






**Eine gute Basis:
SAS Base und SAS Graph**

Nils-Henrik Wolf

**THE
POWER
TO KNOW®**

Copyright © 2006, SAS Institute Inc. All rights reserved.

Agenda

- Erweiterungen für SAS Graph
- Neue Graph Prozeduren
- Neuerungen in SAS Base
- Live Demo ausgesuchter Codes

Copyright © 2006, SAS Institute Inc. All rights reserved.

SAS Language Features

- **Prozeduren**
- Functions
- User-Defined Functions
- External Functions

Copyright © 2006, SAS Institute Inc. All rights reserved.

Prozeduren

- **Erweiterungen von BASE Prozeduren**
- Like Operator Escape Character

Copyright © 2006, SAS Institute Inc. All rights reserved.

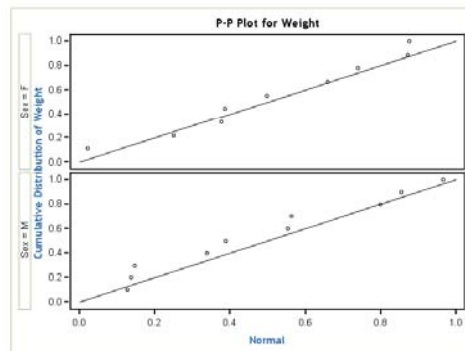
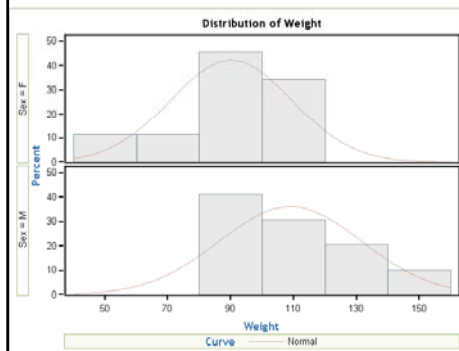
Erweiterte Prozeduren

PROC Univariate unterstützt ODS Graphics

```
ods listing close;
ods graphics;
ods html style=meadow;
title 'Example of ODS Graphics Added to PROC Univariate';
title2 'Provided by GBIDI';
proc univariate data=sashelp.class noprint;
  class sex;
  histogram weight / normal nrows = 2;
  ppplot weight;
run;
ods _all_ close;
```

Example of ODS Graphics Added to PROC Univariate
Provided by GBIDI

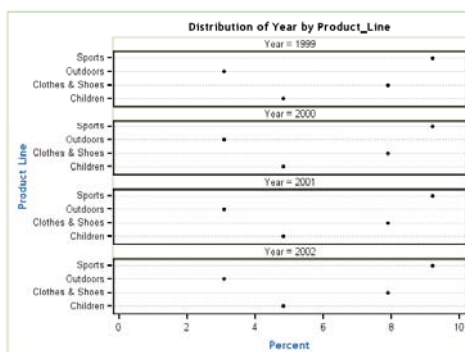
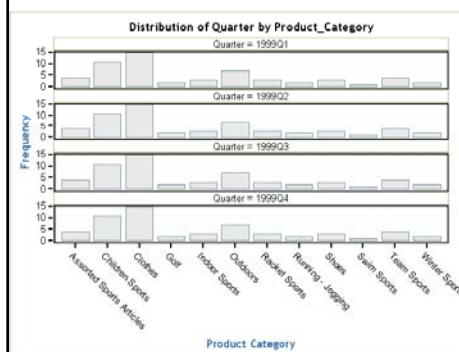
Example of ODS Graphics Added to PROC Univariate
Provided by GBIDI



Erweiterte Prozeduren

PROC FREQ unterstützt ODS Graphics

```
ods html style=meadow;
ods graphics on;
proc freq data=sashelp.orsales;
  title1 'Example of ODS Graphics Added to PROC FREQ';
  title2 'Provided by GBIDI';
  tables year*quarter*product_category / plots=freqplot(type=bar);
  tables year*product_line / plots=freqplot(type=dot scale=percent);
run;
ods graphics off;
ods html close;
```



SET Statement Data Set Listen

- Dash - Bereich
- Colon - "Startet mit" liste

Bereichs Beispiel

Vor 9.2

```
data results;
  set sales1 sales2 sales3 sales4;
run;
```

Mit 9.2

```
data results;
  set sales1 - sales4;
run;
```

Colon List Beispiel

Vor 9.2

```
data results;
  set salesJan salesFeb salesMar;
run;
```

Mit 9.2

```
data results;
  set sales: ;
run;
```

Copyright © 2006, SAS Institute Inc. All rights reserved.

SET Statement INDSNAME Option

INDSNAME=variable

- Erzeugt eine Variable welche den Namen des Datasets speichert aus dem sie gelesen wurde
- Speichert automatisch auch die Library mit

Copyright © 2006, SAS Institute Inc. All rights reserved.

SET Statement INDSNAME Option

```
data results;  
  set prod_BW123_03012008  
      prod_BW123_03022008  
      prod_MH987_03032008  
  indsname = cur_dataset;
```

Erzeugt und benennt eine Variable welche den Dataset Namen speichert

```
Environment = scan(cur_dataset, 2, '_');  
Machine_Number = scan(cur_dataset, 3, '_');  
Date = scan(cur_dataset, 4, '_');
```

```
put cur_dataset= Environment= Machine_Number= Date= ;  
put;  
Run;
```

```
cur_dataset=WORK.PROD_BW123_03012008 Environment=PROD Machine_Number=BW123 Date=03012008  
cur_dataset=WORK.PROD_BW123_03022008 Environment=PROD Machine_Number=BW123 Date=03022008  
cur_dataset=WORK.PROD_MH987_03032008 Environment=PROD Machine_Number=MH987 Date=03032008
```

Copyright © 2006, SAS Institute Inc. All rights reserved.

Prozeduren

- Erweiterungen von BASE Prozeduren
- Like Operator Escape Character

Copyright © 2006, SAS Institute Inc. All rights reserved.

LIKE Operator

- Der LIKE Operator unterstützt jetzt „escape character“.
- Ermöglicht die Suche von Prozentzeichen (%) Unterstrichen (_)

Beispiel: Wenn x folgende Werte enthält

VIEWTABLE: Work.Example	
	x
1	abc
2	a_b
3	axb

Mit escape character

Where x like 'a/_b' escape '/' ;

Nur **a_b** wird selektiert

Ohne escape character

Where x like 'a_b';

a_b und **axb** wird selektiert

Copyright © 2006, SAS Institute Inc. All rights reserved.

SAS Language Features

- Prozeduren
- Functions
- User-Defined Functions
- External Functions

Copyright © 2006, SAS Institute Inc. All rights reserved.

Base Language Features

Neue Numerische Funktionen

GEODIST Function

Rechnet die Distanz zwischen zwei geodätischen Koordinaten.

Neue Datums Funktionen

INTFIT Function

Gibt ein Zeitintervall zurück welches zwischen 2 Terminen.

Beispiel:

```
Interval = intfit('01jan06'd, '03jan06', 'd');
Interval = DAY2
```

Copyright © 2006, SAS Institute Inc. All rights reserved.

Base Language Features

Neue Datums Funktionen

INTGET Function

Gibt ein Zeitintervall zwischen 3 Terminen zurück

- Alle Möglichen Intervalle zwischen 1 und 2
- Alle Möglichen Intervalle zwischen 2 und 3
- Wenn die Intervalle gleich sind werden sie zurückgegeben
- Wenn ein Intervall ein Vielfaches des anderen ist wird das kleinere zurückgegeben
- Ansonsten wird missing zurückgegeben

Beispiel:

2 days
6 days

```
Interval = intget('01mar08'd, '03mar08'd, '09mar08'd);
Interval = DAY2
```

Copyright © 2006, SAS Institute Inc. All rights reserved.

SAS Language Features

- Prozeduren
- Functions
- User-Defined Functions
- External Functions

Copyright © 2006, SAS Institute Inc. All rights reserved.

Die FCMP Prozedur: Beispiel

Erzeugen einer Funktion

```
proc FCMP outlib = sasuser.funcs.trial;  
  function Study_Day(intervention_date, event_date);  
    if event_date < intervention_date then  
      return(event_date - intervention_date);  
    else  
      return(event_date - intervention_date + 1);  
    endsub;  
run;
```

Copyright © 2006, SAS Institute Inc. All rights reserved.

The FCMP Procedure: Beispiel

Erzeugen einer Funktion

```
proc FCMP outlib = sasuser.funcs.trial;
  function Study_Day(intervention_date, event_date);
    If event_date < intervention_date then
      return(event_date - intervention_date);
    else
      return(event_date - intervention_date + 1);
  endsub;
run;
```

Aufrufen einer Funktion

```
options cmplib = sasuser.funcs;
data _null_;
  start = '15Feb2006'd;
  today = '27Mar2006'd;
  sd = study_day(start, today);
  put sd=;
run;
```

Copyright © 2006, SAS Institute Inc. All rights reserved.

The FCMP Procedure: Beispiel

Erzeugen einer Funktion

```
proc FCMP outlib = sasuser.funcs.trial;
  function Study_Day(intervention_date, event_date);
    If event_date < intervention_date then
      return(event_date - intervention_date);
    else
      return(event_date - intervention_date + 1);
  endsub;
run;
```

Aufrufen einer Funktion

```
options cmplib = sasuser.funcs;
data _null_;
  start = '15Feb2006'd;
  today = '27Mar2006'd;
  sd = study_day(start, today);
  put sd=;
run;
```

Zurückgegebener
Wert für sd war 41

```
152 Options cmplib = sasuser.funcs;
153 Data _null_;
154   start = '15Feb2006'd;
155   today = '27Mar2006'd;
156   sd = study_day(start, today);
157
158   put sd=;
159   Run;

sd=41
NOTE: DATA statement used (Total process time):
      real time           0.06 seconds
      cpu time            0.04 seconds
```

Copyright © 2006, SAS Institute Inc. All rights reserved.

SAS Language Features

- Prozeduren
- Functions
- User-Defined Functions
- External Functions

Copyright © 2006, SAS Institute Inc. All rights reserved.

Die PROTO Prozedur External Functions

- PROC PROTO erlaubt es in C oder C++ geschriebene Funktionen zu registrieren

```
PROC PROTO PACKAGE=catalog-entry <option(s)>;
```

```
LINK load-module;
```

LINK- gibt den Namen Pfad und Modul an welcher die Funktion enthält.

Copyright © 2006, SAS Institute Inc. All rights reserved.

PROC PROTO Example

```
proc proto package = sasuser.myfuncs.mathfun
  label = "package of math functions";
  link "math.dll";
  int split(int x "number to split")
    label = "splitter function";
  int cashflow(double amt,
               double rate,
               int periods,
               double * flows / iotype=0)
    label = "cash flow function";
run;
```

Copyright © 2006, SAS Institute Inc. All rights reserved.

PROC PROTO Beispiel

```
proc fcmp libname=sasuser.myfuncs;
  array flows[20];
  a = split(32);
  put a;
  b = cashflow(1000, .07, 20, flows);
  put b;
  put flows;
run;
```

Output: Listing

Output from the SPLIT and CASHFLOW Functions

```

                                     The SAS System                               1
                                     The FCMP Procedure

16
12
70 105 128.33333333 145.83333333 159.83333333 171.5 181.5 190.25 198.02777778 205.02777778
211.39141414 217.22474747 222.60936286 227.60936286 232.27602953 236.65102953 240.76867658
244.65756547 248.341776 251.841776
```

Copyright © 2006, SAS Institute Inc. All rights reserved.

SAS Code Analyzer

- Erzeugt Metadaten über einen SAS Code
 - SORT
 - EXTRACT
 - METADATEN über Tabellen
 - ...
- Wird in einem File aufgezeichnet
- Kann automatisch in SAS Dataintegration Studio eingefügt werden

Copyright © 2006, SAS Institute Inc. All rights reserved.

SAS Code Analyzer

```
proc scaproc;
  record 'C:\temp\restart\record.txt' attr ;
run;

Data EAST_1994_Sales;
set sashelp.prdsale;
Keep actual product year month;
If year=1994 and region='EAST' then output;
run;

Proc sql;
create table total_monthly_sales as
select distinct month,
sum(actual) as total format=dollar20.
from work.east_1994_sales
group by month;
quit;

Proc means data=EAST_1994_Sales noprint nway;
class year month product;
var actual;
output out=East_Monthly_Sales(drop=_type_ _freq_) sum=;
run;

Data east_1994_percent_monthly_sales;
Merge east_monthly_sales total_monthly_sales;
By month;
percent_of_sales = actual / total * 100;
format percent_of_sales 5.2;
drop total;
run;

proc scaproc;
write;
run;
```

Aufruf der SCAPROC Prozedur
mit dem Filenamen in dem
aufgezeichnet werden soll

Mit dem write Statement
wird die Aufzeichnung
abgeschlossen

Copyright © 2006, SAS Institute Inc. All rights reserved.

SAS Code Analyzer: DI Studio SAS Code Importer Plugin

