

NEW FEATURES IN FOUNDATION SAS 9.4

CYNTHIA JOHNSON
CUSTOMER LOYALTY



THE POWER TO KNOW.

Agenda

- Base SAS
- SAS/ACCESS Interface to PC Files
- SAS Support for Hadoop
- SAS/GRAPH
- SAS Studio

BASE SAS 9.4



BASE SAS PROGRAMMING LANGUAGES

Two new programming languages:

- DS2
- FedSQL

BASE SAS NEW PROGRAMMING LANGUAGE

DS2 is a new SAS proprietary programming language. It

- Enables DS2 language statements from Base SAS
- Includes additional data types, ANSI SQL types, programming structure elements, and user-defined methods and packages.
- Allows embedded FedSQL in some statements
- Runs anywhere – Base, In-Database (via In-Database Code Accelerator), High Performance Analytics (via HPDS2)

DS2 WHY USE IT?

DS2 is beneficial in applications that

- require the precision that results from using the new supported data types
- benefit from using new expressions or write methods or packages
- need to execute SAS FedSQL from within DS2
- execute outside a SAS session, for example on High-Performance Analytics Server or the SAS Federation Server
- take advantage of threaded processing in products such as the SAS In-Database Code Accelerator, SAS High-Performance Analytics Server, and SAS Enterprise Miner

DS2 NEW PROCEDURE

The DS2 Procedure

- The DS2 procedure enables you to submit DS2 language statements from a Base SAS session.

DS2 SAMPLE SYNTAX

Similar to the Data step, but different.....

```
proc ds2;  
data _null_;  
    method init();  
        dcl varchar(16) str;  
        str = 'Hello World!';  
        put str;  
    end;  
enddata;  
run;  
quit;
```


BASIC DS2 SYNTAX

```
PROC DS2;  
data _null_;  
  method init();  
    dcl varchar(20) foo;  
    foo = '**> Starting';  
    put foo;  
  end;  
  method run();  
    set ds2_sas.banks;  
    put _all_;  
  end;  
  method term();  
    dcl char(11) bar;  
    bar = '**> I quit!';  
    put bar;  
  end;  
run; quit;
```

Initial processing

Execution loop

Final processing

BASE SAS PROGRAMMING LANGUAGES

FedSQL is a SAS proprietary implementation of the ANSI SQL:1999 core standard. It provides

- A scalable, threaded, high-performance way to access, manage, and share relational data in multiple data sources
- A common SQL syntax across all data sources
- Support for new data types

The FEDSQL Procedure

- The FEDSQL procedure enables you to submit FedSQL language statements from a Base SAS session. The FedSQL language is the SAS implementation of ANSI SQL:1999 core standard.

FEDSQL AND DS2 EXAMPLE

You can use an embedded FedSQL query to generate data within a DS2 method block.

```
method run();  
  set {select * from work.titles natural join work.price  
      order by publisher};  
  by publisher;  
  ... DS2 statements ...  
end;
```

```
proc ds2;
data sales (overwrite=YES);
  keep Customer_ID Total;
method run();
  set {select c.Customer_ID
        ,Total_Retail_Price
      from ds2_sas.order_fact f
        full join
        ds2_sas.customer_dim c
      on f.Customer_ID=c.Customer_ID
      order by 1};
  by customer_id;
  if first.customer_id then Total=0;
  Total+total_retail_price;
  if last.customer_id then output;
end; enddata;
run; quit;
```

EMBEDDED SQL IN DS2

Returns SQL
result set as input
stream

BY group
processing on
results

OUTPUT OUTPUT DELIVERY SYSTEM

Enhancements to the Output Delivery System enable you to

- Create EPUB, HTML5, Microsoft PowerPoint and Excel files.
- Use the ODS Report Writing Interface (RWI) to create and manipulate predefined ODS objects in a DATA step to create highly customized output.
- Arrange ODS output objects exactly where you want them on a page, or use dynamic placement of objects by using a grid structure. (ODS Layout)
- Animate multi-page GIF images and SVG files by setting system options.

SAMPLE CODE

The ODS PowerPoint destination allows you to send SAS output directly to PowerPoint.

Like other ODS destinations, simply specify the POWERPOINT keyword on the ODS statement and use FILE= to name the resulting file.

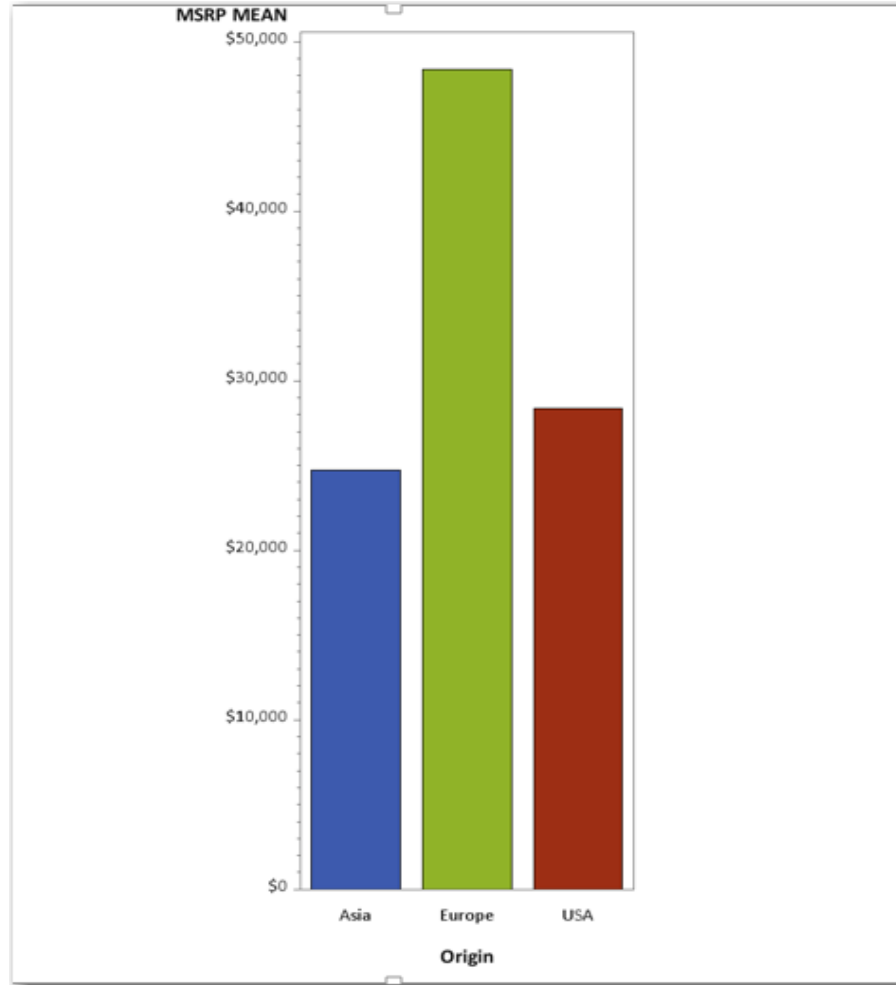
```
ODS POWERPOINT FILE="example.pptx";  
  <SAS code>  
ODS POWERPOINT CLOSE;
```

ODS POWERPOINT SAMPLE CODE

```
ods powerpoint file='c:\test.pptx';
```

```
proc gchart data=sashelp.cars;  
    vbar origin / sumvar=msrp type=mean patternid=midpoint;  
run; quit;
```

```
ods powerpoint close;
```

PowerPoint Using Template Layout Two Content with ODS LIST/GMAP

- Pre-defined template
- Side-by-side output
- Use:
 - ✓ Tables
 - ✓ Graphs
 - ✓ Lists
 - ✓ Text



EXAMPLE ODS POWERPOINT

```
ods powerpoint file="Layout2List.ppt" layout=twocontent nogtitle nogfootnote  
style=powerpointlight;
```

```
proc odslist;
```

```
  item 'Pre-defined template';
```

```
  item 'Side-by-side output';
```

```
  item;
```

```
    p 'Use:';
```

```
    list / style=[bullet=check];
```

```
      item 'Tables';
```

```
      item 'Graphs';
```

```
      item 'Lists';
```

```
      item 'Text';
```

```
    end;
```

```
run;
```

```
goptions hsize=4.5in vsize=4.5in;
```

```
proc gmap map=maps.us data=maps.us all;
```

```
  id state;
```

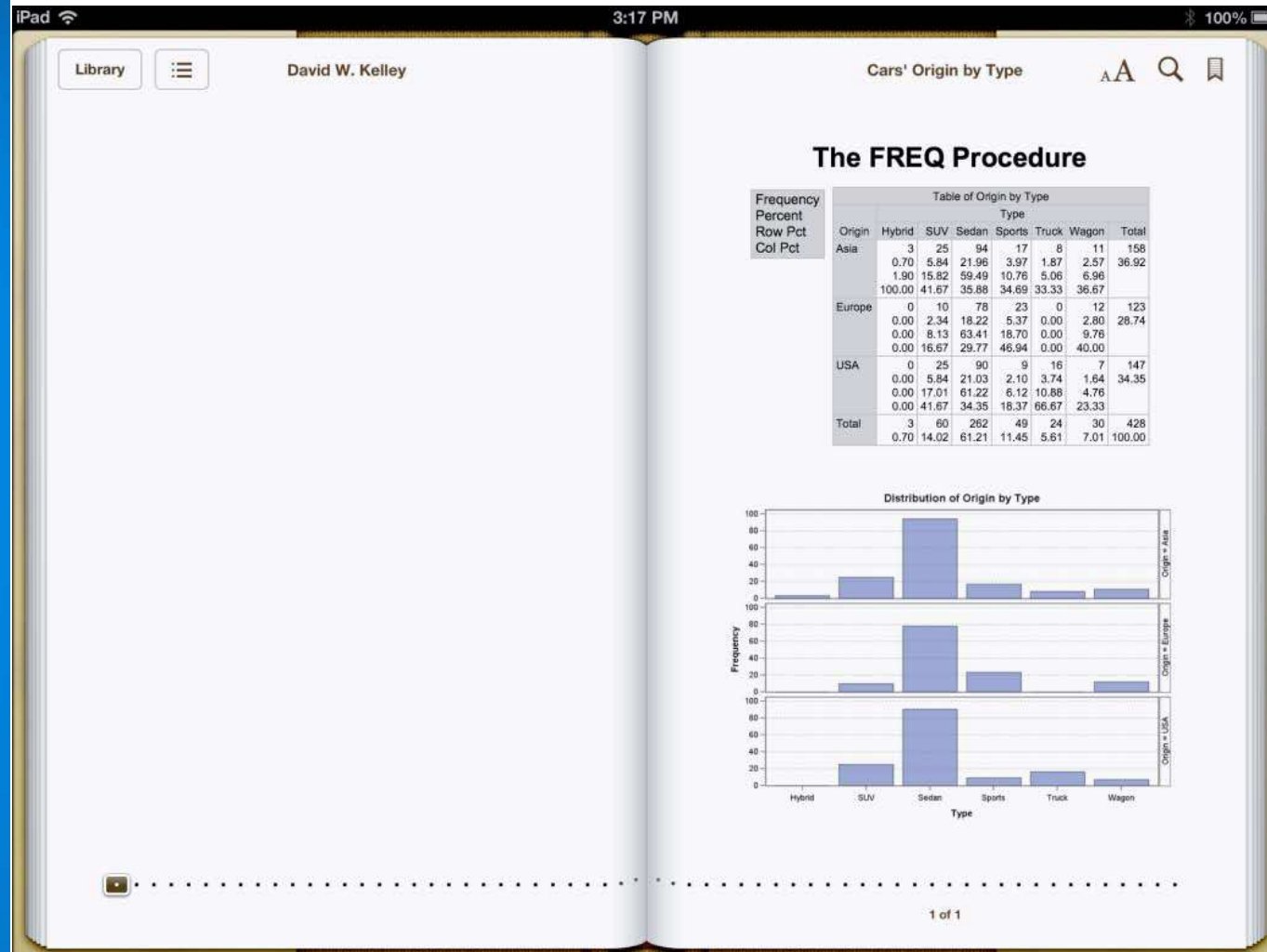
```
  choro statecode/statistic=frequency discrete nolegend;
```

```
run; quit;
```

```
ods _all_ close;
```

SAS ODS EPUB

Output SAS Reports to eBook format to be viewed on iPad & iPhone



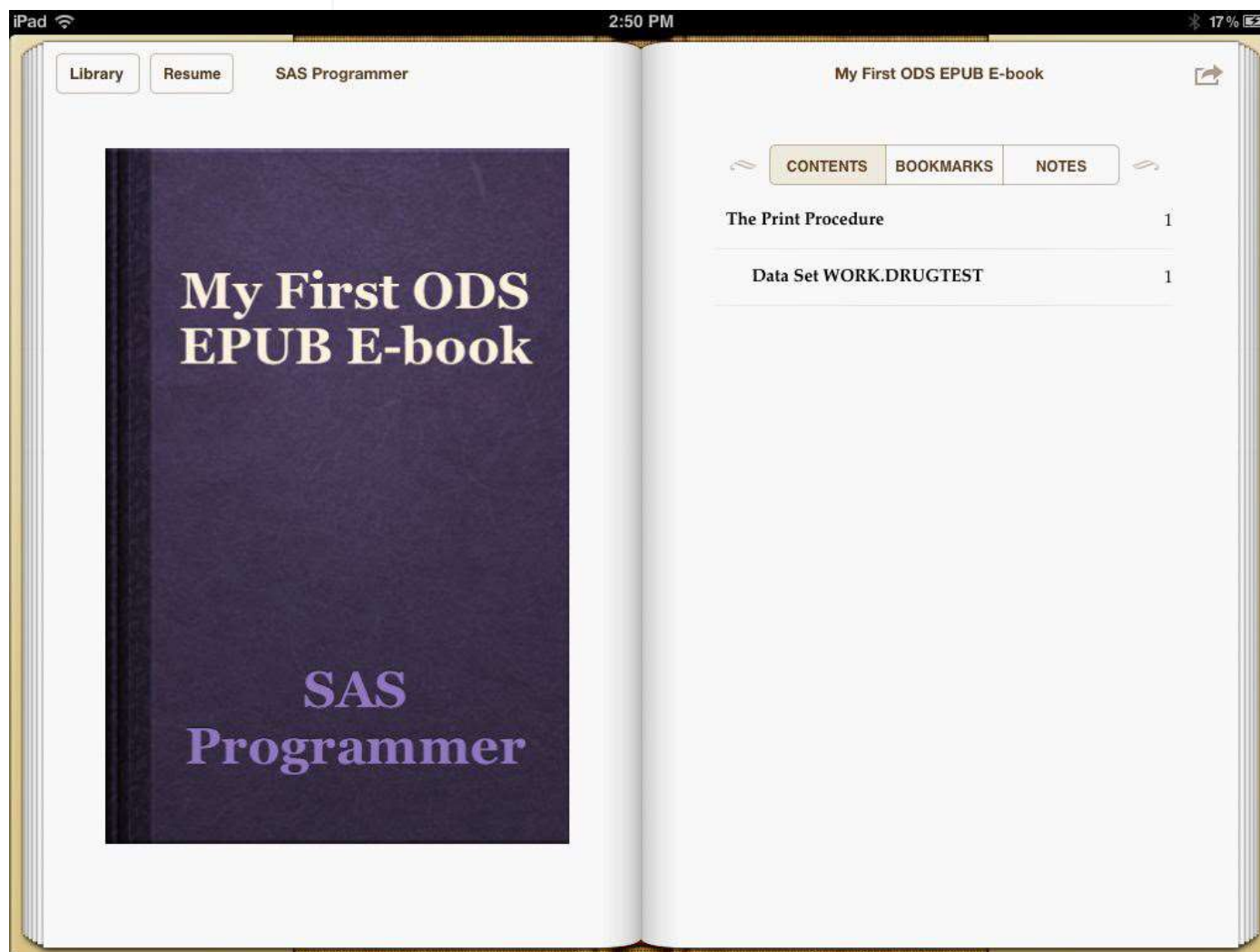
EPUB DESTINATION EXAMPLE

```
ods epub file="glm.epub" title="My First ODS EPUB E-book"  
options(creator="SAS Programmer" description="My First ODS  
EPUB Book" subject="PROC GLM" type="ODS EPUB book");
```

```
ods graphics on;  
proc glm data=DrugTest;  
  class Drug;  
  model PostTreatment = Drug|PreTreatment;  
run;  
quit;
```

```
ods epub close;
```

EPUB DESTINATION SAMPLE OUTPUT

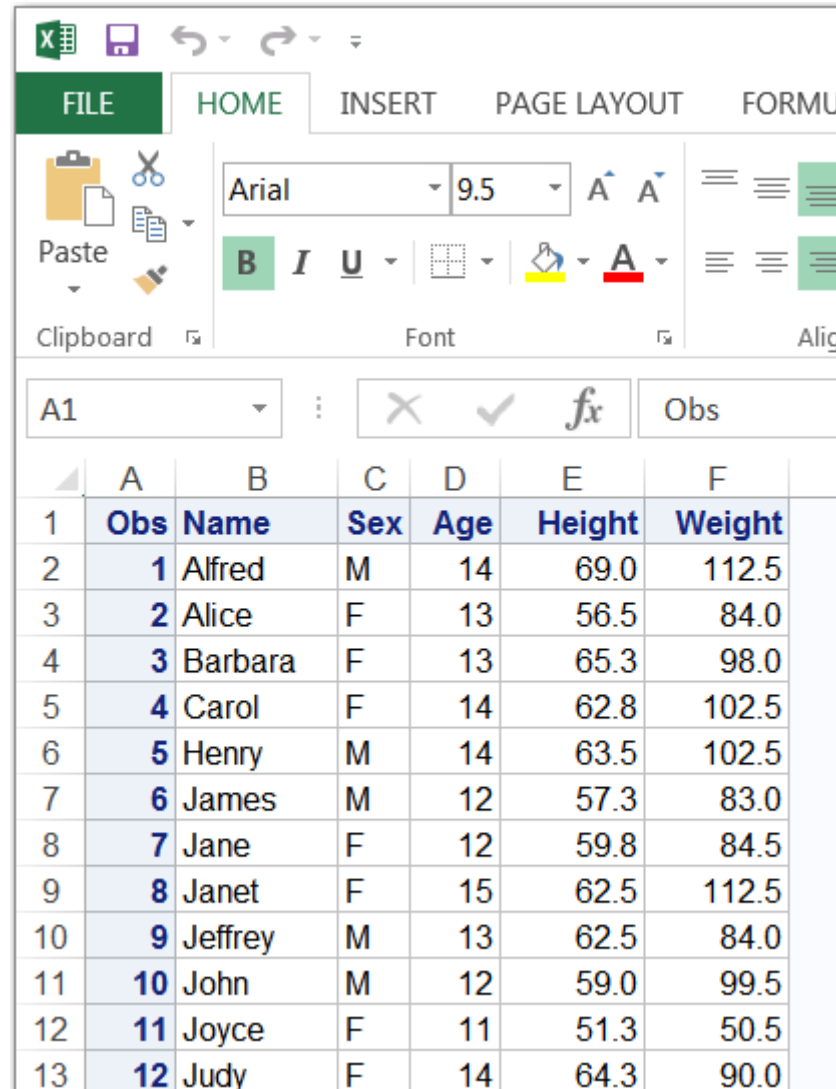


ODS EXCEL EXAMPLE

The ODS Excel destination opens, manages, or closes the ODS destination for Excel, which produces Excel spreadsheet files compatible with Microsoft Office 2010 and later versions.

```
ods excel file="c:\temp\test.xlsx"  
    options(sheet_name='OFAC'  
            zoom='100' row_heights='12.75');  
  
proc print data=sashelp.class;  
run;  
  
ods excel close;
```

ODS EXCEL PARTIAL OUTPUT



The image shows a partial view of an Excel spreadsheet. The ribbon includes FILE, HOME, INSERT, PAGE LAYOUT, and FORMULAS. The HOME ribbon is active, showing the Clipboard group (Paste) and the Font group (Arial, 9.5, Bold, Italic, Underline, Text Color, Background Color). The active cell is A1. The data table below is an ODS output with columns labeled Obs, Name, Sex, Age, Height, and Weight. The data rows are numbered 1 through 13.

	A	B	C	D	E	F
1	Obs	Name	Sex	Age	Height	Weight
2	1	Alfred	M	14	69.0	112.5
3	2	Alice	F	13	56.5	84.0
4	3	Barbara	F	13	65.3	98.0
5	4	Carol	F	14	62.8	102.5
6	5	Henry	M	14	63.5	102.5
7	6	James	M	12	57.3	83.0
8	7	Jane	F	12	59.8	84.5
9	8	Janet	F	15	62.5	112.5
10	9	Jeffrey	M	13	62.5	84.0
11	10	John	M	12	59.0	99.5
12	11	Joyce	F	11	51.3	50.5
13	12	Judy	F	14	64.3	90.0

OUTPUT ODS STATISTICAL GRAPHICS

Enhancements to ODS Graphics provide

- Several new plot types, including axis tables that create an axis-aligned row or column of textual data.
- The addition of numerous plot layout, panel, and axis options to control and enhance the output of your graphs.
- A new sub-pixel rendering feature provides smoother curves for line charts and more consistent spacing in bar charts.

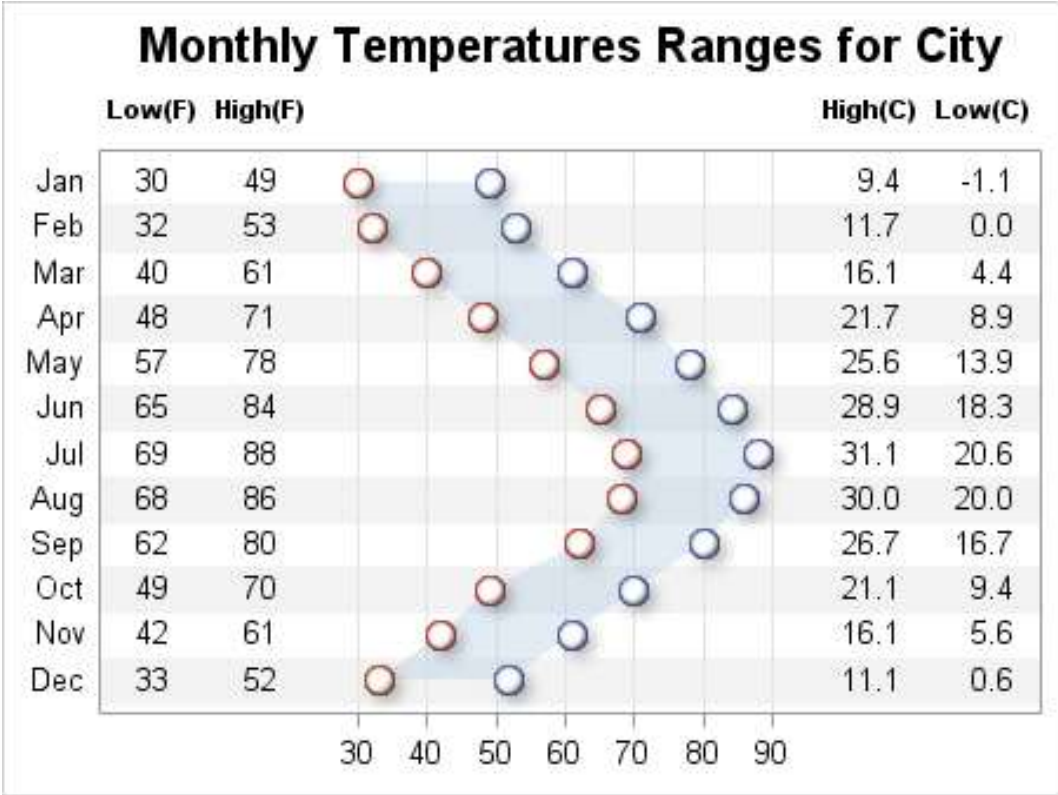
OUTPUT ODS STATISTICAL GRAPHICS

Additional Information:

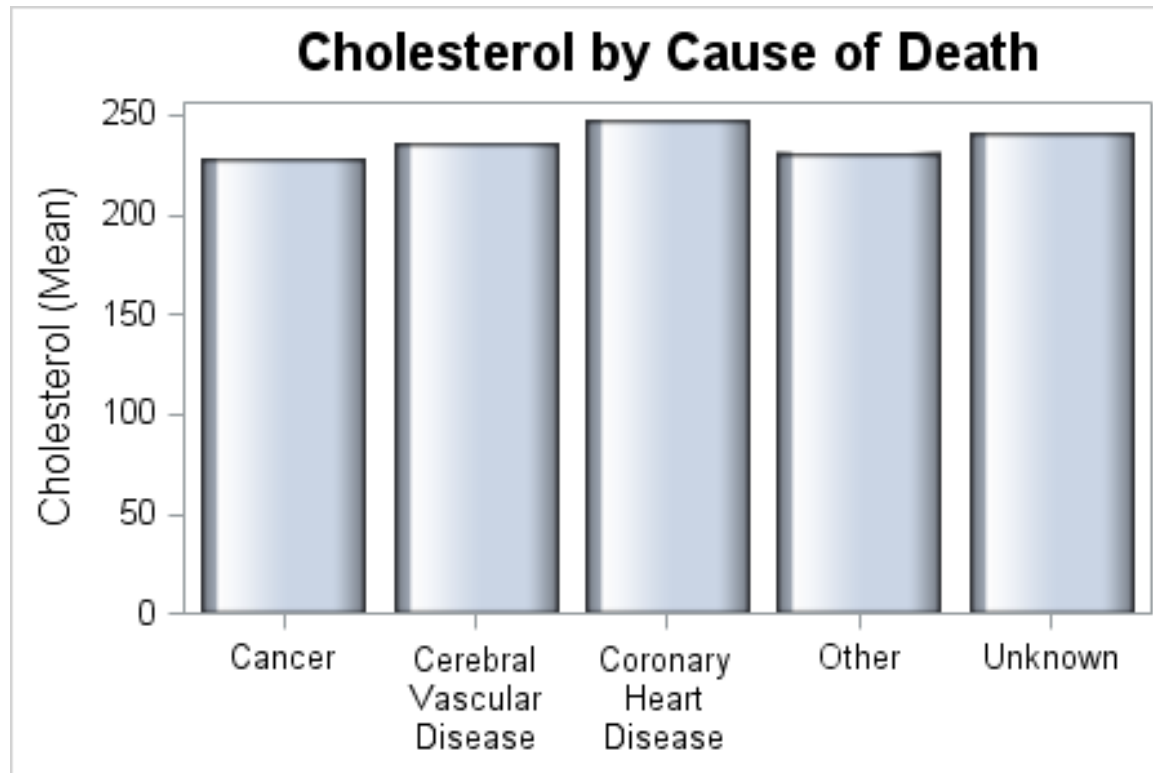
- The ODS Graphics products provide more options for fitting or splitting data labels, curve labels, and axis tick values when there is not enough room to display the text normally.
- The ODS Graphics Designer introduces an Auto Charts feature that generates a variety of graphs automatically, based on your data.

AXISTABLE STATEMENT

AXIS ALIGNED TEXT VALUES



ENHANCED GRAPHS



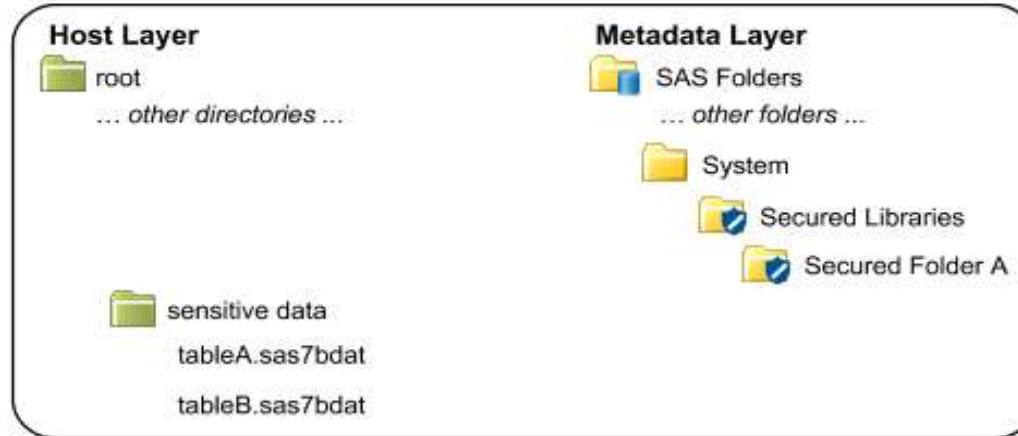
SECURITY

- SAS/SECURE is now delivered free of charge with Base SAS.
 - Uses the industry standard Advanced Encryption Standard (AES) with 64-bit salt.
- **Metadata-bound libraries** universally enforce metadata-layer permission requirements for physical tables—regardless of how a user requests access from SAS – closing the “LIBNAME hole”
 - The **AUTHLIB procedure** is a new utility procedure that enables you to manage metadata-bound libraries.

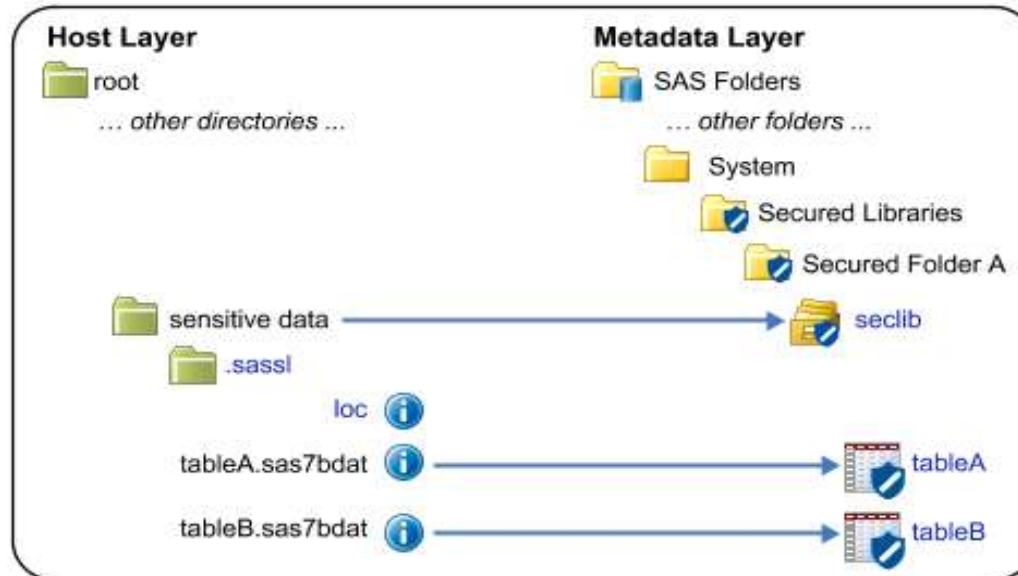
SECURITY METADATA BOUND LIBRARIES

Depiction of a Metadata-Bound Library

Before:



After:



ⓘ = security location information

→ = security binding

EXTENDED ATTRIBUTES

CUSTOMIZED ATTRIBUTES FOR DATA SETS AND VARIABLES

Extended attributes are customized metadata for your SAS files. They

- Can be defined on a data set or on an individual variable
- Are organized into (name, value) pairs
- Can be numeric or character – no pre-defined limit on the number of bytes allowed for a character value
- Are managed by PROC DATASETS
- Base engine support with more to follow (MLE, ACCESS)

EXTENDED ATTRIBUTES

EXAMPLE

```
data mysales;
  purchase = "car"; age = 37; cars = 3;
run;

proc datasets nolist ;
  modify mysales;
  xattr add ds mylabel="This label can be as long
                                as I need it to be, with lots
                                of details about this data set.";
  xattr add var age ( mean = 30 )
              cars ( maker = "Ferrari" models=4 );
run;
quit;
```


OTHER NEW FEATURES

The Work library data sets and catalogs, and the values of global statements, macro variables, and system options can be **preserved between SAS sessions**.

The PRESENV Procedure

- The PRESENV procedure preserves all global statements and macro variables in your SAS code from one SAS session to another.

Set SAS System Option

```
options presenv;
```

Creates data to be used in subsequent session- submitted before exiting SAS

```
proc presenv save permdir=permdir sascode=sascode;  
run;
```

Restore

```
%include 'restore-file'; run;
```

SAS data sets can be written to an external file in JSON representation.

The JSON Procedure

- The JSON procedure reads data from a SAS data set and writes it to an external file in JSON representation.

SAS now processes arbitrary text that contains SAS macro specifications in an input stream. The macro code in the stream can be expanded and stored in a file.

The STREAM Procedure

- The STREAM procedure enables you to process an input stream that consists of arbitrary text that can contain SAS macro specifications. It can expand macro code and store it in a file.

OTHER NEW FEATURES

The SAS language now supports time zones based on Universal Coordinate Time (UTC).

- Data sets and catalog time stamps can specify the time based on a specific time zone.
- SAS can also determine the time for an area, taking into account Daylight Savings Time.

Examples of new functions to support UTC include:

- TZONEID-returns the current time zone ID.
- TZONENAME-returns the current standard or daylight savings time and the time zone name.
- TZONES2U-converts a SAS datetime value to a UTC datetime value.

SAS/ACCESS TO PC FILE FORMATS



THE POWER TO KNOW.

XLSX FILES ENHANCEMENTS

- Add a new Microsoft Excel XLSX worksheet to an existing workbook
- Update an existing worksheet in a workbook
- Export XLSX files directly to UNIX
- Specify DBMS=XLSX to read and write to Excel workbooks under UNIX and Microsoft Windows without accessing the PC Files Server.

EXAMPLE

EXPORT SAS DATA SETS TO EXCEL 2010 WORKBOOK AND REPLACE SHEETS

```
LIBNAME SDF V9 "&sasdir";  
PROC EXPORT DATA=SDF.INVOICE  
    FILE="&tmpdir.text.xlsx"  
    DBMS=XLSX REPLACE;  
    SHEET=' Invoice' ;  
RUN;
```

```
PROC EXPORT DATA=SDF.ORDERES  
    FILE="&tmpdir.text.xlsx"  
    DBMS=XLSX REPLACE;  
    SHEET=' Orders' ;  
RUN;
```


EXAMPLE

EXPORT SAS DATA SETS TO EXCEL 2010 WORKBOOK AND ADD A NEW SHEET

```
LIBNAME SDF V9 "&sasdir";  
PROC EXPORT DATA=SDF.INVOICE  
    FILE="&tmpdir.text.xlsx"  
    DBMS=XLSX;  
    SHEET='Invoice_%2013';  
RUN;
```

SAS SUPPORT FOR HADOOP



SUPPORT FOR HADOOP

FOUNDATION SAS

Foundation SAS offers support for Hadoop through

- Base SAS
- SAS/Access Interface to Hadoop (Hive)
- SAS/Access Interface to HAWQ
- SAS/Access Interface to Impala



FILENAME statement – DATA step can read and write HDFS files.

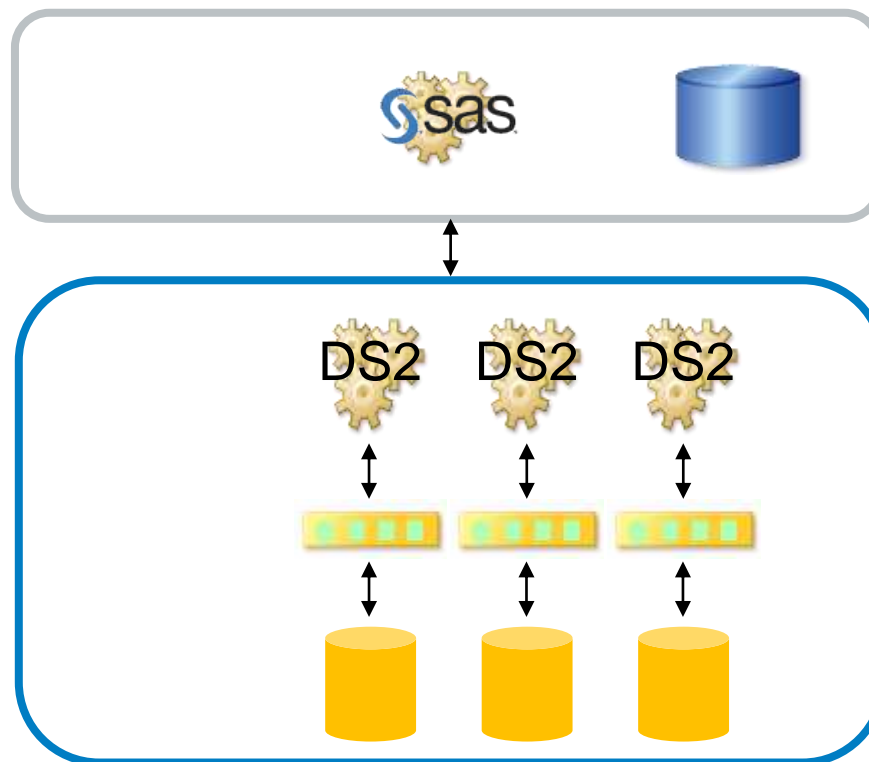
PROC HADOOP - Copy or move files, execute MapReduce and Pig code, execute file system commands

SAS/Access Interface to Hadoop supports

- SQL Pass-through
 - HiveQL queries passed to Hive for processing
- LIBNAME statement for Hadoop
 - Hive tables appear as SAS data sets
 - Access engine can translate to HiveQL to optimize data processing
 - Capable of significant in-database processing

HADOOP SUPPORT SAS IN-DATABASE CODE ACCELERATOR

With the SAS In-Database Code Accelerator, DS2 code can be executed in-database. This allows for more complex processing than with SQL alone.



Note: This technology is not part of the standard Foundation SAS offering.

SAS/GRAPH

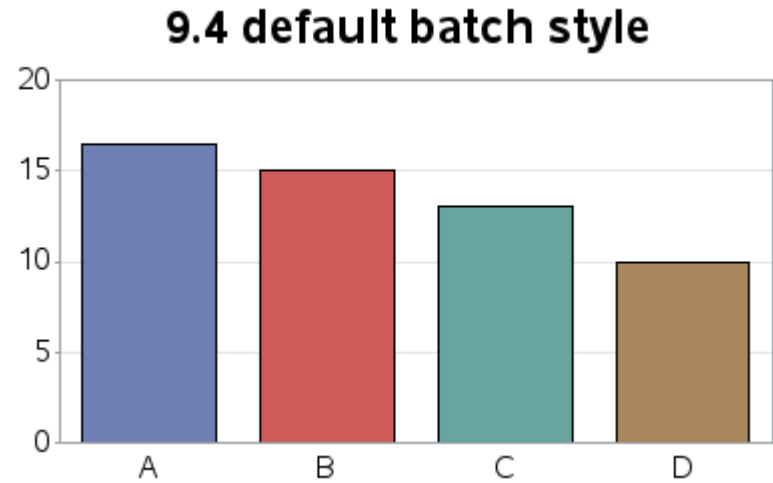
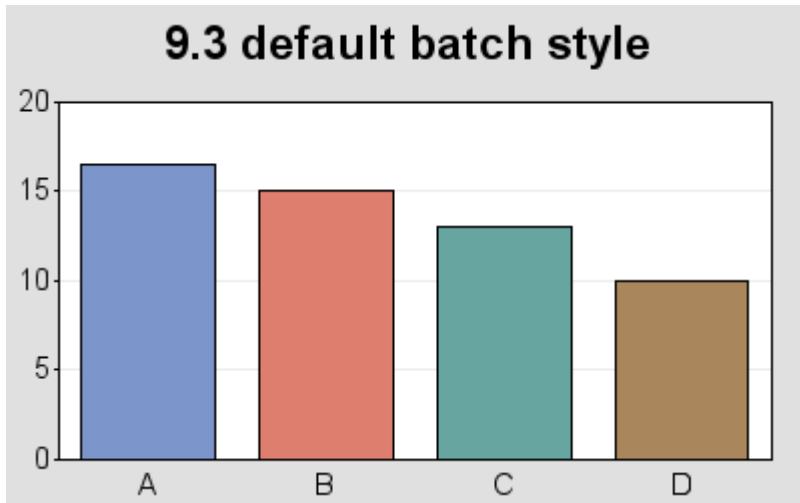


SAS/GRAPH SELECTED ENHANCEMENTS

- New Default Style for batch mode
- New Graph styles (DOVE and RAVEN)
- The GIF device now supports RGBA color mode (transparency) and anti-aliasing.
- The SVG and GIF devices now support animation.
- Maps data sets have been updated
- The GEOCODE procedure now supports non-U.S. street geocoding

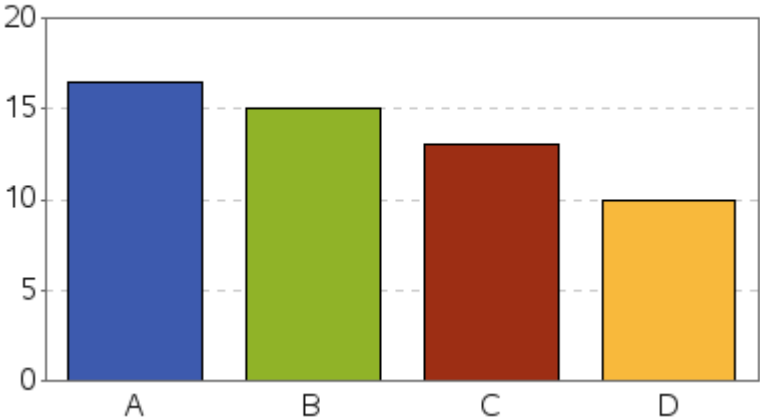
SAS/GRAPH DEFAULT STYLE

New Default Style for batch mode

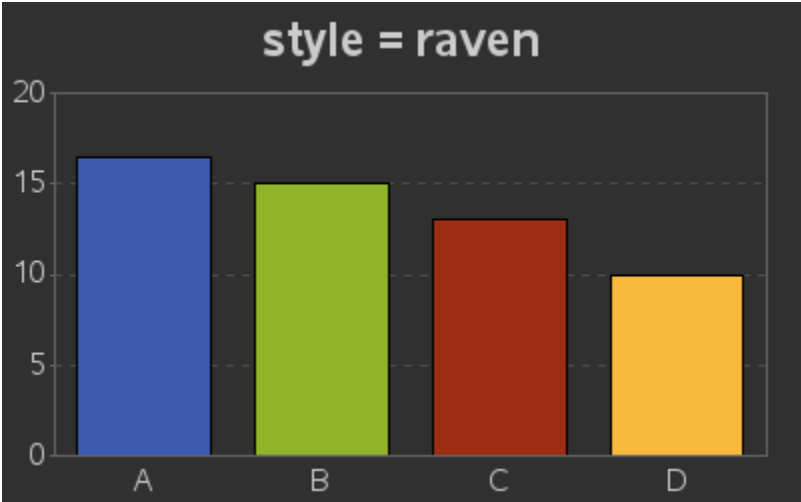


SAS/GRAPH NEW GRAPH STYLES

style = dove



style = raven



SAS/GRAPH ANTI-ALIASING

The GIF device now supports RGBA color mode (transparency) and anti-aliasing.

SAS 9.3 device=gif
jagged lines

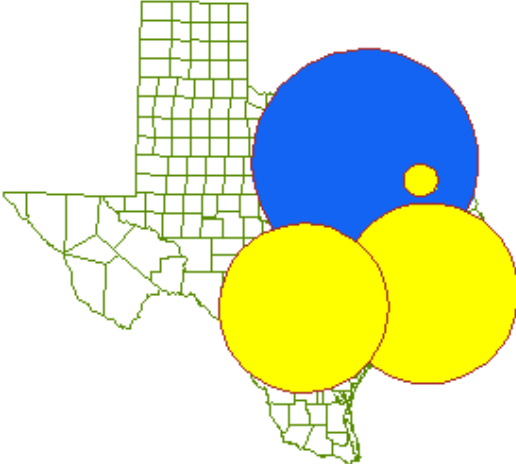


SAS 9.4 device=gif
anti-aliased lines

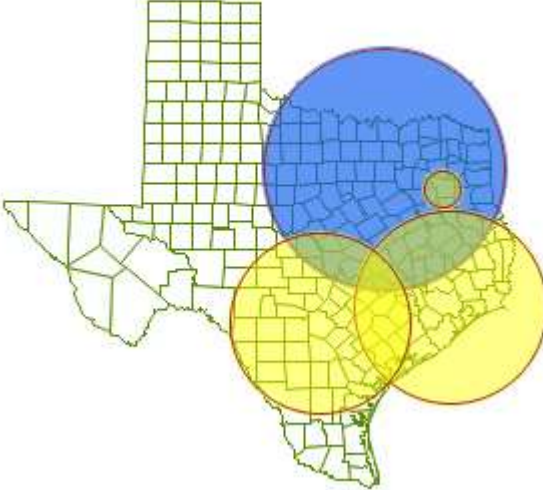


SAS/GRAPH GIF ALPHA-TRANSPARENCY

SAS 9.3 device=gif
solid colors



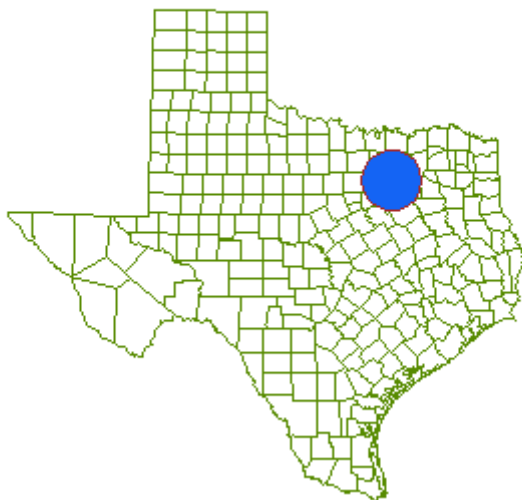
SAS 9.4 device=gif
alpha transparent colors



SAS/GRAPH BETTER GIF ANIMATIONS

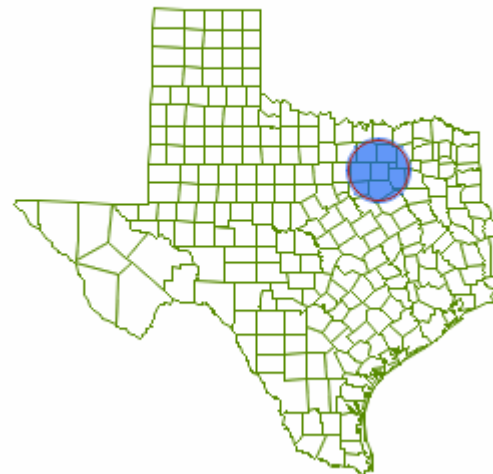
SAS 9.3 GIF Animation

No smooth lines or alpha-transparency



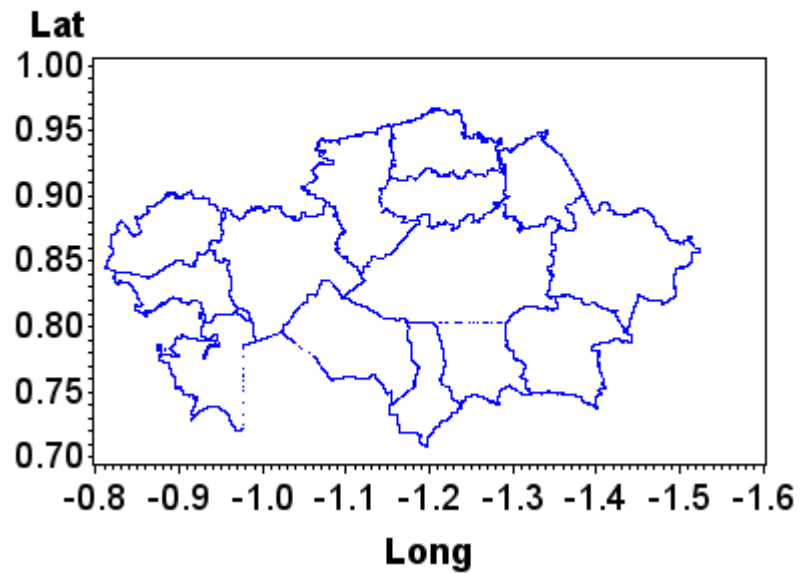
SAS 9.4 GIF Animation

With smooth lines & alpha-transparency

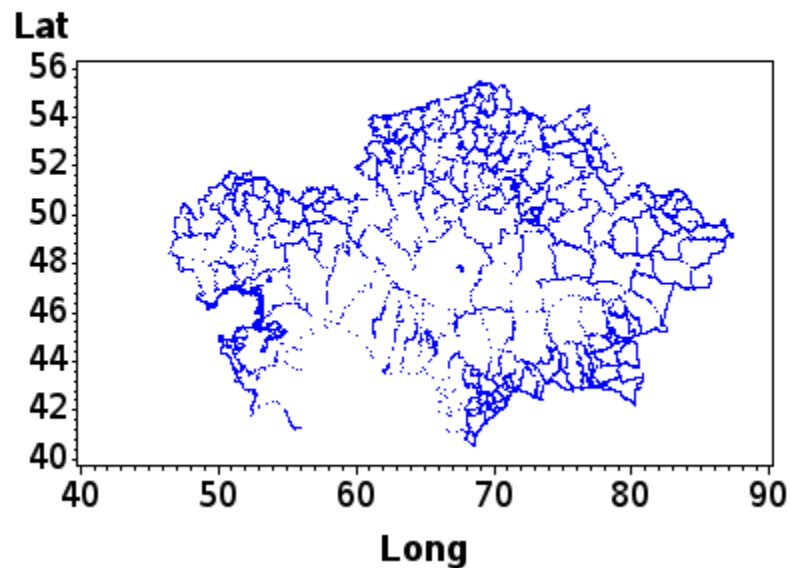


SAS/GRAPH NEW MAPS

9.3 maps.Kazakhst
westlong radians

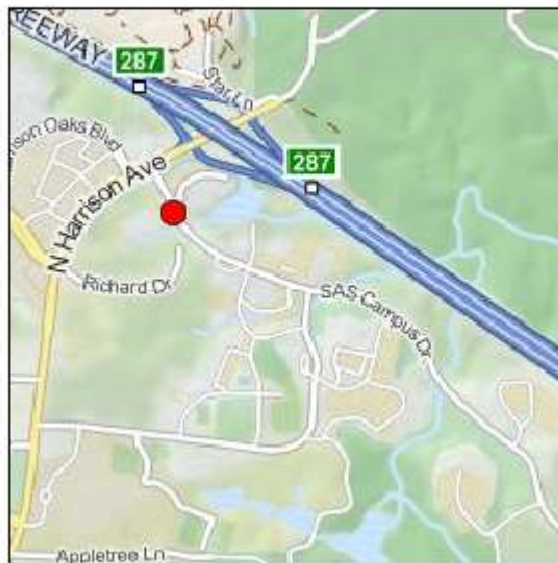


9.4 mapsgfk.Kazakhstan
eastlong degrees



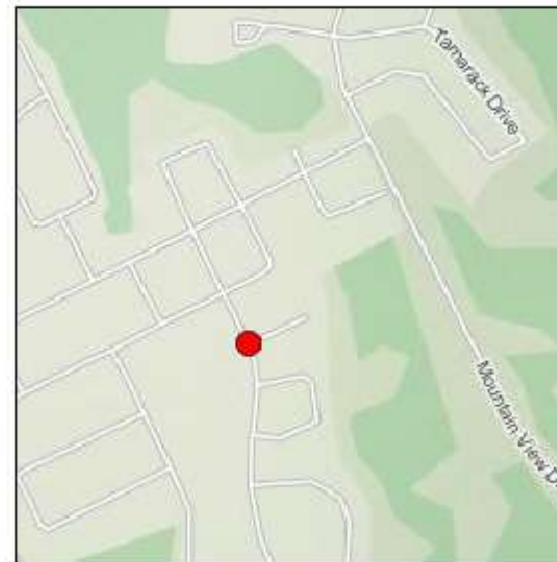
- The GEOCODE procedure now supports non-U.S. street geocoding

9.3 Street-Level Geocoding in United States



100 SAS Campus Drive, Cary, NC

9.4 Street-Level Geocoding in Canada



55 Almond Place, Whitehorse, YT

SAS STUDIO



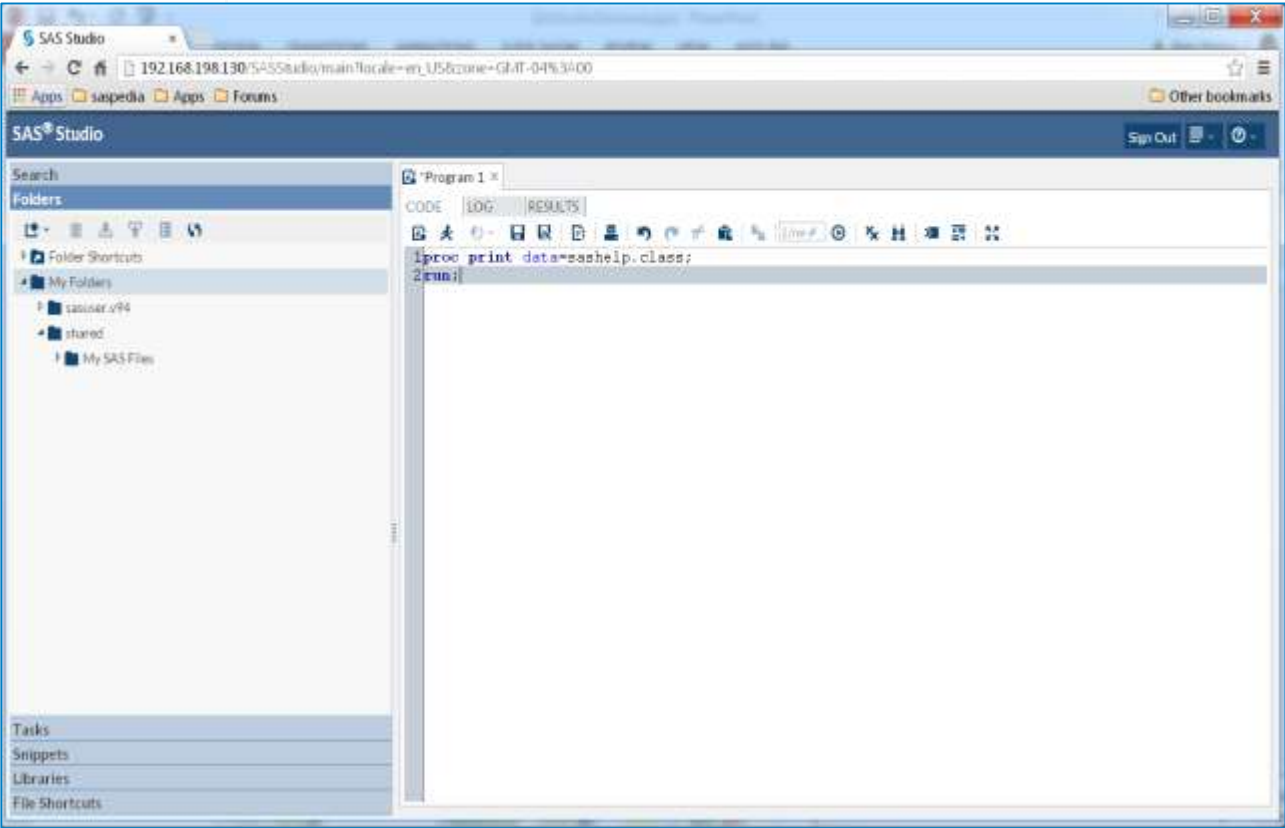
SAS STUDIO

SAS® Studio is a SAS developer environment that runs in a Web browser, enabling developers to program and interact with SAS wherever and whenever they want.

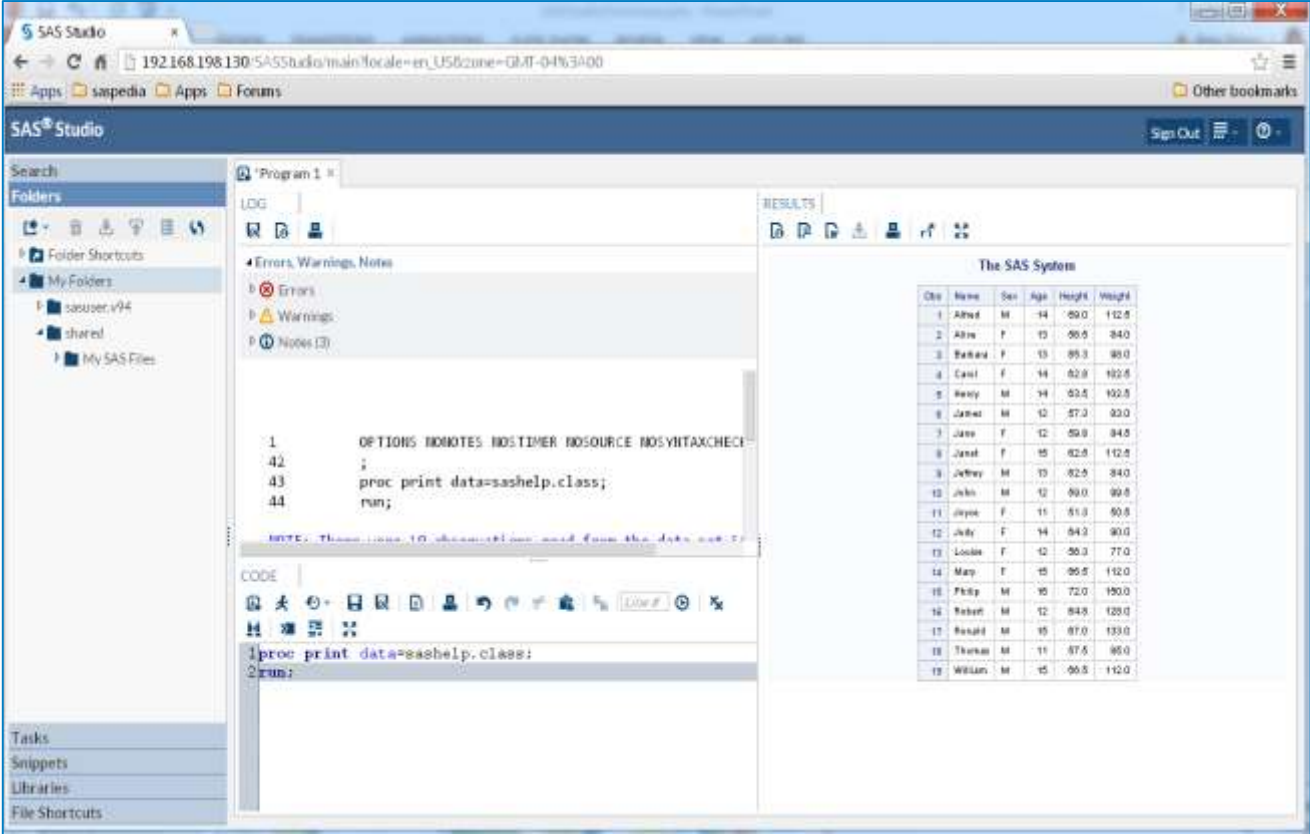
What's so great about SAS® Studio?

- **Availability.** SAS Studio allows SAS programmers to submit SAS code from a wide range of devices, from wherever they happen to be.
- **Consistency.** Become familiar with the SAS Studio user interface once and use it throughout your career.
- **Assistance.** In SAS Studio, SAS code is front and center. To speed development and promote consistent and efficient coding practices, functions similar to SAS® Enterprise Guide® (such as code-generating tasks and auto-complete) are available.

SAS STUDIO AVAILABLE THROUGH ALL MODERN WEB BROWSERS



SAS STUDIO ONE INTERFACE FOR SAS PROGRAMMING



SAS STUDIO INCREASED DEVELOPER PRODUCTIVITY

The screenshot displays the SAS Studio interface for a regression analysis. The left sidebar contains a 'Tasks' pane with categories like 'My Tasks', 'Correlations and Regression', 'Data', 'Descriptive', 'Econometrics', 'Graph', 'High Performance', 'Introductory Statistics', 'Regression', 'Two-sample t Test', 'Multivariate', and 'Survival Analysis'. The main workspace is titled 'Regression 1' and is divided into several panes:

- DATA | OPTIONS | INFORMATION:** Contains settings for the regression model, including checkboxes for 'Scatter Plots', 'Diagnostic and Residual Plots', and 'METHODS'.
- LOG | RESULTS:** Displays a grid of diagnostic plots: 'Residual vs Quantile', 'Residual vs Predicted Value', 'Cook's D', 'Histogram of Residuals', 'Fitted-Value vs Residual', and 'Proportion Less'. A summary table is also present:

Observations	12
Parameters	4
Error DF	8
MSR	16.805
R-Square	0.8305
Adj R-Square	0.8047

- CODE:** Shows the SAS code used for the regression analysis:

```
31  
32 proc reg data=SASHELP.ELECTRIC plots(only)=(diagnostics residuals d  
33   observedbypredicted);  
34   model Revenue=Nuclear NaturalGas Hydro / stb clb alpha=0.65 wpe  
35   output out=work._residuals_ r=_Residual_;  
36   run;  
37 quit;  
38  
39 .....  
40 *** Test for normality ***  
41 .....  
42 ods select TestsForNormality;  
43  
44 proc univariate data=work._residuals_normal_;  
45   var _Residual_;
```

RESOURCES

- Online documentation:
<http://support.sas.com/documentation/onlinedoc/base/index.html>
- ODS EPUB Destination
<http://support.sas.com/resources/papers/proceedings13/368-2013.pdf>
- ODS PowerPoint Destination
<http://support.sas.com/resources/papers/proceedings13/041-2013.pdf>

THANK YOU FOR ATTENDING!



**THE
POWER
TO KNOW.**