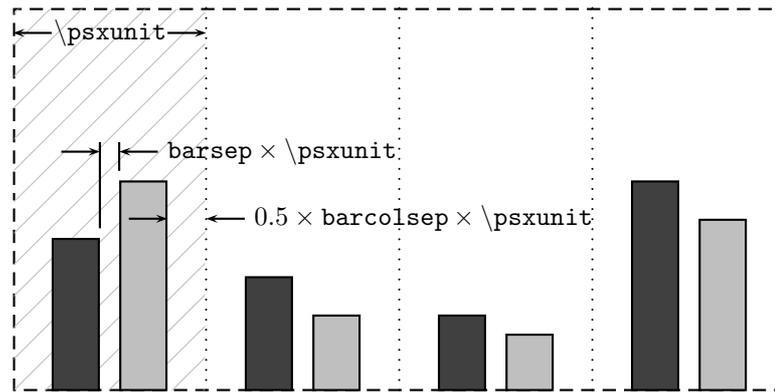


Abbildung 1: Schematische Darstellung für das Layout des Balkendiagramms. Die schraffierte Fläche markiert den Bereich, in dem die Daten der ersten Spalte der Datei dargestellt sind. Die Breite der Balken entspricht $[(1 - \text{barsep} - 0.5 \times \text{barcolsep}) \times \text{\psxunit}] / N$, wobei N für die Anzahl der Balken in einer Spalte steht.

Die Anzahl der angegebenen `barstyles` muss der Anzahl von Balken in jeder Spalte entsprechen oder größer als diese sein. Andernfalls entsteht ein Postscript-Fehler. Bei den `cluster` und `stack` `chartstyles` entspricht dies den Datenreihen in der Datei; beim `block` `chartstyle` entspricht es der halben Anzahl der Datenreihen. Außerdem ist bei der Angabe der `barstyles` darauf zu achten, dass die Stile in geschweiften Klammern angegeben werden, wie z. B.:

```
\psbarchart [Options] {...}
```

Die verfügbaren `barstyles` sind standardmäßig `red`, `green`, `blue`, `black`, `white`, `gray`, `lightgray`, und `darkgray`. Jeder davon erzeugt einen Balken in der angegebenen Farbe mit rechtwinkligen Ecken und schwarzem Umriss. Mit dem Befehl `\newsbarstyle` können zusätzlich neue `barstyles` definiert werden, siehe dazu Kapitel 2.4.

Abbildung 1 zeigt, wie `\psbarchart` die Balken im Standard-`chartstyle`, `cluster`, anlegt. Beim `stack` `chartstyle` wird die `barsep` Option nicht berücksichtigt.

2.4 Anpassung des Diagramms

Um einen angefertigten Balkentyp hinzuzufügen, ist der Befehl `\newsbarstyle` zu verwenden:

```
\newsbarstyle{<name>}{<definition>}
```

Diese Bezeichnung ist eine Textfolge, die derzeit von keinem anderen `barstyle` verwendet wird. Die Definition kann sich aus jedem `pstricks` Schlüssel bzw. aus jeder Gruppe von Schlüsseln ergeben, die mit `\psframe` anwendbar sind. Zum Beispiel wird `red` `barstyle` wie folgt definiert:

```
\newsbarstyle{red}{fillcolor=red,fillstyle=solid,framearc=0}
```

Um den Typ, mit dem die Spaltenbezeichnungen festgelegt werden, anzupassen, wird `\psbarlabel` neu definiert. Wenn die Beschriftung aus kleinen Kursivbuchstaben be-

stehen soll, sieht diese so aus:

```
\renewcommand* \psbarlabel{}
```

Üblicherweise erstellt `\psbarlabel` die Spaltenbeschriftung in der aktuellen Schriftart.

Um den Abstand zwischen den Säulen und der Spaltenbeschriftung zu regulieren, wird `\psbarlabelsep` neu definiert. Dabei ist es als Befehl, nicht als Umfang, definiert und muss daher mit `\renewcommand*` wiederdefiniert werden. Als Standard benutzt man `\theta`.

Am Ende ist es möglich, die Daten der Datei mit `\psbarscale` anzupassen und zu manipulieren, wie z. B.

```
\psbarscale(<scale>){<Postscript code>}
```

welches von Herbert Voß' `\pstScale` aus `pstricks-add` beeinflusst wurde [7]. Die Daten werden durch den Wert in Klammern eingestuft und können im Weiteren mit dem Postscript Code gesteuert werden. Für den Fall, dass man den Logarithmus der eingegebenen Daten grafisch darstellen will, würde man `\psbarscale(1){log}` benutzen. Der Postscript Code bezieht sich auf die Daten vor der Skalierung.

3 Die Arten von Diagrammen

Wie bereits in 2.3 beschrieben, gibt es drei verschiedene Möglichkeiten für den Parameter `chartstyle`. Der Standard-Typ `cluster` ist ein allgemeines Balkendiagramm, bei dem die Balken aufgrund ähnlicher Daten gruppiert werden. Die zweite Option, `stack`, zeichnet eine Reihe von Säulen aufeinander gestapelt, statt sie nebeneinander in einer Gruppe anzuordnen. Benutzt man `block`, die dritte Alternative, so werden angehäufte Balken zwischen zwei Datenpunkten eingezeichnet und somit eine Spanne von Werten anstelle eines einzelnen Wertes veranschaulicht.

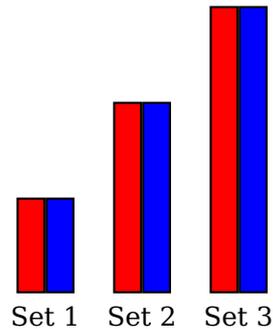
Die folgenden Abschnitte beschreiben jeden `chartstyle` im Detail und erklären, wie die Eingabedaten verwendet werden. Zusätzlich werden für jeden Fall Muster gezeigt. Im Kapitel 4 sind vollständige Beispiele einschließlich der Achsen aufgezeigt.

3.1 cluster

In einem `cluster` Diagramm wird jede Variable in einer Reihe, die durch Kommas getrennt ist, als ein Balken in einer einzelnen Spalte dargestellt. Jede Reihe steht daher für eine neue Balkeneinheit. Die Datei

```
Set 1, Set 2, Set 3
1, 2, 3
1, 2, 3
```

erzeugt nachstehendes Diagramm:



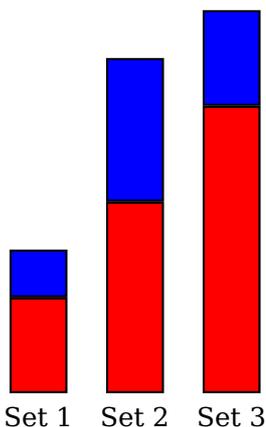
```
\psset{unit=0.5in}%
\begin{pspicture}(0,-0.5)(3,3)%
  \readpsbardata{\data}{example1.csv}%
  \psbarchart[barstyle={red,blue}]{\data}%
\end{pspicture}
```

3.2 stack

Bei einem stack Diagramm befindet sich in jeder Spalte lediglich ein Balken. Dieser eine Balken besteht aus so vielen Abschnitten, wie die Datendatei Reihen hat. Jede Reihe wird auf die Oberkante der Vorigen „gestapelt“. Die Datei

```
Set 1, Set 2, Set 3
1, 2, 3
0.5, 1.5, 1
```

erzeugt folgendes Diagramm:



```
\psset{unit=0.5in}%
\begin{pspicture}(0,-0.5)(3,4)%
  \readpsbardata{\data}{example2.csv}%
  \psbarchart[barstyle={red,blue},%
  chartstyle=stack]{\data}%
\end{pspicture}
```

Beachten Sie, dass diese Säulen hier breiter sind als bei `chartstyle=cluster`, da sie bei `stack` vertikal aufeinander gestapelt werden, bei `cluster` allerdings horizontal nebeneinander auf der x-Achse angeordnet sind. Um eine ähnliche Balkenbreite wie im `cluster`-Diagramm zu erhalten, stellt man `xunit=0.25in` ein.

3.3 block

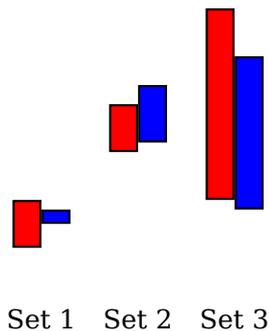
In einem block Diagramm kann an jedem Balken eine Reihe von Werten abgelesen werden. Daher benötigt jede Säule zwei Datenlinien aus der Datei, eine für die Markierung der oberen und eine für die Markierung der unteren Grenze. Gibt es mehrere

Paare von Datenlinien, werden sie gesammelt dargestellt. Existiert eine ungerade Anzahl von Datenreihen, so wird die letzte nicht beachtet.

Aus:

```
Set 1, Set 2, Set 3
1, 2, 3
0.5, 1.5, 1
0.75, 1.6, 0.9
0.9, 2.2, 2.5
```

entsteht dieses Diagramm:

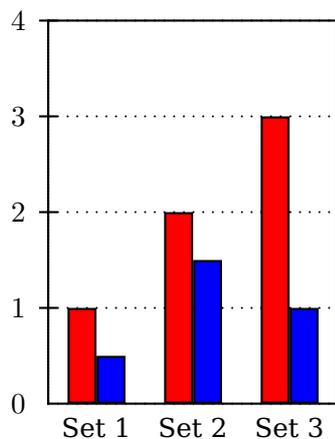


```
\psset{unit=0.5in}%
\begin{pspicture}(0,-0.5)(3,3.5)%
\readpsbardata{\data}{example3.csv}%
\psbarchart[barstyle={red,blue},%
chartstyle=block]{\data}%
\end{pspicture}
```

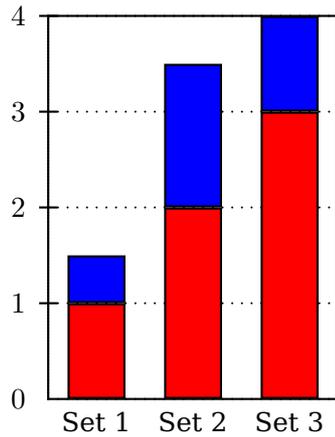
4 Beispiele

Grundlegende Beispiele für jeden Typ einschließlich Achsen und Gitternetzlinien:

```
\begin{filecontents*}{example2.csv}
Set 1, Set 2, Set 3
1, 2, 3
0.5, 1.5, 1
\end{filecontents*}
```



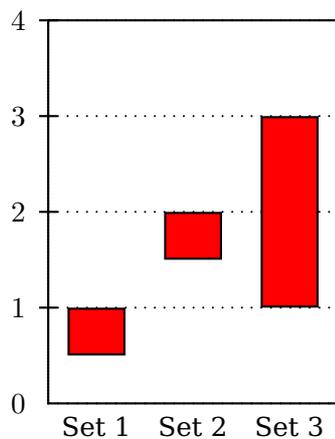
```
\psset{unit=0.5in}%
\begin{pspicture}(0,-0.5)(3,4.5)%
\psgrid[xunit=1.5in,gridlabels=0,%
subgriddiv=0,griddots=30](0,0)(1,4)%
\psaxes[axesstyle=frame,0x=0,Dx=1,labels=y,%
ticks=y](0,0)(3,4)%
\readpsbardata{\data}{example2.csv}%
\psbarchart[barstyle={red,blue}]{\data}%
\end{pspicture}
```



```

\psset{unit=0.5in}%
\begin{pspicture}(0,-0.5)(3,4.5)%
  \psgrid[xunit=1.5in,gridlabels=0,%
    subgriddiv=0,griddots=30](0,0)(1,4)%
  \psaxes[axesstyle=frame,0x=0,Dx=1,labels=y,%
    ticks=y](0,0)(3,4)%
  \readpsbardata{\data}{example2.csv}%
  \psbarchart[barstyle={red,blue},%
    chartstyle=stack]{\data}%
\end{pspicture}

```

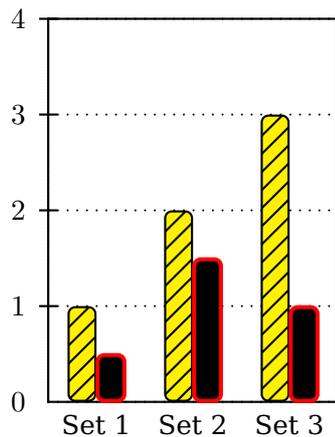


```

\psset{unit=0.5in}%
\begin{pspicture}(0,-0.5)(3,4.5)%
  \psgrid[xunit=1.5in,gridlabels=0,%
    subgriddiv=0,griddots=30](0,0)(1,4)%
  \psaxes[axesstyle=frame,0x=0,Dx=1,labels=y,%
    ticks=y](0,0)(3,4)%
  \readpsbardata{\data}{example2.csv}%
  \psbarchart[barstyle={red,blue},%
    chartstyle=block]{\data}%
\end{pspicture}

```

Verwendung von `\newsbarstyle`:

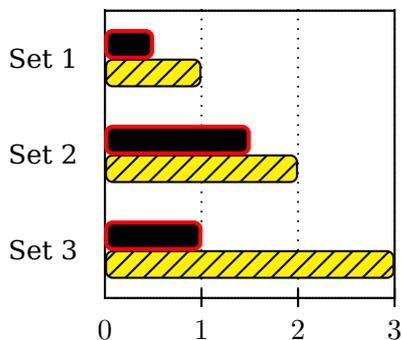


```

\psset{unit=0.5in}%
\newsbarstyle{yellowhatch}{framearc=0.5,%
  fillstyle=hlines*,rot=45,fillcolor=yellow}%
\newsbarstyle{redoutline}{framearc=0.5,%
  fillcolor=black,linecolor=red,%
  linewidth=1.5pt}%
\begin{pspicture}(0,-0.5)(3,4.5)%
  \psgrid[xunit=1.5in,gridlabels=0,%
    subgriddiv=0,griddots=30](0,0)(1,4)%
  \psaxes[axesstyle=frame,0x=0,Dx=1,labels=y,%
    ticks=y](0,0)(3,4)%
  \readpsbardata{\data}{example2.csv}%
  \psbarchart[barstyle={yellowhatch,%
    redoutline}]{\data}%
\end{pspicture}

```

Verwendung von [orientation=horizontal]:



```

\psset{unit=0.5in}%
\newsbarstyle{yellowhatch}{framearc=0.5,%
  fillstyle=hlines*,rot=45,fillcolor=yellow}%
\newsbarstyle{redoutline}{framearc=0.5,%
  fillcolor=black,linecolor=red,%
  linewidth=1.5pt}%
\begin{pspicture}(0,-0.5)(3,4.5)%
  \psgrid[xunit=1.5in,gridlabels=0,%
    subgriddiv=0,griddots=30](0,0)(1,4)%
  \psaxes[axesstyle=frame,0x=0,Dx=1,labels=y,%
    ticks=y](0,0)(3,4)%
  \readpsbardata{\data}{example2.csv}%
  \psbarchart[barstyle={yellowhatch,%
    redoutline}]{\data}%
\end{pspicture}

```

Komplexe Effekte sind unter mehrfacher Benennung von `\psbarchart` in einem einzelnen Diagramm möglich. Zum Beispiel kann ein besonders wichtiger Balken in grün hervorgehoben werden, indem man die Daten in zwei Dateien aufgespaltet. Eine enthält eine Null für den Datenwert, der hervorgehoben werden soll und die andere enthält Nullen für alle anderen Daten *außer* der Hervorzuhebenden.

Datei 1 (example4.csv):

```

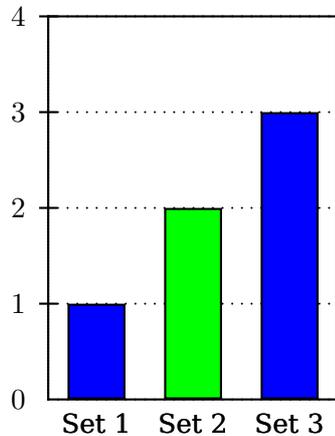
Set 1, Set 2, Set 3
1, 0, 3

```

Datei 2 (example5.csv):

0, 2, 0

Beachten Sie, dass die zweite Datei keine Kopfzeile hat, damit die Balkenbeschriftung nicht doppelt erfolgt.



```
\psset{unit=0.5in}%
\begin{pspicture}(0,-0.5)(3,4.5)%
  \psgrid[xunit=1.5in,gridlabels=0,%
    subgriddiv=0,griddots=30](0,0)(1,4)%
  \psaxes[axesstyle=frame,0x=0,Dx=1,labels=y,%
    ticks=y](0,0)(3,4)%
  \readpsbardata{\data}{example4.csv}%
  \psbarchart[barstyle={blue}]{\data}%
  \readpsbardata[header=false]{\data}%
  {example5.csv}%
  \psbarchart[barstyle={green}]{\data}%
\end{pspicture}
```

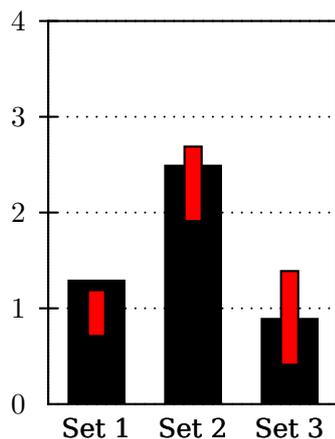
Verschiedene Diagrammarten können auch in einem Balkendiagramm kombiniert werden.

Datei 1 (example6.csv):

Set 1, Set 2, Set 3
1.3, 2.5, 0.9

Datei 2 (example7.csv):

0.7, 1.9, 0.4
1.2, 2.7, 1.4



```
\psset{unit=0.5in}%
\begin{pspicture}(0,-0.5)(3,4.5)%
  \psgrid[xunit=1.5in,gridlabels=0,%
    subgriddiv=0,griddots=30](0,0)(1,4)%
  \psaxes[axesstyle=frame,0x=0,Dx=1,labels=y,%
    ticks=y](0,0)(3,4)%
  \readpsbardata{\data}{example6.csv}%
  \psbarchart[barstyle={black}]{\data}%
  \readpsbardata[header=false]{\data}%
  {example7.csv}%
  \psbarchart[barstyle={red},chartstyle=block,%
    barcolsep=0.8]{\data}%
\end{pspicture}
\end{pspicture}
```

5 Tipps

- Informieren Sie sich über Befehle, um die Erstellung und Platzierung der Beschriftung zu erleichtern.
- Erlauben Sie die automatische Beschriftung der Balken mit den Werten der Datendatei.
- Erlauben Sie die automatische Beschriftung der Balken nach Ihrer Wahl.
- Verbessern Sie die Syntaxanalyse der Überschrift, um Kommas zwischen den Einträgen der Kopfzeile zu erlauben.
- Fügen Sie eine Fehler-Überprüfung hinzu, um sicherzustellen, dass jede Reihe der Datei dieselbe Anzahl an Einträgen enthält; falls nicht, wird der Fehler in $\text{T}_{\text{E}}\text{X}$ oder $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ übertragen.
- Verbessern Sie die Dokumentation.

Literatur

- [1] Michel Goosens, Frank Mittelbach, Sebastian Raetz, Denis Roegel, and Herbert Voß. *The $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ Graphics Companion*. Addison-Wesley Publishing Company, Reading, Mass., 2. edition, 2007.
- [2] Nikolai G. Kollock. *PostScript richtig eingesetzt: vom Konzept zum praktischen Einsatz*. IWT, Vaterstetten, 1989.
- [3] Herbert Voss. *PSTricks Support for pdf*. <http://PSTricks.de/pdf/pdfoutput.phtml>, 2002.
- [4] Herbert Voß. *$\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ Referenz*. DANTE – Lehmanns, Heidelberg/Hamburg, 1. edition, 2007.
- [5] Herbert Voß. *PSTricks – Grafik für $\text{T}_{\text{E}}\text{X}$ und $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$* . DANTE – Lehmanns, Heidelberg/Hamburg, 5. edition, 2008.
- [6] Michael Wiedmann and Peter Karp. *References for $\text{T}_{\text{E}}\text{X}$ and Friends*. <http://www.miwie.org/tex-refs/>, 2003.
- [7] Timothy Van Zandt. *PSTricks - PostScript macros for Generic TeX*. <http://www.tug.org/application/PSTricks>, 1993.

Index

B

bar, 3
barcolsep, 5
barddiag, 3
barlabelrot, 5
barsep, 5, 6
barstyle, 5, 6
black, 6
block, 5-7
blue, 6

C

chartstyle, 5-8
cluster, 5-8

D

darkgray, 6

F

fillcolor, 6
fillstyle, 6
framearc, 6

G

gnuplot, 3
gray, 6
green, 6

H

header, 5
histogr, 3
horizontal, 5

K

Keyvalue
- black, 6
- block, 5-7
- blue, 6
- cluster, 5-8
- darkgray, 6
- gray, 6
- green, 6
- horizontal, 5
- lightgray, 6
- red, 6
- stack, 5-8

- vertical, 5
- white, 6
Keyword
- barcolsep, 5
- barlabelrot, 5
- barsep, 5, 6
- barstyle, 5, 6
- chartstyle, 5-8
- fillcolor, 6
- fillstyle, 6
- framearc, 6
- header, 5
- orientation, 5
- xunit, 8

L

LaPrint, 3
lightgray, 6

M

Macro
- \newsbarstyle, 6
- \par, 4
- \psbarchart, 6, 11
- \psbarlabel, 6, 7
- \psbarlabelsep, 7
- \psbarscale, 7
- \psframe, 3, 6
- \psxunit, 5
- \readpsbardata, 5
- \renewcommand*, 7
- \renewcommand*, 7
Matlab, 3
mfpic, 3

N

\newsbarstyle, 6

O

orientation, 5

P

Package
- bar, 3
- barddiag, 3

- histogr, 3
- mfpic, 3
- psfrag, 3
- pst-bar, 2, 3
- pst-plot, 4
- pstricks, 2, 3, 6
- pstricks-add, 7
- \par, 4
- \psbarchart, 6, 11
- \psbarlabel, 6, 7
- \psbarlabelsep, 7
- \psbarscale, 7
- psfrag, 3
- \psframe, 3, 6
- pst-bar, 2, 3
- pst-plot, 4
- pstricks, 2, 3, 6
- pstricks-add, 7
- \psxunit, 5

R

- \readpsbardata, 5
- red, 6
- \renewcommand*, 7

S

- solid, 6
- stack, 5–8

V

- Value
 - cluster, 8
 - red, 6
 - solid, 6
- vertical, 5

W

- white, 6

X

- xunit, 8