Tutorium

Allgemeines Lineares Modell

BSc Psychologie SoSe 2023

6. Termin: (4) Normalverteilungen

Sean Mulready

Follow-up

Fragen von/während der letzten Woche:

- Indexierung bei der Definition zu Einheitsmatrizen
- Können Diagonalmatrizen auch Nullen auf der Diagonale haben?
- Müssen Diagonalmatrizen immer quadratisch sein?

Follow-up : Korrektur

- Korrektur der Definition zu Einheitsmatrizen (Indizes):

Definition (Einheitsmatrizen und Einheitsvektoren)

- Wir bezeichnen die *Einheitsmatrix* mit

$$I_n := (i_{jk})_{1 \leq j \leq n, 1 \leq k \leq n} \in \mathbb{R}^{(n \times n)} \text{ mit } i_{jk} = 1 \text{ für } j = k \text{ und } i_{jk} = 0 \text{ für } j \neq k \tag{1}$$

- Wir bezeichnen die *Einheitsvektoren* $e_i, i = 1, ..., n$ mit

$$e_i := (e_{ij})_{1 \leq j \leq n} \in \mathbb{R}^n \text{ mit } e_{ij} = 1 \text{ für } i = j \text{ und } e_{ij} = 0 \text{ für } i \neq j \tag{2}$$

- In den Vorlesungs- und Tutoriumsfolien korrigiert

Follow-up - Diagonalmatrizen

- Können Diagonalmatrizen auch Nullen auf der Diagonale haben?

Ja, können sie. Diagonalmatrizen haben per Definition nur die Bedingung dass **nichtdiagonale** Elemente 0 sind. Eine Nullmatrix ist also auch eine Diagonalmatrix.

- Sind Diagonalmatrizen **immer** quadratisch?

Nein. Normalerweise sind Diagonalmatrizen zwar quadaratisch aber es gibt auch nichtquadratische Diagonalmatrizen, insbesondere bei der singular value decomposition . Siehe auch hier. Entsprechend wurde die Definition der Tutoriumsfolien geändert (siehe Definition zu Diagonalmatrizen)

Selbstkontrollfragen - Normalverteilungen

1. Geben Sie die Definitionen des Erwartungswerts und der Kovarianzmatrix eines Zufallsvektors wieder.
2. Was repräsentieren die Diagonalelemente der Kovarianzmatrix eines Zufallsvektors?
3. Was repräsentieren die Nichtdiagonalelemente der Kovarianzmatrix eines Zufallsvektors?
4. Definieren Sie die WDF eines multivariaten normalverteilten Zufallsvektors und erläutern Sie diese.
5. Welche Werte haben der Erwartungswert und die Kovarianzmatrix eines normalverteilten Zufallsvektors?
6. Visualisieren Sie die WDF eines 2-dimensionalen normalverteilten Zufallsvektors mit den Parameterwerten
$$\mu := \begin{pmatrix} 10 \\ 15 \end{pmatrix} \text{ und } \Sigma := \begin{pmatrix} 3 & 1 \\ 1 & 2 \end{pmatrix}.$$
7. Generieren Sie 100 Realisierungen aus dieser Verteilung und visualisieren Sie diese.
8. Geben Sie das Theorem zur invertierbaren linearen Transformation multivariater Normalverteilungen wieder.
9. Geben Sie das Theorem zur linear-affinen Transformation multivariater Normalverteilungen wieder.
10. Geben Sie das Theorem zu sphärischen Normalverteilungen wieder.
11. Erläutern Sie den Begriff des sphärischen Kovarianzmatrixparameters.
12. Skizzieren Sie den Beweis des Theorem zu Unabhängigen normalverteilten Zufallsvariablen.

1. Geben Sie die Definitionen des Erwartungswerts und der Kovarianzmatrix eines Zufallsvektors wieder.

Definition (Erwartungswert und Kovarianzmatrix von Zufallsvektoren)

ξ sei ein n-dimensionaler Zufallvektor. Dann ist der *Erwartungwert* von ξ definiert als der n-dimensionale Vektor

$$\mathbb{E}(\xi) := \begin{pmatrix} \mathbb{E}(\xi_1) \\ \vdots \\ \mathbb{E}(\xi_n) \end{pmatrix} \tag{3}$$

und die *Kovarianzmatrix* von ξ ist definiert als die $n \times n$ Matrix

$$\mathbb{C}(\xi) := \mathbb{E}\left((\xi - \mathbb{E}(\xi))(\xi - \mathbb{E}(\xi))^T\right) \tag{4}$$

Anmerkung

Zur Erinnerung

Theorem (Kovarianzmatrix eines Zufallsvektors)

ξ sei ein n-dimensionaler Zufallvektor und $\mathbb{C}(\xi)$ sei seine Kovarianzmatrix. Dann gilt

$$\mathbb{C}(\xi) = \Big(\mathbb{C}(\xi_i, \xi_j)\Big)_{1 \leq i,j \leq n} = \begin{pmatrix} \mathbb{C}(\xi_1, \xi_1) & \mathbb{C}(\xi_1, \xi_2) & \cdots & \mathbb{C}(\xi_1, \xi_n) \\ \mathbb{C}(\xi_2, \xi_1) & \mathbb{C}(\xi_2, \xi_2) & \cdots & \mathbb{C}(\xi_2, \xi_n) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbb{C}(\xi_n, \xi_1) & \mathbb{C}(\xi_n, \xi_2) & \cdots & \mathbb{C}(\xi_n, \xi_n) \end{pmatrix}. \tag{5}$$

2.Was repräsentieren die Diagonalelemente der Kovarianzmatrix eines Zufallsvektors?

Die Diagonalelemente $\mathbb{C}(\xi_i, \xi_j), i = j$ der Kovarianzmatrix von ξ repräsentieren die Kovarianzen der Elemente des Zufallsvektors ξ mit sich selbst und damit die *Varianzen der einzelnen Elemente von* ξ.

Anmerkung: Damit spezifizieren die Diagonalelemente der Kovarianzmatrix Σ die Breite der WDFen der jeweiligen Elemente $\xi_1, ..., \xi_n$ des Zufallsvektors

3. Was repräsentieren die Nichtdiagonalelemente der Kovarianzmatrix eines Zufallsvektors?

Die Nichtdiagonalelemente $\mathbb{C}(\xi_i, \xi_j), i \neq j$ der Kovarianzmatrix von ξ repräsentieren die *Kovarianzen der Elemente des Zufallsvektors* ξ

4.Definieren Sie die WDF eines multivariaten normalverteilten Zufallsvektors und erläutern Sie diese.

Teil 1/2 : Definition

Definition (Multivariate Normalverteilung)

ξ sei ein n-dimensionaler Zufallsvektor mit Ergebnisraum $\mathbb{R}^n$ und WDF

$$p : \mathbb{R}^n \to \mathbb{R}_{>0}, x \mapsto p(x) := (2\pi)^{-\frac{n}{2}} |\Sigma|^{-\frac{1}{2}} exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right). \quad (6)$$

Dann sagen wir, dass ξ einer *multivariaten* (oder n*-dimensionalen*) *Normalverteilung* mit *Erwartungswertparameter* $\mu \in \mathbb{R}^n$ und positiv-definitem *Kovarianzmatrixparameter* $\Sigma \in \mathbb{R}^{n\times n}$ unterliegt und nennen ξ einen *(multivariat) normalverteilten Zufallsvektor*. Wir kürzen dies mit $\xi \sim N(\mu, \Sigma)$ ab. Die WDF eines multivariat normalverteilten Zufallsvektors bezeichnen wir mit

$$N(x; \mu, \Sigma) := (2\pi)^{-\frac{n}{2}} |\Sigma|^{-\frac{1}{2}} exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right). \quad (7)$$

4.Definieren Sie die WDF eines multivariaten normalverteilten Zufallsvektors und erläutern Sie diese.

Teil 2/2 : Erläuterung

In der WDF sind

- $\mu \in \mathbb{R}^n$ der *Erwartungswertparameter*, welcher dem Wert höchster WD entspricht
- $\Sigma \in \mathbb{R}^{n \times n}$ der positive-definite *Kovarianzmatrixparameter*
- $(2\pi)^{-\frac{n}{2}} |\Sigma|^{-\frac{1}{2}}$ die Normalisierungskonstante, wobei $|\Sigma|$ die Determinante von Σ ist, (Anm.: die Normalisierungskonstante wird verwendet, sodass die Fläche unter dem Graphen der WDF 1 ergibt)
- und Σ^{-1} die (ebenfalls positiv-definite) Inverse des Kovarianzmatrixparameters

Die WDF nimmt (als Definitionsmenge/"input") einen Wert $x \in \mathbb{R}^n$ aus dem Ergebnisraum von ξ (also einen n-dimensionalen Wert, den der Zufallsvektor ξ annehmen kann) und gibt eine reelle Zahl ($\mathbb{R}_{>0}$) zurück, welche die Wahrscheinlichkeitsdichte beschreibt.

Anmerkung:

- Werte der WDF für bestimmte $x \in \mathbb{R}^n$ sind *nicht* die Wahrscheinlichkeiten dafür, dass die Zufallsvariable ξ diesen Wert x annimmt, sondern die Wahrscheinlichkeitsdichte für bestimmte x.
- Die Inverse einer p.d. Matrix ist auch p.d.

5.Welche Werte haben der Erwartungswert und die Kovarianzmatrix eines normalverteilten Zufallsvektors?

Der Erwartungswert eines normalverteilten Zufallsvektors entspricht seinem Erwartungswertparameter, formal $\mathbb{E}(\xi) = \mu$ und die Kovarianzmatrix eines normalverteilten Zufallsvektors entspricht seinem Kovarianzmatrixparameter, formal $\mathbb{C}(\xi) = \Sigma$.

Anmerkung: siehe auch das Theorem zu Erwartungswert und Kovarianzmatrix

Konstruktion und Definition - Selbstkontrollfragen

6.Visualisieren Sie die WDF eines 2-dimensionalen normalverteilten Zufallsvektors mit den Parameterwerten $\mu := \begin{pmatrix} 10 \\ 15 \end{pmatrix}$ und $\Sigma := \begin{pmatrix} 3 & 1 \\ 1 & 2 \end{pmatrix}$.

Teil 1/3 - Visualisierung der WDF

```
# R Paket für multivariate Normalverteilungen
library(mvtnorm)
library(latex2exp)

# Parameterdefinition
mu     = c(10,15)                # \mu \in \mathbb{R}^2
Sigma  = matrix(c(3,1,1,2), 2)  # \Sigma in \mathbb{R}^{2 \times 2}

# Ergebnisraumdefintion
x_min  = 0                                  # x_i Minimum
x_max  = 20                                 # x_i Maximum
x_res  = 1e3                                # x_i Auflösung (1e3 --> 1000 Werte)
x_1    = seq(x_min, x_max, length.out = x_res)   # x_1 Raum
x_2    = seq(x_min, x_max, length.out = x_res)   # x_2 Raum
x      = expand.grid(x_1,x_2)                    # x = (x_1,x_2)^T Raum

# Wahrscheinlichkeitsdichtefunktionauswertung
WDF = dmvnorm(as.matrix(x), mu, Sigma)  # Multivariate WDF
p      = matrix(WDF, nrow = x_res)      # Matrixkonversion der WDF
```

Konstruktion und Definition - Selbstkontrollfragen

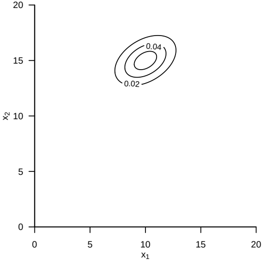
6. Visualisieren Sie die WDF eines 2-dimensionalen normalverteilten Zufallsvektors mit den Parameterwerten $\mu := \begin{pmatrix} 10 \\ 15 \end{pmatrix}$ und $\Sigma := \begin{pmatrix} 3 & 1 \\ 1 & 2 \end{pmatrix}$.

Teil 2/3 - Visualisierung der WDF

```
# Visualisierung
contour(
  x_1,
  x_2,
  p,
  xlim      =  c(x_min,x_max),
  ylim      =  c(x_min,x_max),
  xlab      = TeX("$x_1$"),
  ylab      = TeX("$x_2$"),
  nlevels   = 5
  )
```

6. Visualisieren Sie die WDF eines 2-dimensionalen normalverteilten Zufallsvektors mit den Parameterwerten $\mu := \begin{pmatrix} 10 \\ 15 \end{pmatrix}$ und $\Sigma := \begin{pmatrix} 3 & 1 \\ 1 & 2 \end{pmatrix}$.

Teil 3/3 - Visualisierung der WDF



Konstruktion und Definition - Selbstkontrollfragen

7.Generieren Sie 100 Realisierungen aus dieser Verteilung und visualisieren Sie diese.

Teil 1/6 - Generierung der 100 Realisierungen

```
# R Paket für multivariate Normalverteilungen
library(mvtnorm)

# Parameterdefinition
mu     = c(10,15)               # \mu \in \mathbb{R}^2
Sigma  = matrix(c(3,1,1,2), 2)  # \Sigma in \mathbb{R}^{2 \times 2}

# Zufallsvektorrealisierungen
Realisierungen = rmvnorm(n = 100, mu, Sigma)
print(Realisierungen[1:8,])    # Ausgabe der ersten 8 Realisierungen
```

```
>       [,1] [,2]
> [1,]  7.70 14.9
> [2,]  5.85 14.2
> [3,] 11.43 16.8
> [4,]  6.82 14.1
> [5,]  9.49 14.5
> [6,]  9.26 15.7
> [7,] 12.98 13.4
> [8,] 10.27 12.6
```

Konstruktion und Definition - Selbstkontrollfragen

7.Generieren Sie 100 Realisierungen aus dieser Verteilung und visualisieren Sie diese.

Teil 2/6 - Visualisierung der 100 Realisierungen

```r
# R Paket für latex
library(latex2exp)

# Abbildungsparameter
par(
  family      = "sans",
  pty         = "s",      # plotting type "s" = square plotting region
  bty         = "l",      # boxtype
  lwd         = 1,
  las         = 1,        # Achsenbeschriftung 1 = horizontal
  mgp         = c(2,1,0), # margin line für Achsenbeschriftung und Linie
  xaxs        = "i",
  yaxs        = "i",
  font.main   = 1,
  cex         = .7,
  cex.main    = 1.2)
```

Konstruktion und Definition - Selbstkontrollfragen

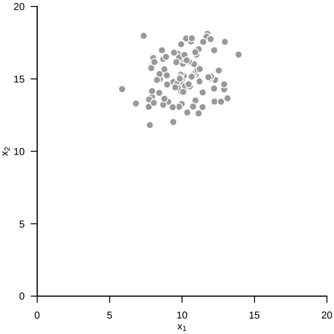
7.Generieren Sie 100 Realisierungen aus dieser Verteilung und visualisieren Sie diese.

Teil 3/6 - Visualisierung der 100 Realisierungen

```
# Visualisierung
plot(
  Realisierungen,
  xlim  = c(min(Realisierungen[,1])-1,max(Realisierungen[,1])+1),
  ylim  = c(min(Realisierungen[,2])-1,max(Realisierungen[,2])+1),
  xlab  = TeX("$x_1$"),
  ylab  = TeX("$x_2$"),
  pch   = 21,        # plotting character 21 = kl. Kreis
  col   = "white",   # color Umrandung
  bg    = "gray60",  # background color
  cex   = 1.5)       # Größe der pch
```

7.Generieren Sie 100 Realisierungen aus dieser Verteilung und visualisieren Sie diese.

Teil 4/6 - Visualisierung der 100 Realisierungen



Konstruktion und Definition - Selbstkontrollfragen

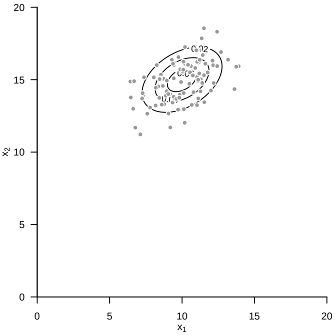
7.Generieren Sie 100 Realisierungen aus dieser Verteilung und visualisieren Sie diese.

Teil 5/6 - Gemeinsame Visualisierung der 100 Realisierungen und der WDF

```
# Visualisierung der WDF und der 100 Realisierung
contour(
  x_1,
  x_2,
  p,
  xlim      =  c(x_min,x_max),
  ylim      =  c(x_min,x_max),
  xlab      = TeX("$x_1$"),
  ylab      = TeX("$x_2$"),
  nlevels   = 5)
points(
  Realisierungen,
  pch   = 21,
  col   = "white",
  bg    = "gray60",
  cex   = 1)
```

7.Generieren Sie 100 Realisierungen aus dieser Verteilung und visualisieren Sie diese.

Teil 6/6 - Gemeinsame Visualisierung der 100 Realisierungen und der WDF



8.Geben Sie das Theorem zur invertierbaren linearen Transformation multivariater Normalverteilungen wieder.

Theorem (Invertierbare lineare Transformation)

$\xi \sim N(\mu, \Sigma)$ sei ein normalverteilter n-dimensionaler Zufallsvektor und es sei $\upsilon := A\xi$ mit einer invertierbaren Matrix $A \in \mathbb{R}^{n \times n}$. Dann gilt

$$\upsilon \sim N\left(A\mu, A\Sigma A^T\right) \tag{8}$$

Anmerkung:

- Eine Matrix ist invertierbar, wenn die Determinante $\neq 0$ ist.

9.Geben Sie das Theorem zur linear-affinen Transformation multivariater Normalverteilungen wieder.

Theorem (Linear-affine Transformation)

$\xi \sim N(\mu, \Sigma)$ sei ein normalverteilter n-dimensionaler Zufallsvektor und es sei $\upsilon := A\xi + b$ mit $A \in \mathbb{R}^{m \times n}$ und $b \in \mathbb{R}^m$. Dann gilt

$$\upsilon \sim N\left(A\mu + b, A\Sigma A^T\right) \tag{9}$$

10. Geben Sie das Theorem zu sphärischen Normalverteilungen wieder.

Theorem (Sphärische multivariate Normalverteilung)

Für $i = 1, ..., n$ seien $N(x_i; \mu_i, \sigma^2)$ die WDFen von n unabhängigen univariaten normalverteilten Zufallsvariablen $\xi_1, ..., \xi_n$ mit $\mu_1, ..., \mu_n \in \mathbb{R}$ und $\sigma^2 > 0$. Weiterhin sei $N(x; \mu, \sigma^2 I_n)$ die WDF eines n-variaten Zufallsvektors ξ mit Erwartungswertparameter $\mu := (\mu_1, ..., \mu_n) \in \mathbb{R}^n$. Dann gilt

$$p_\xi(x) = p_{\xi_1,...,\xi_n}(x_1, ..., x_n) = \prod_{i=1}^{n} p_{\xi_i}(x_i) \tag{10}$$

und insbesondere

$$N\left(x; \mu, \sigma^2 I_n\right) = \prod_{i=1}^{n} N\left(x_i; \mu_i, \sigma^2\right). \tag{11}$$

11.Erläutern Sie den Begriff des sphärischen Kovarianzmatrixparameters.

- Ein *sphärischer* Kovarianzmatrixparameter hat die Form $\sigma^2 I_n$
- Sphärische Kovarianzmatrixparamter von n-variaten Normalverteilungen entsprechen n unabhängigen univariaten Normalverteilungen und umgekehrt.
- Die sphärische Form spiegelt die Unabhängigkeit der univariaten Zufallsvariablen wieder, da alle nicht-Diagonal-Elemente gleich 0 sind (i.e. $(\sigma^2 I_n)_{ij} = 0$ für $i \neq j$) und ensprechend keine Kovarianzen existieren.

Beispiel

Für $i = 1, .., 5$ seien seien $N(x_i; \mu_i, \sigma^2)$ die WDFen von n unabhängigen **univariaten** normalverteilten Zufallsvariablen $\xi_1, ..., \xi_n$ mit $\mu_1, ..., \mu_n \in \mathbb{R}$ und $\sigma^2 = 9$. Weiterhin sei $N(x; \mu, \sigma^2 I_n)$ die WDF eines n-variaten (**multivariaten**) Zufallsvektors ξ mit Erwartungswertparameter $\mu := (\mu_1, ..., \mu_n) \in \mathbb{R}^n$. Dann gilt

$$N\left(x; \mu, \sigma^2 I_n\right) = \prod_{i=1}^{n} N\left(x_i; \mu_i, \sigma^2 = 9\right).$$

mit

$$\sigma^2 I_n = 9 I_n = 9 \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 9 & 0 & 0 & 0 & 0 \\ 0 & 9 & 0 & 0 & 0 \\ 0 & 0 & 9 & 0 & 0 \\ 0 & 0 & 0 & 9 & 0 \\ 0 & 0 & 0 & 0 & 9 \end{pmatrix}$$

12.Skizzieren Sie den Beweis des Theorems zu sphärischen Normalverteilungen.

Zuerst schreiben wir die multivariate WDF $N\left(x;\mu,\sigma^2 I_n\right)$ nach der Definition einer WDF eines multivariaten normalverteilen Zufallsvektors auf, wobei wir anstelle des Kovarianzmatrixparameters Σ den sphärischen Kovarianzmatrixparameter $\sigma^2 I_n$ einsetzen.

Da der sphärische Kovarianzmatrixparameter eine Diagonalmatrix ist, entspricht dessen Determinante dem Produkt der Diagonaleinträge und der **Term vor der Exponantialfunktion** lässt sich umschreiben zu einem n-fachen Produkt der Normalisierungskonstante.

Der **Term der Exponentialfunktion** wird durch Matrixmultiplikation zu einem Term, der aus einer multiplikativen Konstante und einer Summe der quadrierten Differenzen zwischen x_i und μ_i besteht. Mit der Eigenschaft von Exponentialfunktion (i.e. $\exp(x+y)=\exp(x)\exp(y)$) können wir die Exponentialfunktion umschreiben zu einem Produkt von Exponentialfunktionen des Terms.

Mit den Eigenschaften von Produkten können wir beide Produkte zusammenfassen und erhalten ein Produkt von univariaten Normalverteilungen.

Anhang

Vorwort: Zu Aufgaben 6 und 7 war eine nähere Erläuterung des R-Codes gewünscht, die ich in Präsenz nicht so geben konnte. Diesem Wunsch soll mit dem Anhang nachgekommen werden. Vorab sei gesagt: Zur Erläuterung des Codes habe ich mich ChatGPT bedient. Dazu habe ich vor dem Code (siehe nächste Folie) folgende Anweisung gestellt: "Erkläre mir folgenden R-Code Blockweise. Als einzelne Blöcke nimmst du die Abschnitte, die durch die Kommentare mit dem"#" ganz links in der Zeile getrennt sind." Damit konnte ChatGPT schon viel machen, jedoch stellenweise nicht in meiner gewünschten Tiefe. So konnte ich nach der ersten Erklärung bspw. nochmal näher nachfragen mit der Folgefrage "was bewirkt das Argument length.out in Block 3?"

Damit möchte ich zwei Dinge sagen:

Erstens, wenn Verständnisprobleme da sind, ist es nicht verboten, auch mal eine künstliche Intelligenz wie ChatGPT zu fragen. Das Programm sollte nicht zur vollständigen Lösung herangezogen werden, kann aber vieles erklären. Auch wenn man Hilfe bei der Hypothesenbildung oder dem Finden einer Fragestellung benötigt, kann ein Dialog mit ChatGPT in die richtige Richtung lenken.

Zweitens, ähnlich wie beim Programmieren arbeitet ChatGPT am besten, wenn ich genau und logisch schreibe, was ich erwarte. Ich habe in meiner Anfrage nicht nur danach gefragt, mir den Code zu erklären sondern auch wie (nämlich in Abschnitten, dazu habe ich auch geschrieben, wie die Abschnitte getrennt sind).

Zunächst der Code im ganzen. Hier Aufgaben 6 und 7 in einem (auf 2 Slides, aus Platzgründen):

```
# R Paket für multivariate Normalverteilungen
library(mvtnorm)
library(latex2exp)

# Parameterdefinition
mu      = c(10,15)                # \mu \in \mathbb{R}^2
Sigma   = matrix(c(3,1,1,2), 2)   # \Sigma in \mathbb{R}^{2 \times 2}

# Zufallsvektorrealisierungen
Realisierungen = rmvnorm(n = 100, mu, Sigma)
print(Realisierungen[1:8,])     # Ausgabe der ersten 8 Realisierungen

# Abbildungsparameter
graphics.off()
fdir          =  file.path(getwd(), "Abbildungen")
dev.new()
par(
  family        = "sans",
  pty           = "s",
  bty           = "l",
  lwd           = 1,
  las           = 1,
  mgp           = c(2,1,0),
  xaxs          = "i",
  yaxs          = "i",
  font.main     = 1,
  cex           = .7,
  cex.main      = 1.2)
```

```r
# Ergebnisraumdefintion
x_min  = 0                                        # x_i Minimum
x_max  = 20                                       # x_i Maximum
x_res  = 1e3                                      # x_i Auflösung (1e3 --> 1000 Werte)
x_1    = seq(x_min, x_max, length.out = x_res)    # x_1 Raum
x_2    = seq(x_min, x_max, length.out = x_res)    # x_2 Raum
x      = expand.grid(x_1,x_2)                     # x = (x_1,x_2)^T Raum

# Wahrscheinlichkeitsdichtefunktionauswertung
WDF = dmvnorm(as.matrix(x), mu, Sigma)  # Multivariate WDF
p      = matrix(WDF, nrow = x_res)      # Matrixkonversion der WDF

# Visualisierung
contour(
  x_1,
  x_2,
  p,
  xlim      =  c(x_min,x_max),
  ylim      =  c(x_min,x_max),
  xlab      = TeX("$x_1$"),
  ylab      = TeX("$x_2$"),
  nlevels   = 5)
points(
  Realisierungen,
  pch   = 21,
  col   = "white",
  bg    = "gray60",
  cex   = 1)

# Speichern
dev.copy2pdf(file=file.path(abb_dir,"alm_4_skf_6_overlapp.pdf"))
```

```r
# R Paket für multivariate Normalverteilungen
library(mvtnorm)
library(latex2exp)
```

Laden der notwendigen R-Pakete: **“mvtnorm”** für die Arbeit mit multivariaten Normalverteilungen, **“latex2exp”** für die Verwendung von LaTeX-Ausdrücken.

```r
# Parameterdefinition
mu     = c(10,15)                # \mu \in \mathbb{R}^2
Sigma  = matrix(c(3,1,1,2), 2)  # \Sigma in \mathbb{R}^{2 \times 2}
```

Definition der Parameter für die multivatiate Normalverteilung wie in der Aufgabenstellung, mit mu für μ (Erwartungswertparameter) und Sigma für Σ (Kovarianzmatrixparameter)

```r
# Zufallsvektorrealisierungen
Realisierungen = rmvnorm(n = 100, mu, Sigma)
print(Realisierungen[1:8,])     # Ausgabe der ersten 8 Realisierungen
```

Hier werden mit **rmvnorm** nun 100 Zufallsvektoren (jeder Vektor hat zwei Werte) aus der multivariaten Normalverteilung (mit den zuvor definierten Parametern) generiert. Diese Zufallsvektoren werden in der Variable **Realisierungen** gespeichert. Zur Veranschaulichung werden die ersten 8 dieser Vektoren ausgegeben.

```
# Abbildungsparameter
graphics.off()
fdir        =  file.path(getwd(), "Abbildungen")
dev.new()
par(
  family      = "sans",
  pty         = "s",
  bty         = "l",
  lwd         = 1,
  las         = 1,
  mgp         = c(2,1,0),
  xaxs        = "i",
  yaxs        = "i",
  font.main   = 1,
  cex         = .7,
  cex.main    = 1.2)
```

Hier werden Parameter für die Abbildung festgelegt. Dabei wird **graphics.off()** verwendet, um alle vorhandenen Grafikfenster zu schließen, mit **fdir** wird eine Variable definiert, die das Speicherverzeichnis angibt (der Name kann hier selbst gewählt werden, z.B. auch "abb_verzeichnis") und letztlich öffnet **dev.new()** ein neues Grafikfenster. Mit **par()** werden nun verschiedene Grafikparameter gesetzt, u.a. die Schriftart mit **family**, der Rahmentyp mit **bty** (für "boxtype"). Alle möglichen setzbaren Parameter können in der R-Dokumentation nachgelesen werden oder in der entsprechenden Literatur. Oder auch über ChatGPT erfragt werden.

```
# Ergebnisraumdefintion
x_min  = 0                                     # x_i Minimum
x_max  = 20                                    # x_i Maximum
x_res  = 1e3                                   # x_i Auflösung (1e3 --> 1000 Werte)
x_1    = seq(x_min, x_max, length.out = x_res) # x_1 Raum
x_2    = seq(x_min, x_max, length.out = x_res) # x_2 Raum
x      = expand.grid(x_1,x_2)                  # x = (x_1,x_2)^T Raum
```

Festlegen des Ergebnisraums, mit **x_min** und **x_max** werden, wie man im weiteren Verlauf des Codes sieht, die Grenzen der x_1- und x_2-Werte gesetzt. Mit **x_res** wird die Auflösung definiert, hier die Anzahl der Werte. Dazu schauen wir uns die folgenden Variablen **x_1** und **x_2** an. Diese werden definiert als Sequenz von Werten zwischen den zuvor definierten Grenzen, **length.out = x_res** bedeutet hier, dass beide Vektoren x_1 und x_2 je 1000 Werte enthalten, die sich gleichmäßig zwischen den Grenzen verteilen. Wichtig: Ergebnisraum heißt nicht, dass es sich bei x_1 und x_2 um Realisationen handelt. Wir definieren hier lediglich, in welchem Bereich sich die Realisationen bewegen *können*. Ziel ist es, sicherzustellen, dass ausreichend Punkte vorhanden sind, um die WDF genau darstellen zu können. Letzlich wird mit **x** ein *Gitter* (“grid”) über die Vektoren gespannt und damit eine Matrix erstellt, die alle möglichen Kombinationen der Elemente von **x_1** und **x_2** enthält.

```
# Wahrscheinlichkeitsdichtefunktionauswertung
WDF = dmvnorm(as.matrix(x), mu, Sigma)  # Multivariate WDF
p      = matrix(WDF, nrow = x_res)      # Matrixkonversion der WDF
```

Berechnung der WDF für jede Wertekombination in x mit der Funktion **dmvnorm**. Da **x** als Gitter von Koordinaten definiert wurde und damit in Form eines *data frame* vorliegt, die Funktion **dmvnorm()** aber als Argument eine Matrix erwartet, verwenden wir die **as.datatype()**-Funktion um den Datentyp den wir brauchen, zu erzwingen (ich könnte auch andere Datentypen erzwingen, z.B. mit "as.string()"). Hier **as.matrix(x)**. Das Ergebnis der Berechnungen wird dann als Variable **WDF** gespeichert und in der Variable **p** zu einer Matrix mit 1000 Zeilen (=x_res) umgewandelt.

```r
# Visualisierung
contour(
  x_1,
  x_2,
  p,
  xlim      =  c(x_min,x_max),
  ylim      =  c(x_min,x_max),
  xlab      = TeX("$x_1$"),
  ylab      = TeX("$x_2$"),
  nlevels   = 5)
points(
  Realisierungen,
  pch    = 21,
  col    = "white",
  bg     = "gray60",
  cex    = 1)
```

Jetzt zur Visualisierung. Die Funktion **contour()** wird verwendet, um die Konturlinienen der WDF auf den Ebenen **x_1** und **x_2** zu zeichnen. Als erwähnenswerter Paramter bleibt **nlevels**, damit wird die Anzahl der Konturlinien bestimmt. Im Anschluss wird mit der Funktion **ponts()** die generierten Realisierungen dargestellt.

Literaturempfehlung:

Murrell, Paul. 2019. *R Graphics.* Third edition. The R Series. Boca Raton: CRC Press, Taylor & Francis Group