

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

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

processed June 4, 2019

Contents

1	The Examples	1
1.1	bgjs-basic1.tex	1
1.2	bgjs-basic2.tex	5
1.3	bgjs-basic3.tex	8
1.4	bgjs-basic4.tex	12
1.5	bgjs-adv1.tex	13
1.6	bgjs-comma1.tex	16
1.7	bgjs-comma2.tex	17
1.8	bgjs-pro1.tex	20
1.9	bgjs-dyn1.tex	24

1 <*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.

2 </package>

3 <*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.

4 \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.

5 % \usepackage{web}

Input “the package”

```
6 \usepackage{bargraph-js}
7 \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.

```
8 % \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.

```
9 \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.

```
10 \begin{insDLJS}{lbl}{Labeling function}
11 %% fld="<bgenv-name>@<bg-name>.<bar-name>"
12 %% v=value of field
13 function customLabelsForBars(fld,v){
14 var pos=fld.indexOf(".");
15 var bargraph=fld.substring(0,pos);
16 var bar=fld.substring(pos+1);
17 switch(bargraph) {
18 case "vehiclesH@hBar":
19 switch(bar) {
20 case "auto":
21 return "Automobiles: "+v;
22 case "truck":
23 return "Pickup trucks: "+v;
24 case "suv":
25 return "SUV types: "+v+", costing big \$$s";
26 case "van":
27 return "Family vans: "+v
28 +", these cost some serious \u20AC\u20ACs";
29 default:
30 return simpleBarLabels(fld,v);
31 }
32 // other cases can be included
33 default:
34 return simpleBarLabels(fld,v);
35 }
36 }
37 \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.

```
38 % \barLabelsTU{"Within the \@env@ environment, within the
39 %   \@barname@ environment, the bar \@bar@"}
40 %   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.

```
41 \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,

```
42 % \barLabelsNoTU{"Vehicle Data: "+o.barname+": "
43 %   + 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.

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

```
45
```

```
46 \parindentOpt
```

```
47 \begin{document}
```

```
48
```

```
49 \textbf{Comments.} The bar graphs here are based on count data.
```

```
50 Input natural numbers into the text fields. If the width of a horizontal
```

```
51 bar or the height of a vertical bar goes outside the graphing area, the
```

```
52 bar graph is \emph{automatically re-scaled} so the widest one (for
```

```
53 horizontal) or highest one (for vertical) now fits. Try the
```

```
54 \textsf{Optimize} buttons as well.\medskip
```

```
55
```

```
56 \textbf{Instructions:} Stand at an intersection for one hour and count
```

```
57 the number of vehicles of each of the specified types passing you going
```

```
58 in one direction. Enter your results in the fields below.\medskip
```

```
59
```

```
60 %% Here we use the default method of labeling the bar graphs
```

```
61 \fbox{\begin{bargraphenv}[width=.67\linewidth,height=2in,
```

```
62   o=horiz]{vehiclesH}
```

```
63 \presetsbarfor{hBar}{auto}{\BG{red}}
```

```
64 \presetsbarfor{hBar}{truck}{\BG{green}}
```

```
65 \presetsbarfor{hBar}{suv}{\BG{yellow}}
```

```
66 \presetsbarfor{hBar}{van}{\BG{magenta}}
```

```
67 \begin{bargraph}[nbars=4,gap=3]{hBar}
```

```

68 \barfor{auto}\barfor{truck}\barfor{suv}\barfor{van}
69 \end{bargraph}
70 \end{bargraphenv}}\hfill
71 \begin{minipage}[b][2in][c]
72   {.33\linewidth-2\fbboxsep-2\fbboxrule-10pt}\kern0pt\parskip3pt
73 \makebox[\widthof{Truck:}][l]{Auto:}
74   \inputFor{vehiclesH}{hBar}{auto}{.5in}{11bp}\vcgBdry[3bp]
75 \makebox[\widthof{Truck:}][l]{Truck:}
76   \inputFor{vehiclesH}{hBar}{truck}{.5in}{11bp}\vcgBdry[3bp]
77 \makebox[\widthof{Truck:}][l]{SUV:}
78   \inputFor{vehiclesH}{hBar}{suv}{.5in}{11bp}\vcgBdry[3bp]
79 \makebox[\widthof{Truck:}][l]{Van:}
80   \inputFor{vehiclesH}{hBar}{van}{.5in}{11bp}\vcgBdry[3bp]
81   \pushButton[\TU{This button re-scales the bar graph so that the
82   longest bar takes the entire width of the region.}\CA{Optimize}
83   \AAMouseup{optimizeScaling("vehiclesH");}]{optimize1}{}{13bp}
84 \end{minipage}\medskip
85
86 Reset horizontal bar graph: \pushButton[\CA{Reset}
87   \AAMouseup{resetBargraphs("vehiclesH");}]{reset}{}{13bp}\bigskip
88
89 \fbox{\begin{bargraphenv}[width=2in,height=2in,o=vert]{vehiclesV}
90 \presetsbarfor{vBar}{auto}{\BG{red}}
91   \TU{There were @v@ automobiles observed}}
92 \presetsbarfor{vBar}{truck}{\BG{green}}
93   \TU{There were @v@ pickups observed during time period}}
94 \presetsbarfor{vBar}{suv}{\BG{yellow}}
95   \TU{There were @v@ SUVs observed, costing big
96     \textdollar\textdollar s}}
97 \presetsbarfor{vBar}{van}{\BG{magenta}}
98   \TU{There were @v@ family vans filled with happy people;
99     reminds me of J\"{u}rgen's family}}
100 \begin{bargraph}[nbars=4,gap=3]{vBar}
101 \barfor{auto}\barfor{truck}\barfor{suv}\barfor{van}
102 \end{bargraph}
103 \end{bargraphenv}}\hfill
104 \begin{minipage}[b][2in][c]
105   {\linewidth-2\fbboxsep-2\fbboxrule-2in-10pt}\kern0pt\parskip3pt
106 \makebox[\widthof{Truck:}][l]{Auto:}
107   \inputFor{vehiclesV}{vBar}{auto}{.5in}{11bp}\vcgBdry[3bp]
108 \makebox[\widthof{Truck:}][l]{Truck:}
109   \inputFor{vehiclesV}{vBar}{truck}{.5in}{11bp}\vcgBdry[3bp]
110 \makebox[\widthof{Truck:}][l]{SUV:}
111   \inputFor{vehiclesV}{vBar}{suv}{.5in}{11bp}\vcgBdry[3bp]
112 \makebox[\widthof{Truck:}][l]{Van:}
113   \inputFor{vehiclesV}{vBar}{van}{.5in}{11bp}\vcgBdry[4bp]
114 \pushButton[\TU{This button re-scales the bar graph so that the
115 tallest bar takes the entire height of the region.}\CA{Optimize}
116   \AAMouseup{optimizeScaling("vehiclesV");}
117 ]]{optimize2}{}{13bp}

```

```

118 \end{minipage}\medskip
119
120 Reset vertical bar graph:
121 \pushButton[\CA{Reset}\AAmouseup{resetBargraphs("vehiclesV");}
122 ]{reset}{-}{13bp}\vcgBdry[6bp]
123 Reset all bar graphs:
124 \pushButton[\CA{Reset}
125 \AAmouseup{resetBargraphs("vehiclesH","vehiclesV");}
126 ]{reset}{-}{13bp}
127 \end{document}
128 </basic1>
129 <*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.

```

130 \documentclass{article}
131 % \usepackage{web}
132 \usepackage{bargraph-js}
133 \hypersetup{pdfpagemode=UseNone} % don't need to see bookmarks

```

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

```

134 \usepackage{graphicx}
135 % \previewOn\pmpvOn
136 \makePDasXOn
137
138 \begin{insDLJS}{lbl}{Labeling function}
139 %% fld="<bgenv-name>@<bg-name>.<bar-name>"
140 %% v=value of field
141 function customLabelsForBars(fld,v){
142 var pos=fld.indexOf(".");
143 var bargraph=fld.substring(0,pos);
144 var bar=fld.substring(pos+1);
145 switch(bargraph) {
146 case "vehiclesH@hBar":
147 switch(bar) {
148 case "auto":
149 return "Automobiles: "+v;
150 case "truck":
151 return "Pickup trucks: "+v;
152 case "suv":
153 return "SUV types: "+v+", costing big \$$\$s";
154 case "van":
155 return "Family vans: "+v
156 +", these cost some serious \u20AC\u20ACs";
157 default:
158 return simpleBarLabels(fld,v);
159 }
160 // other cases can be included

```

```

161     default:
162         return simpleBarLabels(fld,v);
163     }
164 }
165 \end{insDLJS}
166 % \barLabelsTU{"Within the \\@env@" environment,
167 % within the \\@barname@" environment, the bar \\@bar@" has a
168 % value of @v@"}
169 \barLabelsTU{customBarLabels} % applies to all bars with a \TU key
170
171 % \barLabelsNoTU{customLabelsForBars} % applies to all other bars
172 \barLabelsNoTU{"Vehicle Data: "+o.barname+": "
173 +o.bar+", Value: "+o.value}
174
175 \parindentOpt
176 \begin{document}
177
178 The bar graphs here are based on count data. Input natural numbers
179 into the text fields. If the width of a horizontal bar or the height
180 of a vertical bar goes outside the graphing area, the bar graph is
181 \emph{automatically re-scaled} so the widest one (for horizontal)
182 or highest one (for vertical) now fits. Try the Optimize buttons as
183 well.\medskip
184
185 %% These bars have no \TU key, so current value of \barLabelsTU
186 %% applies.
187 \fbox{\begin{bargraphenv}
188   [width=.67\linewidth,height=2in,o=horiz]{vehiclesH}
189 \presetsbarfor{hBar}{auto}{\BG{red}}
190 \presetsbarfor{hBar}{truck}{\BG{green}}
191 \presetsbarfor{hBar}{suv}{\BG{yellow}}
192 \presetsbarfor{hBar}{van}{\BG{magenta}}
193 \begin{bargraph}[nbars=4,gap=3]{hBar}
194 \barfor{auto}
195 \cmd{\vs{-2bp}}{\small{Automobile (two or four door)}}\vs{3bp}}
196 \barfor{truck}
197 \cmd{\vs{-2bp}}{\small{Pickup truck}}\vs{3bp}}
198 \barfor{suv}
199 \cmd{\vs{-2bp}}{\small{Sport utility vehicle (SUV)}}\vs{3bp}}
200 \barfor{van}
201 \cmd{\vs{1bp}}{\small{Passenger van}}}}
202 \end{bargraph}
203 \end{bargraphenv}}\hfill
204 \begin{minipage}[b][2in][c]
205   {.33\linewidth-2\fboxsep-2\fboxrule-10pt}\kern0pt\parskip3pt
206 \makebox[\widthof{Truck:}][l]{Auto:}
207   \inputFor{vehiclesH}{hBar}{auto}{.5in}{11bp}\vcgBdry[3bp]
208 \makebox[\widthof{Truck:}][l]{Truck:}
209   \inputFor{vehiclesH}{hBar}{truck}{.5in}{11bp}\vcgBdry[3bp]
210 \makebox[\widthof{Truck:}][l]{SUV:}

```

```

211 \inputFor{vehiclesH}{hBar}{suv}{.5in}{11bp}\vcgBdry[3bp]
212 \makebox[\widthof{Truck:}][l]{Van:}
213 \inputFor{vehiclesH}{hBar}{van}{.5in}{11bp}\vcgBdry[3bp]
214 \pushButton[\TU{This button re-scales the bar graph so that the
215 longest bar takes the entire width of the region.}\CA{Optimize}
216 \AAmouseup{optimizeScaling("vehiclesH");}{optimize1}]{13bp}
217 \end{minipage}\medskip
218
219 Reset horizontal bar graph: \pushButton[\CA{Reset}
220 \AAmouseup{resetBargraphs("vehiclesH");}{reset}]{13bp}
221
222 \bigskip
223
224 \fbox{\begin{bargraphenv}[width=2in,height=2in,o=vert]{vehiclesV}
225 \presetsbarfor{vBar}{auto}{\BG{red}
226 \TU{There were @v@ automobiles observed}}
227 \presetsbarfor{vBar}{truck}{\BG{green}
228 \TU{There were @v@ pickups observed during time period}}
229 \presetsbarfor{vBar}{suv}{\BG{yellow}
230 \TU{There were @v@ SUVs observed, costing big
231 \textdollar\textdollar s}}
232 \presetsbarfor{vBar}{van}{\BG{magenta}
233 \TU{There were @v@ family vans filled with happy people;
234 reminds me of J\{u}rgen's family}}
235 \begin{bargraph}[nbars=4,gap=3]{vBar}
236 \barfor{auto}
237 \cmd{\hs{-2bp}\smash{\rotatebox[origin=lb]{90}
238 {\small{Automobile (two or four door)}}}\hs{3bp}}
239 \barfor{truck}
240 \cmd{\hs{-2bp}\smash{\rotatebox[origin=lb]{90}
241 {\small{Pickup truck}}}\hs{3bp}}
242 \barfor{suv}
243 \cmd{\hs{-2bp}\smash{\rotatebox[origin=lb]{90}
244 {\small{Sport utility vehicle (SUV)}}}\hs{3bp}}
245 \barfor{van}
246 \cmd{\hs{1bp}\smash{\rotatebox[origin=lb]{90}
247 {\small{Passenger van}}}}
248 \end{bargraph}
249 \end{bargraphenv}}\hfill
250 \begin{minipage}[b][2in][c]
251 {\linewidth-2\fbboxsep-2\fbboxrule-2in-10pt}\kern0pt\parskip3pt
252 \makebox[\widthof{Truck:}][l]{Auto:}
253 \inputFor{vehiclesV}{vBar}{auto}{.5in}{11bp}\vcgBdry[3bp]
254 \makebox[\widthof{Truck:}][l]{Truck:}
255 \inputFor{vehiclesV}{vBar}{truck}{.5in}{11bp}\vcgBdry[3bp]
256 \makebox[\widthof{Truck:}][l]{SUV:}
257 \inputFor{vehiclesV}{vBar}{suv}{.5in}{11bp}\vcgBdry[3bp]
258 \makebox[\widthof{Truck:}][l]{Van:}
259 \inputFor{vehiclesV}{vBar}{van}{.5in}{11bp}\vcgBdry[4bp]
260 \pushButton[\TU{This button re-scales the bar graph so that the

```

```

261 tallest bar takes the entire height of the region.}\CA{Optimize}
262 \AAmouseup{optimizeScaling("vehiclesV");}{optimize2}{13bp}
263 \end{minipage}\medskip
264
265 Reset vertical bar graph:
266 \pushButton[\CA{Reset}
267   \AAmouseup{resetBargraphs("vehiclesV");}
268 ]{reset}{13bp}\vcgBdry[6bp]
269 Reset all bar graphs:
270 \pushButton[\CA{Reset}
271   \AAmouseup{resetBargraphs("vehiclesH","vehiclesV");}{reset}{13bp}
272
273 \end{document}
274 \</basic2>
275 \< *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.

```

276 \documentclass{article}
277 % \usepackage{web}
278 \usepackage[usealtadobe]{eforms}
279 \hypersetup{pdfpagemode=UseNone} % don't need to see bookmarks
280 \usepackage{bargraph-js}
281 % \previewOn\pmpvOn
282 \makePDasXOn
283
284 \begin{insDLJS}{lbl}{Labeling function}
285 %% fld="<bgenv-name>@<bg-name>.<bar-name>"
286 %% v=value of field
287 function customLabelsForBars(fld,v){
288 var pos=fld.indexOf(".");
289 var bargraph=fld.substring(0,pos);
290 var bar=fld.substring(pos+1);
291 switch(bargraph) {
292 case "vehiclesH@hBar":
293   switch(bar) {
294     case "auto":
295       return "Automobiles: "+v;
296     case "truck":
297       return "Pickup trucks: "+v;
298     case "suv":
299       return "SUV types: "+v+", costing big \$$\$s";
300     case "van":
301       return "Family vans: "+v
302         +", these cost some serious \u20AC\u20ACs";
303     default:
304       return simpleBarLabels(fld,v);

```



```

305     }
306 // other cases can be included
307     default:
308         return simpleBarLabels(fld,v);
309 }
310 }
311 \end{insDLJS}
312 % applies to all bars with a \TU key
313 % \barLabelsTU{customBarLabels}
314 % applies to all other bars
315 % \barLabelsNoTU{customLabelsForBars}
316
317 \parindentOpt
318 \begin{document}
319
320 In this example, we label the bar graph using text fields (using the
321 \verb~\labelFld~ command defined in this package). The fields are
322 initially hidden, then made visible when data is entered into the
323 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.

```

324 %% Here we use the default method of labeling the bar graphs
325 %% This bar graph is horizontally oriented
326 \fbox{\begin{bargraphenv}[width=.67\linewidth,height=2in,
327   o=horiz]{vehiclesH}\def\WD{2in}
328   \presetsbarfor{hBar}{auto}{\BG{red}}
329   \presetsbarfor{hBar}{truck}{\BG{green}}
330   \presetsbarfor{hBar}{suv}{\BG{yellow}}
331   \presetsbarfor{hBar}{van}{\BG{magenta}}
332   \begin{bargraph}[nbars=4,gap=3]{hBar}
333   \barfor{auto}
334   \cmd{\vs{-3bp}\labelFld[\textSize{10}]
335     {Automobiles (two or four door)}{hBar.auto}{\WD}{13bp}\vs{3bp}}
336   \barfor{truck}
337   \cmd{\vs{-3bp}\labelFld[\textSize{10}]
338     {Pickup trucks}{hBar.truck}{\WD}{13bp}\vs{3bp}}
339   \barfor{suv}
340   \cmd{\vs{-3bp}\labelFld[\textSize{10}]
341     {Sport utility vehicle (SUV)}{hBar.suv}{\WD}{13bp}\vs{3bp}}
342   \barfor{van}
343   \cmd{\vs{0bp}\labelFld[\textSize{10}]
344     {Passenger van}{hBar.van}{\WD}{13bp}}
345   \end{bargraph}
346 \end{bargraphenv}}\hfill
347 \begin{minipage}[b][2in][c]

```

```

348 { .33\linewidth-2\fbboxsep-2\fbboxrule-10pt}\kern0pt\parskip3pt
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
toggleFldVisibility() <bg-name>.<bar-name>@<bgenv-name>.
349 \def\presetinputfor#1#2{%
The arguments for the \presetinputfor command are,

#1=<bgenv-name>

#2=<bg-name>.<bar-name>

350 \AAkeystroke{EFNumber_Keystroke(0, 0, 0, 0, "", true);}
351 \AAformat{try{EFNumber_Format(0, 0, 0, 0, "", true);}r
352 if(event.rc)toggleFldVisibility("#2@#1",%
353 (event.value!=0));}catch(e){}}
354 \AAvalidate{EFRRange_Validate(true, 0, false, 0);}
355 }
356 \makebox[\widthof{Truck:}] [1]{Auto:}
357 \inputFor{vehiclesH}{hBar}{auto}{.5in}{11bp}\vcgBdry[3bp]
358 \makebox[\widthof{Truck:}] [1]{Truck:}
359 \inputFor{vehiclesH}{hBar}{truck}{.5in}{11bp}\vcgBdry[3bp]
360 \makebox[\widthof{Truck:}] [1]{SUV:}
361 \inputFor{vehiclesH}{hBar}{suv}{.5in}{11bp}\vcgBdry[3bp]
362 \makebox[\widthof{Truck:}] [1]{Van:}
363 \inputFor{vehiclesH}{hBar}{van}{.5in}{11bp}\vcgBdry[3bp]
364 \pushButton[\TU{This button re-scales the bar graph so that the
365 longest bar takes the entire width of the region.}\CA{Optimize}
366 \AAMouseup{optimizeScaling("vehiclesH");}]{optimize1}{}{13bp}
367 \end{minipage}\medskip
368
369 Reset horizontal bar graph: \pushButton[\CA{Reset}
370 \AAMouseup{resetBargraphs("hBar","vehiclesH");}\r
371 ]}{reset}{}{13bp}
372
373 \bigskip
374
375 %% This bar graph is vertical so we must insert \R{90} in the
376 %% options argument of \labelFld.
377 \fbox{\begin{bargraphenv}[width=2in,height=2in,%
378 o=vert]{vehiclesV}\def\WD{2in}
379 \presetsbarfor{vBar}{auto}{\BG{red}
380 \TU{There were @v@ automobiles observed}}
381 \presetsbarfor{vBar}{truck}{\BG{green}
382 \TU{There were @v@ pickups observed during time period}}
383 \presetsbarfor{vBar}{suv}{\BG{yellow}
384 \TU{There were @v@ SUVs observed, costing big
385 \textdollar\textdollar s}}

```

```

386 \presetsbarfor{vBar}{van}{\BG{magenta}
387 \TU{There were @v@ family vans filled with happy people;
388   reminds me of J\{u}rgen's family}}
389 \begin{bargraph}[nbars=4,gap=3]{vBar}
390 \barfor{auto}
391 \cmd{\hs{-3bp}\labelFld[\R{90}\textSize{10}]
392   {Automobiles (two or four door)}{vBar.auto}{\WD}{13bp}\hs{3bp}}
393 \barfor{truck}
394 \cmd{\hs{-3bp}\labelFld[\R{90}\textSize{10}]
395   {Pickup trucks}{vBar.truck}{\WD}{13bp}\hs{3bp}}
396 \barfor{suv}
397 \cmd{\hs{-3bp}\labelFld[\R{90}\textSize{10}]
398   {Sport utility vehicle (SUV)}{vBar.suv}{\WD}{13bp}\hs{3bp}}
399 \barfor{van}
400 \cmd{\hs{0bp}\labelFld[\R{90}\textSize{10}]
401   {Passenger van}{vBar.van}{\WD}{13bp}}
402 \end{bargraph}
403 \end{bargraphenv}}\hfill
404 \begin{minipage}[b][2in][c]
405   {\linewidth-2\fbboxsep-2\fbboxrule-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>`.

```

406 \def\presetinputfor#1#2{%
The arguments for the \presetinputfor command are,

#1=<bgenv-name>

#2=<bg-name>.<bar-name>

407 \AAkeystroke{EFNumber_Keystroke(0, 0, 0, 0, "", true);}
408 \AAformat{try{EFNumber_Format(0, 0, 0, 0, "", true);\r
409   if(event.rc)toggleFldVisibility("#2@#1",%
410 (event.value!=0));}catch(e){}}
411 \AAvalidate{EFRRange_Validate(true, 0, false, 0);}
412 }
413 \makebox[\widthof{Truck:}] [l]{Auto:}
414 \inputFor{vehiclesV}{vBar}{auto}{.5in}{11bp}\vcgBdry[3bp]
415 \makebox[\widthof{Truck:}] [l]{Truck:}
416 \inputFor{vehiclesV}{vBar}{truck}{.5in}{11bp}\vcgBdry[3bp]
417 \makebox[\widthof{Truck:}] [l]{SUV:}
418 \inputFor{vehiclesV}{vBar}{suv}{.5in}{11bp}\vcgBdry[3bp]
419 \makebox[\widthof{Truck:}] [l]{Van:}
420 \inputFor{vehiclesV}{vBar}{van}{.5in}{11bp}\vcgBdry[4bp]
421 \pushButton[\TU{This button re-scales the bar graph so that the
422   tallest bar takes the entire height of the region.}\CA{Optimize}]
423 \AAmouseup{optimizeScaling("vehiclesV");}

```

```

424 ]{optimize2}{}{13bp}
425 \end{minipage}\medskip
426
427 Reset vertical bar graph:
428 \pushButton[\CA{Reset}
429 \AAmouseup{resetBargraphs("vBar","vehiclesV");}]
430 ]{reset}{}{13bp}\vcgBdry[6bp]
431 Reset all bar graphs:
432 \pushButton[\CA{Reset}
433 \AAmouseup{resetBargraphs("hBar","vBar","vehiclesH","vehiclesV");}]
434 ]{reset}{}{13bp}
435 \end{document}
436 </basic3>
437 <*basic4>

```

1.4 bgjs-basic4.tex

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

```

438 \documentclass{article}
439 % \usepackage{web}
440 \usepackage{bargraph-js}
441 \hypersetup{pdfpagemode=UseNone} % don't need to see bookmarks
442 % \previewOn\mpmvOn
443 \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.

```

444 \barLabelsNoTU{"Profits (" + o.barname.substring(1) + ") for "
445 + o.bar + ": " + ((o.value < 0) ? "-" : "") + "$" + Math.abs(o.value)
446 + " (million)"}
447
448 \parindent0pt
449 \begin{document}
450
451 \textbf{Instructions:} Enter the profits for the year 2018, enter
452 a negative number if there was a loss for a particular quarter.\medskip
453
454 %% origin=.5,showaxis=true : change o=vert to o=horiz
455 \fbox{\begin{bargraphenv}[width=2in,height=2in,o=vert,%
456 origin=.5,showaxis=true]{profits}
457 \presetsbarfor{y2018}{Q1}{\BG{red}}
458 \presetsbarfor{y2018}{Q2}{\BG{green}}
459 \presetsbarfor{y2018}{Q3}{\BG{yellow}}
460 \presetsbarfor{y2018}{Q4}{\BG{magenta}}
461 \begin{bargraph}[nbars=4,gap=3]{y2018}
462 \barfor{Q1}\barfor{Q2}\barfor{Q3}\barfor{Q4}
463 \end{bargraph}
464 \end{bargraphenv}}\hfill

```

```

465 \begin{minipage}[b][2in][c]
466   {\linewidth-2\fbboxsep-2\fbboxrule-2in-10pt}\kern0pt\parskip3pt
467 \def\presetinputfor#1#2{%
468   \AAkeystroke{AFNumber_Keystroke(0, 0, 0, 0, "", true);}
469   \AAformat{AFNumber_Format(0, 0, 0, 0, "", true);}
470 }
471 \makebox[\linewidth][l]{\textbf
472   {Profits for year 2018 (in millions)}}\ [3pt]
473 \makebox[\widthof{Q1:}][l]{Q1:}
474   \inputFor{profits}{y2018}{Q1}{.5in}{13bp}\vcgBdry[3bp]
475 \makebox[\widthof{Q1:}][l]{Q2:}
476   \inputFor{profits}{y2018}{Q2}{.5in}{13bp}\vcgBdry[3bp]
477 \makebox[\widthof{Q1:}][l]{Q3:}
478   \inputFor{profits}{y2018}{Q3}{.5in}{13bp}\vcgBdry[3bp]
479 \makebox[\widthof{Q1:}][l]{Q4:}
480   \inputFor{profits}{y2018}{Q4}{.5in}{13bp}\vcgBdry[4bp]
481 \pushButton[\TU{This button re-scales the bar graph so that the
482 tallest bar takes the entire height of the region.}\CA{Optimize}
483   \AAMouseup{optimizeScaling("profits");}
484 ]{optimize2}{}{13bp}
   Supply manual re-scaling text field and push button
485 \displaysfFor{profits}{.5in}{13bp}\olBdry
486 \manualsfFor[\CA{Rescale}\TU{Enter a new scale factor in the
487 text field, then press this button}
488 ]{profits}{}{13bp}
489 \end{minipage}\medskip
490
491 Reset vertical bar graph:
492   \pushButton[\CA{Reset}\AAMouseup{resetBargraphs("profits");}
493 ]{reset}{}{13bp}
494 \end{document}
495 \</basic4>
496 \<*adv1>

```

1.5 bgjs-adv1.tex

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

```

497 \documentclass{article}
498 % \usepackage{web}
499 \usepackage[usealtadobe,setcorder]{eforms}
500 \usepackage{bargraph-js}
501 \hypersetup{pdfpagemode=UseNone} % don't need to see bookmarks
502 % \previewOn\pmpvOn
503 \makePDasXOn
504

```

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.

```

505 \calcOrder{totalClass}
506
507 \begin{insDLJS}{lbl}{Labeling function}
508 %% fld="<bgenv-name>@<bg-name>.<bar-name>"
509 %% v=value of field
510 function customLabelsForBars(fld,v){
511 var pos=fld.indexOf(".");
512 var bargraph=fld.substring(0,pos);
513 var bar=fld.substring(pos+1);
514 switch(bargraph) {
515 case "math@class":
516     var f=this.getField("totalClass");
517     var tot=f.value;
518     var V=(tot!=0)?((v/tot)*100):(v*100);
519     V=util.printf("%.1f",V);
520     switch(bar) {
521     case "A":
522         return V+"% of the class received an 'A'";
523     case "B":
524         return V+"% of the class received an 'B'";
525     case "C":
526         return V+"% of the class received an 'C'";
527     case "D":
528         return V+"% of the class received an 'D'";
529     case "F":
530         return V+"% of the class received an 'F'";
531     default:
532         return "unknown result";
533     }
534 default:
535     return simpleBarLabels(fld,v);
536 }
537 }
538 \end{insDLJS}
539
540 %% applies to all bars without a \TU key
541 \barLabelsNoTU{customLabelsForBars}
542
543 \parindent0pt
544 \begin{document}
545
546 Enter the number of class members that received the grades
547 A, B, C, D, and~F.\medskip
548

```

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.

```

549 \begin{defineJS}[\catcode'\*=0\relax]{\presetinputforcalc}
550 var f=this.getField("pctFor.*p(2)*p(1)");
551 var g=this.getField("totalClass");
552 if (event.value!=0)
553   f.value=( g.value==0 )?1:(event.value/g.value);
554 \end{defineJS}
555
556 \def\pctThisGrade#1#2#3{\textField[\Ff{\FfReadOnly}\Q{2}
557   \AAkeystroke{EFPercent_Keystroke(1, 1);}
558   \AAformat{if (event.value!=0)EFPercent_Format(1, 1);\r
559   else event.value="";}

```

The name of this field shall be

`pctFor.<bg-name>.<bar-name>@<bgenv-name>`

```

560 ]{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.

```

561 \fbox{\scaleFactorDef{dataForEnv["math"].height/100}%
562 \begin{bargraphenv}[width=2in,height=2in,o=vert]{math}
563 \begin{bargraph}[nbars=5,gap=3]{class}
564 \barfor{A}\barfor{B}\barfor{C}\barfor{D}\barfor{F}
565 \end{bargraph}
566 \end{bargraphenv}}\hfill
567 \begin{minipage}[b][2in][c]
568   {\linewidth-2\fboxsep-2\fboxrule-2in-10pt}\kern0pt\parskip3pt
569 \def\presetinputfor#1#2{\AAcalculate{\presetinputforcalc}
570   \AAkeystroke{AFNumber_Keystroke(0, 0, 0, 0, "", true);}
571   \AAformat{AFNumber_Format(0, 0, 0, 0, "", true);}
572   \AAvalidate{AFRange_Validate(true, 0, false, 0);}
573 }%
574 \makebox[\widthof{A:}][1]{A:} \inputFor{math}{class}{A}{.5in}{11bp}
575   \pctThisGrade{math}{class}{A}\vcgbdry[3bp]
576 \makebox[\widthof{A:}][1]{B:} \inputFor{math}{class}{B}{.5in}{11bp}
577   \pctThisGrade{math}{class}{B}\vcgbdry[3bp]
578 \makebox[\widthof{A:}][1]{C:} \inputFor{math}{class}{C}{.5in}{11bp}
579   \pctThisGrade{math}{class}{C}\vcgbdry[3bp]
580 \makebox[\widthof{A:}][1]{D:} \inputFor{math}{class}{D}{.5in}{11bp}
581   \pctThisGrade{math}{class}{D}\vcgbdry[3bp]
582 \makebox[\widthof{A:}][1]{F:} \inputFor{math}{class}{F}{.5in}{11bp}
583   \pctThisGrade{math}{class}{F}\vcgbdry[3bp]
584 \makebox[\widthof{A:}][1]{\phantom{A:}}
585 \textField[\Ff{\FfReadOnly}
586   \AAcalculate{AFSimple_Calculate("SUM",new Array("math"))};}
587 ]{totalClass}{.5in}{13bp}\vcgbdry[4bp]
588 \pushbutton[\TU{This button re-scales the bar graph so that the

```

```

589 tallest bar takes the entire height of the region.}
590 \CA{Optimize}\AAmouseup{optimizeScaling("math");}
591 ]{optimize3}{-}{13bp}
592 \end{minipage}\medskip
593
594 Reset vertical bar graph:
595 \pushButton[\CA{Reset}
596 \AAmouseup{resetBargraphs("math","pctFor","totalClass");}
597 ]{reset}{-}{13bp}
598 \end{document}
599 </adv1>
600 <*comma1>

```

1.6 bgjs-comma1.tex

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

```

601 \documentclass{article}
602 % \usepackage{web}
603 \usepackage{eforms}
604 \hypersetup{pdfpagemode=UseNone} % don't need to see bookmarks
605 \usepackage{bargraph-js}
606 % \previewOn\pmpvOn
607 \makePDasXOn
608
609 \parindent0pt
610 \begin{document}
611
612 This file demonstrates comma-delimited data, manually enter the data
613 in the text field or by press on one of the buttons to give
614 pre-packaged examples.\medskip
615
616 \bgroup\setlength{\fboxrule}{1bp}\fbox
617 {\begin{bargraphenv}[width=(14bp*20),height=2in,%
618 o=vert]{statdemo}%
619 \begin{bargraph}[nbars=20,gap=0,bardimen=14]{histogram}
620 \barfor{bar1}\barfor{bar2}\barfor{bar3}\barfor{bar4}\barfor{bar5}
621 \barfor{bar6}\barfor{bar7}\barfor{bar8}\barfor{bar9}\barfor{bar10}
622 \barfor{bar11}\barfor{bar12}\barfor{bar13}\barfor{bar14}
623 \barfor{bar15}\barfor{bar16}\barfor{bar17}\barfor{bar18}
624 \barfor{bar19}\barfor{bar20}
625 \end{bargraph}%
626 \end{bargraphenv}}\egroup\vcgBdry[6pt]
627
628 \hglue1bp\textField[\TU{Enter up to twenty nonnegative numbers
629 separated by commas}
630 \AAvalidate{resetBargraphs("statdemo");\r
631 \populateCommaData("statdemo","histogram",event.value,%
Require each entry in the array to be non-negative numbers.
632 validateArrayNonNegNums)}

```



```

633 ]{commaed}{(\bardimen*\nbars+2\fbboxsep+2bp)}{13bp}\vcgBdry[6pt]
634
635 \hglue1bp\pushButton[\CA{Symmetrical}\AAmouseup{%
636 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.
637 \populateCommaData("statdemo","histogram",str);
638 }]{symmetrical}{13bp}\cgBdry[.5em]
639 \pushButton[\CA{Skew left}\AAmouseup{%
640 var str="1,2,2,3,3,4,5,6,8,10,12,14,16,19,20,19,17,15,13,11";\r
641 \populateCommaData("statdemo","histogram",str);}
642 ]{skewleft}{13bp}\cgBdry[.5em]
643 \pushButton[\CA{Skew right}\AAmouseup{%
644 var str="17,18,19,20,19,18,16,14,12,10,8,7,7,6,6,4,4,3,2,1";\r
645 \populateCommaData("statdemo","histogram",str);}
646 ]{skewright}{13bp}\vcgBdry[6pt]
647
648 \hglue1bp\pushButton[\TU{This button re-scales the bar graph so
649 that the longest bar takes the entire width of the region.}
650 \CA{Optimize}\AAmouseup{optimizeScaling("statdemo");}
651 ]{optimize1}{13bp}\cgBdry[.5em]
652 \displayfFor{statdemo}{.5in}{13bp}\olBdry
653 \manualsfFor[\CA{Rescale}\TU{Enter a new scale factor in the
654 text field, then press this button}]{statdemo}{13bp}\vcgBdry[6bp]
655
656 \hglue1bp\pushButton[\CA{Reset}
657 \AAmouseup{resetBargraphs("statdemo","commaed","rescale.statdemo")}
658 ]{reset}{13bp}\vcgBdry[6pt]
659 \end{document}
660 </comma1>
661 <*comma2>

```

1.7 bgjs-comma2.tex

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

```

662 \documentclass{article}
663 % \usepackage{web}
664 \usepackage{eforms}
665 \hypersetup{pdfpagemode=UseNone} % don't need to see bookmarks
666 \usepackage{bargraph-js}
667 % \previewOn\pmpvOn
668 \makePDasXOn
669
670 \parindent0pt
671 \begin{document}
672
673 This file demonstrates comma-delimited data, manually enter the

```

```

674 data in the text field or by press on one of the buttons to give
675 pre-packaged examples. More than one \texttt{ bargraph } environment
676 can appear within a \texttt{ bargraphenv }, as illustrated in the next
677 two examples.\medskip
678
679 \fbox{\begin{ bargraphenv }[width=23bp*10,height=1.4in,o=vert]{math1}
680 \presetsbarfor{class1}{A}{\BG{red}
681 \TU{Class1: @v@ students received an 'A'}}
682 \presetsbarfor{class1}{B}{\BG{red}
683 \TU{Class1: @v@ students received an 'B'}}
684 \presetsbarfor{class1}{C}{\BG{red}
685 \TU{Class1: @v@ students received an 'C'}}
686 \presetsbarfor{class1}{D}{\BG{red}
687 \TU{Class1: @v@ students received an 'D'}}
688 \presetsbarfor{class1}{F}{\BG{red}
689 \TU{Class1: @v@ students received an 'F'}}
690 \presetsbarfor{class2}{A}{\BG{blue}
691 \TU{Class2: @v@ students received an 'A'}}
692 \presetsbarfor{class2}{B}{\BG{blue}
693 \TU{Class2: @v@ students received an 'B'}}
694 \presetsbarfor{class2}{C}{\BG{blue}
695 \TU{Class2: @v@ students received an 'C'}}
696 \presetsbarfor{class2}{D}{\BG{blue}
697 \TU{Class2: @v@ students received an 'D'}}
698 \presetsbarfor{class2}{F}{\BG{blue}
699 \TU{Class2: @v@ students received an 'F'}}
700 \begin{ bargraph }[nbars=5,gap=3]{class1}
701 \barfor{A}\barfor{B}\barfor{C}\barfor{D}\barfor{F}
702 \end{ bargraph }
703 \begin{ bargraph }[nbars=5,gap=3]{class2}\cmd{\hs{ bargap }}
704 \barfor{A}\barfor{B}\barfor{C}\barfor{D}\barfor{F}
705 \end{ bargraph }
706 \end{ bargraphenv }\vcgBdry[6pt]
707
708 \begin{ minipage }[c]{222pt}
709 class1:
710 \textField[\TU{Enter five natural numbers separated
711 by commas}
712 \AAvalidate{\populateCommaData("math1","class1",event.value,%
713 validateArrayNonNegNums)}
714 ]{txtclass3}{2in}{13bp}\cgBdry[.5em]
715 \pushButton[\CA{Class 1}\AAmouseup{%
716 var str="12,15,23,10,15";\r
717 this.getField("txtclass3").value=str;\r
718 \populateCommaData("math1","class1",str);
719 }]{math1class1}{13bp}\vcgBdry[4bp]
720 class2:
721 \textField[\TU{Enter five natural numbers separated by commas}
722 \AAvalidate{\populateCommaData("math1","class2",event.value,%
723 validateArrayNonNegNums)}

```

```

724 ]{txtclass4}{2in}{13bp}\cgBdry[.5em]
725 \pushButton[\CA{Class 2}\AAMouseup{%
726   var str="10,17,29,10,20";\r
727   this.getField("txtclass4").value=str;\r
728   \populateCommaData("math1","class2",str);
729 }]{math1class1}{-}{13bp}
730 \end{minipage}\cgBdry[.5em]
731 \pushButton[\TU{This button re-scales the bar graph so that the
732 tallest bar takes the entire height of the region.}\CA{Optimize}
733 \AAMouseup{optimizeScaling("math1");}
734 ]{optimize3}{-}{13bp}\cgBdry[.5em]
735 \pushButton[\CA{Reset}
736 \AAMouseup{resetBargraphs("math1","txtclass3","txtclass4");}
737 ]{reset}{-}{13bp}\vcgBdry[6pt]
738
739
740 You can adjust the positions of the bar graph to have a more
741 side-by-side comparison.\vcgBdry[4bp]
742
743 \fbox{\begin{bargraphenv}[width=33bp*5,height=2in,%
744   o=vert]{math2}
745 \presetsbarfor{class1}{A}{\BG{red}}
746 \presetsbarfor{class1}{B}{\BG{red}}
747 \presetsbarfor{class1}{C}{\BG{red}}
748 \presetsbarfor{class1}{D}{\BG{red}}
749 \presetsbarfor{class1}{F}{\BG{red}}
750 \presetsbarfor{class2}{A}{\BG{blue}}
751 \presetsbarfor{class2}{B}{\BG{blue}}
752 \presetsbarfor{class2}{C}{\BG{blue}}
753 \presetsbarfor{class2}{D}{\BG{blue}}
754 \presetsbarfor{class2}{F}{\BG{blue}}
755 \begin{bargraph}[nbars=5,gap=13]{class1}
756 \barfor{A}\barfor{B}\barfor{C}\barfor{D}\barfor{F}
757 \end{bargraph}%
758 \begin{bargraph}[nbars=5,gap=13]{class2}\cmd{\hs{13bp}}
759 \cmd{\color{red}\hs{-33bp*5+10bp}}
760 \barfor{A}\barfor{B}\barfor{C}\barfor{D}\barfor{F}
761 \end{bargraph}
762 \end{bargraphenv}}\vcgBdry[6pt]
763
764 \begin{defineJS}{\pbaction}
765 var f=this.getField("txtclass5");
766 for (var value="",i=0; i<5; i++)
767   value+=" "+(Math.round(Math.random()*200))+",";
768 f.value=value.substring(0,value.length-1);
769 for (var value="",i=0; i<5; i++)
770   value+=" "+(Math.round(Math.random()*200))+",";
771 var g=this.getField("txtclass6");
772 g.value=value.substring(0,value.length-1);
773 \end{defineJS}

```

```

774
775 \begin{minipage}[c]{222pt}
776 class1:
777 \textField[TU{Enter five natural numbers separated by commas}
778 \AAvalidate{\populateCommaData("math2","class1",event.value,%
779   validateArrayNonNegNums)}
780 ]{txtclass5}{2in}{13bp}\cgBdry[.5em]
781 \pushButton[\CA{Class 1}\AAmouseup{%
782   var str="12,15,23,10,15";\r
783   this.getField("txtclass5").value=str;\r
784   \populateCommaData("math2","class1",str);
785 }]{math1class1}{13bp}\vcgBdry[4bp]
786 class2:
787 \textField[TU{Enter five natural numbers separated by commas}
788 \AAvalidate{\populateCommaData("math2","class2",event.value,%
789   validateArrayNonNegNums)}
790 ]{txtclass6}{2in}{13bp}\cgBdry[.5em]
791 \pushButton[\CA{Class 2}\AAmouseup{%
792   var str="10,17,29,10,20";\r
793   this.getField("txtclass6").value=str;\r
794   \populateCommaData("math2","class2",v);
795 }]{math1class1}{13bp}
796 \end{minipage}\cgBdry[.5em]
797 \pushButton[\CA{Test}\TU{Press to automatically enter data into the
798   two fields above}\AAmouseup{\pbaction}]{tstdata}{13bp}\cgBdry[.5em]
799 \pushButton[\TU{This button re-scales the bar graph so that the
800   tallest bar takes the entire height of the region.}\CA{Optimize}
801   \AAmouseup{optimizeScaling("math2");}]{optimize3}{13bp}\cgBdry[.5em]
802 \pushButton[\CA{Reset}
803   \AAmouseup{resetBargraphs("math2","txtclass5","txtclass6");}
804 ]{reset}{13bp}
805 \end{document}
806 </comma2>
807 <*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.

```

808 \documentclass{article}
809 %%%%%%%%%%%%%%%
810 % Requires a dvips/Distiller workflow %
811 %%%%%%%%%%%%%%%
812 \usepackage[web,eforms={usealtadobe},uselayers]{aeb_pro}
813 % \hypersetup{pdfpagemode=UseNone} % web executes this
814 \usepackage{bargraph-js}
815 \usepackage{graphicx}
816

```

```

817 % \previewOn\pmpvOn
818 \makePDasXOn
819
820 \begin{insDLJS}{lbl}{Labeling function}
821 %% fld="<bgenv-name>@<bg-name>.<bar-name>"
822 %% v=value of field
823 function customLabelsForBars(fld,v){
824 var pos=fld.indexOf(".");
825 var bargraph=fld.substring(0,pos);
826 var bar=fld.substring(pos+1);
827 switch(bargraph) {
828 case "vehiclesH@hBar":
829     switch(bar) {
830     case "auto":
831         return "Automobiles: "+v;
832     case "truck":
833         return "Pickup trucks: "+v;
834     case "suv":
835         return "SUV types: "+v+", costing big \$$\$s";
836     case "van":
837         return "Family vans: "+v
838             +", these cost some serious \u20AC\u20ACs";
839     default:
840         return simpleBarLabels(fld,v);
841     }
842 // other cases can be included
843     default:
844         return simpleBarLabels(fld,v);
845 }
846 }
847 \end{insDLJS}
848 %% applies to all bars with a \TU key
849 %% \barLabelsTU{customBarLabels}
850 %% applies to all other bars
851 \barLabelsNoTU{customLabelsForBars}
852
853 \parindent0pt
854 \begin{document}
855
856 In this example, we label the bar graph using typeset content which
857 is placed in layers (OCGs). The layers are initially hidden, then
858 made visible when data is entered into the input fields.
859 \medskip
860
861 %% These bars have not \TU key, so the are handled by the argument
862 %% of \barLabelsNoTU{customLabelsForBars}.
863 \fbox{\begin{bargraphenv}[width=.67\linewidth,height=2in,
864 o=horiz]{vehiclesH}
865 \presetsbarfor{hBar}{auto}{\BG{red}}
866 \presetsbarfor{hBar}{truck}{\BG{green}}

```

```

867 \presetsbarfor{hBar}{suv}{\BG{yellow}}
868 \presetsbarfor{hBar}{van}{\BG{magenta}}
869 \begin{bargraph}[nbars=4,gap=3]{hBar}
870 \barfor{auto}\cmd{\vs{-3bp}\xBld{hBar.auto}
871 {\small Automobiles (two or four door)}\eBld\vs{3bp}}
872 \barfor{truck}\cmd{\vs{-3bp}\xBld{hBar.truck}
873 {\small Pickup trucks}\eBld\vs{3bp}}
874 \barfor{suv}\cmd{\vs{-3bp}\xBld{hBar.suv}
875 {\small Sport utility vehicle (SUV)}\eBld\vs{3bp}}
876 \barfor{van}\cmd{\vs{0bp}\xBld{hBar.van}
877 {\small Passenger van}\eBld}
878 \end{bargraph}
879 \end{bargraphenv}}\hfill
880 \begin{minipage}[b][2in][c]
881 { .33\linewidth-2\fbboxsep-2\fbboxrule-10pt}\kern0pt\parskip3pt
882 \def\presetinputfor#1#2{%
883 \AAkeystroke{EFNumber_Keystroke(0, 0, 0, 0, "", true);}
884 \AAformat{try{EFNumber_Format(0, 0, 0, 0, "", true)}\r
885 if(event.rc)toggleSetThisLayer("#2", (event.value!=0));}catch(e){}
886 \AAvalidate{EFRRange_Validate(true, 0, false, 0);}
887 }
888 \makebox[\widthof{Truck:}][l]{Auto:}
889 \inputFor{vehiclesH}{hBar}{auto}{.5in}{11bp}\vcgBdry[3bp]
890 \makebox[\widthof{Truck:}][l]{Truck:}
891 \inputFor{vehiclesH}{hBar}{truck}{.5in}{11bp}\vcgBdry[3bp]
892 \makebox[\widthof{Truck:}][l]{SUV:}
893 \inputFor{vehiclesH}{hBar}{suv}{.5in}{11bp}\vcgBdry[3bp]
894 \makebox[\widthof{Truck:}][l]{Van:}
895 \inputFor{vehiclesH}{hBar}{van}{.5in}{11bp}\vcgBdry[3bp]
896 \pushButton[\TU{This button re-scales the bar graph so that the
897 longest bar takes the entire width of the region.}\CA{Optimize}
898 \AAMouseup{optimizeScaling("vehiclesH");}]{optimize1}{13bp}
899 \end{minipage}\medskip
900
901 Reset horizontal bar graph: \pushButton[\CA{Reset}
902 \AAMouseup{resetBargraphs("hBar","vehiclesH");}
903 ]{reset}{13bp}
904
905 \bigskip
906
907 %% These bars have a \TU key
908 \fbbox{\begin{bargraphenv}[width=2in,height=2in,%
909 o=vert]{vehiclesV}\def\WD{2in}
910 \presetsbarfor{vBar}{auto}{\BG{red}
911 \TU{There were @v@ automobiles observed}}
912 \presetsbarfor{vBar}{truck}{\BG{green}
913 \TU{There were @v@ pickups observed during time period}}
914 \presetsbarfor{vBar}{suv}{\BG{yellow}
915 \TU{There were @v@ SUVs observed, costing big
916 \textdollar\textdollar s}}

```

```

917 \presetsbarfor{vBar}{van}{\BG{magenta}
918 \TU{There were @v@ family vans filled with happy people;
919 reminds me of J\{u}rgen's family}}
920 \begin{bargraph}[nbars=4,gap=3]{vBar}
921 \barfor{auto}
922 \cmd{\hs{-3bp}\xBld{vBar.auto}\smash{\raisebox{1pt}
923 {\rotatebox[origin=lb]{90}
924 {\small Automobiles (two or four door)}}}\eBld\hs{3bp}}
925 \barfor{truck}
926 \cmd{\hs{-3bp}\xBld{vBar.truck}\smash{\raisebox{1pt}
927 {\rotatebox[origin=lb]{90}{\small Pickup trucks}}}\eBld\hs{3bp}}
928 \barfor{suv}
929 \cmd{\hs{-3bp}\xBld{vBar.suv}\smash{\raisebox{1pt}
930 {\rotatebox[origin=lb]{90}
931 {\small Sport utility vehicle (SUV)}}}\eBld\hs{3bp}}
932 \barfor{van}
933 \cmd{\hs{0bp}\xBld{vBar.van}\smash{\raisebox{1pt}
934 {\rotatebox[origin=lb]{90}{\small Passenger van}}}\eBld}
935 \end{bargraph}
936 \end{bargraphenv}}\hfill
937 \begin{minipage}[b][2in][c]
938 {\linewidth-2\fbboxsep-2\fbboxrule-2in-10pt}\kern0pt\parskip3pt
939 \def\presetinputfor#1#2{%

```

The arguments for the `\presetinputfor` command are,

```

#1=<bgenv-name>
#2=<bg-name>.<bar-name>

```

```

940 \AAkeystroke{EFNumber_Keystroke(0, 0, 0, 0, "", true);}
941 \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.

```

942 if(event.rc)toggleSetThisLayer("#2",(event.value!=0));}catch(e){}
943 \AAvalidate{EFRRange_Validate(true, 0, false, 0);}
944 }
945 \makebox[\widthof{Truck:}][l]{Auto:}
946 \inputFor{vehiclesV}{vBar}{auto}{.5in}{11bp}\vcgBdry[3bp]
947 \makebox[\widthof{Truck:}][l]{Truck:}
948 \inputFor{vehiclesV}{vBar}{truck}{.5in}{11bp}\vcgBdry[3bp]
949 \makebox[\widthof{Truck:}][l]{SUV:}
950 \inputFor{vehiclesV}{vBar}{suv}{.5in}{11bp}\vcgBdry[3bp]
951 \makebox[\widthof{Truck:}][l]{Van:}
952 \inputFor{vehiclesV}{vBar}{van}{.5in}{11bp}\vcgBdry[4bp]
953 \pushButton[\TU{This button re-scales the bar graph so that the
954 tallest bar takes the entire height of the region.}\CA{Optimize}
955 \AAmouseup{optimizeScaling("vehiclesV");}
956 ]{optimize2}{}{13bp}

```

```

957 \end{minipage}\medskip
958
959 Reset vertical bar graph:
960 \pushButton[\CA{Reset}
961   \AAmouseup{resetBargraphs("vBar","vehiclesV");}
962 ]{reset}{\13bp}\vcgBdry[6bp]
963 Reset all bar graphs:
964 \pushButton[\CA{Reset}
965   \AAmouseup{resetBargraphs("hBar","vBar","vehiclesH","vehiclesV");}
966 ]{reset}{\13bp}
967
968 \end{document}
969 </pro1>
970 <*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.

```

971 \documentclass{article}
972 % \usepackage{web}
973 \usepackage[usealtadobe]{eforms}
974 \hypersetup{pdfpagemode=UseNone} % don't need to see bookmarks
975 \usepackage[dynamic]{bargraph-js}
976 % \previewOn\pmpvOn
977 \makePDasXOn
978

```

`\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.

```

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

```

Adjust the left margin and `\textwidth`

```

981 \def\addWdth{.6in}
982 \addtolength{\oddsidemargin}{\addWdth}
983 \addtolength{\marginparwidth}{0pt}
984 \setlength{\textwidth}{\paperwidth-2in-\oddsidemargin}

```

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

```

985 \begin{insDLJS}{dtable}{Display Table of Probabilities}
986 function displayTable(fld,aDistr) {
987   var tableHead=util.printf("  \s","k ")
988   +util.printf("  \s      ","pmf")
989   +util.printf("  \s      ","cdf")+"\r";
990   var str=tableHead;
991   for (var i=0; i<aDistr.length;i++) {
992     str +=(util.printf("%4d",aDistr[i][0])

```



```

993     + util.printf("\%10.6f",aDistr[i][1])
994     + util.printf("\%10.6f",aDistr[i][2])
995     + "\r"
996   );
997 }
998 var f=this.getField(fld);
999 if (f!=null) f.value=str;
1000 }

```

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.

```

1001 function customDyLabels(pr,v,bPmf) {
1002   return bPmf?("P(Y="+pr+")="+v):("P(Y<="+pr+")="+v);
1003 }
1004 \end{insDLJS}
1005
1006 \parindentOpt
1007 \begin{document}
1008 \begin{center}\bfseries
1009   The bar graph for the probability mass function (pmf)
1010 \end{center}
1011 \noindent\makebox[Opt][r]{\smash
1012   {\raisebox{-4in+2\fbboxsep+2\fbboxrule}{\displayTable}%
1013     \hspace{\marginparsep}}}
1014 \fbox{\begin{bargraphenv}[width=\linewidth-2\fbboxsep-2\fbboxrule,%
1015   height=2in,o=vert]{Pmf}
1016 \begin{bargraph}{pmfBar}\isdynamic\end{bargraph}
1017 \end{bargraphenv}}\vcgBdry[\medskipamount]
1018 \pushButton[\TU{This button re-scales the bar graph so that the
1019   tallest bar takes the entire height of the region; shift-click
1020   reverts bar graph to its original scaling.}\CA{Optimize}\AAmouseup{%
1021   try{\displayDyBargraph("Pmf",aPmfCdf,true,!event.shift)}
1022     catch(e){};}
1023 ]{optimize}{-}{13bp}\cgBdry[1em]
1024 Under normal scaling, the height of this region is 1 unit, when the
1025 bar graph is optimized, the height is the height of the tallest bar.
1026
1027 \medskip
1028 \begin{center}\bfseries
1029   The bar graph for the cumulative distribution function (cdf)
1030 \end{center}
1031 \fbox{\begin{bargraphenv}[width=\linewidth-2\fbboxsep-2\fbboxrule,%
1032   height=2in,o=vert]{Cdf}
1033 \begin{bargraph}{cdfBar}\isdynamic\end{bargraph}
1034 \end{bargraphenv}}\vcgBdry[\medskipamount]
1035 The height of the region above is 1~unit.
1036

```

```

1037 \begin{flushleft}
1038 \textbf{Generate some probability distributions}\medskip
1039
1040 \pushButton[\CA{Dist1}\AAmouseup{%
    Here we hard-wire the aPmfCdf matrix
1041   var aPmfCdf=new Array(
1042     [1,.2,.2],
1043     [2,.1,.3],
1044     [3,.2,.5],
1045     [4,.2,.7],
1046     [5,.1,.8],
1047     [6,.2,1]
1048   );\r
1049   displayTable("displayTable",aPmfCdf);\r

```

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

```

1050   displayDyBargraph("Pmf",aPmfCdf,true,false,%
1051 {bc:color.blue,fc:color.red,lbl:customDyLabels});\r
1052   displayDyBargraph("Cdf",aPmfCdf,false,false,%
1053 {bc:color.blue,fc:color.red,lbl:customDyLabels});
1054 }]{Distr1}{-}{13bp}\cgBdry[.5em]
1055 \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.

```

1056   var aValues=[-2,-1,0,1,2,3,4];\r
1057   var apmfs=[2/20,3/20,6/20,1/20,2/20,6/20,2/20];\r
1058   var acdfs=new Array();\r
1059   var aPmfCdf=new Array();\r
1060   acdfs[0]=apmfs[0];\r
1061   aPmfCdf[0]=[aValues[0],apmfs[0],acdfs[0]];\r
1062   var l=aValues.length-1;\r
1063   for (var i=1; i<l; i++) {\r\t
1064     acdfs[i]=apmfs[i]+acdfs[i-1];\r\t
1065     aPmfCdf[i]=[aValues[i],apmfs[i],acdfs[i]]\r
1066   }\r
1067   displayTable("displayTable",aPmfCdf);\r
1068   displayDyBargraph("Pmf",aPmfCdf,true,false,%
1069 {bc:color.blue,fc:color.red});\r
1070   displayDyBargraph("Cdf",aPmfCdf,false,false,%
1071 {bc:color.blue,fc:color.red});
1072 }]{Distr2}{-}{13bp}\cgBdry[.5em]
1073 \pushButton[\TU{Randomly generate a probability distribution}
1074   \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`.

```

1075   var maxN=40;\r

```

```

1076 var aPmfCdf=new Array();\r
1077 var n=Math.round(Math.random()*maxN);\r
1078 var aValues=[],apmfs[],acdfs=[];\r
1079 var total=0;\r
1080 for (var i=0; i<n; i++) {\r\t
1081     aValues[i]=i;\r\t
1082     apmfs[i]=Math.round(Math.random()*maxN);\r\t
1083     total+=(apmfs[i]);\r
1084 }\r
1085 for (var i=0; i<n; i++) {\r\t
1086     apmfs[i]=apmfs[i]/total;\r\t
1087     acdfs[i]=apmfs[i]+((i==0)?0:acdfs[i-1]);\r\t
1088     aPmfCdf[i]=[aValues[i],apmfs[i],acdfs[i]];\r
1089 }\r
1090 displayTable("displayTable",aPmfCdf);\r
1091 displayDyBargraph("Pmf",aPmfCdf,true,false,
1092 {bc:color.blue,fc:color.red});\r
1093 displayDyBargraph("Cdf",aPmfCdf,false,false,
1094 {bc:color.blue,fc:color.red});
1095 }]{Distr3}{-}{13bp}\cgBdry[.5em]
1096 \pushButton[\CA{Reset}
1097     \TU{Press to clear the fields of this page, and shift-press to
1098     clear all fields.}
1099     \AAmouseup{%
1100     this.calculate=true;\r
1101     if (event.shift)\r\t
1102     this.resetForm();\r
1103     else {\r\t
1104     this.removeField("Pmf@pmfBar");\r\t
1105     this.removeField("Cdf@cdfBar");\r\t
1106     this.resetForm("displayTable");\r
1107     }
1108 }]{reset}{-}{13bp}\cgBdry[2em]
1109 \pushButton[\CA{Toggle Border}\AAmouseup{%
1110 var f=this.getField("Pmf@pmfBar");\r
1111 if ( f!=null ) {\r\t
1112     var a=f.getArray();\r\t\t
1113     if (f.saveStrokeColor==undefined) f.saveStrokeColor=color.red;\r\t
1114     if (!color.equal(a[0].strokeColor,color.transparent))\r\t\t
1115     f.saveStrokeColor=a[0].strokeColor;\r\t
1116     var g=getField("Cdf@cdfBar");\r\t
1117     if (color.equal(a[0].strokeColor,color.transparent)) {\r\t\t\t
1118     f.strokeColor=f.saveStrokeColor;\r\t\t
1119     g.strokeColor=f.saveStrokeColor;\r\t
1120     } else {\r\t\t\t
1121     f.strokeColor=color.transparent;\r\t\t
1122     g.strokeColor=color.transparent;\r\t
1123     }\r
1124 }
1125 }]{toggleBdry}{-}{13bp}

```

```
1126 \end{flushleft}
1127 \end{document}
1128 </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`.