



United States Steel Corporation

PROC GREPLAY With Templates

September 26, 2008

Barry Hong
Process Analysis and Simulation
U. S. Steel Canada

© 2008 United States Steel Corporation

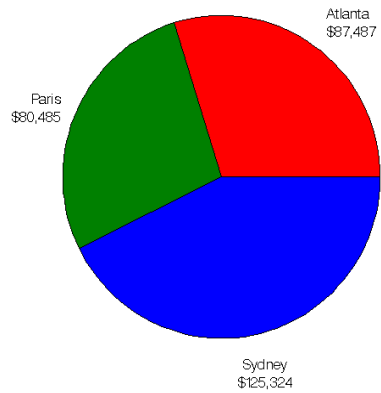




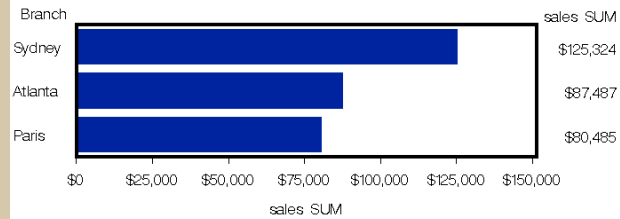
PROC GREPLAY With Templates

Create a lot of graphs?

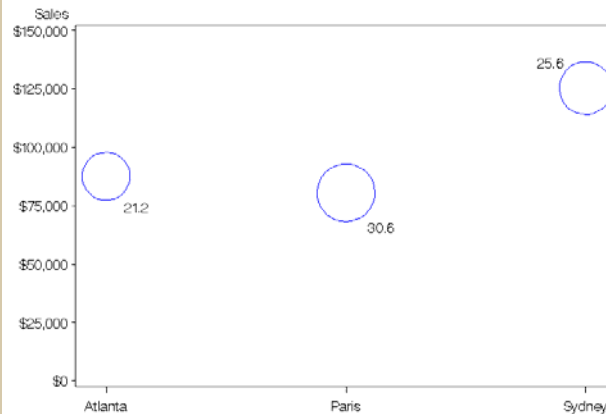
Total Sales



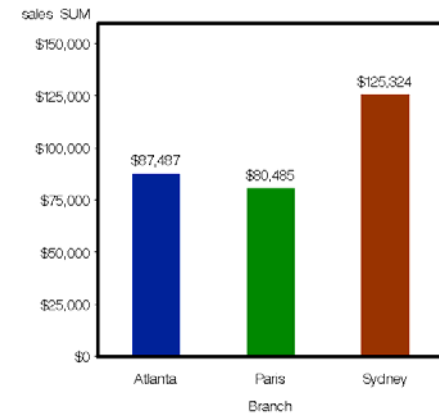
Total Sales



Sales and Margins by Site



Total Sales

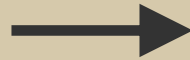




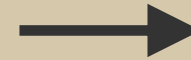
PROC GREPLAY

Concept

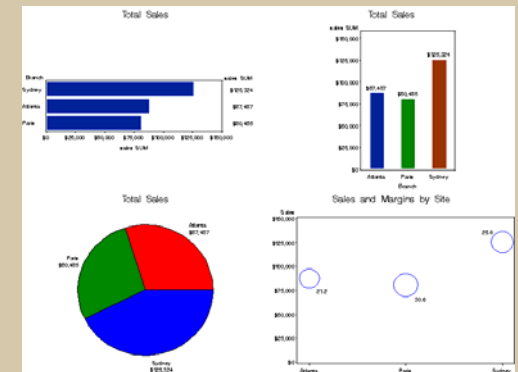
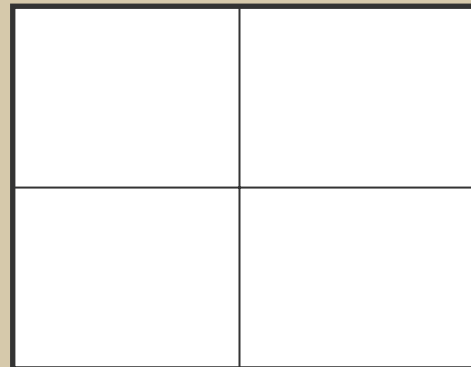
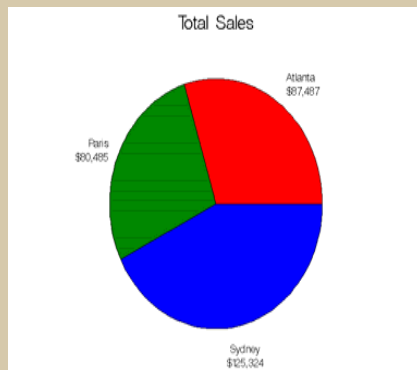
Create graphs
in a catalog
Eg. work.charts



Replay graphs
through a
template



VOILA!
Four charts on
one page





Creating Graphics

Default graphics catalog is work.gseg

Default name is based on the procedure

In SAS 9.1 long name sequence is template
templat1
templat2 ...
last character dropped

Basics

Graphic procedures have the option GOUT=lib.cat to store graphic images

Procedure option NAME='gname' permits you to name the graphic segment for future reference. Highly recommended

In SAS 9.1 names are limited to 8 characters. If the name exists SAS will add numbers to the end of the name.
In SAS 9.2 names can be up to 256 characters long



Creating Graphics

Output graphics catalog is work.charts

Graphic entry name is hbar.

Default name would have been gchart.

Example

```
proc gchart data=site_sales gout=work.charts;  
  format sales dollar8.;  
  hbar site /frame  
      sumvar=sales  
      maxis=axis1  
      raxis=0 to 150000 by 25000  
      minor=0  
      descending  
      name='hbar';  
run; quit;
```



Graphics Template

Page Coordinates

Page is defined using an (X,Y) coordinate system with values in percent of the page.

Coordinates are the same for both portrait and landscape orientation.

(0,100)
(ULX,ULY)

(100,100)
(URX,URY)

(0,0)
(LLX,LLY)

(100,0)
(LRX,LRY)

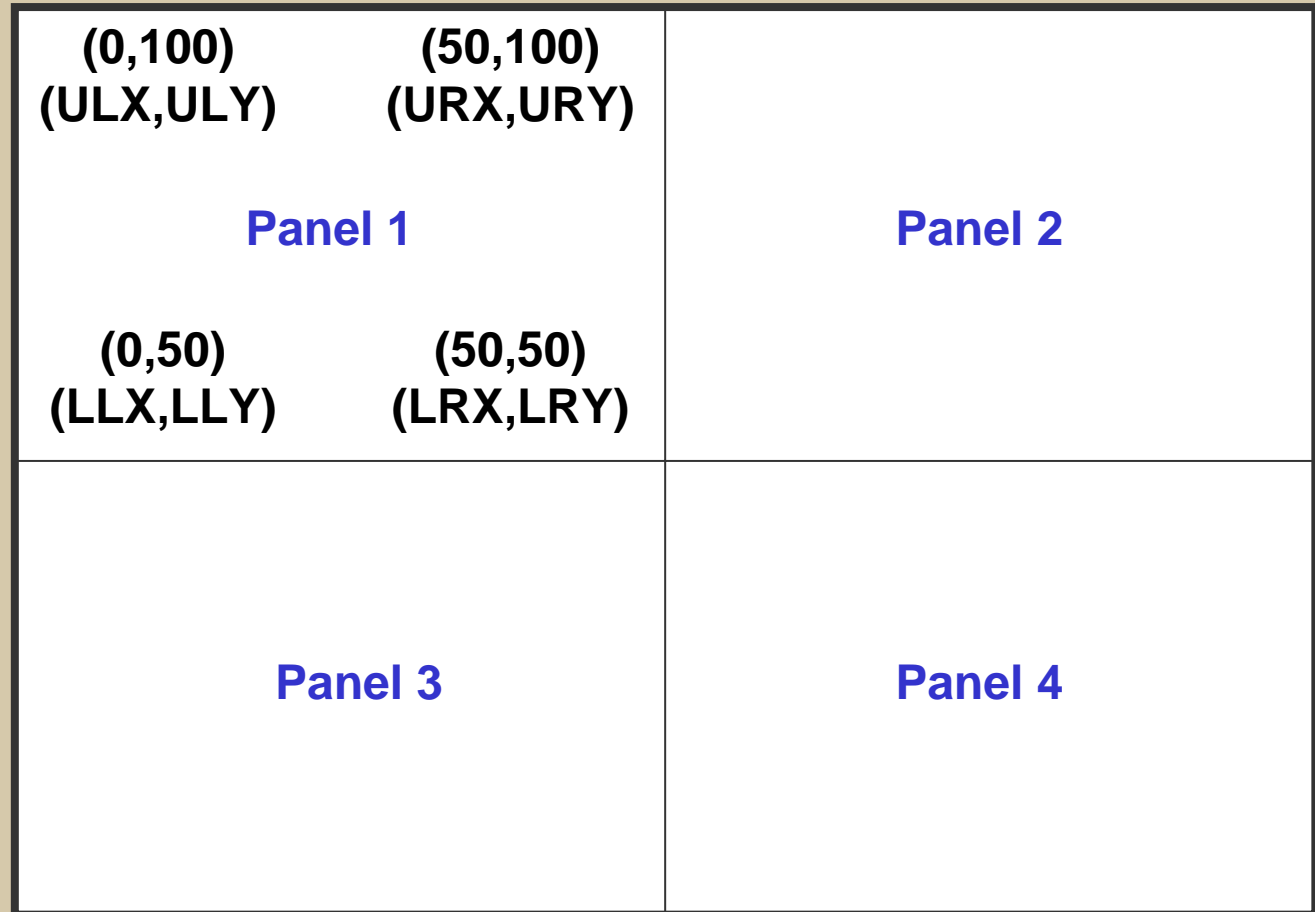


Graphics Template

Layout your plan on a sheet of paper and determine the four corner coordinates for each panel. Panels can be numbered in any order.

There is no limit to the number of panels you can create

Preparation





Graphics Template

Code terms:

- tc= the template catalog. Save in permanent catalog for reuse

SAS Code to Create a Template

```
PROC GREPLAY tc=data.template  
nofs;
```

```
tdef four des='Four panels (2x2)'
```

```
/* define panel 1 */
```

```
1/   llx=0   lly=50  
      ulx=0   uly=100  
      urx=50  ury=100  
      lrx=50  lry=50
```

```
/* define panel 2 */
```

```
2/   llx=50   lly=50  
      ulx=50   uly=100  
      urx=100  ury=100  
      lrx=100  lry=50
```

```
/* define panel 3 */
```

```
3/   llx=0   lly=0  
      ulx=0   uly=50  
      urx=50  ury=50  
      lrx=50  lry=0
```

```
/* define panel 4 */
```

```
4/   llx=50   lly=0  
      ulx=50   uly=50  
      urx=100  ury=50  
      lrx=100  lry=0;
```

```
Run; quit;
```



Graphics Template

Code terms:

- tc= the template catalog. Save in permanent catalog for reuse

- nofs = no full screen editing turns off the interactive window. Templates can be created in the interactive window.

SAS Code to Create a Template

```
PROC GREPLAY tc=data.template  
nofs;
```

```
tdef four des='Four panels (2x2)'
```

```
/* define panel 1 */
```

```
1/   llx=0   lly=50  
      ulx=0   uly=100  
      urx=50  ury=100  
      lrx=50  lry=50
```

```
/* define panel 2 */
```

```
2/   llx=50   lly=50  
      ulx=50   uly=100  
      urx=100  ury=100  
      lrx=100  lry=50
```

```
/* define panel 3 */
```

```
3/   llx=0   lly=0  
      ulx=0   uly=50  
      urx=50  ury=50  
      lrx=50  lry=0
```

```
/* define panel 4 */
```

```
4/   llx=50   lly=0  
      ulx=50   uly=50  
      urx=100  ury=50  
      lrx=100  lry=0;
```

```
Run; quit;
```



Graphics Template

Code terms:

- tc= the template catalog. Save in permanent catalog for reuse
- nofs = no full screen editing
- tdef defines the template, name is "four" Each panel requires eight values. Description is optional.

SAS Code to Create a Template

```
PROC GREPLAY tc=data.template  
nofs;
```

```
tdef four des='Four panels (2x2)'
```

```
/* define panel 1 */
```

```
1/ llx=0 lly=50  
ulx=0 uly=100  
urx=50 ury=100  
lrx=50 lry=50
```

```
/* define panel 2 */
```

```
2/ llx=50 lly=50  
ulx=50 uly=100  
urx=100 ury=100  
lrx=100 lry=50
```

```
/* define panel 3 */
```

```
3/ llx=0 lly=0  
ulx=0 uly=50  
urx=50 ury=50  
lrx=50 lry=0
```

```
/* define panel 4 */
```

```
4/ llx=50 lly=0  
ulx=50 uly=50  
urx=100 ury=50  
lrx=100 lry=0;
```

```
Run; quit;
```



Graphics Template

Code terms:

- tc= the template catalog. Save in permanent catalog for reuse
- nofs = no full screen editing
- tdef defines the template, name is "four"
- Only one semicolon after the last panel definition

SAS Code to Create a Template

```
PROC GREPLAY tc=data.template  
nofs;
```

```
tdef four des='Four panels (2x2)'
```

```
  /* define panel 1 */
```

```
    1/   llx=0   lly=50  
        ulx=0   uly=100  
        urx=50  ury=100  
        lrx=50  lry=50
```

```
  /* define panel 2 */
```

```
    2/   llx=50   lly=50  
        ulx=50   uly=100  
        urx=100  ury=100  
        lrx=100  lry=50
```

```
  /* define panel 3 */
```

```
    3/   llx=0   lly=0  
        ulx=0   uly=50  
        urx=50  ury=50  
        lrx=50  lry=0
```

```
  /* define panel 4 */
```

```
    4/   llx=50   lly=0  
        ulx=50   uly=50  
        urx=100  ury=50  
        lrx=100  lry=0 ;
```

```
Run; quit;
```



PROC GREPLAY

Code terms:

- `igout=` input graphics catalog

Replay graphs through the template

PROC GREPLAY

`igout=work.charts`

`tc=data.template`

`template=four`

`gout=work.composite`

`nofs;`

`treplay 1:hbar 2: vbar 3:pie 4:bubble`

`name='fourplot'`

`des='Four Sales Plots';`

`run; quit;`



PROC GREPLAY

Code terms:

- igout= input graphics catalog
- tc= the template catalog

Replay graphs through the template

PROC GREPLAY

```
igout=work.charts
```

```
tc=data.template
```

```
template=four
```

```
gout=work.composite
```

```
nofs;
```

```
treplay 1:hbar 2: vbar 3:pie 4:bubble
```

```
name='fourplot'
```

```
des='Four Sales Plots';
```

```
run; quit;
```



PROC GREPLAY

Code terms:

- igout= input graphics catalog
- tc= the template catalog
- **template=** template name in the template catalog

Replay graphs through the template

PROC GREPLAY

igout=work.charts

tc=data.template

template=four

gout=work.composite

nofs;

treplay 1:hbar 2: vbar 3:pie 4:bubble

name='fourplot'

des='Four Sales Plots';

run; quit;



PROC GREPLAY

Code terms:

- igout= input graphics catalog
- tc= the template catalog
- template= template name
- gout= output graphics catalog, default work.gseg

Replay graphs through the template

PROC GREPLAY

igout=work.charts

tc=data.template

template=four

gout=work.composite

nofs;

treplay 1:hbar 2: vbar 3:pie 4:bubble

name='fourplot'

des='Four Sales Plots';

run; quit;



PROC GREPLAY

Code terms:

- `treplay` links the template panels with the graphs

Replay graphs through the template

PROC GREPLAY

```
igout=work.charts
```

```
tc=data.template
```

```
template=four
```

```
gout=work.composite
```

```
nofs;
```

```
treplay 1:hbar 2:vbar 3:pie 4:bubble
```

```
name='fourplot'
```

```
des='Four Sales Plots';
```

```
run; quit;
```



PROC GREPLAY

Code terms:

- treplay links the template panels with the graphs
- `name=` option to name the composite graph, default is `template`

Replay graphs through the template

PROC GREPLAY

```
igout=work.charts
```

```
tc=data.template
```

```
template=four
```

```
gout=work.composite
```

```
nofs;
```

```
treplay 1:hbar 2: vbar 3:pie 4:bubble
```

```
name='fourplot'
```

```
des='Four Sales Plots';
```

```
run; quit;
```



PROC GREPLAY

Replay graphs through the template

PROC GREPLAY

```
igout=work.charts
```

```
tc=data.template
```

```
template=four
```

```
gout=work.composite
```

```
nofs;
```

```
treplay 1:hbar 2:vbar 3:pie 4:bubble
```

```
name='fourplot'
```

```
des='Four Sales Plots';
```

```
run; quit;
```

Code terms:

- treplay links the template panels with the graphs

- name= option to name the composite graph, default is template

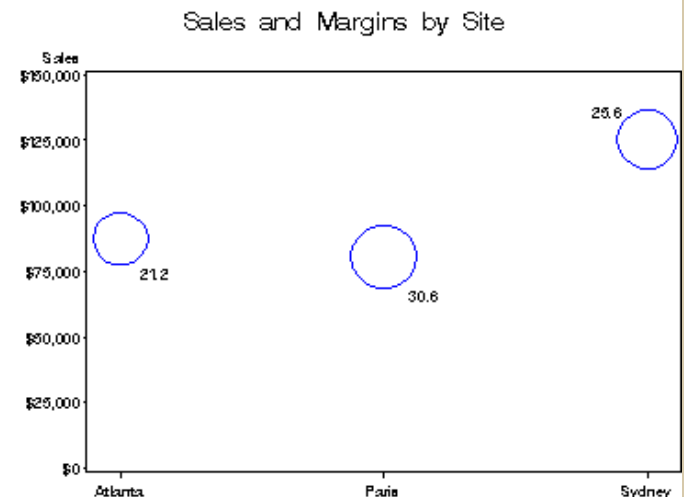
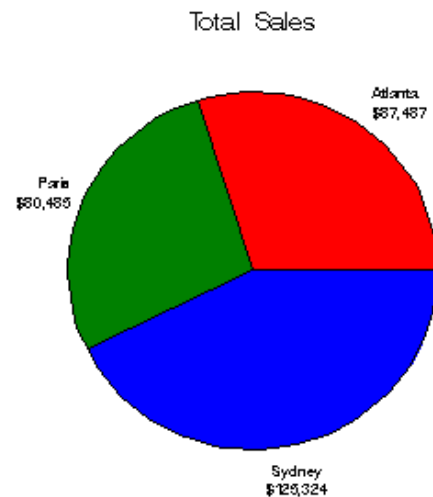
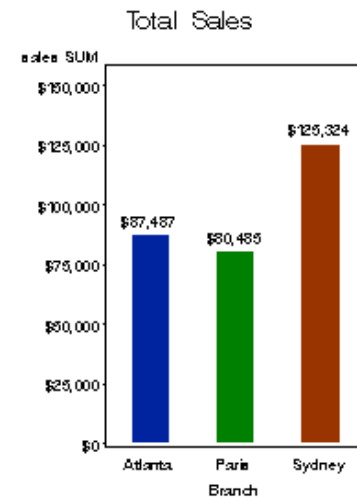
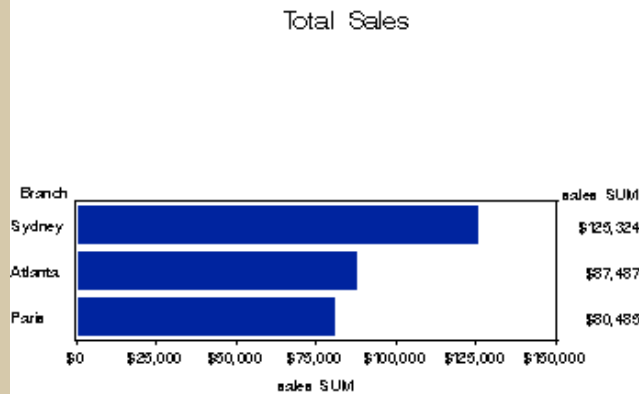
- des= optional description



PROC GREPLAY With Templates

work.composite.fourplot.gseg

Panel coordinates can be set to create borders and white space between charts



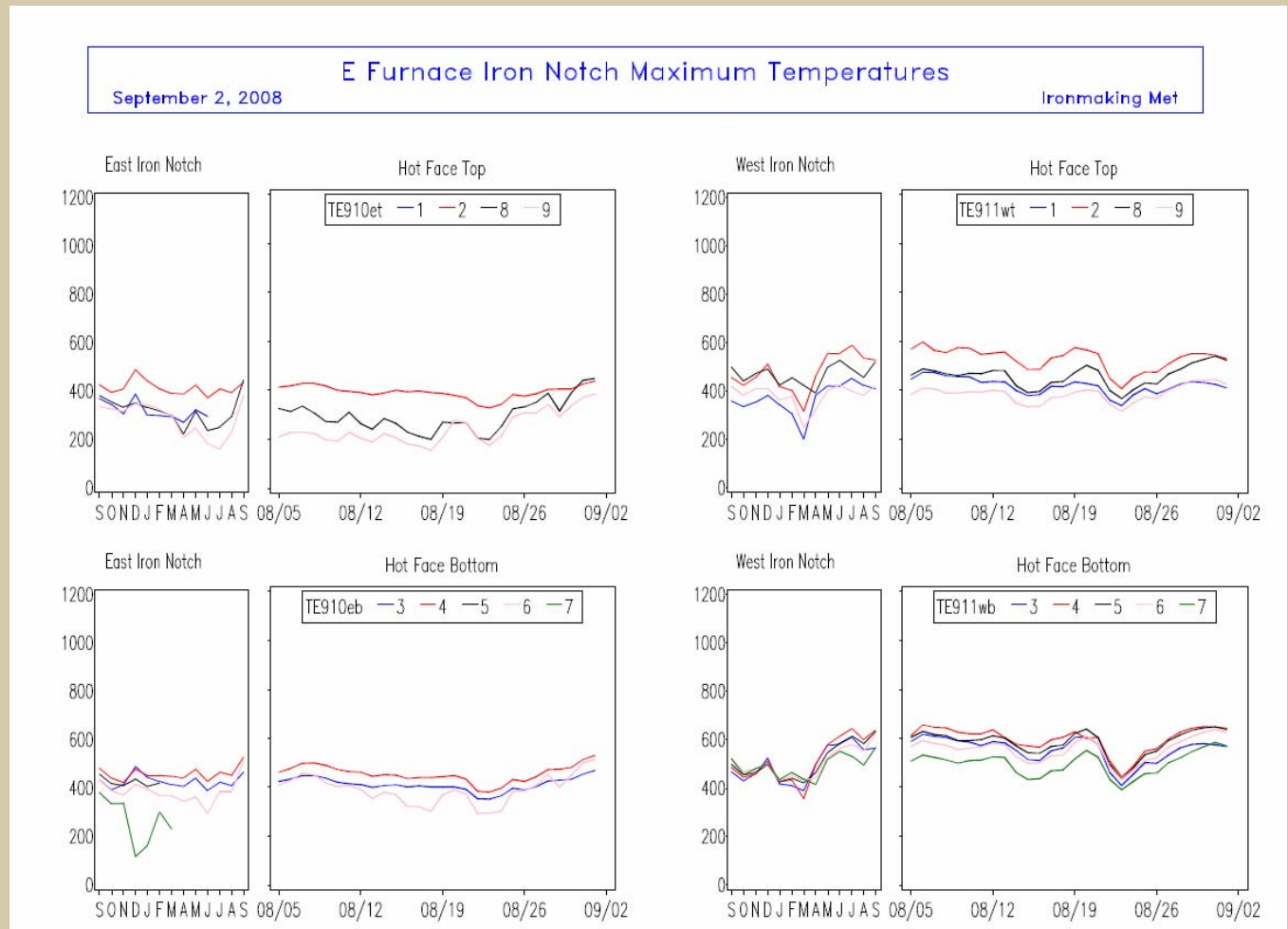


PROC GREPLAY With Templates

Composite page with a banner

Panels do not have to be the same size.

Panels can overlap, the banner panel covers the whole page.

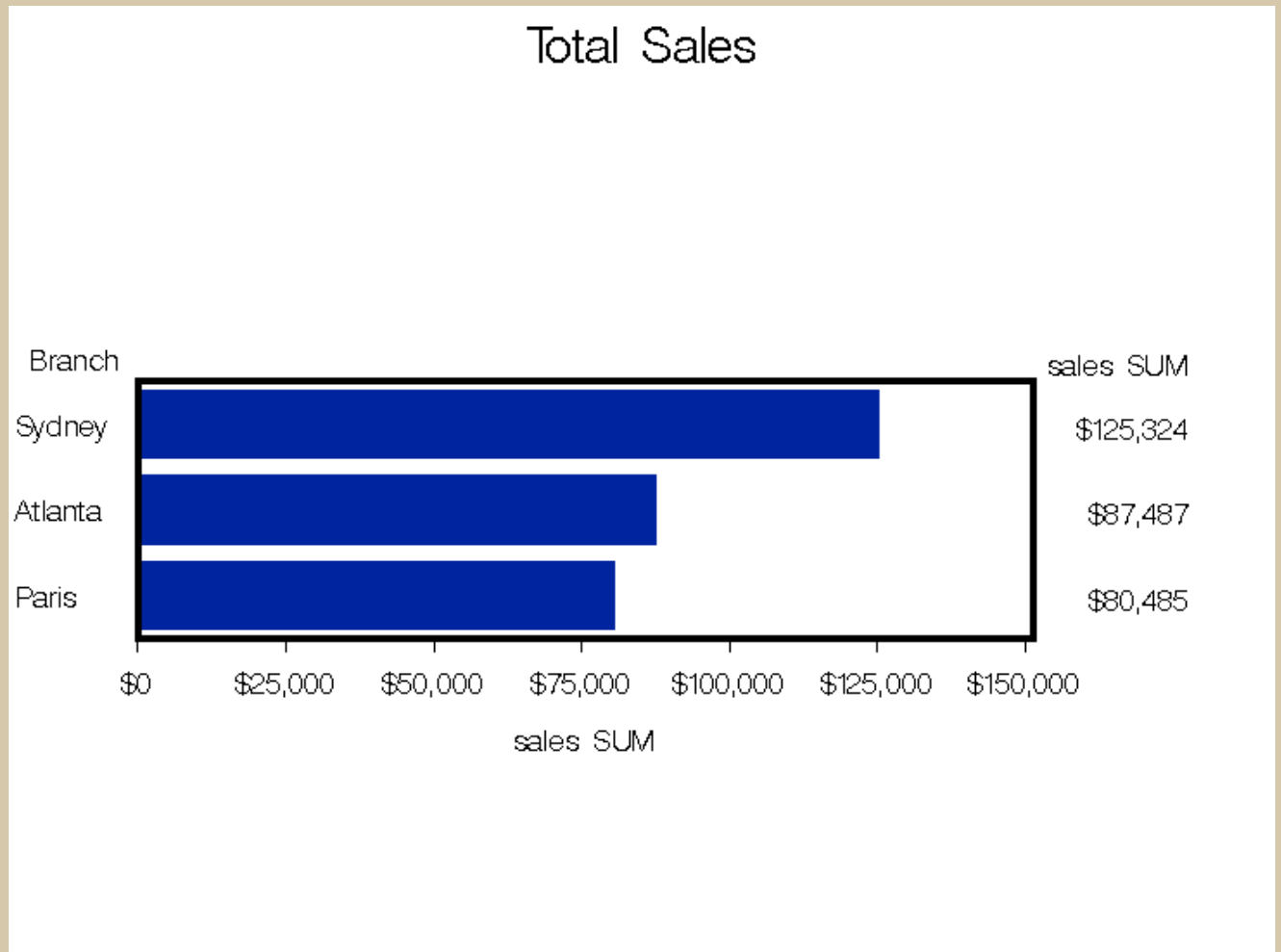




PROC GREPLAY With Templates

Pay attention to panel coordinates

Mistakes in defining a panel can result in funny looking charts.

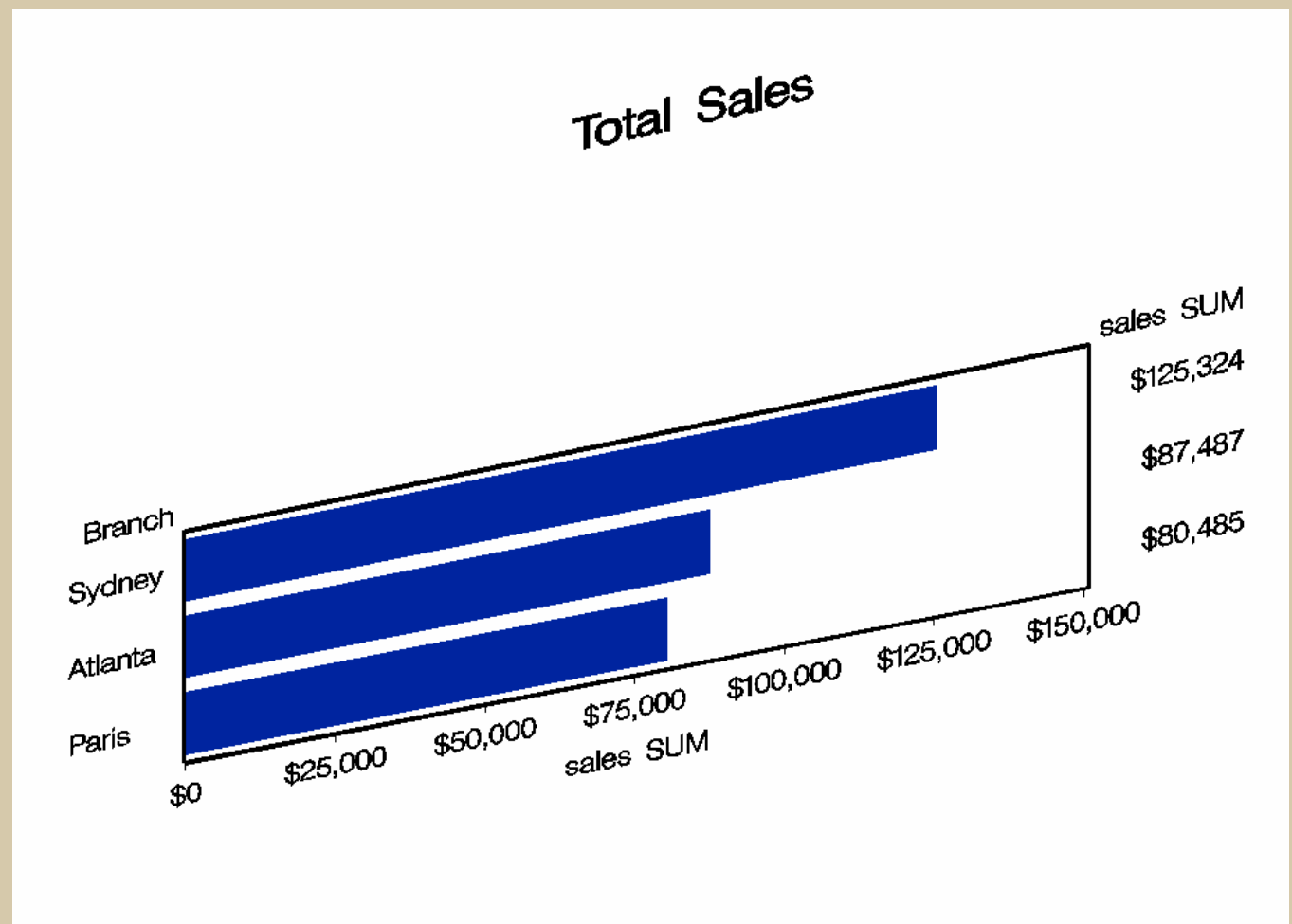




PROC GREPLAY With Templates

Panel coordinate errors

Panel corners did not define a rectangle

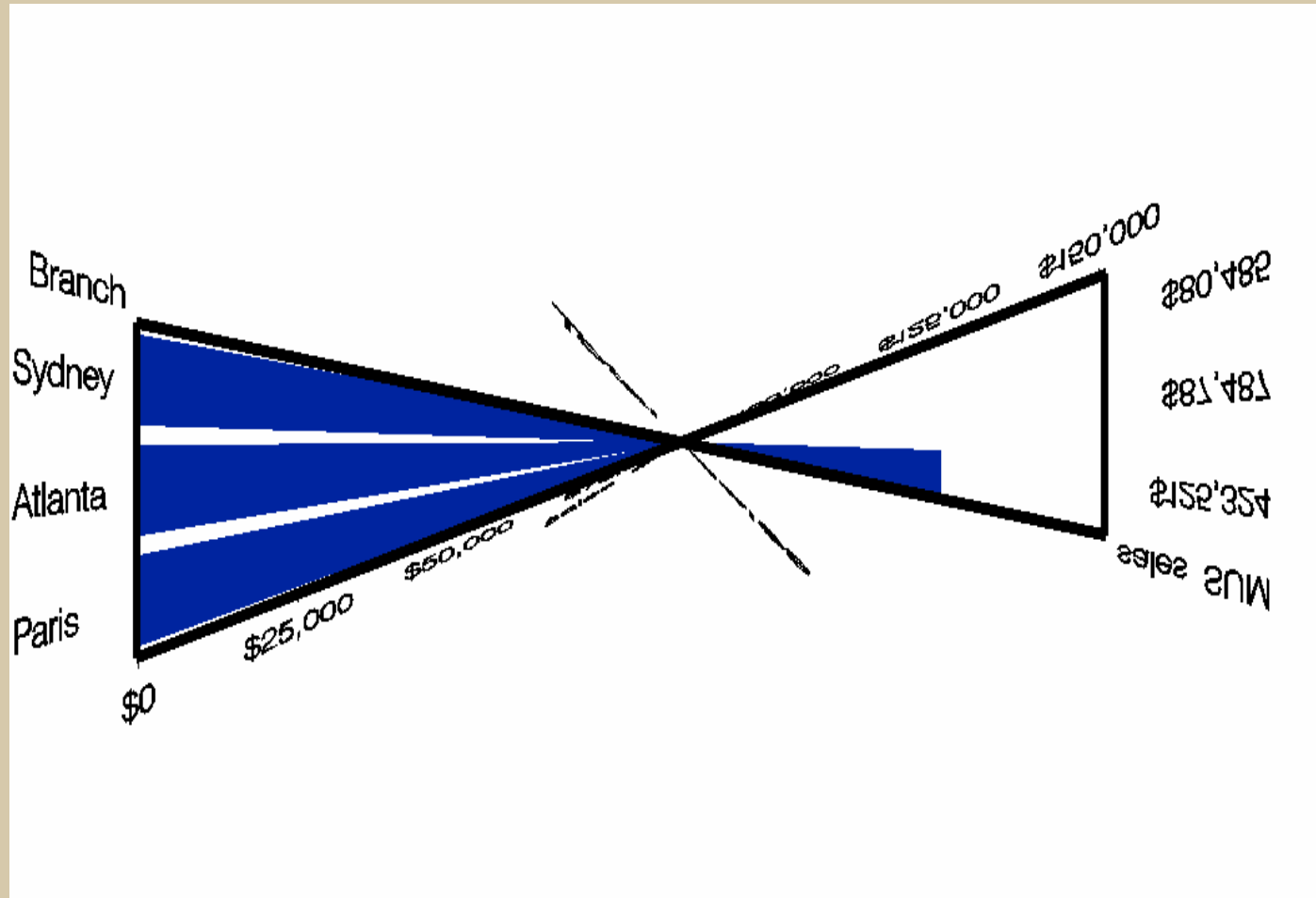




PROC GREPLAY With Templates

Panel coordinate errors

My favourite:
the
dimensional
twist. Right
side
coordinates
transposed





Summary

Using PROC
GREPLAY
With
Templates
won't get
you this
yacht but it
will make
you look like
a big shot to
your
information
consumers.

