bgjs-examples: Documentation and examples for the bargraph-js package

D. P. Story
Email: dpstory@acrotex.net

processed June 4, 2019

Contents

```
⟨*package⟩
```

1 The Examples

This DTX file contains the source files for the examples of the bargraph-js package. It has the added advantage of enabling me to provide verbose comments without messing up the source files.

```
⟨/package⟩
⟨*basic1⟩
```

1.1 `bgjs-basic1.tex`

This is a minimal example, it illustrates the basic capability of the package: horizontal and vertical bars are shown.

```
\documentclass{article}
```

Use of the web package is optional, my favorite package. If loaded, you can remove `\hypersetup{pdfpagemode=UseNone}` below, as web does not show the bookmarks by default.

```
 % \usepackage{web}
```

Input “the package”

```
\usepackage{bargraph-js}
\hypersetup{pdfpagemode=UseNone} % don’t need to see bookmarks
```

When developing the document use `\previewOn\pmpvOn`, the latter command is new (to eforms). It provides captions for the buttons. With this combination, outlines of form fields and captions for push buttons are seen in SumatraPDF, for example.

```
 % \previewOn\pmpvOn
```

For the bargraph-js package, I prefer to specify `\makePDasXon` (from eforms), this attempts to make the dimensions of form fields correspond to the true dimensions, as xelatex does.

```
\makePDasXOn
```

Below is “sample” document JavaScript to provide labeling to bars that do not have a `\TU` key specified (tool tip). There is a default labeling system, this overwrites that.

```
\begin{insDLJS}{lbl}{Labeling function}
%%   fld="<bgenv-name>@<bg-name>.<bar-name>"
%%   v=value of field
function customLabelsForBars(fld,v){
var pos=fld.indexOf(".");
var bargraph=fld.substring(0,pos);
var bar=fld.substring(pos+1);
  switch(bargraph) {
    case "vehiclesH@hBar":
      switch(bar) {
        case "auto":
          return "Automobiles: "+v;
        case "truck":
         return "Pickup trucks: "+v;
        case "suv":
          return "SUV types: "+v+", costing big \$\$s";
        case "van":
          return "Family vans: "+v
            +", these cost some serious \u20AC\u20ACs";
        default:
          return simpleBarLabels(fld,v);
      }
// other cases can be included
    default:
      return simpleBarLabels(fld,v);
  }
}
\end{insDLJS}
```

`\barLabelsTU` The `\barLabelsTU` takes either a string argument or a JavaScript function name. `\barLabelsTU` applies to all bars with a `\TU` key. The `\barLabelsTU` can be used within the body of the document to change the method of assigning labels to bars.

When the argument is a (JavaScript) string, use the variables `@env@`, `@barname@`, `@bar@`, and `@v@` to compose the string, as seen below. When a string is provided, *the \TU key is ignored.*

```
% \barLabelsTU{"Within the \\"@env@\\" environment, within the
%   \\"@barname@\\" environment, the bar \\"@bar@\\"
%   has a value of @v@"}
```

When the argument is a JavaScript function, you can specify the built-in function `customBarLabels`, or define your own. The function is expected to take two arguments, fld and v, eg, `customBarLabels(fld,v)`. Such a function should return a string.

```
\barLabelsTU{customBarLabels} % applies to all bars with a \TU key
```

`\barLabelsNoTU` The `\barLabelsNoTU` takes a string or a function as its argument. The command `\barLabelsNoTU` is specified in the preamble and cannot be changed in the body of the document. This labeling system applies to all bars with no `\TU` key. When the argument is a string, use `o.barname`, `o.bar`, and `o.value` to compose the string. For example,

```
% \barLabelsNoTU{"Vehicle Data: "+o.barname+": "
%    + o.bar+", Value: "+o.value}
```

When the argument is a JavaScript function, the referenced function must be written. In this demo, `customLabelsForBars(fld,v)` is reference below; however, we choose the default, which is `\barLabelsNoTU{""}` (or `barLabelsNoTU{}`). The results are the same; however, the difference is that you can write your own handler.

```
\barLabelsNoTU{customLabelsForBars} % applies to all other bars

\parindent0pt
\begin{document}

\textbf{Comments.} The bar graphs here are based on count data.
Input natural numbers into the text fields. If the width of a horizontal
bar or the height of a vertical bar goes outside the graphing area, the
bar graph is \emph{automatically re-scaled} so the widest one (for
horizontal) or highest one (for vertical) now fits. Try the
\textsf{Optimize} buttons as well.\medskip

\textbf{Instructions:} Stand at an intersection for one hour and count
the number of vehicles of each of the specified types passing you going
in one direction. Enter your results in the fields below.\medskip

%% Here we use the default method of labeling the bar graphs
\fbox{\begin{bargraphenv}[width=.67\linewidth,height=2in,
  o=horiz]{vehiclesH}
\presetsbarfor{hBar}{auto}{\BG{red}}
\presetsbarfor{hBar}{truck}{\BG{green}}
\presetsbarfor{hBar}{suv}{\BG{yellow}}
\presetsbarfor{hBar}{van}{\BG{magenta}}
\begin{bargraph}[nbars=4,gap=3]{hBar}
\barfor{auto}\barfor{truck}\barfor{suv}\barfor{van}
\end{bargraph}
\end{bargraphenv}}\hfill
\begin{minipage}[b][2in][c]
  {.33\linewidth-2\fboxsep-2\fboxrule-10pt}\kern0pt\parskip3pt
\makebox[\widthof{Truck:}][l]{Auto:}
  \inputFor{vehiclesH}{hBar}{auto}{.5in}{11bp}\vcgBdry[3bp]
\makebox[\widthof{Truck:}][l]{Truck:}
  \inputFor{vehiclesH}{hBar}{truck}{.5in}{11bp}\vcgBdry[3bp]
\makebox[\widthof{Truck:}][l]{SUV:}
  \inputFor{vehiclesH}{hBar}{suv}{.5in}{11bp}\vcgBdry[3bp]
\makebox[\widthof{Truck:}][l]{Van:}
  \inputFor{vehiclesH}{hBar}{van}{.5in}{11bp}\vcgBdry[3bp]
  \pushButton[\TU{This button re-scales the bar graph so that the
  longest bar takes the entire width of the region.}\CA{Optimize}
  \AAmouseup{optimizeScaling("vehiclesH");}]{optimize1}{}{13bp}
\end{minipage}\medskip

Reset horizontal bar graph: \pushButton[\CA{Reset}
  \AAmouseup{resetBargraphs("vehiclesH");}]{reset}{}{13bp}\bigskip

\fbox{\begin{bargraphenv}[width=2in,height=2in,o=vert]{vehiclesV}
\presetsbarfor{vBar}{auto}{\BG{red}
  \TU{There were @v@ automobiles observed}}
\presetsbarfor{vBar}{truck}{\BG{green}
  \TU{There were @v@ pickups observed during time period}}
\presetsbarfor{vBar}{suv}{\BG{yellow}
  \TU{There were @v@ SUVs observed, costing big
    \textdollar\textdollar s}}
\presetsbarfor{vBar}{van}{\BG{magenta}
  \TU{There were @v@ family vans filled with happy people;
    reminds me of J\"{u}rgen’s family}}
\begin{bargraph}[nbars=4,gap=3]{vBar}
\barfor{auto}\barfor{truck}\barfor{suv}\barfor{van}
\end{bargraph}
\end{bargraphenv}}\hfill
\begin{minipage}[b][2in][c]
  {\linewidth-2\fboxsep-2\fboxrule-2in-10pt}\kern0pt\parskip3pt
\makebox[\widthof{Truck:}][l]{Auto:}
  \inputFor{vehiclesV}{vBar}{auto}{.5in}{11bp}\vcgBdry[3bp]
\makebox[\widthof{Truck:}][l]{Truck:}
  \inputFor{vehiclesV}{vBar}{truck}{.5in}{11bp}\vcgBdry[3bp]
\makebox[\widthof{Truck:}][l]{SUV:}
  \inputFor{vehiclesV}{vBar}{suv}{.5in}{11bp}\vcgBdry[3bp]
\makebox[\widthof{Truck:}][l]{Van:}
  \inputFor{vehiclesV}{vBar}{van}{.5in}{11bp}\vcgBdry[4bp]
\pushButton[\TU{This button re-scales the bar graph so that the
tallest bar takes the entire height of the region.}\CA{Optimize}
  \AAmouseup{optimizeScaling("vehiclesV");
}]{optimize2}{}{13bp}
```

```
\end{minipage}\medskip

Reset vertical bar graph:
  \pushButton[\CA{Reset}\AAmouseup{resetBargraphs("vehiclesV");}
]{reset}{}{13bp}\vcgBdry[6bp]
Reset all bar graphs:
  \pushButton[\CA{Reset}
    \AAmouseup{resetBargraphs("vehiclesH","vehiclesV");}
]{reset}{}{13bp}
\end{document}
⟨/basic1⟩
⟨*basic2⟩
```

1.2 bgjs-basic2.tex

This is the same as `bgjs-basic1.tex`, additionally, this example illustrates hard-wired (typeset) labels of the bars.

```
\documentclass{article}
 % \usepackage{web}
\usepackage{bargraph-js}
\hypersetup{pdfpagemode=UseNone} % don’t need to see bookmarks
```

In this file we use `\rotatebox`, so graphicx is required.

```
\usepackage{graphicx}
 % \previewOn\pmpvOn
\makePDasXOn

\begin{insDLJS}{lbl}{Labeling function}
%%   fld="<bgenv-name>@<bg-name>.<bar-name>"
%%   v=value of field
function customLabelsForBars(fld,v){
var pos=fld.indexOf(".");
var bargraph=fld.substring(0,pos);
var bar=fld.substring(pos+1);
  switch(bargraph) {
    case "vehiclesH@hBar":
      switch(bar) {
        case "auto":
          return "Automobiles: "+v;
        case "truck":
         return "Pickup trucks: "+v;
        case "suv":
          return "SUV types: "+v+", costing big \$\$s";
        case "van":
          return "Family vans: "+v
            +", these cost some serious \u20AC\u20ACs";
        default:
          return simpleBarLabels(fld,v);
      }
// other cases can be included
```

```
    default:
      return simpleBarLabels(fld,v);
  }
}
\end{insDLJS}
 % \barLabelsTU{"Within the \\"@env@\\" environment,
 %  within the \\"@barname@\\" environment, the bar \\"@bar@\\" has a
 %  value of @v@"}
\barLabelsTU{customBarLabels} % applies to all bars with a \TU key

 % \barLabelsNoTU{customLabelsForBars} % applies to all other bars
\barLabelsNoTU{"Vehicle Data: "+o.barname+": "
  +o.bar+", Value: "+o.value}

\parindent0pt
\begin{document}

The bar graphs here are based on count data. Input natural numbers
into the text fields. If the width of a horizontal bar or the height
of a vertical bar goes outside the graphing area, the bar graph is
\emph{automatically re-scaled} so the widest one (for horizontal)
or highest one (for vertical) now fits. Try the Optimize buttons as
well.\medskip

%% These bars have no \TU key, so current value of \barLabelsTU
%% applies.
\fbox{\begin{bargraphenv}
  [width=.67\linewidth,height=2in,o=horiz]{vehiclesH}
\presetsbarfor{hBar}{auto}{\BG{red}}
\presetsbarfor{hBar}{truck}{\BG{green}}
\presetsbarfor{hBar}{suv}{\BG{yellow}}
\presetsbarfor{hBar}{van}{\BG{magenta}}
\begin{bargraph}[nbars=4,gap=3]{hBar}
\barfor{auto}
\cmd{\vs{-2bp}{\small{Automobile (two or four door)}}\vs{3bp}}
\barfor{truck}
\cmd{\vs{-2bp}{\small{Pickup truck}}\vs{3bp}}
\barfor{suv}
\cmd{\vs{-2bp}{\small{Sport utility vehicle (SUV)}}\vs{3bp}}
\barfor{van}
\cmd{\vs{1bp}{\small{Passenger van}}}
\end{bargraph}
\end{bargraphenv}}\hfill
\begin{minipage}[b][2in][c]
  {.33\linewidth-2\fboxsep-2\fboxrule-10pt}\kern0pt\parskip3pt
\makebox[\widthof{Truck:}][l]{Auto:}
  \inputFor{vehiclesH}{hBar}{auto}{.5in}{11bp}\vcgBdry[3bp]
\makebox[\widthof{Truck:}][l]{Truck:}
  \inputFor{vehiclesH}{hBar}{truck}{.5in}{11bp}\vcgBdry[3bp]
\makebox[\widthof{Truck:}][l]{SUV:}
```

```
  \inputFor{vehiclesH}{hBar}{suv}{.5in}{11bp}\vcgBdry[3bp]
\makebox[\widthof{Truck:}][l]{Van:}
  \inputFor{vehiclesH}{hBar}{van}{.5in}{11bp}\vcgBdry[3bp]
\pushButton[\TU{This button re-scales the bar graph so that the
longest bar takes the entire width of the region.}\CA{Optimize}
\AAmouseup{optimizeScaling("vehiclesH");}]{optimize1}{}{13bp}
\end{minipage}\medskip

Reset horizontal bar graph: \pushButton[\CA{Reset}
  \AAmouseup{resetBargraphs("vehiclesH");}]{reset}{}{13bp}

\bigskip

\fbox{\begin{bargraphenv}[width=2in,height=2in,o=vert]{vehiclesV}
\presetsbarfor{vBar}{auto}{\BG{red}
  \TU{There were @v@ automobiles observed}}
\presetsbarfor{vBar}{truck}{\BG{green}
  \TU{There were @v@ pickups observed during time period}}
\presetsbarfor{vBar}{suv}{\BG{yellow}
  \TU{There were @v@ SUVs observed, costing big
  \textdollar\textdollar s}}
\presetsbarfor{vBar}{van}{\BG{magenta}
  \TU{There were @v@ family vans filled with happy people;
    reminds me of J\"{u}rgen’s family}}
\begin{bargraph}[nbars=4,gap=3]{vBar}
\barfor{auto}
\cmd{\hs{-2bp}\smash{\rotatebox[origin=lb]{90}
  {\small{{Automobile (two or four door)}}}}\hs{3bp}}
\barfor{truck}
\cmd{\hs{-2bp}\smash{\rotatebox[origin=lb]{90}
  {\small{Pickup truck}}}\hs{3bp}}
\barfor{suv}
\cmd{\hs{-2bp}\smash{\rotatebox[origin=lb]{90}
  {\small{Sport utility vehicle (SUV)}}}\hs{3bp}}
\barfor{van}
\cmd{\hs{1bp}\smash{\rotatebox[origin=lb]{90}
  {\small{Passenger van}}}}
\end{bargraph}
\end{bargraphenv}}\hfill
\begin{minipage}[b][2in][c]
  {\linewidth-2\fboxsep-2\fboxrule-2in-10pt}\kern0pt\parskip3pt
\makebox[\widthof{Truck:}][l]{Auto:}
  \inputFor{vehiclesV}{vBar}{auto}{.5in}{11bp}\vcgBdry[3bp]
\makebox[\widthof{Truck:}][l]{Truck:}
  \inputFor{vehiclesV}{vBar}{truck}{.5in}{11bp}\vcgBdry[3bp]
\makebox[\widthof{Truck:}][l]{SUV:}
  \inputFor{vehiclesV}{vBar}{suv}{.5in}{11bp}\vcgBdry[3bp]
\makebox[\widthof{Truck:}][l]{Van:}
  \inputFor{vehiclesV}{vBar}{van}{.5in}{11bp}\vcgBdry[4bp]
\pushButton[\TU{This button re-scales the bar graph so that the
tallest bar takes the entire height of the region.}\CA{Optimize}
\AAmouseup{optimizeScaling("vehiclesV");}]{optimize2}{}{13bp}
\end{minipage}\medskip

Reset vertical bar graph:
  \pushButton[\CA{Reset}
    \AAmouseup{resetBargraphs("vehiclesV");}
  ]{reset}{}{13bp}\vcgBdry[6bp]
Reset all bar graphs:
  \pushButton[\CA{Reset}
    \AAmouseup{resetBargraphs("vehiclesH","vehiclesV");}]{reset}{}{13bp}

\end{document}
⟨/basic2⟩
⟨*basic3⟩
```

1.3 bgjs-basic3.tex

This example illustrates using form fields to label each bar; the captions appear or are hidden depending on whether there are any data.

```
\documentclass{article}
 % \usepackage{web}
\usepackage[usealtadobe]{eforms}
\hypersetup{pdfpagemode=UseNone} % don't need to see bookmarks
\usepackage{bargraph-js}
 % \previewOn\pmpvOn
\makePDasXOn

\begin{insDLJS}{lbl}{Labeling function}
%%   fld="<bgenv-name>@<bg-name>.<bar-name>"
%%   v=value of field
function customLabelsForBars(fld,v){
var pos=fld.indexOf(".");
var bargraph=fld.substring(0,pos);
var bar=fld.substring(pos+1);
  switch(bargraph) {
    case "vehiclesH@hBar":
      switch(bar) {
        case "auto":
          return "Automobiles: "+v;
        case "truck":
         return "Pickup trucks: "+v;
        case "suv":
          return "SUV types: "+v+", costing big \$\$s";
        case "van":
          return "Family vans: "+v
            +", these cost some serious \u20AC\u20ACs";
        default:
          return simpleBarLabels(fld,v);
```

```
   }
// other cases can be included
  default:
   return simpleBarLabels(fld,v);
 }
}
\end{insDLJS}
 % applies to all bars with a \TU key
 % \barLabelsTU{customBarLabels}
 % applies to all other bars
 % \barLabelsNoTU{customLabelsForBars}

\parindent0pt
\begin{document}

In this example, we label the bar graph using text fields (using the
\verb~\labelFld~ command defined in this package). The fields are
initially hidden, then made visible when data is entered into the
input fields.\par\medskip
```

`\labelFld` The syntax for `\labelFld` is as follows:

`\labelFld[`⟨*options*⟩`]{`⟨*text*⟩`}{`⟨*bg-name*⟩`.`⟨*bar-name*⟩`}`⟨*width*⟩`.{`⟨*height*⟩`}`

Where, ⟨*text*⟩ is the labeling text this is to appear with this bar. See the examples below. When the bars are vertical (`o=vert`) then amongst the ⟨*options*⟩ for the field is key-value pair `\R{90}`, again, see below for examples.

```
%% Here we use the default method of labeling the bar graphs
%% This bar graph is horizontally oriented
\fbox{\begin{bargraphenv}[width=.67\linewidth,height=2in,
 o=horiz]{vehiclesH}\def\WD{2in}
\presetsbarfor{hBar}{auto}{\BG{red}}
\presetsbarfor{hBar}{truck}{\BG{green}}
\presetsbarfor{hBar}{suv}{\BG{yellow}}
\presetsbarfor{hBar}{van}{\BG{magenta}}
\begin{bargraph}[nbars=4,gap=3]{hBar}
\barfor{auto}
\cmd{\vs{-3bp}\labelFld[\textSize{10}]
 {Automobiles (two or four door)}{hBar.auto}{\WD}{13bp}\vs{3bp}}
\barfor{truck}
\cmd{\vs{-3bp}\labelFld[\textSize{10}]
 {Pickup trucks}{hBar.truck}{\WD}{13bp}\vs{3bp}}
\barfor{suv}
\cmd{\vs{-3bp}\labelFld[\textSize{10}]
 {Sport utility vehicle (SUV)}{hBar.suv}{\WD}{13bp}\vs{3bp}}
\barfor{van}
\cmd{\vs{0bp}\labelFld[\textSize{10}]
 {Passenger van}{hBar.van}{\WD}{13bp}}
\end{bargraph}
\end{bargraphenv}}\hfill
\begin{minipage}[b][2in][c]
```

```
  {.33\linewidth-2\fboxsep-2\fboxrule-10pt}\kern0pt\parskip3pt
```

toggleFldVisibility()

To get the field to appear and hide in synchronization with the data input by the `\inputFor` commands, we create a custom `\presetinputfor` command. We use a built-in JavaScript function `toggleFldVisibility`, the argument of which is the field name of the targeted bar: the full name is ⟨*bg-name*⟩`.`⟨*bar-name*⟩`@`⟨*bgenv-name*⟩.

```
\def\presetinputfor#1#2{%
```

The arguments for the `\presetinputfor` command are,

`#1=`⟨*bgenv-name*⟩

`#2=`⟨*bg-name*⟩`.`⟨*bar-name*⟩

```
  \AAkeystroke{EFNumber_Keystroke(0, 0, 0, 0, "", true);}
  \AAformat{try{EFNumber_Format(0, 0, 0, 0, "", true);\r
  if(event.rc)toggleFldVisibility("#2@#1",%
(event.value!=0));}catch(e){}}
  \AAvalidate{EFRange_Validate(true, 0, false, 0);}
}
\makebox[\widthof{Truck:}][l]{Auto:}
  \inputFor{vehiclesH}{hBar}{auto}{.5in}{11bp}\vcgBdry[3bp]
\makebox[\widthof{Truck:}][l]{Truck:}
  \inputFor{vehiclesH}{hBar}{truck}{.5in}{11bp}\vcgBdry[3bp]
\makebox[\widthof{Truck:}][l]{SUV:}
  \inputFor{vehiclesH}{hBar}{suv}{.5in}{11bp}\vcgBdry[3bp]
\makebox[\widthof{Truck:}][l]{Van:}
  \inputFor{vehiclesH}{hBar}{van}{.5in}{11bp}\vcgBdry[3bp]
\pushButton[\TU{This button re-scales the bar graph so that the
  longest bar takes the entire width of the region.}\CA{Optimize}
  \AAmouseup{optimizeScaling("vehiclesH");}]{optimize1}{}{13bp}
\end{minipage}\medskip

Reset horizontal bar graph: \pushButton[\CA{Reset}
  \AAmouseup{resetBargraphs("hBar","vehiclesH");\r
}]{reset}{}{13bp}

\bigskip

%% This bar graph is vertical so we must insert \R{90} in the
%% options argument of \labelFld.
\fbox{\begin{bargraphenv}[width=2in,height=2in,%
  o=vert]{vehiclesV}\def\WD{2in}
\presetsbarfor{vBar}{auto}{\BG{red}
  \TU{There were @v@ automobiles observed}}
\presetsbarfor{vBar}{truck}{\BG{green}
  \TU{There were @v@ pickups observed during time period}}
\presetsbarfor{vBar}{suv}{\BG{yellow}
  \TU{There were @v@ SUVs observed, costing big
    \textdollar\textdollar s}}
```

```
\presetsbarfor{vBar}{van}{\BG{magenta}
  \TU{There were @v@ family vans filled with happy people;
    reminds me of J\"{u}rgen’s family}}
\begin{bargraph}[nbars=4,gap=3]{vBar}
\barfor{auto}
\cmd{\hs{-3bp}\labelFld[\R{90}\textSize{10}]
  {Automobiles (two or four door)}{vBar.auto}{\WD}{13bp}\hs{3bp}}
\barfor{truck}
\cmd{\hs{-3bp}\labelFld[\R{90}\textSize{10}]
  {Pickup trucks}{vBar.truck}{\WD}{13bp}\hs{3bp}}
\barfor{suv}
\cmd{\hs{-3bp}\labelFld[\R{90}\textSize{10}]
  {Sport utility vehicle (SUV)}{vBar.suv}{\WD}{13bp}\hs{3bp}}
\barfor{van}
\cmd{\hs{0bp}\labelFld[\R{90}\textSize{10}]
  {Passenger van}{vBar.van}{\WD}{13bp}}
\end{bargraph}
\end{bargraphenv}}\hfill
\begin{minipage}[b][2in][c]
  {\linewidth-2\fboxsep-2\fboxrule-2in-10pt}\kern0pt\parskip3pt
```

toggleFldVisibility()

To get the field to appear and hide in synchronization with the data input by the `\inputFor` commands, we create a custom `\presetinputfor` command. We use a built-in JavaScript function `toggleFldVisibility`, the argument of which is the field name of the targeted bar: the full name is ⟨*bg-name*⟩`.`⟨*bar-name*⟩`@`⟨*bgenv-name*⟩.

```
\def\presetinputfor#1#2{%
```

The arguments for the `\presetinputfor` command are,

`#1=`⟨*bgenv-name*⟩

`#2=`⟨*bg-name*⟩`.`⟨*bar-name*⟩

```
  \AAkeystroke{EFNumber_Keystroke(0, 0, 0, 0, "", true);}
  \AAformat{try{EFNumber_Format(0, 0, 0, 0, "", true);\r
  if(event.rc)toggleFldVisibility("#2@#1",%
(event.value!=0));}catch(e){}}
  \AAvalidate{EFRange_Validate(true, 0, false, 0);}
}
\makebox[\widthof{Truck:}][l]{Auto:}
  \inputFor{vehiclesV}{vBar}{auto}{.5in}{11bp}\vcgBdry[3bp]
\makebox[\widthof{Truck:}][l]{Truck:}
  \inputFor{vehiclesV}{vBar}{truck}{.5in}{11bp}\vcgBdry[3bp]
\makebox[\widthof{Truck:}][l]{SUV:}
  \inputFor{vehiclesV}{vBar}{suv}{.5in}{11bp}\vcgBdry[3bp]
\makebox[\widthof{Truck:}][l]{Van:}
  \inputFor{vehiclesV}{vBar}{van}{.5in}{11bp}\vcgBdry[4bp]
\pushButton[\TU{This button re-scales the bar graph so that the
  tallest bar takes the entire height of the region.}\CA{Optimize}
  \AAmouseup{optimizeScaling("vehiclesV");}
```

```
]{optimize2}{}{13bp}
\end{minipage}\medskip

Reset vertical bar graph:
  \pushButton[\CA{Reset}
    \AAmouseup{resetBargraphs("vBar","vehiclesV");}
]{reset}{}{13bp}\vcgBdry[6bp]
Reset all bar graphs:
  \pushButton[\CA{Reset}
    \AAmouseup{resetBargraphs("hBar","vBar","vehiclesH","vehiclesV");}
]{reset}{}{13bp}
\end{document}
⟨/basic3⟩
⟨*basic4⟩
```

1.4 `bgjs-basic4.tex`

This example highlights the options `origin=.5` and `showaxis=true`. Try changing `o=vert` to `o=horiz`.

```
\documentclass{article}
 % \usepackage{web}
\usepackage{bargraph-js}
\hypersetup{pdfpagemode=UseNone} % don't need to see bookmarks
 % \previewOn\pmpvOn
\makePDasXOn
```

Here we develop a custom JavaScript string to label each bar when the user runs the mouse over a bar in the bar graph.

```
\barLabelsNoTU{"Profits ("+o.barname.substring(1) +") for "
    + o.bar +": "+((o.value<0)?"-\$":"\$")+Math.abs(o.value)
    + " (million)"}

\parindent0pt
\begin{document}

\textbf{Instructions:} Enter the profits for the year 2018, enter
a negative number if there was a loss for a particular quarter.\medskip

%% origin=.5,showaxis=true : change o=vert to o=horiz
\fbox{\begin{bargraphenv}[width=2in,height=2in,o=vert,%
  origin=.5,showaxis=true]{profits}
\presetsbarfor{y2018}{Q1}{\BG{red}}
\presetsbarfor{y2018}{Q2}{\BG{green}}
\presetsbarfor{y2018}{Q3}{\BG{yellow}}
\presetsbarfor{y2018}{Q4}{\BG{magenta}}
\begin{bargraph}[nbars=4,gap=3]{y2018}
\barfor{Q1}\barfor{Q2}\barfor{Q3}\barfor{Q4}
\end{bargraph}
\end{bargraphenv}}\hfill
```

```
\begin{minipage}[b][2in][c]
  {\linewidth-2\fboxsep-2\fboxrule-2in-10pt}\kern0pt\parskip3pt
\def\presetinputfor#1#2{%
  \AAkeystroke{AFNumber_Keystroke(0, 0, 0, 0, "", true);}
  \AAformat{AFNumber_Format(0, 0, 0, 0, "", true);}
}
\makebox[\linewidth][l]{\textbf
  {Profits for year 2018 (in millions)}}\\[3pt]
\makebox[\widthof{Q1:}][l]{Q1:}
  \inputFor{profits}{y2018}{Q1}{.5in}{13bp}\vcgBdry[3bp]
\makebox[\widthof{Q1:}][l]{Q2:}
  \inputFor{profits}{y2018}{Q2}{.5in}{13bp}\vcgBdry[3bp]
\makebox[\widthof{Q1:}][l]{Q3:}
  \inputFor{profits}{y2018}{Q3}{.5in}{13bp}\vcgBdry[3bp]
\makebox[\widthof{Q1:}][l]{Q4:}
  \inputFor{profits}{y2018}{Q4}{.5in}{13bp}\vcgBdry[4bp]
\pushButton[\TU{This button re-scales the bar graph so that the
tallest bar takes the entire height of the region.}\CA{Optimize}
  \AAmouseup{optimizeScaling("profits");}
]{optimize2}{}{13bp}
```

Supply manual re-scaling text field and push button

```
\displaysfFor{profits}{.5in}{13bp}\olBdry
\manualsfFor[\CA{Rescale}\TU{Enter a new scale factor in the
  text field, then press this button}
]{profits}{}{13bp}
\end{minipage}\medskip

Reset vertical bar graph:
  \pushButton[\CA{Reset}\AAmouseup{resetBargraphs("profits");}
]{reset}{}{13bp}
\end{document}
⟨/basic4⟩
⟨*adv1⟩
```

1.5 `bgjs-adv1.tex`

We make an auxiliary calculations when data are entered into the `\inputFor` command.

```
\documentclass{article}
 % \usepackage{web}
\usepackage[usealtadobe,setcorder]{eforms}
\usepackage{bargraph-js}
\hypersetup{pdfpagemode=UseNone} % don’t need to see bookmarks
 % \previewOn\pmpvOn
\makePDasXOn

```

In this example, we calculate the percentage of the class that receives each grade. When data is entered, it is important to calculate the field `titalClass` first, this will force all calculate that depend on this field to be calculated. As a result, `\calcOrder{totalClass}` set the calculateion order. `\calcOrder` is an eforms command.

```
\calcOrder{totalClass}

\begin{insDLJS}{lbl}{Labeling function}
%%   fld="<bgenv-name>@<bg-name>.<bar-name>"
%%   v=value of field
function customLabelsForBars(fld,v){
var pos=fld.indexOf(".");
var bargraph=fld.substring(0,pos);
var bar=fld.substring(pos+1);
  switch(bargraph) {
    case "math@class":
      var f=this.getField("totalClass");
      var tot=f.value;
      var V=(tot!=0)?((v/tot)*100):(v*100);
      V=util.printf("\%.1f",V);
      switch(bar) {
        case "A":
          return V+"\% of the class received an ’A’";
        case "B":
          return V+"\% of the class received an ’B’";
        case "C":
          return V+"\% of the class received an ’C’";
        case "D":
          return V+"\% of the class received an ’D’";
        case "F":
          return V+"\% of the class received an ’F’";
        default:
          return "unknown result";
      }
    default:
      return simpleBarLabels(fld,v);
  }
}
\end{insDLJS}

%% applies to all bars without a \TU key
\barLabelsNoTU{customLabelsForBars}

\parindent0pt
\begin{document}

Enter the number of class members that received the grades
A, B, C, D, and~F.\medskip

```

We define `\presetinputforcalc`, which is the calculate code for the `\inputFor` commands. Here, `p(1)` is ⟨*bgenv-name*⟩ and `p(2)` is ⟨*bg-name*⟩.⟨*bar-name*⟩. We use these to form the name of the `pctFor` fields.

```
\begin{defineJS}[\catcode`\*=0\relax]{\presetinputforcalc}
var f=this.getField("pctFor.*p(2)@*p(1)");
var g=this.getField("totalClass");
if (event.value!=0)
  f.value=( g.value==0 )?1:(event.value/g.value);
\end{defineJS}

\def\pctThisGrade#1#2#3{\textField[\Ff{\FfReadOnly}\Q{2}
  \AAkeystroke{EFPercent_Keystroke(1, 1);}
  \AAformat{if (event.value!=0)EFPercent_Format(1, 1);\r
  else event.value="";}
```

The name of this field shall be

`pctFor.`⟨*bg-name*⟩`.`⟨*bar-name*⟩`@`⟨*bgenv-name*⟩

```
]{pctFor.#2.#3@#1}{.6in}{11bp}\cgBdry[.5em]}
```

`\scaleFactorDef` is used to reset initial scale factor the next environment. Here we set it to be the height of the environment (in bp points) divided by 100, that way, data are re-scaled as a proportion of the height of the bar graph. We make this declaration inside the `\fbox` to make the definition local.

```
\fbox{\scaleFactorDef{dataForEnv["math"].height/100}%
\begin{bargraphenv}[width=2in,height=2in,o=vert]{math}
\begin{bargraph}[nbars=5,gap=3]{class}
\barfor{A}\barfor{B}\barfor{C}\barfor{D}\barfor{F}
\end{bargraph}
\end{bargraphenv}}\hfill
\begin{minipage}[b][2in][c]
  {\linewidth-2\fboxsep-2\fboxrule-2in-10pt}\kern0pt\parskip3pt
\def\presetinputfor#1#2{\AAcalculate{\presetinputforcalc}
  \AAkeystroke{AFNumber_Keystroke(0, 0, 0, 0, "", true);}
  \AAformat{AFNumber_Format(0, 0, 0, 0, "", true);}
  \AAvalidate{AFRange_Validate(true, 0, false, 0);}
}%
\makebox[\widthof{A:}][l]{A:} \inputFor{math}{class}{A}{.5in}{11bp}
  \pctThisGrade{math}{class}{A}\vcgBdry[3bp]
\makebox[\widthof{A:}][l]{B:} \inputFor{math}{class}{B}{.5in}{11bp}
  \pctThisGrade{math}{class}{B}\vcgBdry[3bp]
\makebox[\widthof{A:}][l]{C:} \inputFor{math}{class}{C}{.5in}{11bp}
  \pctThisGrade{math}{class}{C}\vcgBdry[3bp]
\makebox[\widthof{A:}][l]{D:} \inputFor{math}{class}{D}{.5in}{11bp}
  \pctThisGrade{math}{class}{D}\vcgBdry[3bp]
\makebox[\widthof{A:}][l]{F:} \inputFor{math}{class}{F}{.5in}{11bp}
  \pctThisGrade{math}{class}{F}\vcgBdry[3bp]
\makebox[\widthof{A:}][l]{\phantom{A:}}
\textField[\Ff{\FfReadOnly}
  \AAcalculate{AFSimple_Calculate("SUM",new Array("math"));}
]{totalClass}{.5in}{13bp}\vcgBdry[4bp]
\pushButton[\TU{This button re-scales the bar graph so that the
```

```
  tallest bar takes the entire height of the region.}
  \CA{Optimize}\AAmouseup{optimizeScaling("math");}
]{optimize3}{}{13bp}
\end{minipage}\medskip

Reset vertical bar graph:
\pushButton[\CA{Reset}
  \AAmouseup{resetBargraphs("math","pctFor","totalClass");}
]{reset}{}{13bp}
\end{document}
⟨/adv1⟩
⟨*comma1⟩
```

1.6 bgjs-comma1.tex

This file illustrates populating a bar graph using comma-delimited data.

```
\documentclass{article}
 % \usepackage{web}
\usepackage{eforms}
\hypersetup{pdfpagemode=UseNone} % don't need to see bookmarks
\usepackage{bargraph-js}
 % \previewOn\pmpvOn
\makePDasXOn

\parindent0pt
\begin{document}

This file demonstrates comma-delimited data, manually enter the data
in the text field or by press on one of the buttons to give
pre-packaged examples.\medskip

\bgroup\setlength{\fboxrule}{1bp}\fbox
{\begin{bargraphenv}[width=(14bp*20),height=2in,%
  o=vert]{statdemo}%
\begin{bargraph}[nbars=20,gap=0,bardimen=14]{histogram}
\barfor{bar1}\barfor{bar2}\barfor{bar3}\barfor{bar4}\barfor{bar5}
\barfor{bar6}\barfor{bar7}\barfor{bar8}\barfor{bar9}\barfor{bar10}
\barfor{bar11}\barfor{bar12}\barfor{bar13}\barfor{bar14}
\barfor{bar15}\barfor{bar16}\barfor{bar17}\barfor{bar18}
\barfor{bar19}\barfor{bar20}
\end{bargraph}%
\end{bargraphenv}}\egroup\vcgBdry[6pt]

\hglue1bp\textField[\TU{Enter up to twenty nonnegative numbers
  separated by commas}
  \AAvalidate{resetBargraphs("statdemo");\r
  \populateCommaData("statdemo","histogram",event.value,%
```

Require each entry in the array to be non-negative numbers.

```
    validateArrayNonNegNums)}
```

```
  ]{commaed}{(\bardimen*\nbars+2\fboxsep+2bp)}{13bp}\vcgBdry[6pt]

\hglue1bp\pushButton[\CA{Symmetrical}\AAmouseup{%
var str="1,2,3,4,5,6,7,8,9,10,10,9,8,7,6,5,4,3,2,1";\r
```

Here, we don't include `validateArrayNonNegNums()` because it is assumed the document author knows what he/she is doing.

```
\populateCommaData("statdemo","histogram",str);
}]{symmetrical}{}{13bp}\cgBdry[.5em]
\pushButton[\CA{Skew left}\AAmouseup{%
var str="1,2,2,3,3,4,5,6,8,10,12,14,16,19,20,19,17,15,13,11";\r
\populateCommaData("statdemo","histogram",str);}
]{skewleft}{}{13bp}\cgBdry[.5em]
\pushButton[\CA{Skew right}\AAmouseup{%
var str="17,18,19,20,19,18,16,14,12,10,8,7,7,6,6,4,4,3,2,1";\r
\populateCommaData("statdemo","histogram",str);}
]{skewright}{}{13bp}\vcgBdry[6pt]

\hglue1bp\pushButton[\TU{This button re-scales the bar graph so
  that the longest bar takes the entire width of the region.}
  \CA{Optimize}\AAmouseup{optimizeScaling("statdemo");}
]{optimize1}{}{13bp}\cgBdry[.5em]
\displaysfFor{statdemo}{.5in}{13bp}\olBdry
\manualsfFor[\CA{Rescale}\TU{Enter a new scale factor in the
  text field, then press this button}]{statdemo}{}{13bp}\vcgBdry[6bp]

\hglue1bp\pushButton[\CA{Reset}
  \AAmouseup{resetBargraphs("statdemo","commaed","rescale.statdemo")}
]{reset}{}{13bp}\vcgBdry[6pt]
\end{document}
⟨/comma1⟩
⟨*comma2⟩
```

1.7 `bgjs-comma2.tex`

This file demonstrates multiple **bargraph** environments placed within a single **bargraphenv** environment. Comma-delimited data can populates the bar graphs.

```
\documentclass{article}
 % \usepackage{web}
\usepackage{eforms}
\hypersetup{pdfpagemode=UseNone} % don't need to see bookmarks
\usepackage{bargraph-js}
 % \previewOn\pmpvOn
\makePDasXOn

\parindent0pt
\begin{document}

This file demonstrates comma-delimited data, manually enter the
```

```
data in the text field or by press on one of the buttons to give
pre-packaged examples. More than one \texttt{bargraph} environment
can appear within a \texttt{bargraphenv}, as illustrated in the next
two examples.\medskip

\fbox{\begin{bargraphenv}[width=23bp*10,height=1.4in,o=vert]{math1}
\presetsbarfor{class1}{A}{\BG{red}
  \TU{Class1: @v@ students received an ’A’}}
\presetsbarfor{class1}{B}{\BG{red}
  \TU{Class1: @v@ students received an ’B’}}
\presetsbarfor{class1}{C}{\BG{red}
  \TU{Class1: @v@ students received an ’C’}}
\presetsbarfor{class1}{D}{\BG{red}
  \TU{Class1: @v@ students received an ’D’}}
\presetsbarfor{class1}{F}{\BG{red}
  \TU{Class1: @v@ students received an ’F’}}
\presetsbarfor{class2}{A}{\BG{blue}
  \TU{Class2: @v@ students received an ’A’}}
\presetsbarfor{class2}{B}{\BG{blue}
  \TU{Class2: @v@ students received an ’B’}}
\presetsbarfor{class2}{C}{\BG{blue}
  \TU{Class2: @v@ students received an ’C’}}
\presetsbarfor{class2}{D}{\BG{blue}
  \TU{Class2: @v@ students received an ’D’}}
\presetsbarfor{class2}{F}{\BG{blue}
  \TU{Class2: @v@ students received an ’F’}}
\begin{bargraph}[nbars=5,gap=3]{class1}
\barfor{A}\barfor{B}\barfor{C}\barfor{D}\barfor{F}
\end{bargraph}
\begin{bargraph}[nbars=5,gap=3]{class2}\cmd{\hs{\bargap}}
\barfor{A}\barfor{B}\barfor{C}\barfor{D}\barfor{F}
\end{bargraph}
\end{bargraphenv}}\vcgBdry[6pt]

\begin{minipage}[c]{222pt}
class1:
  \textField[\TU{Enter five natural numbers separated
    by commas}
    \AAvalidate{\populateCommaData("math1","class1",event.value,%
      validateArrayNonNegNums)}
  ]{txtclass3}{2in}{13bp}\cgBdry[.5em]
  \pushButton[\CA{Class 1}\AAmouseup{%
    var str="12,15,23,10,15";\r
    this.getField("txtclass3").value=str;\r
    \populateCommaData("math1","class1",str);
}]{math1class1}{}{13bp}\vcgBdry[4bp]
class2:
  \textField[\TU{Enter five natural numbers separated by commas}
    \AAvalidate{\populateCommaData("math1","class2",event.value,%
      validateArrayNonNegNums)}
  ]{txtclass4}{2in}{13bp}\cgBdry[.5em]
  \pushButton[\CA{Class 2}\AAmouseup{%
    var str="10,17,29,10,20";\r
    this.getField("txtclass4").value=str;\r
    \populateCommaData("math1","class2",str);
  }]{math1class1}{}{13bp}
\end{minipage}\cgBdry[.5em]
\pushButton[\TU{This button re-scales the bar graph so that the
  tallest bar takes the entire height of the region.}\CA{Optimize}
  \AAmouseup{optimizeScaling("math1");}
]{optimize3}{}{13bp}\cgBdry[.5em]
\pushButton[\CA{Reset}
  \AAmouseup{resetBargraphs("math1","txtclass3","txtclass4");}
]{reset}{}{13bp}\vcgBdry[6pt]


You can adjust the positions of the bar graph to have a more
side-by-side comparison.\vcgBdry[4bp]

\fbox{\begin{bargraphenv}[width=33bp*5,height=2in,%
  o=vert]{math2}
\presetsbarfor{class1}{A}{\BG{red}}
\presetsbarfor{class1}{B}{\BG{red}}
\presetsbarfor{class1}{C}{\BG{red}}
\presetsbarfor{class1}{D}{\BG{red}}
\presetsbarfor{class1}{F}{\BG{red}}
\presetsbarfor{class2}{A}{\BG{blue}}
\presetsbarfor{class2}{B}{\BG{blue}}
\presetsbarfor{class2}{C}{\BG{blue}}
\presetsbarfor{class2}{D}{\BG{blue}}
\presetsbarfor{class2}{F}{\BG{blue}}
\begin{bargraph}[nbars=5,gap=13]{class1}
\barfor{A}\barfor{B}\barfor{C}\barfor{D}\barfor{F}
\end{bargraph}%
\begin{bargraph}[nbars=5,gap=13]{class2}\cmd{\hs{13bp}}
\cmd{\color{red}\hs{-33bp*5+10bp}}
\barfor{A}\barfor{B}\barfor{C}\barfor{D}\barfor{F}
\end{bargraph}
\end{bargraphenv}}\vcgBdry[6pt]

\begin{defineJS}{\pbaction}
var f=this.getField("txtclass5");
for (var value="",i=0; i<5; i++)
  value+=(""+(Math.round(Math.random()*200))+",");
f.value=value.substring(0,value.length-1);
for (var value="",i=0; i<5; i++)
  value+=(""+(Math.round(Math.random()*200))+",");
var g=this.getField("txtclass6");
g.value=value.substring(0,value.length-1);
\end{defineJS}
```

```

\begin{minipage}[c]{222pt}
class1:
  \textField[\TU{Enter five natural numbers separated by commas}
  \AAvalidate{\populateCommaData("math2","class1",event.value,%
    validateArrayNonNegNums)}
  ]{txtclass5}{2in}{13bp}\cgBdry[.5em]
  \pushButton[\CA{Class 1}\AAmouseup{%
    var str="12,15,23,10,15";\r
    this.getField("txtclass5").value=str;\r
    \populateCommaData("math2","class1",str);
  }]{math1class1}{}{13bp}\vcgBdry[4bp]
class2:
  \textField[\TU{Enter five natural numbers separated by commas}
  \AAvalidate{\populateCommaData("math2","class2",event.value,%
    validateArrayNonNegNums)}
  ]{txtclass6}{2in}{13bp}\cgBdry[.5em]
  \pushButton[\CA{Class 2}\AAmouseup{%
    var str="10,17,29,10,20";\r
    this.getField("txtclass6").value=str;\r
    \populateCommaData("math2","class2",v);
  }]{math1class1}{}{13bp}
\end{minipage}\cgBdry[.5em]
\pushButton[\CA{Test}\TU{Press to automatically enter data into the
  two fields above}\AAmouseup{\pbaction}]{tstdata}{}{13bp}\cgBdry[.5em]
\pushButton[\TU{This button re-scales the bar graph so that the
  tallest bar takes the entire height of the region.}\CA{Optimize}
  \AAmouseup{optimizeScaling("math2");}]{optimize3}{}{13bp}\cgBdry[.5em]
\pushButton[\CA{Reset}
  \AAmouseup{resetBargraphs("math2","txtclass5","txtclass6");}
]{reset}{}{13bp}
\end{document}
⟨/comma2⟩
⟨*pro1⟩
```

1.8 `bgjs-pro1.tex`

This file is similar to `bgjs-basic3.tex` but uses layers instead of PDF forms to insert and control the explicit labeling. It requires aeb_pro and a dvips/Adobe Distiller workflow.

```
\documentclass{article}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Requires a dvips/Distiller workflow %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\usepackage[web,eforms={usealtadobe},uselayers]{aeb_pro}
 % \hypersetup{pdfpagemode=UseNone} % web executes this
\usepackage{bargraph-js}
\usepackage{graphicx}

```

```
 % \previewOn\pmpvOn
\makePDasXOn

\begin{insDLJS}{lbl}{Labeling function}
%%   fld="<bgenv-name>@<bg-name>.<bar-name>"
%%   v=value of field
function customLabelsForBars(fld,v){
var pos=fld.indexOf(".");
var bargraph=fld.substring(0,pos);
var bar=fld.substring(pos+1);
  switch(bargraph) {
    case "vehiclesH@hBar":
      switch(bar) {
        case "auto":
          return "Automobiles: "+v;
        case "truck":
         return "Pickup trucks: "+v;
        case "suv":
          return "SUV types: "+v+", costing big \$\$s";
        case "van":
          return "Family vans: "+v
            +", these cost some serious \u20AC\u20ACs";
        default:
          return simpleBarLabels(fld,v);
      }
// other cases can be included
    default:
      return simpleBarLabels(fld,v);
  }
}
\end{insDLJS}
%% applies to all bars with a \TU key
%% \barLabelsTU{customBarLabels}
%% applies to all other bars
\barLabelsNoTU{customLabelsForBars}

\parindent0pt
\begin{document}

In this example, we label the bar graph using typeset content which
is placed in layers (OCGs). The layers are initially hidden, then
made visible when data is entered into the input fields.
\medskip

%% These bars have not \TU key, so the are handled by the argument
%% of \barLabalsNoTU{customLabelsForBars}.
\fbox{\begin{bargraphenv}[width=.67\linewidth,height=2in,
  o=horiz]{vehiclesH}
\presetsbarfor{hBar}{auto}{\BG{red}}
\presetsbarfor{hBar}{truck}{\BG{green}}
```

```
\presetsbarfor{hBar}{suv}{\BG{yellow}}
\presetsbarfor{hBar}{van}{\BG{magenta}}
\begin{bargraph}[nbars=4,gap=3]{hBar}
\barfor{auto}\cmd{\vs{-3bp}\xBld{hBar.auto}
  {\small Automobiles (two or four door)}\eBld\vs{3bp}}
\barfor{truck}\cmd{\vs{-3bp}\xBld{hBar.truck}
  {\small Pickup trucks}\eBld\vs{3bp}}
\barfor{suv}\cmd{\vs{-3bp}\xBld{hBar.suv}
  {\small Sport utility vehicle (SUV)}\eBld\vs{3bp}}
\barfor{van}\cmd{\vs{0bp}\xBld{hBar.van}
  {\small Passenger van}\eBld}
\end{bargraph}
\end{bargraphenv}}\hfill
\begin{minipage}[b][2in][c]
  {.33\linewidth-2\fboxsep-2\fboxrule-10pt}\kern0pt\parskip3pt
\def\presetinputfor#1#2{%
  \AAkeystroke{EFNumber_Keystroke(0, 0, 0, 0, "", true);}
  \AAformat{try{EFNumber_Format(0, 0, 0, 0, "", true);\r
  if(event.rc)toggleSetThisLayer("#2",(event.value!=0));}catch(e){}}
  \AAvalidate{EFRange_Validate(true, 0, false, 0);}
}
\makebox[\widthof{Truck:}][l]{Auto:}
  \inputFor{vehiclesH}{hBar}{auto}{.5in}{11bp}\vcgBdry[3bp]
\makebox[\widthof{Truck:}][l]{Truck:}
  \inputFor{vehiclesH}{hBar}{truck}{.5in}{11bp}\vcgBdry[3bp]
\makebox[\widthof{Truck:}][l]{SUV:}
  \inputFor{vehiclesH}{hBar}{suv}{.5in}{11bp}\vcgBdry[3bp]
\makebox[\widthof{Truck:}][l]{Van:}
  \inputFor{vehiclesH}{hBar}{van}{.5in}{11bp}\vcgBdry[3bp]
\pushButton[\TU{This button re-scales the bar graph so that the
  longest bar takes the entire width of the region.}\CA{Optimize}
  \AAmouseup{optimizeScaling("vehiclesH");}]{optimize1}{}{13bp}
\end{minipage}\medskip

Reset horizontal bar graph: \pushButton[\CA{Reset}
  \AAmouseup{resetBargraphs("hBar","vehiclesH");}
]{reset}{}{13bp}

\bigskip

%% These bars have a \TU key
\fbox{\begin{bargraphenv}[width=2in,height=2in,%
  o=vert]{vehiclesV}\def\WD{2in}
\presetsbarfor{vBar}{auto}{\BG{red}
  \TU{There were @v@ automobiles observed}}
\presetsbarfor{vBar}{truck}{\BG{green}
  \TU{There were @v@ pickups observed during time period}}
\presetsbarfor{vBar}{suv}{\BG{yellow}
  \TU{There were @v@ SUVs observed, costing big
    \textdollar\textdollar s}}
```

```
\presetsbarfor{vBar}{van}{\BG{magenta}
  \TU{There were @v@ family vans filled with happy people;
    reminds me of J\"{u}rgen’s family}}
\begin{bargraph}[nbars=4,gap=3]{vBar}
\barfor{auto}
\cmd{\hs{-3bp}\xBld{vBar.auto}\smash{\raisebox{1pt}
  {\rotatebox[origin=lb]{90}
  {\small Automobiles (two or four door)}}}\eBld\hs{3bp}}
\barfor{truck}
\cmd{\hs{-3bp}\xBld{vBar.truck}\smash{\raisebox{1pt}
  {\rotatebox[origin=lb]{90}{\small Pickup trucks}}}\eBld\hs{3bp}}
\barfor{suv}
\cmd{\hs{-3bp}\xBld{vBar.suv}\smash{\raisebox{1pt}
  {\rotatebox[origin=lb]{90}
  {\small Sport utility vehicle (SUV)}}}\eBld\hs{3bp}}
\barfor{van}
\cmd{\hs{0bp}\xBld{vBar.van}\smash{\raisebox{1pt}
  {\rotatebox[origin=lb]{90}{\small Passenger van}}}\eBld}
\end{bargraph}
\end{bargraphenv}}\hfill
\begin{minipage}[b][2in][c]
  {\linewidth-2\fboxsep-2\fboxrule-2in-10pt}\kern0pt\parskip3pt
\def\presetinputfor#1#2{%
```

The arguments for the `\presetinputfor` command are,

`#1=`⟨*bgenv-name*⟩

`#2=`⟨*bg-name*⟩`.`⟨*bar-name*⟩

```
  \AAkeystroke{EFNumber_Keystroke(0, 0, 0, 0, "", true);}
  \AAformat{try{EFNumber_Format(0, 0, 0, 0, "", true);\r
```

I’ve deliberately named the layers to be ⟨*bg-name*⟩`.`⟨*bar-name*⟩, which is exactly the `#2` argument of `\presetinputfor`. The `toggleSetThisLayer` is a JavaScript function built into the aeb_pro package.

```
  if(event.rc)toggleSetThisLayer("#2",(event.value!=0));}catch(e){}}
  \AAvalidate{EFRange_Validate(true, 0, false, 0);}
}
\makebox[\widthof{Truck:}][l]{Auto:}
  \inputFor{vehiclesV}{vBar}{auto}{.5in}{11bp}\vcgBdry[3bp]
\makebox[\widthof{Truck:}][l]{Truck:}
  \inputFor{vehiclesV}{vBar}{truck}{.5in}{11bp}\vcgBdry[3bp]
\makebox[\widthof{Truck:}][l]{SUV:}
  \inputFor{vehiclesV}{vBar}{suv}{.5in}{11bp}\vcgBdry[3bp]
\makebox[\widthof{Truck:}][l]{Van:}
  \inputFor{vehiclesV}{vBar}{van}{.5in}{11bp}\vcgBdry[4bp]
\pushButton[\TU{This button re-scales the bar graph so that the
tallest bar takes the entire height of the region.}\CA{Optimize}
  \AAmouseup{optimizeScaling("vehiclesV");}
]{optimize2}{}{13bp}
```

```
\end{minipage}\medskip

Reset vertical bar graph:
  \pushButton[\CA{Reset}
    \AAmouseup{resetBargraphs("vBar","vehiclesV");}
]{reset}{}{13bp}\vcgBdry[6bp]
Reset all bar graphs:
  \pushButton[\CA{Reset}
    \AAmouseup{resetBargraphs("hBar","vBar","vehiclesH","vehiclesV");}
]{reset}{}{13bp}

\end{document}
⟨/pro1⟩
⟨*dyn1⟩
```

1.9 bgjs-dyn1.tex

This file gives some ideas how to create discrete probability distributions for discrete random variables that take on evenly spaced values. We present a table of values (both pmf and cdf) and the corresponding pmf and cdf bargraphs.

```
\documentclass{article}
 % \usepackage{web}
\usepackage[usealtadobe]{eforms}
\hypersetup{pdfpagemode=UseNone} % don't need to see bookmarks
\usepackage[dynamic]{bargraph-js}
 % \previewOn\pmpvOn
\makePDasXOn

```

`\displayTable` creates a text field that will contain the tabled values of the discrete distribution. In this demo, we place the table in the left margin.

```
\def\displayTable{\textField[\autoCenter{n}\BC{}\BG{}
  \Ff{\FfMultiline}]{displayTable}{\oddsidemargin}{6in}}
```

Adjust the left margin and `\textwidth`

```
\def\addWdth{.6in}
\addtolength{\oddsidemargin}{\addWdth}
\addtolength{\marginparwidth}{0pt}
\setlength{\textwidth}{\paperwidth-2in-\oddsidemargin}
```

`displayTable` displays the entries in the `aDistr` array and places the string in the field named `fld`.

```
\begin{insDLJS}{dtable}{Display Table of Probabilities}
function displayTable(fld,aDistr) {
  var tableHead=util.printf("   \%s","k ")
    +util.printf("     \%s     ","pmf")
    +util.printf("     \%s     ","cdf")+"\r";
  var str=tableHead;
  for (var i=0; i<aDistr.length;i++) {
    str +=(util.printf("\%4d",aDistr[i][0])
      + util.printf("\%10.6f",aDistr[i][1])
      + util.printf("\%10.6f",aDistr[i][2])
      + "\r"
    );
  }
  var f=this.getField(fld);
  if (f!=null) f.value=str;
}
```

This function is not necessary, but simply demonstrates the feature of writing your own custom labeling function for dynamic bars. This example is modeled after the default function `_labelDyBars`. The argument of this function are `pr` (the current value of the random variable); `v` (the probability associated with `pr`; and `bPmf` is `true` when this is for a `pmf` and `false` when this is for a `cdf`.

```
function customDyLabels(pr,v,bPmf) {
  return bPmf?("P(Y="+pr+")="+v):("P(Y<="+pr+")="+v);
}
\end{insDLJS}

\parindent0pt
\begin{document}
\begin{center}\bfseries
   The bar graph for the probability mass function (pmf)
\end{center}
\noindent\makebox[0pt][r]{\smash
  {\raisebox{-4in+2\fboxsep+2\fboxrule}{\displayTable}%
    \hspace{\marginparsep}}}
\fbox{\begin{bargraphenv}[width=\linewidth-2\fboxsep-2\fboxrule,%
  height=2in,o=vert]{Pmf}
\begin{bargraph}{pmfBar}\isdynamic\end{bargraph}
\end{bargraphenv}}\vcgBdry[\medskipamount]
\pushButton[\TU{This button re-scales the bar graph so that the
  tallest bar takes the entire height of the region; shift-click
  reverts bar graph to its original scaling.}\CA{Optimize}\AAmouseup{%
  try{displayDyBargraph("Pmf",aPmfCdf,true,!event.shift)}
    catch(e){};}
]{optimize}{}{13bp}\cgBdry[1em]
Under normal scaling, the height of this region is 1 unit, when the
bar graph is optimized, the height is the height of the tallest bar.

\medskip
\begin{center}\bfseries
   The bar graph for the cumulative distribution function (cdf)
\end{center}
\fbox{\begin{bargraphenv}[width=\linewidth-2\fboxsep-2\fboxrule,%
  height=2in,o=vert]{Cdf}
\begin{bargraph}{cdfBar}\isdynamic\end{bargraph}
\end{bargraphenv}}\vcgBdry[\medskipamount]
The height of the region above is 1~unit.

```

```
\begin{flushleft}
\textbf{Generate some probability distributions}\medskip

\pushButton[\CA{Dist1}\AAmouseup{%
```

Here we hard-wire the `aPmfCdf` matrix

```
  var aPmfCdf=new Array(
    [1,.2,.2],
    [2,.1,.3],
    [3,.2,.5],
    [4,.2,.7],
    [5,.1,.8],
    [6,.2,1]
  );\r
  displayTable("displayTable",aPmfCdf);\r
```

In the next two `displayDyBargraph`, we use the custom dynamic labeling function `customDyLabels`.

```
  displayDyBargraph("Pmf",aPmfCdf,true,false,%
{bc:color.blue,fc:color.red,lbl:customDyLabels});\r
  displayDyBargraph("Cdf",aPmfCdf,false,false,%
{bc:color.blue,fc:color.red,lbl:customDyLabels});
}]{Distr1}{}{13bp}\cgBdry[.5em]
\pushButton[\CA{Dist2}\AAmouseup{%
```

In this example, we define the values of the distribution and their masses. The cdf is later computed and the `aPmfCdf` is calculated.

```
  var aValues=[-2,-1,0,1,2,3,4];\r
  var apmfs=[2/20,3/20,6/20,1/20,2/20,6/20,2/20];\r
  var acdfs=new Array();\r
  var aPmfCdf=new Array();\r
  acdfs[0]=apmfs[0];\r
  aPmfCdf[0]=[aValues[0],apmfs[0],acdfs[0]];\r
  var l=aValues.length-1;\r
  for (var i=1; i<l; i++) {\r\t
    acdfs[i]=apmfs[i]+acdfs[i-1];\r\t
    aPmfCdf[i]=[aValues[i],apmfs[i],acdfs[i]]\r
  }\r
  displayTable("displayTable",aPmfCdf);\r
  displayDyBargraph("Pmf",aPmfCdf,true,false,%
{bc:color.blue,fc:color.red});\r
  displayDyBargraph("Cdf",aPmfCdf,false,false,%
{bc:color.blue,fc:color.red});
}]{Distr2}{}{13bp}\cgBdry[.5em]
\pushButton[\TU{Randomly generate a probability distribution}
  \CA{Random}\AAmouseup{%
```

We set the maximum values of the distribution through the variable. When `maxN` is too large, it take some time to make all calculations and display all the bar graphs. Experiment with the value of `maxN`. `maxN`.

```
  var maxN=40;\r
  var aPmfCdf=new Array();\r
  var n=Math.round(Math.random()*maxN);\r
  var aValues=[],apmfs=[],acdfs=[];\r
  var total=0;\r
  for (var i=0; i<n; i++) {\r\t
    aValues[i]=i;\r\t
    apmfs[i]=Math.round(Math.random()*maxN);\r\t
    total+=(apmfs[i]);\r
  }\r
  for (var i=0; i<n; i++) {\r\t
    apmfs[i]=apmfs[i]/total;\r\t
    acdfs[i]=apmfs[i]+((i==0)?0:acdfs[i-1]);\r\t
    aPmfCdf[i]=[aValues[i],apmfs[i],acdfs[i]];\r
  }\r
  displayTable("displayTable",aPmfCdf);\r
  displayDyBargraph("Pmf",aPmfCdf,true,false,
{bc:color.blue,fc:color.red});\r
  displayDyBargraph("Cdf",aPmfCdf,false,false,
{bc:color.blue,fc:color.red});
}]{Distr3}{}{13bp}\cgBdry[.5em]
 \pushButton[\CA{Reset}
    \TU{Press to clear the fields of this page, and shift-press to
      clear all fields.}
    \AAmouseup{%
    this.calculate=true;\r
    if (event.shift)\r\t
      this.resetForm();\r
    else {\r\t
      this.removeField("Pmf@pmfBar");\r\t
      this.removeField("Cdf@cdfBar");\r\t
      this.resetForm("displayTable");\r
    }
}]{reset}{}{13bp}\cgBdry[2em]
\pushButton[\CA{Toggle Border}\AAmouseup{%
  var f=this.getField("Pmf@pmfBar");\r
  if ( f!=null ) {\r\t
    var a=f.getArray();\r\t\t
    if (f.saveStrokeColor==undefined) f.saveStrokeColor=color.red;\r\t
    if (!color.equal(a[0].strokeColor,color.transparent))\r\t\t
      f.saveStrokeColor=a[0].strokeColor;\r\t
    var g=getField("Cdf@cdfBar");\r\t
    if (color.equal(a[0].strokeColor,color.transparent)) {\r\t\t
      f.strokeColor=f.saveStrokeColor;\r\t\t
      g.strokeColor=f.saveStrokeColor;\r\t
    } else {\r\t\t
      f.strokeColor=color.transparent;\r\t\t
      g.strokeColor=color.transparent;\r\t
    }\r
  }
}]{toggleBdry}{}{13bp}
```

```
\end{flushleft}
\end{document}
⟨/dyn1⟩
```

Index The index goes here.
Execute `makeindex -s gind.ist -o bgjs-examples.ind bgjs-examples.idx` on the command line and recompile `bgjs-examples.dtx`.

Change History The list of changes goes here.
Execute `makeindex -s gglo.ist -o bgjs-examples.gls bgjs-examples.glo` on the command line and recompile `bgjs-examples.dtx`.