

## Erzeugung nichtnormaler multivariater Zufallsgrößen mit SAS

Armin Tuchscherer<sup>1)</sup>\*, Paul Eberhard Rudolph<sup>1)</sup>,  
Bernd Jäger<sup>2)\*\*</sup>, Margret Tuchscherer<sup>1)</sup>

1) Forschungsinstitut für die Biologie landwirtschaftlicher Nutztiere Dummerstorf-Rostock,  
Wilhelm-Stahl-Allee 2, D-18196 Dummerstorf

\* Telefon: 038208 / 68904

\* eMail: atuschsch@fbn-dummerstorf.de

2) Institut für Biometrie und medizinische Informatik, Ernst-Moritz-Arndt-Universität  
Greifswald Sauerbruchstraße, D-17487 Greifswald

\*\* eMail: bjaeger@biometrie.uni-greifswald.de

### Abstract

Voraussetzung für Robustheitsuntersuchungen multivariater Verfahren mittels Simulation ist die Erzeugung multivariater Zufallsgrößen, die nicht notwendig multivariat normalverteilt sein müssen. Während die Generierung multivariat normalverteilter Zufallsgrößen bereits gut theoretisch und programmtechnisch aufbereitet ist, findet man in der Literatur neben theoretischen Beschreibungen zur allgemeinen Verfahrensweise kaum praktikable Algorithmen zur Erzeugung multivariater nichtnormaler Zufallsgrößen mit abhängigen Komponenten.

In diesem Beitrag werden die Erzeugung multivariater nichtnormaler Zufallsgrößen mit SAS unter bestimmten Voraussetzungen beschrieben und damit zusammenhängende Probleme aufgezeigt.

### Einleitung

Eigenschaften multivariater statistischer Verfahren sind häufig nicht mehr analytisch, sondern nur mittels Simulationen bestimmbar, wenn z. B. bestimmte Verteilungsvoraussetzungen verletzt sind. Zur Durchführung von Simulationen im Zusammenhang mit multivariaten Verfahren benötigt man geeignete Programme zur Erzeugung multivariater Zufallsgrößen. Häufig ist bei praktischen Problemen die Normalverteilungsvoraussetzung verletzt, so daß die Erzeugung von multivariaten nichtnormalen Zufallsgrößen von besonderem Interesse ist.

Um derartige Simulationen in SAS durchzuführen, ist es naheliegend, die entsprechenden Zufallszahlengeneratoren als SAS-Makros bereitzustellen.

Sei  $\underline{Y} = (y_1, \dots, y_n)'$  ein  $n$ -dimensionaler Vektor von kontinuierlichen Zufallsvariablen  $y_i$ ,  $i = 1, \dots, n$ , mit der Verteilungsfunktion  $F(Y)$ . Mit  $f(Y)$  bezeichnen wir die Dichtefunktion,

$$E(\underline{Y}) = (\mu_1, \dots, \mu_n)'$$

sei der Erwartungswertvektor und die Kovarianzmatrix, deren Existenz wir voraussetzen, sei

$$\text{Cov}(\underline{Y}) = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{21} & \sigma_2^2 & \cdots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{n1} & \sigma_{n2} & \cdots & \sigma_n^2 \end{pmatrix}.$$

Mit Hilfe der Korrelationsmatrix

$$\text{Corr}(\underline{Y}) = \begin{pmatrix} 1 & \rho_{12} & \cdots & \rho_{1n} \\ \rho_{21} & 1 & \cdots & \rho_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{n1} & \rho_{n2} & \cdots & 1 \end{pmatrix} \quad \text{mit} \quad \rho_{ij} = \frac{\sigma_{ij}}{\sigma_i \sigma_j} \quad (i, j = 1, \dots, n; i \neq j) \quad \text{läßt sich die}$$

Abhängigkeit der Komponenten von  $\underline{Y}$  leichter beschreiben.

Bei der Erzeugung von n-dimensionalen Vektoren von Pseudozufallszahlen, die der Verteilung von  $\underline{Y}$  genügen, unterscheiden wir zwischen abhängigen und unabhängigen Komponenten von  $\underline{Y}$ .

### Erzeugung multivariater Zufallsgrößen bei unabhängigen Variablen

Ist  $\underline{Y} = (\underline{y}_1, \dots, \underline{y}_n)'$  ein Vektor von unabhängigen Variablen  $\underline{y}_i$ ,  $i = 1, \dots, n$ , so hat die Verteilungsfunktion  $F(\underline{Y})$  die Gestalt

$$F(\underline{Y}) = \prod_{i=1}^n F_i(y_i),$$

wobei die  $F_i(y_i)$  die Randverteilungsfunktionen der  $\underline{y}_i$  sind. Somit können wir die entsprechenden Pseudozufallsvektoren komponentenweise unabhängig entsprechend der Verteilung  $F_i$  der  $\underline{y}_i$  erzeugen. Damit wären wir bei der Erzeugung eindimensionaler Zufallszahlen, auf die hier aber nicht näher eingegangen wird.

### Erzeugung multivariater Zufallsgrößen bei abhängigen Variablen

Praktisch interessanter, aber auch wesentlich komplexer ist das Problem im Falle abhängiger Komponenten von  $\underline{Y}$ .

#### Methode der bedingten Verteilungen

Die Verteilungsfunktion  $F(\underline{Y})$  kann durch die bedingten Verteilungen

$$F(\underline{Y}) = F_1(y_1)F_2(y_2|\underline{y}_1 = y_1) \cdots F_n(y_n|\underline{y}_1 = y_1, \dots, \underline{y}_{n-1} = y_{n-1})$$

ausgedrückt werden. Das Schöne dieser Methode ist, daß man das Problem der Erzeugung eines n-dimensionalen Zufallsvektors auf eine Reihe von n univariaten Problemen nach folgenden n Schritten zurückführen kann (Rosenblatt, 1952):

#### Algorithmus I

1. Erzeuge  $\underline{y}_1 = y_1$  mit der Randverteilung  $F_1$  von  $\underline{y}_1$ .
2. Erzeuge  $\underline{y}_2 = y_2$  mit der bedingten Verteilung  $F_2(y_2|\underline{y}_1 = y_1)$  von  $\underline{y}_2$  unter der Bedingung  $\underline{y}_1 = y_1$ .
- ⋮
- n. Erzeuge  $\underline{y}_n = y_n$  mit der bedingten Verteilung  $F_n(y_n|\underline{y}_1 = y_1, \dots, \underline{y}_{n-1} = y_{n-1})$  von  $\underline{y}_n$  unter der Bedingung  $\underline{y}_1 = y_1, \dots, \underline{y}_{n-1} = y_{n-1}$ .

#### Transformationsmethode

Oftmals ist es sehr schwierig, die bedingten Verteilungen von Algorithmus I abzuleiten. Eine Möglichkeit, dieser Schwierigkeit auszuweichen, ist die Transformationsmethode. Diese Methode besteht darin,  $\underline{Y}$  als Funktion anderer, oftmals unabhängiger, univariater Zufallsvariablen, die in der Regel leicht zu erzeugen sind, darzustellen.

Die Generierung der multivariaten Normalverteilung mittels linearer Transformation unabhängiger univariater Normalverteilungen (Tuchscherer u.a., 1999) ist nur ein Beispiel hierfür. Die multivariate Normalverteilung kann wiederum Ausgangspunkt für die Erzeugung nicht-normaler multivariater Zufallsgrößen sein.

Beispiele dafür sind das System der gemischten multivariaten Normalverteilungen und Johnsons Translationssystem, auf die im folgenden näher eingegangen werden soll.

## Mischung von Normalverteilungen

Gemischte Normalverteilungen, in der Robustheitsliteratur als verschmutzte Normalverteilungen mit einer sehr breiten Anwendung bekannt, entstehen durch Mischen zweier  $n$ -dimensionaler Normalverteilungen mit Wahrscheinlichkeit  $p$  ( $0 \leq p \leq 1$ ):

$$\underline{Y} \sim p \cdot N_n(\underline{\mu}_1, \underline{\Sigma}_1) + (1-p) \cdot N_n(\underline{\mu}_2, \underline{\Sigma}_2).$$

Der Erwartungswert von  $\underline{Y}$  ist

$$\underline{\mu}^* = E(\underline{Y}) = p\underline{\mu}_1 + (1-p)\underline{\mu}_2,$$

die Kovarianzmatrix

$$\underline{\Sigma}^* = \text{Cov}(\underline{Y}) = p\underline{\Sigma}_1 + (1-p)\underline{\Sigma}_2 + p(1-p)(\underline{\mu}_1 - \underline{\mu}_2)(\underline{\mu}_1 - \underline{\mu}_2)'$$

und die Dichte

$$f(\underline{Y}) = p \cdot f_1(\underline{Y}) + (1-p) \cdot f_2(\underline{Y}),$$

mit

$$f_i(\underline{Y}) = (2\pi)^{-n/2} |\underline{\Sigma}_i|^{-1/2} \exp\left(-\frac{1}{2} \cdot (\underline{Y} - \underline{\mu}_i)' \underline{\Sigma}_i^{-1} (\underline{Y} - \underline{\mu}_i)\right), \quad i=1,2.$$

## Algorithmus II

1. Erzeugen einer auf dem Intervall  $[0,1]$  gleichverteilten Zufallsvariablen  $\underline{U} = U$ .  
Falls  $U \leq p$  gehe zu Schritt 2, sonst gehe zu Schritt 3.
2. Erzeuge  $\underline{Y}$  als  $N_n(\underline{\mu}_1, \underline{\Sigma}_1)$ .
3. Erzeuge  $\underline{Y}$  als  $N_n(\underline{\mu}_2, \underline{\Sigma}_2)$ .

## SAS-Makro %V\_NORMAL

Der Algorithmus II wurde als SAS-Makro %V\_NORMAL umgesetzt unter Verwendung der SAS-Prozedur `proc iml`.

In den SAS-Dateien **Mittel1**, **KOVAR1**, **Mittel2**, **KOVAR2** seien die Erwartungswertvektoren bzw. Kovarianzmatrizen der beiden zu vermischenden Normalverteilungen gespeichert (Voraussetzung für den Aufruf des Makros %V\_NORMAL). Im zweidimensionalen Fall könnten diese Dateien zum Beispiel wie folgt erzeugt werden.

```

data Mittel1;
input mue1;
cards;
0
0
run;

data Mittel2;
input mue2;
cards;
1
1
run;

data KOVAR1;
input co1-co2;
cards;
1 0.9
0.9 1
run;

data KOVAR2;
input co1-co2;
cards;
1 -0.5
-0.5 1
run;

```

Das Makro %V\_NORMAL hat dann folgende Gestalt:

### SAS-Makro

```

/*-----*
| %V_NORMAL(M,P,MEANS1,COVAR1,MEANS2,COVAR2, DATA, NAMES) |
| ===== |
| erzeugt n-dimensionale Pseudozufallszahlenvektoren Y=(y(1),...,y(n))' |
| einer verschmutzten Normalverteilung, d.h. |
| Y ~ P * NV(mue1,COV1) + (1-P) * NV(mue2,COV2) , mit |
| |
| mue1=(mue1(1), ..., mue1(n))', mue2=(mue2(1), ..., mue2(n))' und |
| / \ / \ |
| | COV1(1,1) ... COV1(1,n) | | COV2(1,1) ... COV2(1,n) | |
| COV1=| ... | , COV2=| ... | |
| | COV1(n,1) ... COV1(n,n) | | COV2(n,1) ... COV2(n,n) | |
| \ / \ / |
|-----*
| INPUT: |
| ===== |
| M Anzahl der zu erzeugenden Vektoren |
| P Verschmutzungsgrad (0<=P<=1 ) |
| MEANS1 SAS-Datei der Form |
| mue1(1) |
| . |
| . |
| . |
| mue1(n) |
| COVAR1 SAS-Datei in der Form |
| COV1(1,1) ... COV1(1,n) |
| . |
| . |
| . |
| COV1(n,1) ... COV1(n,n) |
| MEANS2 SAS-Datei der Form |
| mue2(1) |
| . |
| . |
| . |
| mue2(n) |
| COVAR2 SAS-Datei in der Form |
| COV2(1,1) ... COV2(1,n) |
| . |
| . |
| . |
| COV2(n,1) ... COV2(n,n) |
| OUTPUT: |
| ===== |
| DATA temporäre SAS-Datei, mit der Datensatzstruktur: |
| Y1 Y2 ... Yn |
| NAMES Namen der Variablen der Datei 'DATA' |
| z.B. bei n=4 in der Form: |
| {'Y1' 'Y2' 'Y3' 'Y4'} |
|-----*/

```

```
%macro V_NORMAL(M, P, MEANS1, COVAR1, MEANS2, COVAR2, DATA, NAMES);
proc iml;
  use &MEANS1; read all into mue1;
  use &COVAR1;  read all into COV1;
  use &MEANS2; read all into mue2;
  use &COVAR2;  read all into COV2;
  de1=det(COV1);
  de2=det(COV2);
  ERROR='Die Kovarianzmatrix ist nicht positiv semidefinit; DET: ';
  if de1<0 then print ERROR de1;
  if de2<0 then print ERROR de2;
  n=ncol(COV1);
  d=mue1-mue2;
  mue=&P*mue1+(1-&P)*mue2;
  cov=&P*COV1+(1-&P)*COV2+&P*(1-&P)*d*D`;
  print 'Mittelwert und Kovarianzmatrix der verschmutzten Normalverteilung';
  print mue cov;
  A1=j(n,n,0);
  A1=root(cov1);
  A1=A1`;
  A2=j(n,n,0);
  A2=root(cov2);
  A2=A2`;
  DAT=j(&m,n,0);
  Do i=1 to &M;
    c=uniform(0);
    Z=j(n,1,0);
    Z=NORMAL(Z);
    if c<=&P then;
      Y=A1*Z+mue1;
    else;
      Y=A2*Z+mue2;
    Do j=1 to n;
      DAT[i,j]=Y[j];
    end;
  end;
  c=&NAMES ;
  create &Data from DAT [colname=c];
  append from DAT;
quit;
%mend V_NORMAL;
```

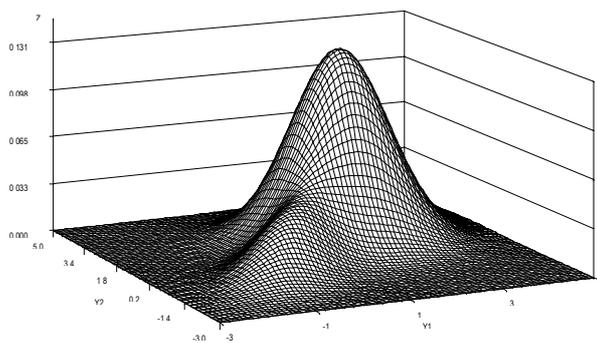
### Aufruf des Makros:

```
%V_NORMAL(10000, 0.5, MITTEL1, KOVAR1, MITTEL2, KOVAR2, Daten, {'Y1' 'Y2'});
```

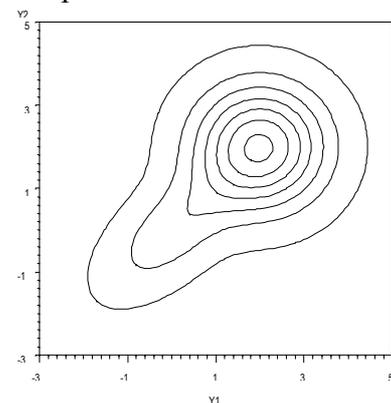
Mit diesem Aufruf des Makros werden **M=10.000** Pseudozufallszahlenpaare der entsprechenden zweidimensionalen Mischnormalverteilung mit der Mischungswahrscheinlichkeit **P=0.5** aus Normalverteilungen mit den Erwartungswertvektoren **MITTEL1** und **MITTEL2** und den Kovarianzmatrizen **KOVAR1** und **KOVAR2** erzeugt und in der temporären SAS-Datei **DATEN** abgelegt. Im folgenden werden Beispiele für zweidimensionale gemischte Normalverteilungen gegeben. Die Dichteplots erfolgten mit `proc g3d`, die Konturplots mit `proc gcontour` und die zweidimensionalen Histogramme mit dem SAS-Makro `%HIST` (Anlage).

### Beispiele für zweidimensionale gemischte Normalverteilungen

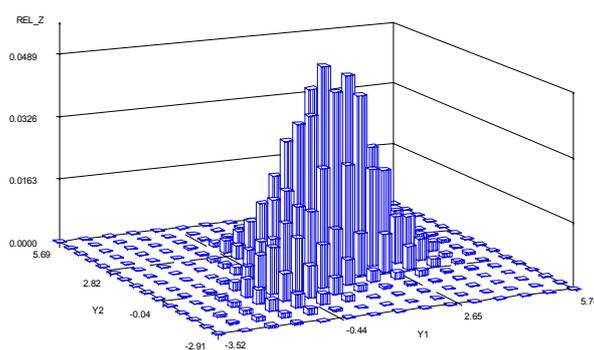
Dichteplot:



Konturplot:



Histogramm (m=10.000 mit %V\_NORMAL):



Parameter:

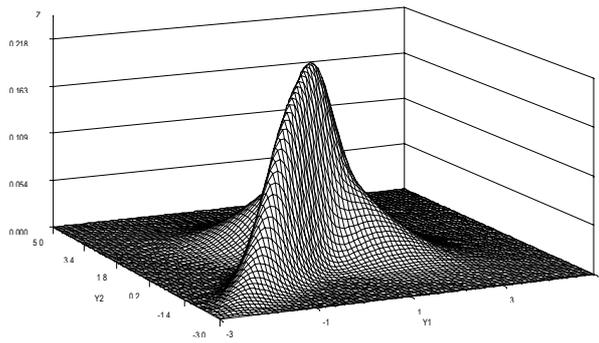
$$\mu_1 = (0,0)', \mu_2 = (2,2)'$$

$$\Sigma_1 = \begin{pmatrix} 1 & 0.6 \\ 0.6 & 1 \end{pmatrix}, \Sigma_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

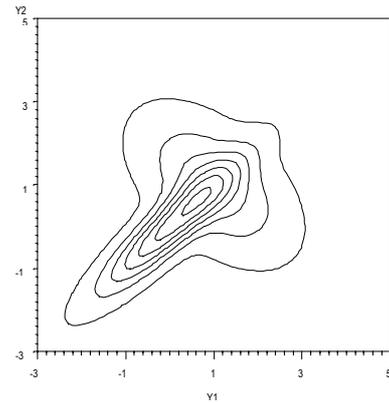
$$p = 0.2$$

$$\mu^* = \begin{pmatrix} 1.6 \\ 1.6 \end{pmatrix}, \Sigma^* = \begin{pmatrix} 1.64 & 0.76 \\ 0.76 & 0.84 \end{pmatrix}$$

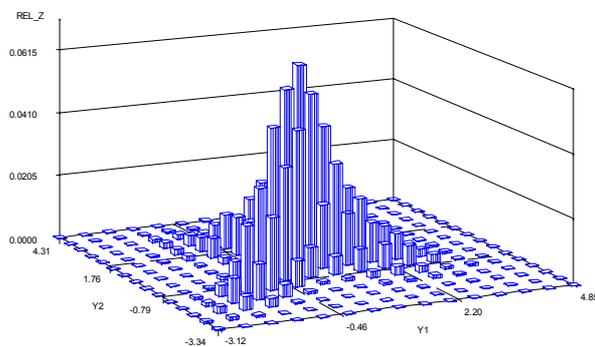
Dichteplot:



Konturplot:



Histogramm (m=10.000 mit %V\_NORMAL):



Parameter:

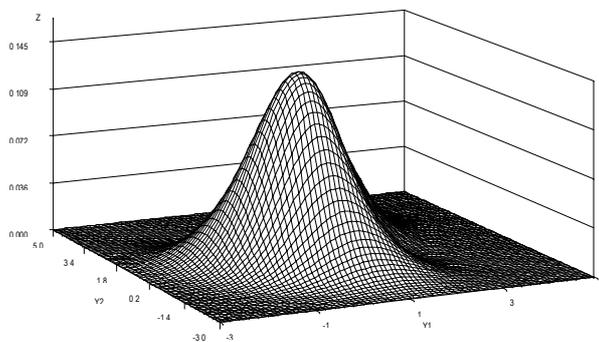
$$\mu_1 = (0,0)', \mu_2 = (1,1)'$$

$$\Sigma_1 = \begin{pmatrix} 1 & 0.9 \\ 0.9 & 1 \end{pmatrix}, \Sigma_2 = \begin{pmatrix} 1 & -0.5 \\ -0.5 & 1 \end{pmatrix}$$

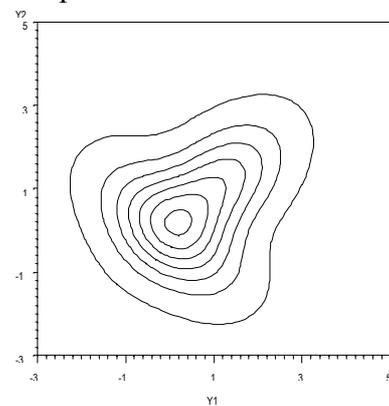
$$p = 0.5$$

$$\mu^* = \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}, \Sigma^* = \begin{pmatrix} 1.25 & 0.45 \\ 0.45 & 1.25 \end{pmatrix}$$

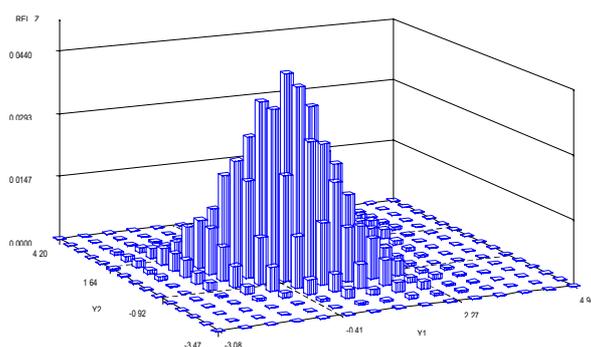
Dichteplot:



Konturplot:



Histogramm (m=10.000 mit %V\_NORMAL):



Parameter:

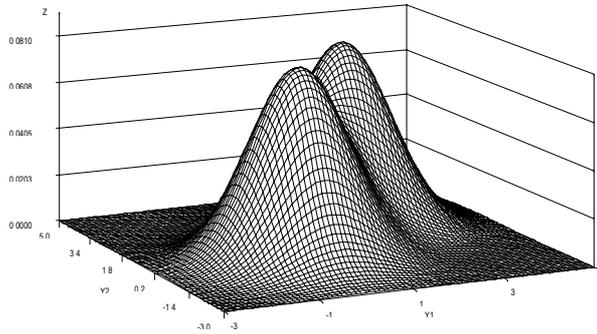
$$\mu_1 = (0,0)', \mu_2 = (1,1)'$$

$$\Sigma_1 = \begin{pmatrix} 1 & -0.5 \\ -0.5 & 1 \end{pmatrix}, \Sigma_2 = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}$$

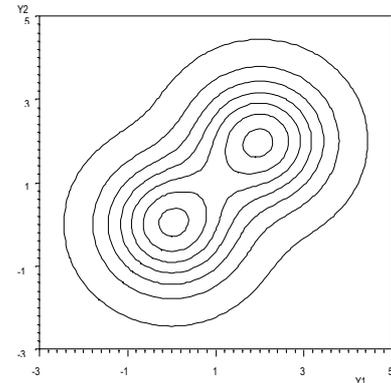
$$p = 0.5$$

$$\mu^* = \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}, \Sigma^* = \begin{pmatrix} 1.25 & 0.25 \\ 0.25 & 1.25 \end{pmatrix}$$

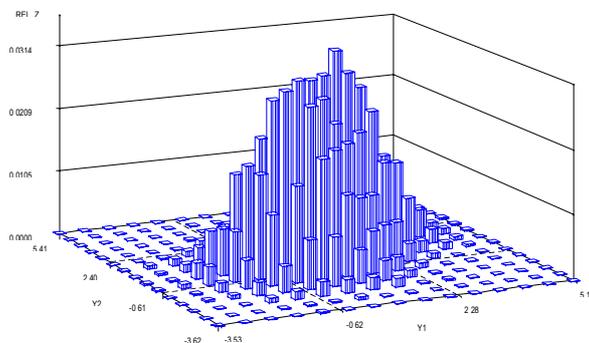
Dichteplot:



Konturplot:



Histogramm (m=10.000 mit %V\_NORMAL):



Parameter:

$$\mu_1 = (0,0)', \mu_2 = (2,2)'$$

$$\Sigma_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \Sigma_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$p = 0.5$$

$$\mu^* = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \Sigma^* = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$$

## Johnsons Translationssystem

Sei  $\underline{X} = (\underline{x}_1, \dots, \underline{x}_n)' \sim N_n(\mu, \Sigma)$ . Wendet man auf die Komponenten von  $\underline{X}$  eine der folgenden Transformationen (Tabelle 1)

Tabelle 1: Transformationen in Johnsons Translationssystem ( $a_1, a_2 \in \mathbb{R}$ ,  $i = 1, \dots, n$ )

Transformation $T_j(\underline{x}_i)$ ; $j \in \{N, L, U, B\}$	Bezeichnung	Kurzbez.	Träger
$\underline{y}_i = T_N(\underline{x}_i) = \underline{x}_i$	Normalverteilung	JS_N	$(-\infty, \infty)$
$\underline{y}_i = T_L(\underline{x}_i) = a_1 \exp(\underline{x}_i) + a_2$	Lognormalverteilung	JS_L	$(a_2, \infty)$
$\underline{y}_i = T_U(\underline{x}_i) = a_1 \sinh(\underline{x}_i) + a_2$	Sinh <sup>-1</sup> -Normalverteilung	JS_U	$(-\infty, \infty)$
$\underline{y}_i = T_B(\underline{x}_i) = a_1 (1 + \exp(\underline{x}_i))^{-1} + a_2$	Logit-Normalverteilung	JS_B	$(a_2, a_2 + a_1)$

an, so hat der resultierende Vektor  $\underline{Y} = (\underline{y}_1, \dots, \underline{y}_n)'$  eine Verteilung im multivariaten Johnsonsystem.

Die Parameter  $a_1, a_2 \in \mathbb{R}$ ,  $i=1, \dots, n$ , wurden zur Kontrolle als Skalen- und Lokationsparameter der  $\underline{y}_i$  eingeführt. Die einfache Form der Transformationsfunktionen führt sofort und sehr einfach zum Erzeugungsalgorithmus für Verteilungen des multivariaten Johnsonsystems.

Hierbei ist noch zu erwähnen, daß auf jede Komponente von  $\underline{X}$  eine beliebige Transformation  $T_j$  aus der obigen Tabelle angewendet werden kann.

### Algorithmus III:

1. Erzeugung von  $\underline{X} = X$  gemäß  $\underline{X} = (\underline{x}_1, \dots, \underline{x}_n)' \sim N_n(\underline{\mu}, \Sigma)$
2. Transformation der Komponenten von  $X$ :  
 $Y = (y_1, \dots, y_n)'$  mit  $y_i = T_j(x_i)$ ;  $j \in \{N, L, U, B\}$ ,  $i=1, \dots, n$

Die einzige Schwierigkeit bei der praktischen Anwendung des Johnsonsystems besteht in der Spezifikation der Verteilungsparameter (siehe Analytische Aspekte, Beispiele für zweidimensionale Johnsonverteilungen).

### Analytische Aspekte

Für Simulationsuntersuchungen ist es günstig, wenn man aus Parametern einer Johnsonverteilung die Parameter der Ausgangsnormalverteilung bestimmen kann, auf die die entsprechende Transformation anzuwenden ist.

Sei  $\underline{X} = (\underline{x}_1, \dots, \underline{x}_n)' \sim N_n(\underline{\mu}, \Sigma)$   $n$ -variater normalverteilt mit  $E(\underline{x}_i) = \mu_i$ ,  $\text{Var}(\underline{x}_i) = \sigma_i^2$ ,  $\text{Cov}(\underline{x}_i, \underline{x}_j) = \sigma_{ij}$ ,  $i \neq j$ ,  $\text{Corr}(\underline{x}_i, \underline{x}_j) = \rho_{ij}$  und sei  $\underline{Y} = (\underline{y}_1, \dots, \underline{y}_n)'$  mit einer Verteilung aus Johnsons Translationssystem, d.h.  $\underline{y}_i = T_j(\underline{x}_i)$  mit einer Transformation aus Tabelle 1 und bezeichne  $E(\underline{y}_i) = \mu_i^*$ ,  $\text{Var}(\underline{y}_i) = \sigma_i^{*2}$ ,  $\text{Corr}(\underline{y}_i, \underline{y}_j) = \rho_{ij}^*$ ,  $i, j=1, \dots, n$ , dann gilt (Johnson, 1987):

Falls  $\underline{Y} = (\underline{y}_1, \dots, \underline{y}_n)'$  mit  $\underline{y}_i = \sinh(\underline{x}_i)$ ,  $i=1, \dots, n$ , (d.h.  $\sinh^{-1}$ -normalverteilt) erhält man für die Parameter dieser Johnsonverteilungen:

$$\begin{aligned} \mu_i^* &= \exp(\sigma_i^2 / 2) \cdot \sinh(\mu_i), \\ \sigma_i^{*2} &= (\exp(\sigma_i^2) - 1) \cdot (1 + \cosh(2\mu_i) \cdot \exp(\sigma_i^2)) / 2, \\ \rho_{ij}^* &= (\sigma_i^* \sigma_j^*)^{-1} \exp((\sigma_i^2 + \sigma_j^2) / 2) \cdot (\exp(\rho_{ij} \sigma_i \sigma_j) \cdot (\cosh(\mu_i + \mu_j) / 2) + \\ &\quad - \exp(-\rho_{ij} \sigma_i \sigma_j) \cdot (\cosh(\mu_i - \mu_j) / 2) - \sinh(\mu_i) \cdot \sinh(\mu_j)) \end{aligned}$$

Mit  $\underline{z}_i = a_1 \cdot \frac{\underline{y}_i - \mu_i^*}{\sigma_i^*} + a_2$  hat man eine  $\sinh^{-1}$ -normalverteilte Zufallsgröße mit Mittelwert  $a_2$  und Varianz  $a_1^2$ . Außerdem kann man

$$\rho_{ij} = \frac{1}{\sigma_i \sigma_j} \cdot \ln \left[ \frac{B + B^2 + \cosh(\mu_i + \mu_j) \cdot \cosh(\mu_i - \mu_j)}{\cosh(\mu_i + \mu_j)} \right], \text{ mit}$$

$$B = \sinh(\mu_i) \cdot \sinh(\mu_j) + \rho_{ij}^* \cdot \exp(-(\sigma_i^2 + \sigma_j^2) / 2)$$

berechnen.

Dabei ist zu testen, ob alle  $|\rho_{ij}| \leq 1$  und im Falle von  $n > 2$  ist auch die positive Definitheit der Kovarianzmatrix  $\Sigma$  zu prüfen.

Falls  $\underline{Y} = (\underline{y}_1, \dots, \underline{y}_n)'$  mit  $\underline{y}_i = \exp(\underline{x}_i)$ ,  $i = 1, \dots, n$  (d.h. lognormalverteilt) erhält man mit  $\mu = 0$  für die Parameter dieser Johnsonverteilungen:

$$\begin{aligned}\mu_i^* &= \exp(\sigma_i^2 / 2) \\ \sigma_i^{*2} &= (\exp(2\sigma_i^2) - \exp(\sigma_i^2)) \\ \rho_{ij}^* &= \frac{\exp(\rho_{ij}\sigma_i\sigma_j) - 1}{(\exp(\sigma_i^2) - 1)^{1/2} \cdot (\exp(\sigma_j^2) - 1)^{1/2}}\end{aligned}$$

Mit  $\underline{z}_i = a1_i \cdot \frac{\underline{y}_i - \mu_i^*}{\sigma_i^*} + a2_i$  hat man eine lognormalverteilte Zufallsgröße mit Mittelwert  $a2_i$  und Varianz  $a1_i^2$ . Außerdem kann man

$$\rho_{ij} = \frac{1}{\sigma_i\sigma_j} \cdot \ln \left[ 1 + \rho_{ij}^* \cdot (\exp(\sigma_i^2) - 1)^{1/2} \cdot (\exp(\sigma_j^2) - 1)^{1/2} \right]$$

berechnen. Dabei ist ebenfalls zu testen, ob alle  $|\rho_{ij}| \leq 1$  und im Falle von  $n > 2$  ist auch die positive Definitheit der Kovarianzmatrix  $\Sigma$  zu prüfen.

Eine sehr breite Anwendung spielt Johnsons Translationssystem bei der Beurteilung von Diskriminanzanalyseverfahren bezüglich einer Abweichung von der Voraussetzung der multivariaten Normalverteilung.

### SAS-Makro %JOHNSON

Der Algorithmus III wurde als SAS-Makro %JOHNSON unter Verwendung der Prozedur `proc iml` umgesetzt. Voraussetzung für den Aufruf des Makros sind die SAS-Dateien **PARA** (enthält Erwartungswertvektor `mue` und den Vektor der Standardabweichungen `sigma` der Ausgangsnormalverteilung sowie die Vektoren der Transformationskoeffizienten `a1` und `a2`), **CORR** (Korrelationskoeffizientenmatrix der Ausgangsnormalverteilung), **TRANS** (enthält Charactervariable `TRANSFOR` für die Festlegung der Transformation der Komponenten der Ausgangsnormalverteilung), deren Erzeugung im zweidimensionalen Fall z.B. folgende Gestalt haben könnte.

```
data PARA;
input  mue sigma a1 a2;
cards;
  0 0.5  1  0
  0 0.5  1  0
run;

data CORR;
input  rho1-rho2;
cards;
  1    0.0
  0.0  1
run;

data TRANS;
input TRANSFOR $;
cards;
  B
  B
run;
```

**SAS-Makro:**

```

/*-----*
| %MACRO JOHNSON(M,PARA,CORR,TRANS,DATA,NAMES) |
| ===== |
| erzeugt n-dimensionale Merkmalsvektoren einer Johnsonverteilung |
| Y=(Y(1),...,Y(n))' durch Transformation einer n-dimensionalen Normal- |
| verteilung X=(X(1),...,X(n))' ~ NV(mue,COV) , mit Y(j)=T(X(j)): |
| Y[j]=X[j]; TRANSFOR='N' = NORMAL |
| Y[j]=a1[j]*exp(X[j])+a2[j]; TRANSFOR='L' = LOGNORMAL |
| Y[j]=a1[j]*sinh(X[j])+a2[j]; TRANSFOR='U' = SINH^-1-NORMAL |
| Y[j]=a1[j]/(1+exp(X[j]))+a2[j]; TRANSFOR='B' = LOGIT-NORMAL |
| ----- |
| INPUT: |
| ===== |
| M Anzahl der Simulationswiederholungen |
| PARA Parameterdatei - SAS-Datei der Form |
| mue(1) sigma(1) a1(1) a2(1) |
| . . . . |
| . . . . |
| mue(n) sigma(n) a1(n) a2(n) |
| CORR Korrelationsmatrix - SAS-Datei in der Form |
| 1 RHO(1,2) ... RHO(1,n) |
| . . . . |
| . . . . |
| RHO(n,1) ... RHO(n,n-1) 1 |
| TRANS Transformationsdatei - SAS-Datei der Form (CHAR) |
| TRANS(1) L |
| . z.B.: N |
| . U |
| TRANS(n) L |
| OUTPUT: |
| ===== |
| DATA temporäre SAS-Datei, mit der Datensatzstruktur: |
| Y1 Y2 ... Yn |
| NAMES Namen der Variablen der Datei 'DATA' |
| z.B. bei n=4 in der Form: |
| {'Y1' 'Y2' 'Y3' 'Y4'} |
| -----*

```

```

%MACRO JOHNSON(M, PARA, CORR, TRANS, DATA, NAMES);
proc iml;
  use &PARA; read all into PAR;
  use &CORR; read all into COR;
  use &TRANS; read all VAR _CHAR_ into JT;
  n=ncol(COR);
  mue=j(n,1,0);
  s=j(n,1,0);
  sg=j(n,1,0);
  a1=j(n,1,0);
  a2=j(n,1,0);
  COV=j(n,n,0);
  A=j(n,n,0);
  DAT=j(&m,n,0);
  Do i=1 to n;
    mue[i]=PAR[i,1] ;
    s[i]=PAR[i,2] ;
    sg[i]=s[i]*s[i];
    a1[i]=PAR[i,3] ;
    a2[i]=PAR[i,4] ;
  end;
  Do i=1 to n;
  Do j=1 to n;
    COV[i,j]=COR[i,j]*s[i]*s[j];
  end;
  end;
  print mue s sg a1 a2 ;
  print cor cov JT ;
  A=root(cov); /* Choleskizerlegung */
  A=A`; /* der Kovarianzmatrix */
  Do i=1 to &M;
    Z=j(n,1,0);
    Z=NORMAL(Z); /* N(0,I) */
    Y=A*Z+mue; /* N(mu,COV) */
    Do j=1 to n;
      IF JT[j]='N' then Y[j]=Y[j]; /* N = NORMAL */
      IF JT[j]='L' then Y[j]=a1[j]*exp(Y[j])+a2[j]; /* L = LOGNORMAL */
      IF JT[j]='U' then Y[j]=a1[j]*sinh(Y[j])+a2[j]; /* U = SINH^-1-NORMAL */
      IF JT[j]='B' then Y[j]=a1[j]/(1+exp(-Y[j]))+a2[j]; /* B = LOGIT-NORMAL */
    end;
  end;
  c=&names;
  create &Data from DAT [colname=c]; /* Ausgabe der n Zufallsvektoren */
  append from DAT; /* als SAS-Datei */
quit;
%mend JOHNSON;

```

**Aufruf des Makros:**

```
%JOHNSON(10000,PARA,CORR,TRANS,DATEN,{'Y1' 'Y2'});
```

Mit diesem Aufruf erzeugt man **10.000** Pseudozufallszahlenpaare der entsprechenden zweidimensionalen Johnsonverteilung, die in die temporäre SAS-Datei **DATEN** mit den Merkmalsbezeichnungen **Y1** und **Y2** geschrieben werden.

**Beispiele für zweidimensionale Verteilungen in Johnsons Translations-system**

Zur Veranschaulichung werden Beispiele für zweidimensionale Johnsonverteilungen gegeben, um den Einfluß der Wahl der Parameter zu demonstrieren. Entsprechende Beispielprogramme zur Erzeugung der Dichte- und Konturplots sind im Anhang zu finden.

**Lognormalverteilungen JS\_LL**

Die bivariaten Lognormalverteilungen mit Erwartungswerten 0 und Varianzen 1, die in den folgenden Abbildungen beschrieben werden, haben die Dichtefunktion

$$f(x_1, x_2) = \frac{b_1 b_2}{2\pi\sigma_1\sigma_2(b_1x_1 + c_1)(b_2x_2 + c_2)(1-\rho^2)^{1/2}} \exp\left(-\frac{(y_1/\sigma_1)^2 - 2\rho y_1 y_2 / (\sigma_1\sigma_2) + (y_2/\sigma_1)^2}{2(1-\rho^2)}\right)$$

für  $x_i > -c_i/b_i$ ,  $i=1,2$ ,

wobei  $y_i = \ln(b_i x_i + c_i)$ ,

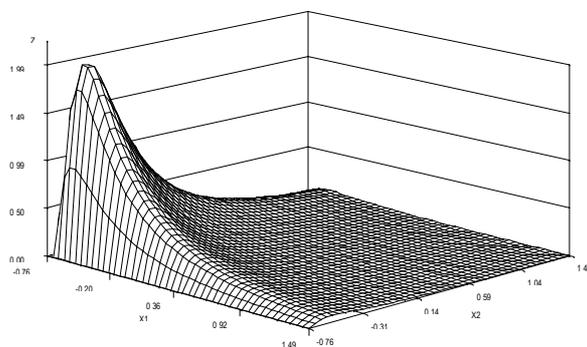
$c_i = \exp(\sigma_i^2/2)$ ,

$b_i = (\exp(2\sigma_i^2) - \exp(\sigma_i^2))^{1/2}$ , d.h.

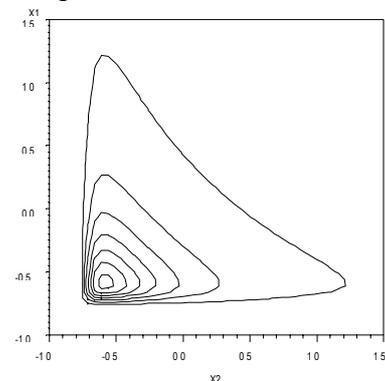
die Transformation von Tabelle 1  $y_i = T_L(x_i) = a1_i \exp(x_i) + a2_i$  hat die Koeffizienten

$$a1_i = \frac{1}{b_i} \text{ und } a2_i = -\frac{c_i}{b_i}, i=1,2.$$

Dichteplot:



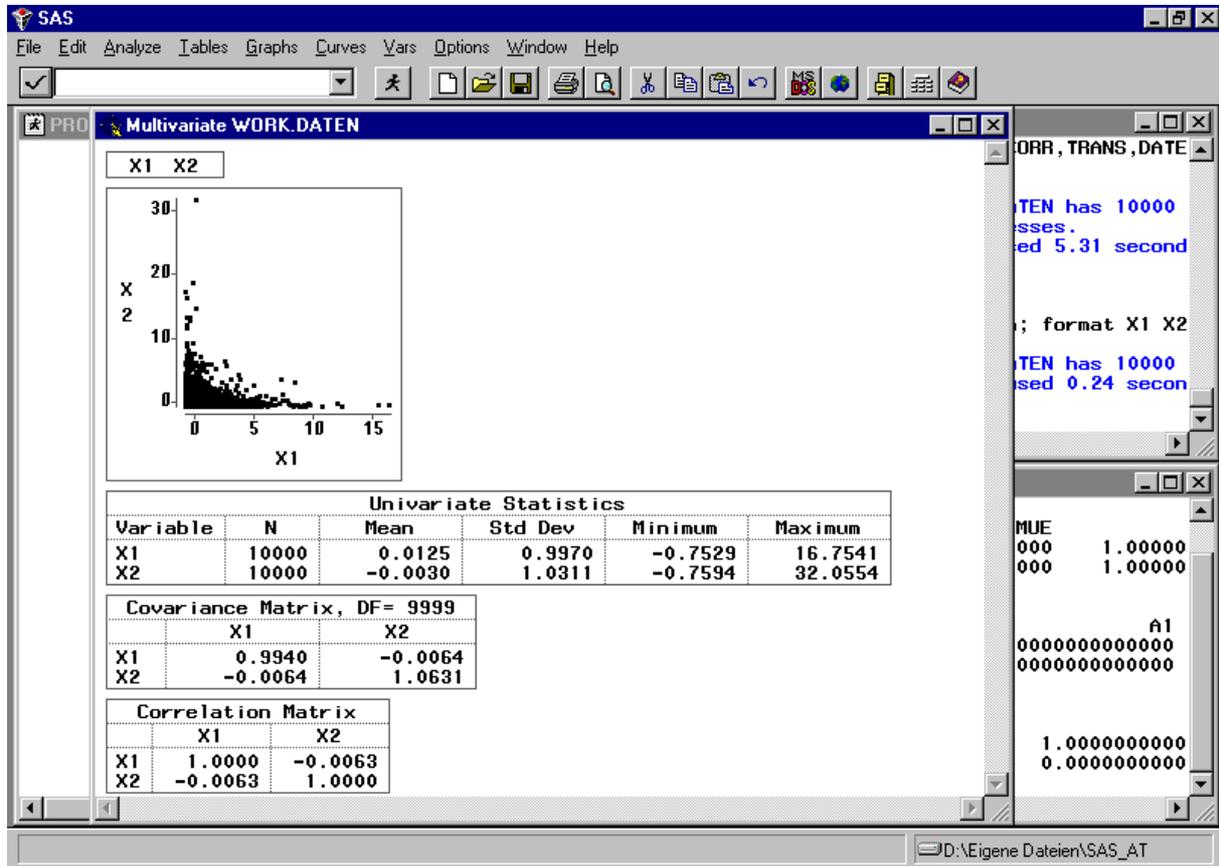
Konturplot:



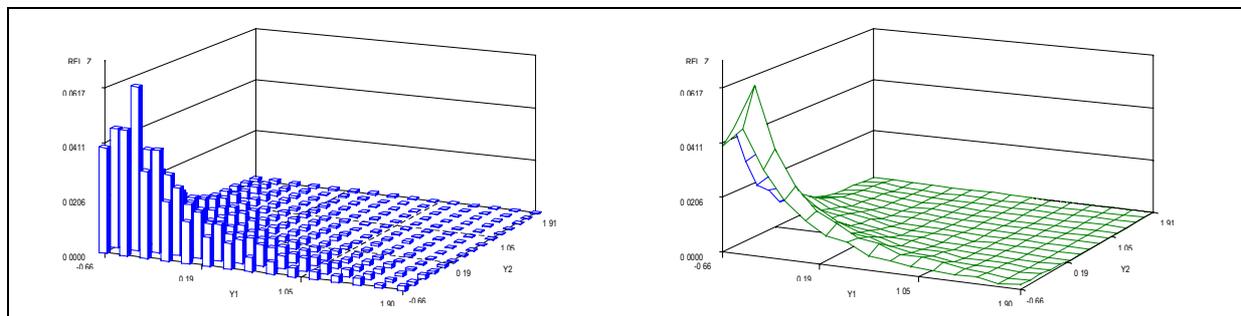
Parameter:  $\mu_1 = 0, \mu_2 = 0, \sigma_1 = 1, \sigma_2 = 1, \rho = 0$

Der Aufruf des Makros %JOHNSON(10000, PARA, CORR, TRANS, DATEN, {'X1' 'X2'}); mit den obigen Parametern müßte dann 10.000 Pseudozufallszahlenpaare einer Johnsonverteilung mit den Parametern  $\mu_1^* = 0, \mu_2^* = 0, \sigma_1^* = 1, \sigma_2^* = 1, \rho^* = 0$  realisieren.

Das Ergebnis ist leicht mit SAS/INSIGHT zu überprüfen:

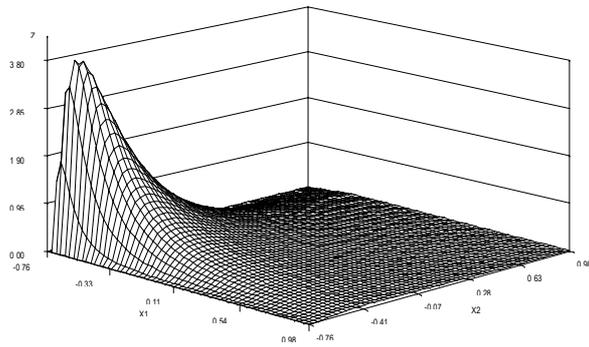


Das Histogramm als Scatter- und als Gridplot mit dem Makro %HIST (Anlage) hat dann folgende Gestalt:

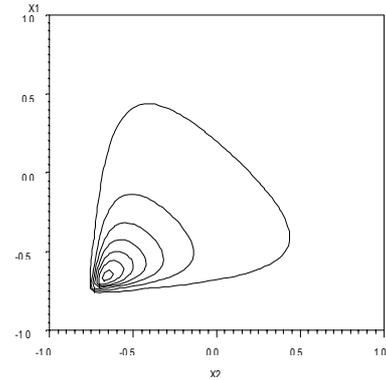


Weitere Beispiele für Dichte- und Konturplots sind:

Dichteplot:

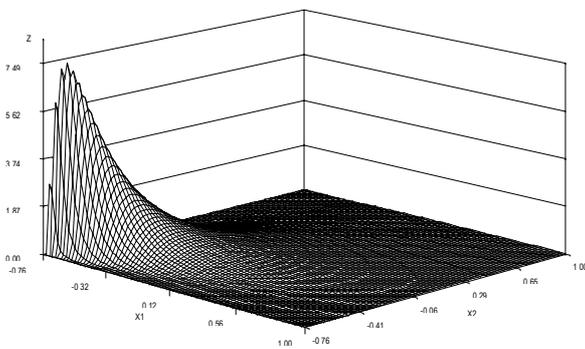


Konturplot:

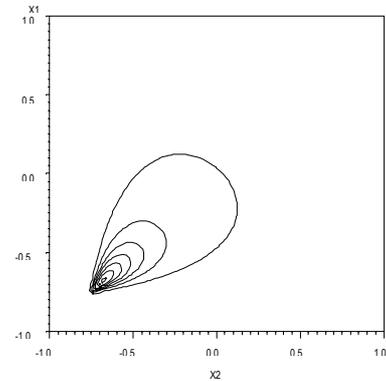


Parameter:  $\mu_1 = 0, \mu_2 = 0, \sigma_1 = 1, \sigma_2 = 1, \rho = 0.5$

Dichteplot:

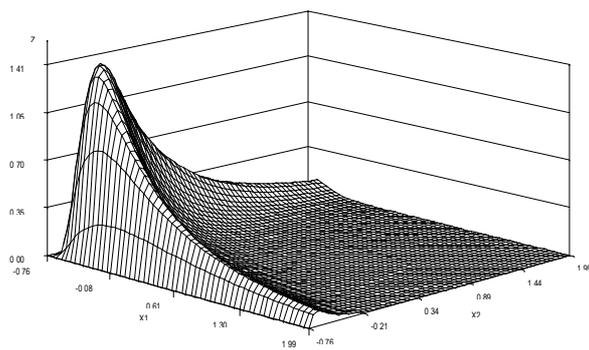


Konturplot:

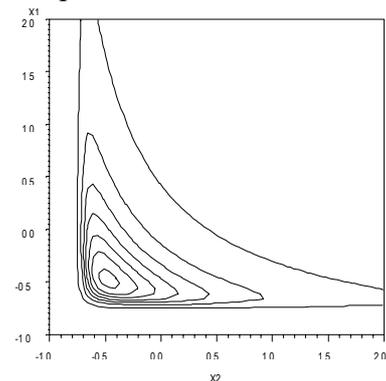


Parameter:  $\mu_1 = 0, \mu_2 = 0, \sigma_1 = 1, \sigma_2 = 1, \rho = 0.8$

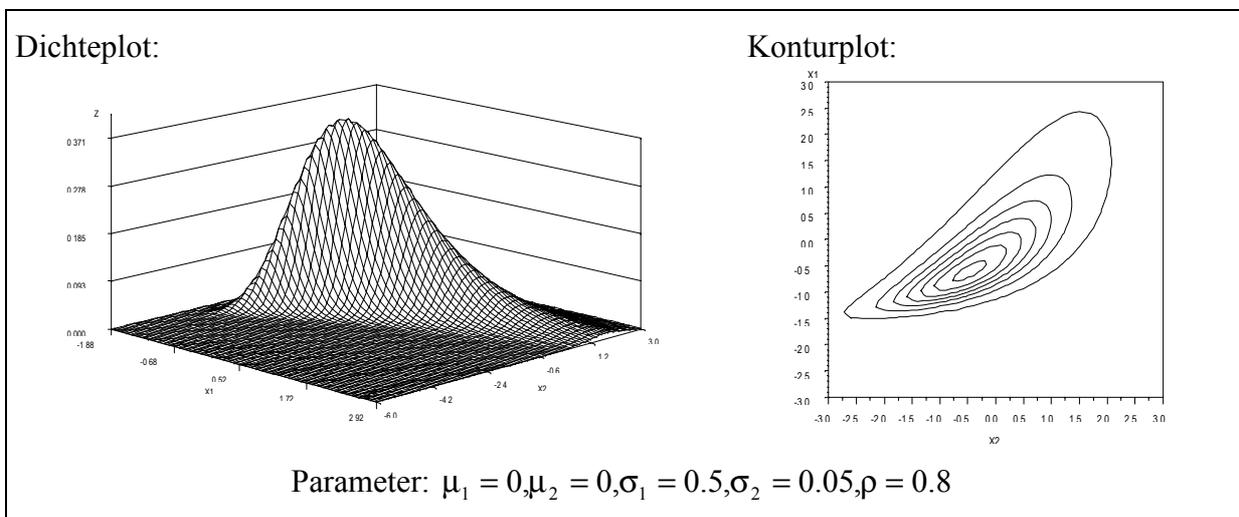
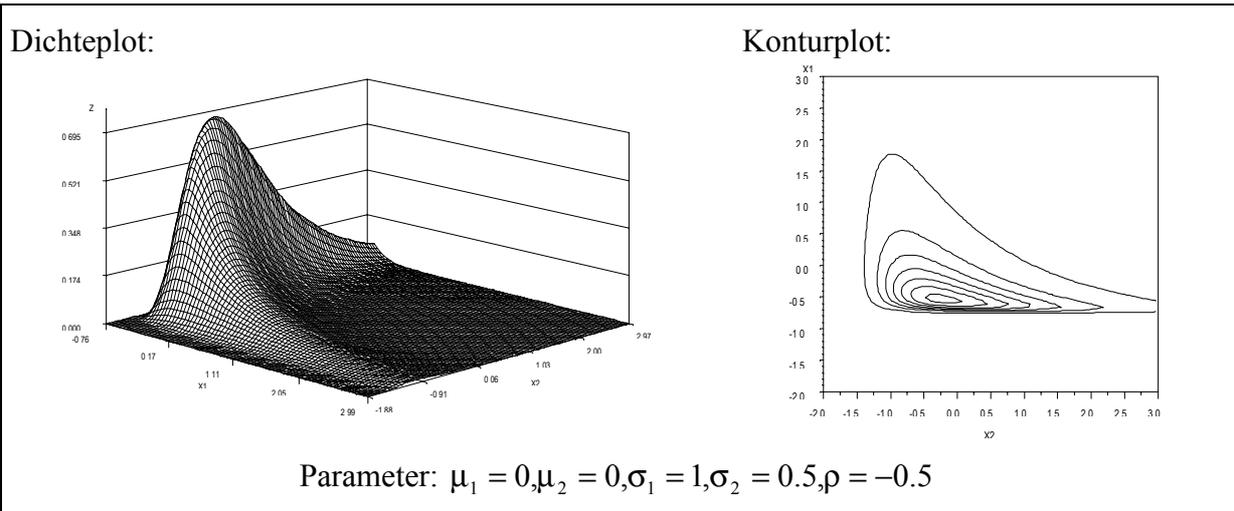
Dichteplot:



Konturplot:



Parameter:  $\mu_1 = 0, \mu_2 = 0, \sigma_1 = 1, \sigma_2 = 1, \rho = -0.5$



### $\text{Sinh}^{-1}$ -Normalverteilung JS\_UU

Die bivariaten  $\text{Sinh}^{-1}$ -Normalverteilungen mit Erwartungswerten 0 und Varianzen 1, die in den folgenden Abbildungen beschrieben werden, haben die Dichtefunktion

$$f(x_1, x_2) = \frac{b_1 b_2 (w_1 + (1 + w_1^2)^{1/2})(w_2 + (1 + w_2^2)^{1/2})}{(1 + w_1^2 + w_1(1 + w_1^2)^{1/2})(1 + w_2^2 + w_2(1 + w_2^2)^{1/2})} \cdot g(\sinh^{-1}(w_1), \sinh^{-1}(w_2)),$$

wobei  $w_i = b_i x_i + c_i$ ,

$$c_i = \exp(\sigma_i^2/2) \sinh(\mu_i),$$

$$b_i = ((\exp(\sigma_i^2) - 1)(\exp(\sigma_i^2) \cosh(2\mu_i) + 1))^{1/2}, \quad i = 1, 2 \text{ sowie}$$

$$g(y_1, y_2) = \frac{\exp(-(y_1^2 - 2y_1 y_2 + y_2^2)/(2(1 - \rho^2)))}{2\sigma_1 \sigma_2 \sqrt{1 - \rho^2}} \text{ mit}$$

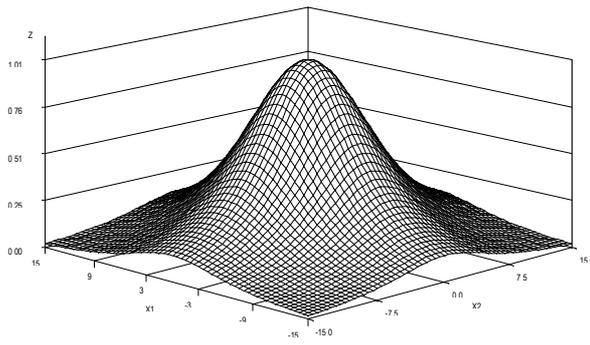
$$y_i = \sinh^{-1}(w_i) = \ln(w_i(1 + w_i^2)^{1/2}), \quad i = 1, 2.$$

Die Transformation von Tabelle 1  $y_i = T_L(x_i) = a_1 \sinh(x_i) + a_2$  hat dementsprechend die Koeffizienten

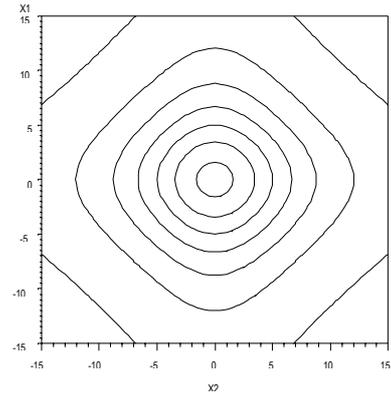
$$a_1 = \frac{\sqrt{2}}{b_i} \text{ und } a_2 = -\frac{\sqrt{2}c_i}{b_i}, \quad i = 1, 2.$$

Beispiele für wesentliche Dichte- und Konturplots sind:

Dichteplot:

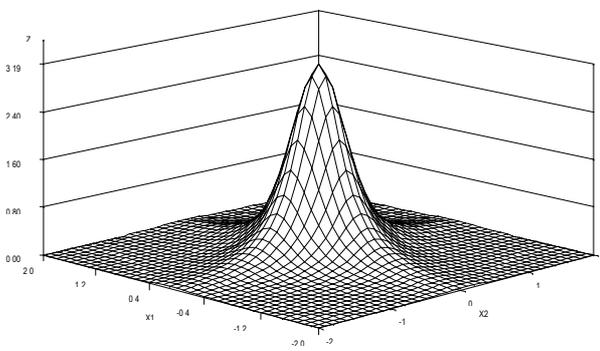


Konturplot:

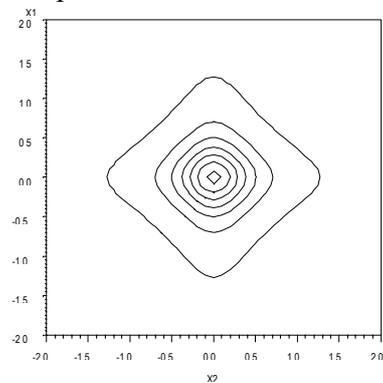


Parameter:  $\mu_1 = 0, \mu_2 = 0, \sigma_1 = 0.1, \sigma_2 = 0.1, \rho = 0.0$

Dichteplot:

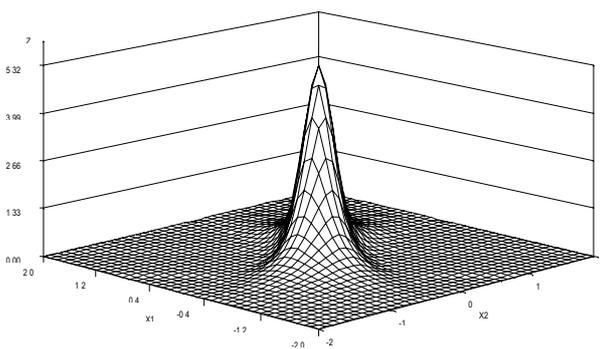


Konturplot:

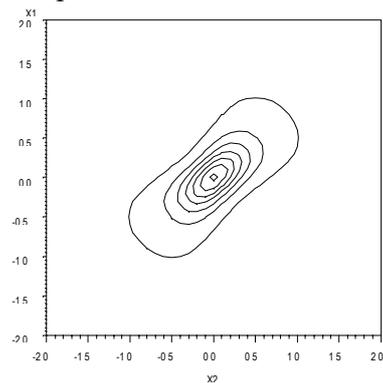


Parameter:  $\mu_1 = 0, \mu_2 = 0, \sigma_1 = 1, \sigma_2 = 1, \rho = 0.0$

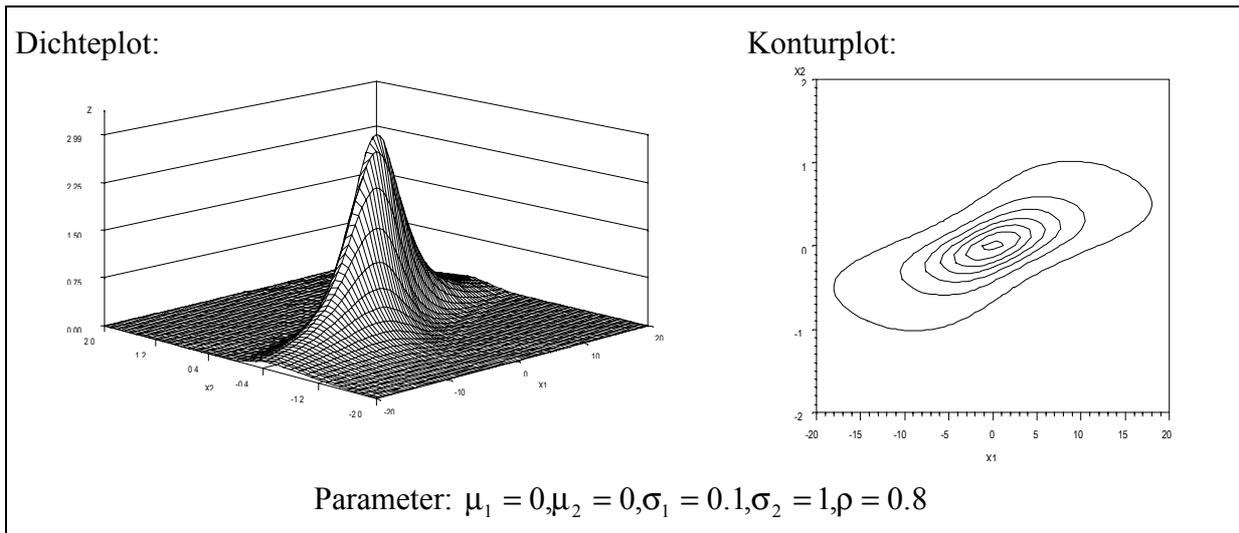
Dichteplot:



Konturplot:



Parameter:  $\mu_1 = 0, \mu_2 = 0, \sigma_1 = 1, \sigma_2 = 1, \rho = 0.8$

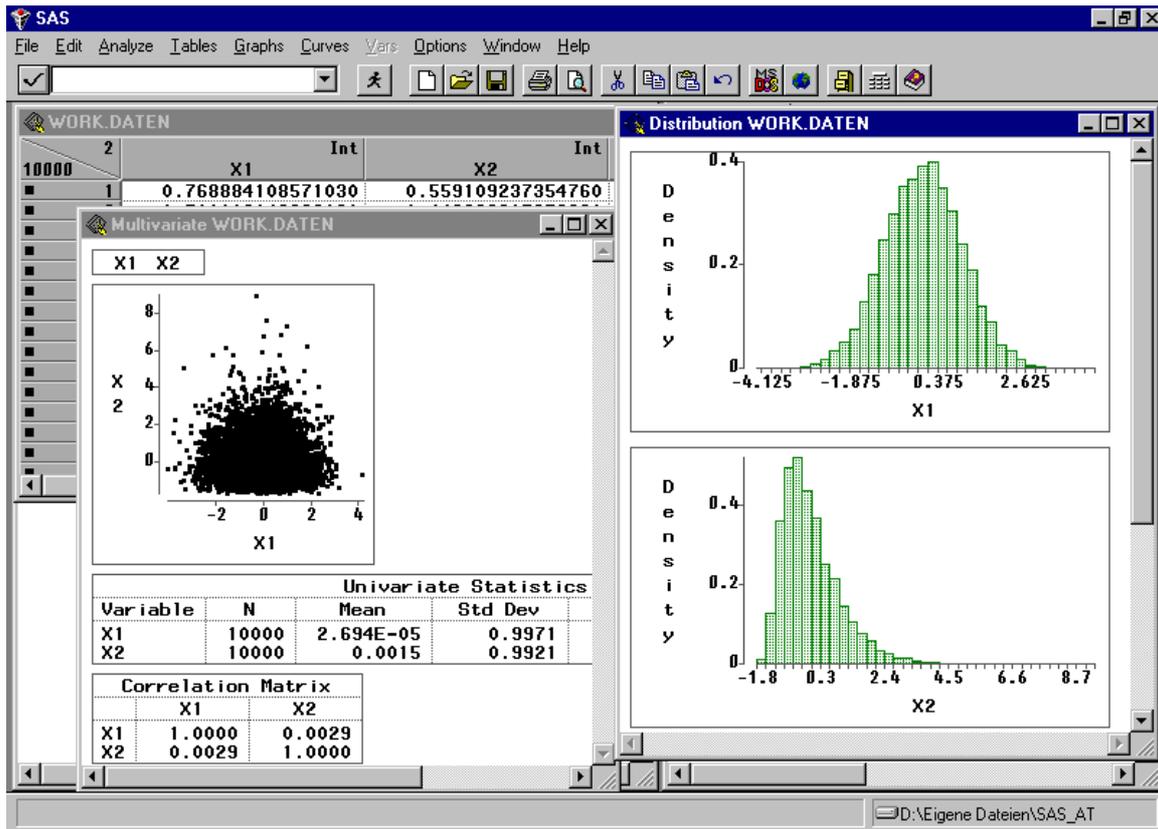


Die wesentlichen unterschiedlichen Fälle der  $\text{Sinh}^{-1}$ -Normalverteilung sind in Tabelle 2 zusammengestellt.

Tabelle 2: Wesentliche unterschiedliche Fälle der  $\text{Sinh}^{-1}$ -Normalverteilung in Abhängigkeit von den Parametern  $\mu_i$  und  $\sigma_i$  der Ausgangsnormalverteilungen

1.	$\mu_i = 0, \sigma_i = 0.1$	JS_UU symmetrisch, nahezu normal
2.	$\mu_i = 0, \sigma_i = 1$	JS_UU symmetrisch, aber stärker tailliert als Normalverteilung
3.	$\mu_i = 2, \sigma_i = 0.5$	JS_UU etwas schief
4.	$\mu_i = 2, \sigma_i = 1$	JS_UU sehr schief

Die Schiefe der JS\_UU für die Parameter  $\mu_1 = 0, \mu_2 = 2, \sigma_1 = 0.1, \sigma_2 = 0.5, \rho = 0$  kann man auch leicht durch 10.000 mit dem Makro %Johnson erzeugte Zufallszahlenpaare und SAS/INSIHT demonstrieren:



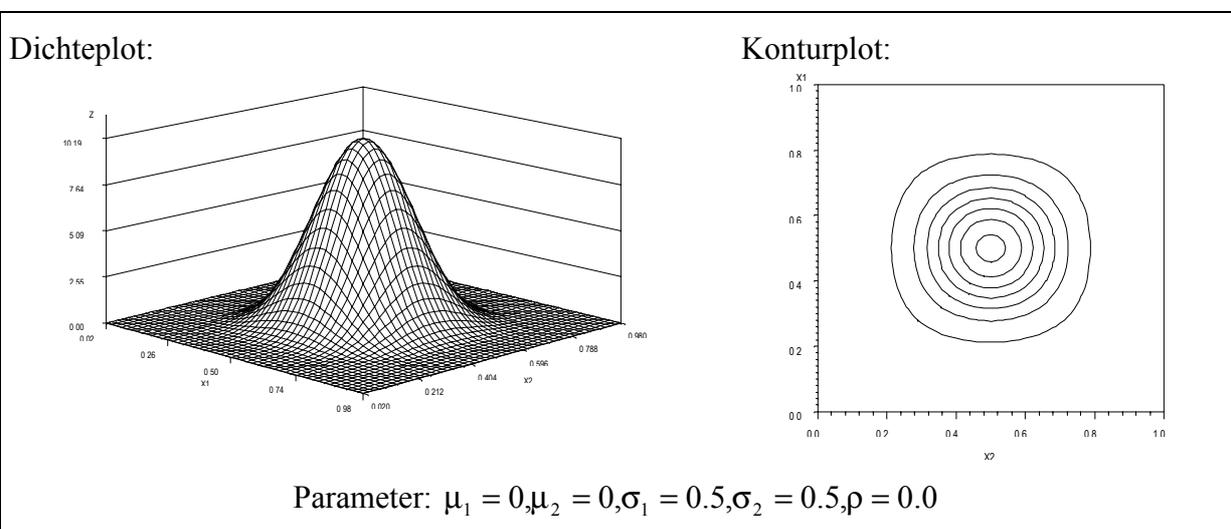
### Logit-Normalverteilungen JS\_BB

Die bivariaten Logit-Normalverteilungen haben keine standardisierte Form der Dichte mit Erwartungswerten 0 und Varianzen 1. Damit der Träger der  $\underline{y}_i$  das Einheitsintervall ist, wählen wir die Transformation von Tabelle 1  $y_i = T_B(x_i) = a_{1i}(1 + \exp(x_i))^{-1} + a_{2i}$  mit den Koeffizienten  $a_{1i} = 1$  und  $a_{2i} = 0$ ,  $i=1,2$ .

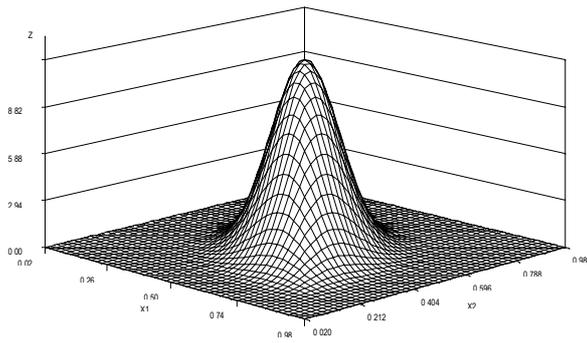
In den folgenden Abbildungen werden die resultierenden Dichtefunktion

$$f(x_1, x_2) = \frac{\exp(-(y_1^2 - 2y_1y_2 + y_2^2)/(2(1-\rho^2)))}{x_1x_2(1-x_1)(1-x_2)2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} \quad \text{mit} \quad y_i = \frac{\ln(x_i/(1-x_i)) - \mu_i}{\sigma_i}, \quad i=1,2.$$

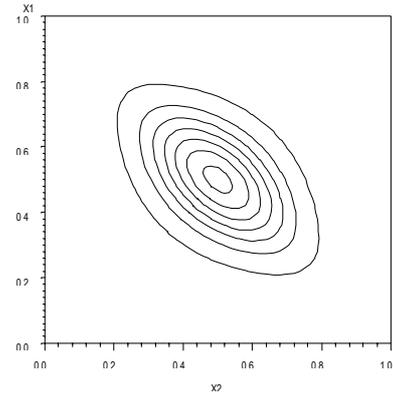
mittels Dichte- und Konturplots beschrieben.



Dichteplot:

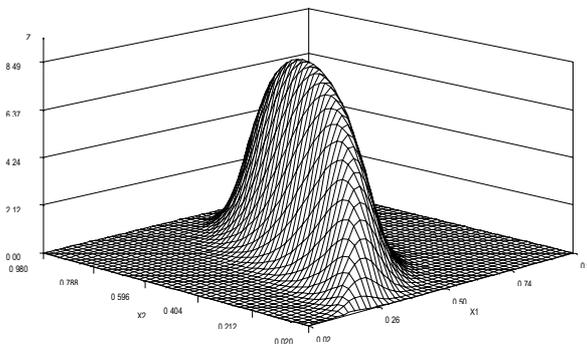


Konturplot:

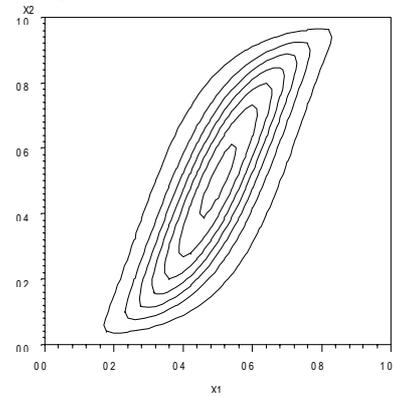


Parameter:  $\mu_1 = 0, \mu_2 = 0, \sigma_1 = 0.5, \sigma_2 = 0.5, \rho = -0.5$

Dichteplot:

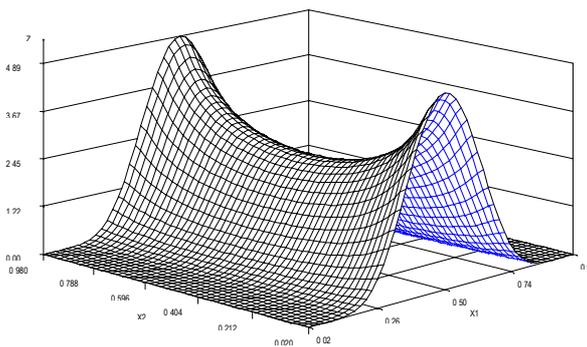


Konturplot:

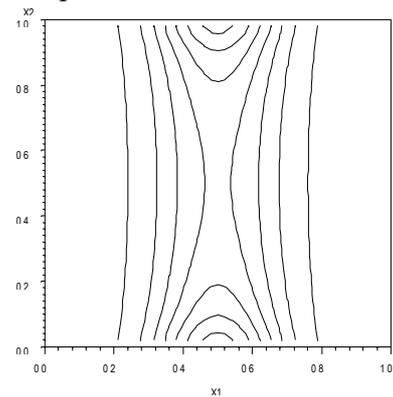


Parameter:  $\mu_1 = 0, \mu_2 = 0, \sigma_1 = 0.5, \sigma_2 = 1, \rho = 0.8$

Dichteplot:

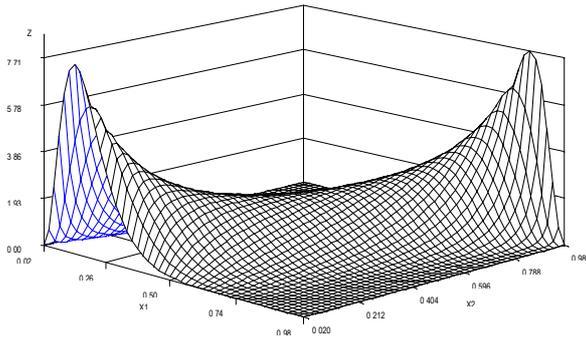


Konturplot:

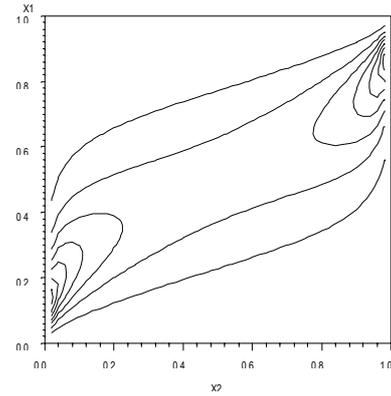


Parameter:  $\mu_1 = 0, \mu_2 = 0, \sigma_1 = 0.5, \sigma_2 = 2, \rho = 0.0$

Dichteplot:

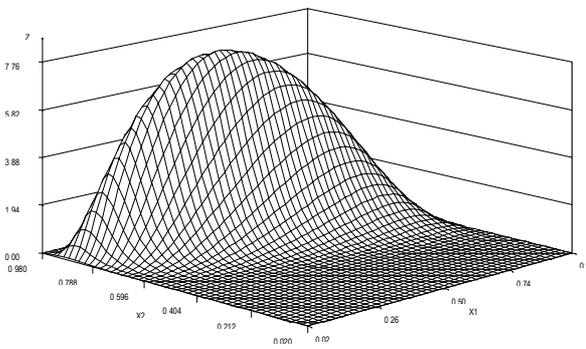


Konturplot:

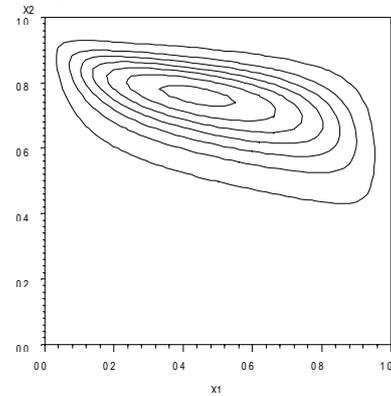


Parameter:  $\mu_1 = 0, \mu_2 = 0, \sigma_1 = 1, \sigma_2 = 2, \rho = 0.8$

Dichteplot:

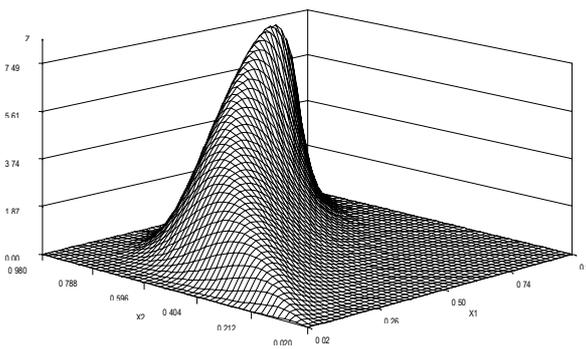


Konturplot:

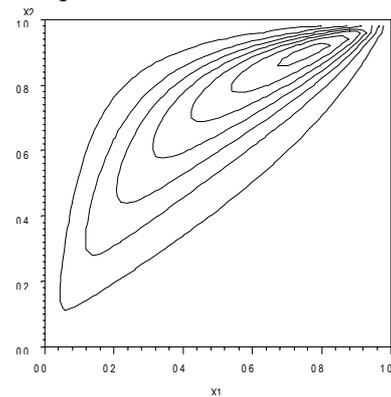


Parameter:  $\mu_1 = 0, \mu_2 = 1, \sigma_1 = 1, \sigma_2 = 0.5, \rho = -0.5$

Dichteplot:

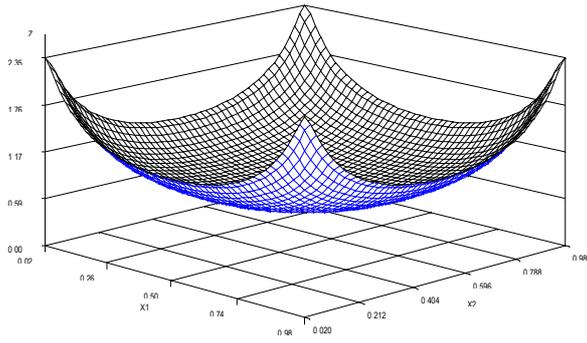


Konturplot:

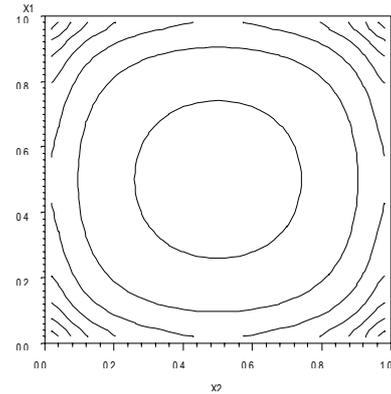


Parameter:  $\mu_1 = 0, \mu_2 = 1, \sigma_1 = 1, \sigma_2 = 1, \rho = 0.8$

Dichteplot:

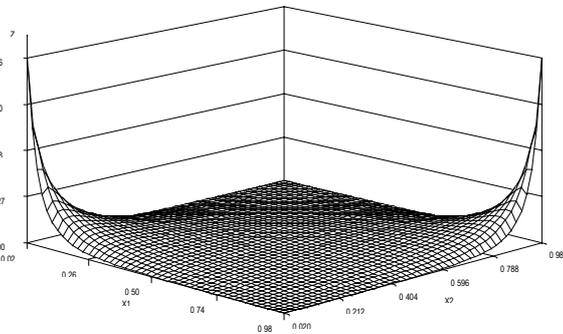


Konturplot:

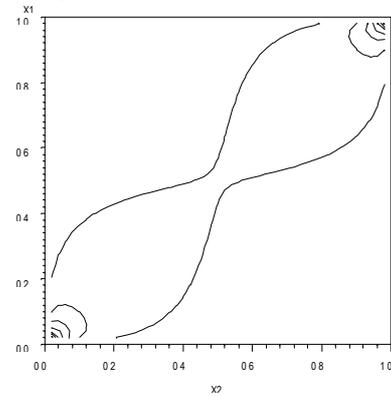


Parameter:  $\mu_1 = 0, \mu_2 = 0, \sigma_1 = 2, \sigma_2 = 2, \rho = 0.0$

Dichteplot:

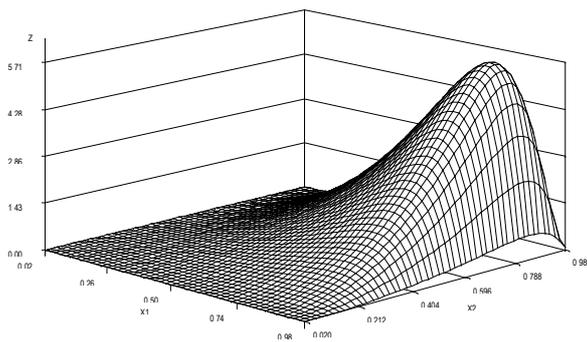


Konturplot:

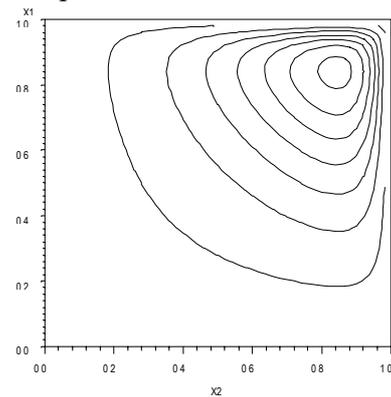


Parameter:  $\mu_1 = 0, \mu_2 = 0, \sigma_1 = 2, \sigma_2 = 2, \rho = 0.8$

Dichteplot:



Konturplot:



Parameter:  $\mu_1 = 1, \mu_2 = 1, \sigma_1 = 1, \sigma_2 = 1, \rho = 0.0$

## Literatur:

- Geißler, H.; Ortseifen, C.: SAS-Makroprogrammierung: Eine Einführung. Universitätsrechenzentrum Heidelberg 1995, 54 S.
- Johnson, M. E.: Multivariate Statistical Simulation. J. Wiley, New York 1987.
- Johnson, N. L.; Kotz, S.: Distributions in Statistics: Continuous Multivariate Distributions. J. Wiley, New York 1972.
- Kleijnen; J.; van Groenendaal, W.: Simulation: A Statistical Perspective. J. Wiley, Chichester 1992, 241 pp.
- Rosenblatt, M.: Remarks on a Multivariate Transformation. Annals of Math. Stat.23 (1952),470-472
- SAS Institute Inc., SAS/IML Software: Usage and Reference, Version 6, First Edition Cary, NC: SAS Institute Inc., 1989, 501 pp.
- SAS Institute Inc., SAS Language: Reference, Version 6, First Edition Cary, NC: SAS Institute Inc., 1990, 1042pp.
- Tuchscherer, A.; Rudolph, P. E.; Jäger, B.; Tuchscherer, M.: Ein SAS-Makro zur Erzeugung multivariat normalverteilter Zufallsgrößen. Konferenzbeiträge der 3. Konferenz für SAS-Benutzer in Forschung und Entwicklung (KSFE), Ruprecht-Karls-Universität Heidelberg 1999, S. 293-306.

## Anhang

### A1 Beispiel Dichteplot und Konturplot - zweidimensionale gemischte Normalverteilungen

```

goptions reset=global gunit=pct border ftext=none htext=3;
axis1
  width=1
  length=50 PCT
  color=BLACK
  order=-3 to 5 by 2
  style=1;
axis2
  width=1
  color=BLACK
  order=-3 to 5 by 2
  style=1;
data snormal;
  do Y1=-3 to 5 by 0.1;
  do y2=-3 to 5 by 0.1;
  pi=3.14159265359;
  mue11=0;
  mue12=0;
  sq11=1;
  sq12=1;
  rho1=0.0;
  mue21=2;
  mue22=2;
  sq21=1;
  sq22=1;
  rho2=0.0;
  p=0.5;
  s11=sqrt(sq11);
  s12=sqrt(sq12);
  a1=1-rho1*rho1;
  b1=(y1-mue11)*(y1-mue11)/sq11;
  c1=2*rho1*(y1-mue11)*(y2-mue12)/(s11*s12);
  d1=(y2-mue12)*(y2-mue12)/sq12;
  z1=exp(-(b1-c1+d1)/(2*a1))/(2*pi*s11*s12*sqrt(a1));
  s21=sqrt(sq21);
  s22=sqrt(sq22);
  a2=1-rho2*rho2;
  b2=(y1-mue21)*(y1-mue21)/sq21;
  c2=2*rho2*(y1-mue21)*(y2-mue22)/(s21*s22);
  d2=(y2-mue22)*(y2-mue22)/sq22;
  z2=exp(-(b2-c2+d2)/(2*a2))/(2*pi*s21*s22*sqrt(a2));
  z=p*z1+(1-p)*z2;
  output;
  end;
end;
run;

proc g3d data=snormal;
  plot y2*y1=z /
    rotate=25
    caxis=black
    ctop=black
    cbottom=blue
    ctext=black
    xtcknum=5
    ytcknum=6

```

```
        zticknum=5
        zmin=0
        grid
    ;
run;

PROC GCONTOUR data=snormal;
    PLOT Y2*Y1=z /
        HAXIS=Axis1
        VAXIS=Axis2
        NOLEGEND;
RUN;
```

## A2 Beispiel Dichteplot und Konturplot - zweidimensionale Johnsonverteilung (S\_LL)

```

goptions reset=global gunit=pct border ftext=none htext=3;
axis1
  width=1
  length=50 PCT
  color=BLACK
  order=-1 to 1.5 by 0.5
  style=1;
axis2
  width=1
  color=BLACK
  order=-1 to 1.5 by 0.5
  style=1;

data SLL;
  mue1=0;
  mue2=0;
  sq1=1;
  sq2=1;
  rho=0.0;
  pi=3.14159265359;
  s1=sqrt(sq1);
  s2=sqrt(sq2);
  a1=exp(sq1/2);
  a2=exp(sq2/2);
  b1=sqrt(exp(2*sq1)-exp(sq1));
  b2=sqrt(exp(2*sq2)-exp(sq2));
  x1min=(-a1/b1)+0.00001;
  x2min=(-a2/b2)+0.00001;
  do x1=x1min to 1.5 by 0.05;
  do x2=x2min to 1.5 by 0.05;
  y1=log(b1*x1+a1);
  y2=log(b2*x2+a2);
  a=b1*b2/(2*pi*s1*s2*(b1*x1+a1)*(b2*x2+a2)*sqrt(1-rho*rho)) ;
  b=((y1/s1)*(y1/s1)-2*rho*y1*y2/(s1*s2)+(y2/s2)*(y2/s2))/(2*(1-rho*rho));
  z=a*exp(-b);
  output;
  end;
end;
run;

proc g3d data=SLL;
  plot x2*x1=z /
    rotate=-45
    caxis=black
    ctop=black
    cbottom=blue
    ctext=black
    xtcknum=5
    ytcknum=6
    ztcknum=5
    zmin=0
    grid;

run;

PROC GCONTOUR data=SLL;
  PLOT x1*x2=z /
    HAXIS=Axis1
    VAXIS=Axis2
    NOLEGEND;

RUN;

```

## A3 Beispiel Dichteplot und Konturplot - zweidimensionale Johnsonverteilung (S\_UU)

```

goptions reset=global gunit=pct border ftext=none htext=3;
axis1
  width=1
  length=50 PCT
  color=BLACK
  order=-15 to 15 by 5
  style=1;
axis2
  width=1
  color=BLACK
  order=-15 to 15 by 5
  style=1;

data SUU;
  mue1=0;
  mue2=0;
  s1=0.1;
  s2=0.1;
  rho=0.0;
  sq1=s1*s1;
  sq2=s2*s2;
  a1=exp(sq1/2)*sinh(mue1);
  a2=exp(sq2/2)*sinh(mue2);
  b1=sqrt((exp(sq1)-1)*(exp(sq1)*cosh(2*mue1)+1));
  b2=sqrt((exp(sq2)-1)*(exp(sq2)*cosh(2*mue2)+1));
  do x1=-15 to 15 by 0.5;
  do x2=-15 to 15 by 0.5;
    w1=b1*x1+a1;
    w2=b2*x2+a2;
    y1=log(w1+sqrt(1+w1*w1));
    y2=log(w2+sqrt(1+w2*w2));
    a=(b1*b1*(w1+sqrt(1+w1*w1))*(w2+sqrt(1+w2*w2)))/((1+w1*w1+w1*sqrt(1+w1*w1))
      *(1+w2*w2+w2*sqrt(1+w2*w2)));
    b=(y1*y1-2*rho*y1*y2+y2*y2)/(2*(1-rho*rho));
    g=exp(-b)/(2*s1*s2*sqrt(1-rho*rho));
    z=a*g;
    output;
  end;
end;
run;

proc g3d data=SUU;
  plot x1*x2=z /
    rotate=45
    caxis=black
    ctop=black
    cbottom=blue
    ctext=black
    xtcknum=5
    ytcknum=6
    ztcknum=5
    zmin=0
    grid;
run;

```

```
PROC GCONTOUR data=SUU;  
  PLOT x1*x2=z /  
    HAXIS=Axis1  
    VAXIS=Axis2  
    NOLEGEND;  
RUN;
```

**A4 Beispiel Dichteplot und Konturplot - zweidimensionale Johnsonverteilung (S\_BB)**

```
goptions reset=global gunit=pct border ftext=none htext=3;
axis1
  width=1
  length=50 PCT
  color=BLACK
  order=0 to 1 by 0.2
  style=1;
axis2
  width=1
  color=BLACK
  order=0 to 1 by 0.2
  style=1;

data SBB;
  mue1=0;
  mue2=0;
  s1=0.5;
  s2=0.5;
  rho=0.0;
  pi=3.14159265359;
  sq1=s1*s1;
  sq2=s2*s2;
  do x1=0.02 to 0.98 by 0.02;
  do x2=0.02 to 0.98 by 0.02;
  y1=(log(x1/(1-x1))-mue1)/s1;
  y2=(log(x2/(1-x2))-mue2)/s2;
  a=(y1*y1-2*rho*y1*y2+y2*y2)/(2*(1-rho*rho));
  b=x1*x2*(1-x1)*(1-x2)*2*pi*s1*s2*sqrt(1-rho*rho) ;
  z=exp(-a)/b;
  output;
  end;
end;

run;
proc g3d data=SBB;
  plot x2*x1=z /
    rotate=-45
    caxis=black
    ctop=black
    cbottom=blue
    ctext=black
    xticknum=5
    yticknum=6
    zticknum=5
    zmin=0
    grid;
run;

PROC GCONTOUR data=SBB;
  PLOT x1*x2=z /
    HAXIS=Axis1
    VAXIS=Axis2
    NOLEGEND;
RUN;
```

## A5 Zweidimensionales Histogramm als Grid und als Scatterplot

```

%macro HIST(DATA, HISTO);
proc iml;
  use &data; read all into y;
  m=nrow(y);
  MY1=j(m,1,0); MY2=j(m,1,0);
  Do i=1 to m;
  MY1[i]=y[i,1] ;
  MY2[i]=y[i,2] ;
  end;
  MY1MIN=MIN(MY1);
  MY2MIN=MIN(MY2);
  MY1Max=Max(MY1);
  MY2Max=Max(MY2);
  print m ;
  print MY1Min MY2MIN ;
  print MY1Max MY2max;
  KL=(2*m)**(1/3);
  KL=INT(KL)+1;
  DY1=MY1max-MY1MIN;
  DY2=MY2max-MY2MIN;
  BKY1=DY1/KL;
  BKY2=DY2/KL;
  FREQ=j(KL*KL,4,0);
  sz=0;
  do i=1 to KL;
    do j=1 to KL;
      sz=sz+1;
      if i=1 then  FREQ[sz,1]=MY1MIN+0.5*BKY1;
      if j=1 then  FREQ[sz,2]=MY2MIN+0.5*BKY2;
      if i>1 then do; if i=KL then FREQ[sz,1]=MY1MIN+0.5*BKY1+(i-1)*BKY1+0.001;
                     else FREQ[sz,1]=MY1MIN+0.5*BKY1+(i-1)*BKY1;
                     end;
      if j>1 then do; if j=KL then FREQ[sz,2]=MY2MIN+0.5*BKY2+(j-1)*BKY2+0.001;
                     else FREQ[sz,2]=MY2MIN+0.5*BKY2+(j-1)*BKY2;
                     end;
    end;
  end;
  print KL DY1 DY2 BKY1 BKY2;
  do i=1 to m;
    do j=1 to KL*KL;
      if y[i,1]>=FREQ[j,1]-0.5*BKY1 then do;
        if y[i,1]<FREQ[j,1]+0.5*BKY1 then do;
          if y[i,2]>=FREQ[j,2]-0.5*BKY2 then do;
            if y[i,2]<FREQ[j,2]+0.5*BKY2 then FREQ[j,3]=FREQ[j,3]+1;
            end;
          end;
        end;
      end;
    end;
  end;
  do i=1 to KL*KL;
  FREQ[i,4]=FREQ[i,3]/m;
  end;
  c={'Y1' 'Y2' 'Z' 'rel_Z'} ;
  create &histo from FREQ [colname=c];
  append from FREQ;
quit;
%mend HIST;

```

```
%HIST(Daten, histogr);
proc g3d data=WORK.histogr ;
  plot Y2 * Y1 = rel_Z /
    rotate=45
    caxis=black
    ctext=black
    grid;

run;
proc g3d data=WORK.histogr ;
  scatter Y2 * Y1 = rel_Z /
    shape="PRISM"
    color="blue"
    rotate=25
    caxis=black
    ctext=black
    grid;

run;
```