

20. SAS Club

Wien, 11. November 2010

„Die Wäscherei“
Albertgasse 49
1080 Wien



THE
POWER
TO KNOW.

Mathematische Modellierung mit SAS

Dr. Gerhard Svolba



THE
POWER
TO KNOW.

Datenmanagement mit SQL

```
PROC SQL;
  CREATE TABLE ordermart
  AS
  SELECT s.ShopID          Format = shop.,
         p.ProductID      Format = product.,
         ProdStatus,
         mdy(month(date),1,year(date)) as MonthYear format = yymmmp7.,
         SUM(quantity) AS Quantity Format = 8.,
         AVG(Price)    AS AvgPrice Format = 8.2
  FROM   products      AS p,
         shops         AS S,
         orders        AS O
  WHERE  p.productid    = o.productid
         AND s.shopid   = o.shopid
  GROUP BY      MonthYear,
               s.ShopID,
               p.ProductID,
  ORDER BY      s.ShopID,
               p.ProductID,
               MonthYear;

QUIT;
```

Datenmanagement mit dem SAS DATASTEP

Erzeugen fortlaufender Nummern und kumulativer Summen

CustID	Date	Points
1	10032004	45
1	04042004	10
1	20042004	20
1	16052004	18
1	01062004	5
2	01022004	10
2	19032004	30
3	05082004	4
3	16082004	16
3	31082004	12
3	10092004	20

```
data customer;
  set customer;
  by CustID;
  if first.custid then do;    Purch_No = 1;
                              Cum_Poi  = Points;    end;

  else do;  Purch_No + 1;
            Cum_Poi + Points;  end;

run;
```

Datenmanagement mit dem SAS DATASTEP

Zugriff auf Werte in vorhergehenden Zeilen (LAG, DIF)

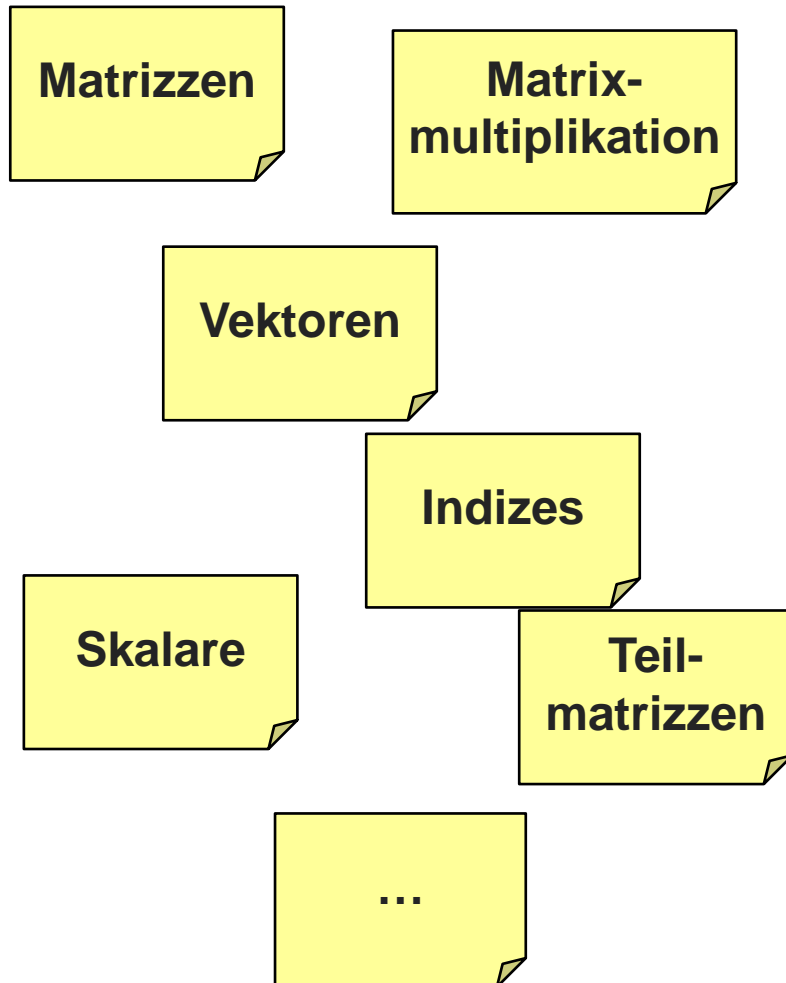
Date	Value
01012004	45
01022004	34
01032004	5
01042004	34
01052004	32
01062004	44

```
data measurements;  
  set measurements;  
  Value_PrevDay = lag(Value);  
run;
```

Datenmanagement mit SAS

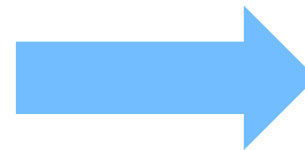
- Datenmanagement
 - SQL
 - SAS Dastep, SAS Procedures, SAS Funktionen
- Features/Funktionen
 - Aggregation von transaktionellen Daten zu Zeitreihendaten; automatisches Auffüllen von Zeitperioden ohne Transaktion
 - Lesen von komplexen Datenstrukturen; Hierarchien, Listen, wechselnde Felddefinitionen
 - Transponieren zwischen unterschiedlichen Datenstrukturen
 - Funktionen: Text-String, Perl Regular Expressions, Statistische und Mathematische Funktionen, Kombinatorik, Wahrscheinlichkeit, Quantile, Zufallszahlen

Was braucht man noch?

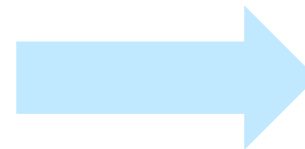


Was bietet SAS?

SAS® IML
(Interactive Matrix Language)



Ermöglicht
Matrizenoperationen
direkt im SAS
(Proc IML)



Bietet ein
Interface zu „R“
(The R Project for
Statistical Computing)

Beispiele für IML Operationen und Funktionen

Operator	Description
` (accent grave)	Transpose (postfix)
- (<i>prefix</i>)	Negative prefix
[]	Subscript
**	Matrix exponentiation
##	Element-wise exponentiation
*	Matrix multiplication
#	Element-wise multiplication
/	Element-wise division
@	Direct (Kronecker) product
+	Addition
-	Subtraction
	Horizontal concatenation
//	Vertical concatenation

Beispiele

- $A+B$: matrix addition
- $A*B$: matrix multiplication,
- $A\#B$: element-wise multiplication
- $A[5,2]$: Element aus der 5. Zeile, 2. Spalte
- $A[1:3,2:10]$:
die ersten drei Spalten für die 2. bis 10. Zeile
- $W = \text{INV}(T(x)^*x);$

Matrizenoperationen direkt im SAS mit Proc IML (Beispiel)

$$\hat{\beta} = (X'X)^{-1}X'y$$

$$\hat{Y}=X*\beta$$

PROC IML;

```
xpxi=inv(t(x)*x);          /* inverse of X'X          */
beta=xpxi*(t(x)*y);        /* parameter estimate      */
yhat=x*beta;               /* predicted values        */
```

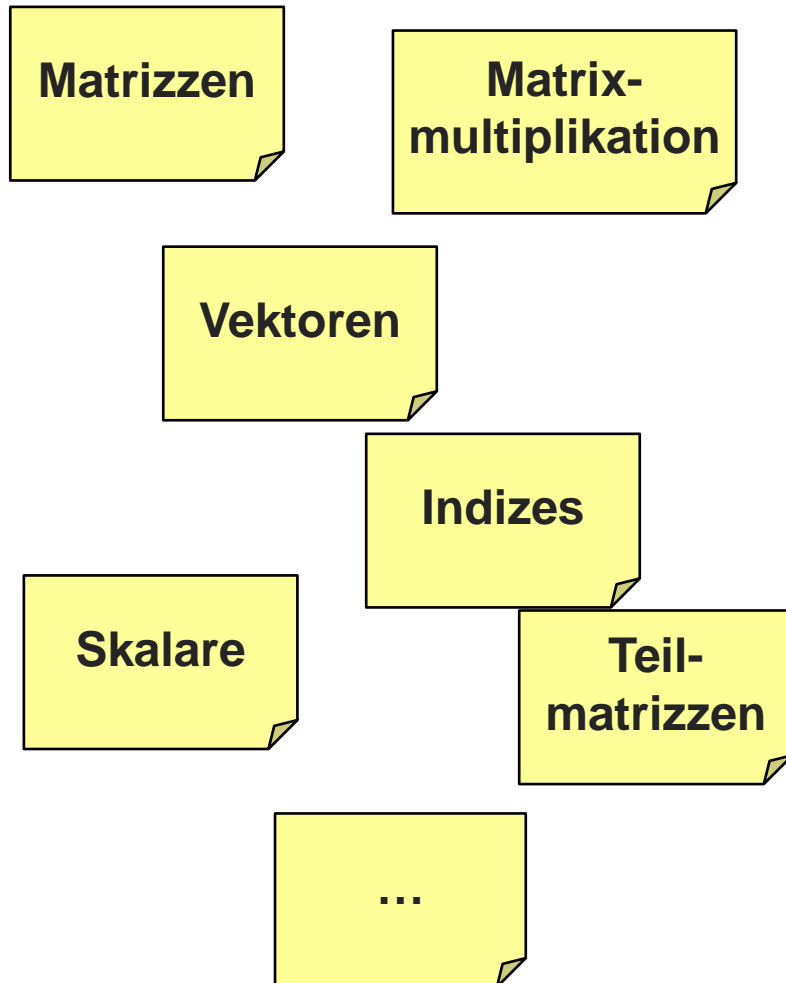
```
resid=y-yhat;              /* residuals               */
sse=ssq(resid);            /* SSE                     */
n=nrow(x);                 /* sample size             */
dfe=nrow(x)-ncol(x);      /* error DF                */
mse=sse/dfe;               /* MSE                     */
cssy=ssq(y-sum(y)/n);     /* corrected total SS     */
rsquare=(cssy-sse)/cssy;   /* RSQUARE                 */
stdb=sqrt(vecdiag(xpxi)*mse); /* std of estimates       */
t=beta/stdb;              /* parameter t tests      */
prob=1-probf(t#t,1,dfe);   /* p-values                */
```

QUIT;

Live Demo:

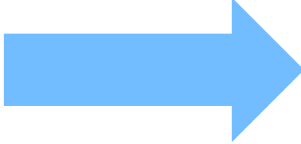
- Lineare Regression mit Proc IML

Was braucht man noch?

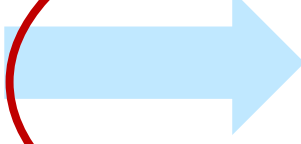


Was bietet SAS?

SAS® IML
(Interactive Matrix Language)

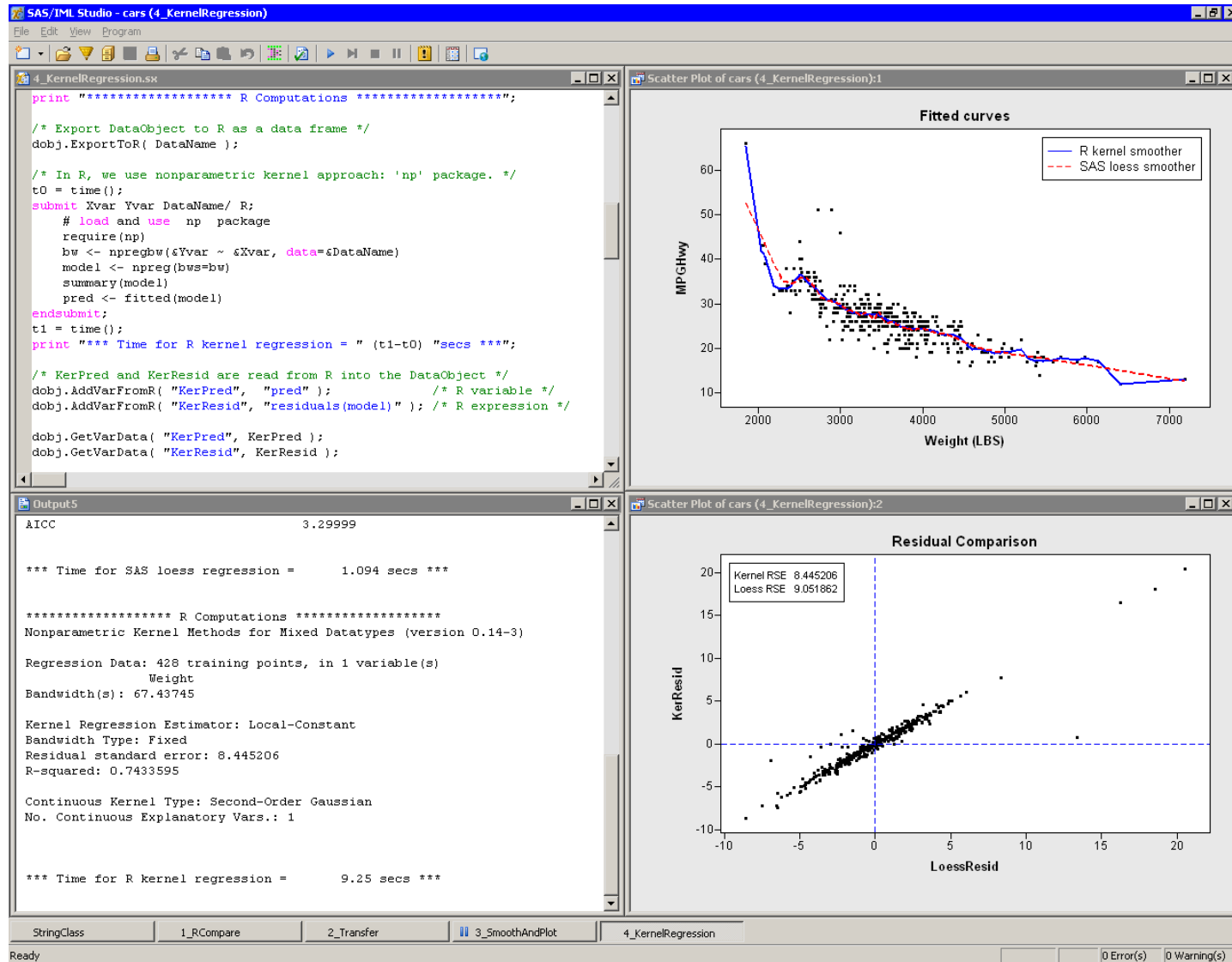


Ermöglicht
Matrizenoperationen
direkt im SAS
(Proc IML)



Bietet ein
Interface zu „R“
(The R Project for
Statistical Computing)

Das SAS IML-Studio



Funktionsweise des R-Aufrufs aus SAS

- `/**... SAS Datastep .. SAS Procedures`
- `... Matrizen-Operationen`
- `... ***/`
- `run ExportMatrixToR(y, "Ry");`
- `submit / R;`
- `Rs <- smooth(Ry)`
- `Rs <- as.matrix(Rs)`
- `endsubmit;`
- `run ImportMatrixFromR(s, "Rs");`
- `/** Aufbereitung der Ergebnisse`
- `... SAS Datastep ... SAS Procedures ... Reporting, Plots, ... ***/`
- `declare ScatterPlot plot;`
- `plot = ScatterPlot.Create("Tukey", x, y);`
- `plot.DrawUseDataCoordinates();`
- `plot.DrawLine(x, s);`

SAS/IML und IML/Studio

- Zielgruppe: “High-end data analysts”
- Flexible Programmierumgebung auf Basis von SAS und SAS/IML. Erweitert um die IML-Plus Sprache
- Aufruf von R functions
 - Kombination der Power von SAS mit dem „Free Tool“ R aus dem akademischen Bereich
 - Zugang zu neu publizierten Methoden, sobald diese erscheinen