(3) Zufallsvektoren

Ziel dieses Seminar ist es, die Begriffe des Zufallsvektors und der multivariaten Wahrscheinlichkeitsverteilung anhand von **R** Abbildungen und Simulationen von bivariaten Normalverteilungen zu verdeutlichen.

Gemeinsame Normalverteilungen

Wir wollen zunächst die WDFen verschiedener bivariater Normalverteilungen visualisieren. Dazu verdeutlichen wir uns folgenden **R** Code.

```r
# R Pakete
library(latex2exp)
library(mvtnorm)

# Abbildungsparameter
pdf(
file        = "./3_Abbildungen_S/mv_3_mvnwdf.pdf",
width       = 8,
height      = 4)
par(
family      = "sans",
mfcol       = c(1,3),
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)

# Ergebnisraumdefintion
x_min  = 0                                                                      # x_i Minimum
x_max  = 2                                                                      # x_i Maximumumum
x_res  = 1e3                                                                    # x_i Auflösung
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

# Parameterdefinition
mu     = c(1,1)                                                                 # \mu \in \mathbb{R}^2
S      = list(matrix(c(0.2,  0.15,  0.15, 0.2), 2),                             # \Sigma in \mathbb{R}^{2 \times 2}
              matrix(c(0.2,  0.00,  0.00, 0.2), 2),                             # \Sigma in \mathbb{R}^{2 \times 2}
              matrix(c(0.2, -0.15, -0.15, 0.2), 2))                             # \Sigma in \mathbb{R}^{2 \times 2}
L      = list(expression(bgroup("(", atop("0.20  0.15", " 0.15  0.20"), ")")),
              expression(bgroup("(", atop("0.20  0.00", " 0.00  0.20"), ")")),
              expression(bgroup("(", atop("0.20 -0.15", "-0.15  0.20"), ")")))

# Kovarianzparametervariantenschleife
i     = 1
for (Sigma in S){

  # Wahrscheinlichkeitsdichtefunktionauswertung
  p      = matrix(                                                               # Matrixkonversion des von
                  dmvnorm(as.matrix(X), mu, Sigma),                              # dmvnorm() ausgegebenen Vektors
                  nrow = x_res)

  # 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)
```

```
  text(1,2.2, L[[i]], xpd = TRUE)
  i = i + 1
}
dev.off()
```

Wir erhalten Abbildung 1, wobei sich die Matrix im Titel der Unterabbildungen auf den jeweiligen Kovarianzmatrixparameter bezieht.

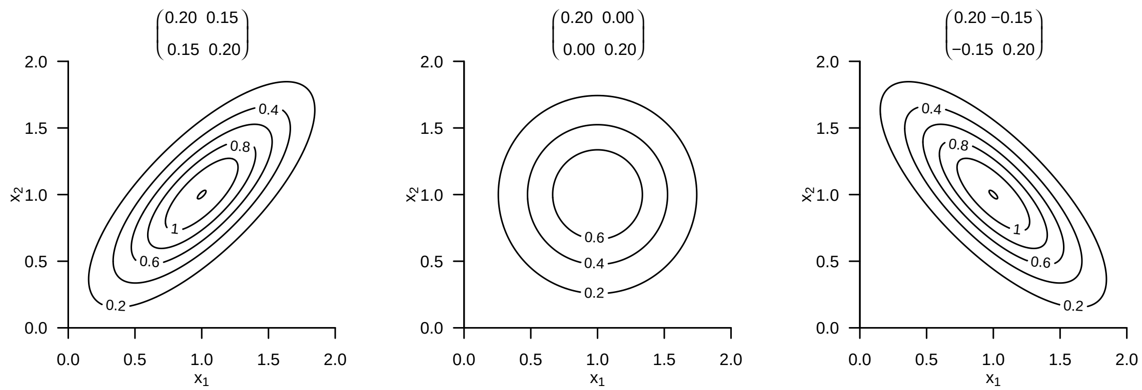

Abbildung 1. WDFen bivariater Normalverteilungen

Als nächstes wollen wir unabhängige und identische verteilte Realisierungen von bivariaten Zufallsvektoren gewinnen und visualisieren, die nach obigen Normalverteilungen verteilt sind. Dazu verdeutlichen wir uns folgenden **R** Code.

```
# R Pakete
library(latex2exp)
library(mvtnorm)

# Abbildungsparameter
pdf(
file        = "./3_Abbildungen_S/mv_3_rmvnorm.pdf",
width       = 8,
height      = 4)
par(
family      = "sans",
mfcol       = c(1,3),
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)

# Parameterdefinition
mu     = c(1,1)                                              # \mu \in \mathbb{R}^2
S      = list(matrix(c(0.2,  0.15,  0.15, 0.2), 2),          # \Sigma in \mathbb{R}^{2 \times 2}
```

```
             matrix(c(0.2,  0.00,  0.00, 0.2), 2),                             # \Sigma in \mathbb{R}^{2 \times 2}
             matrix(c(0.2, -0.15, -0.15, 0.2), 2))                             # \Sigma in \mathbb{R}^{2 \times 2}
L      = list(expression(bgroup("(", atop("0.20  0.15", " 0.15  0.20"), ")")),
             expression(bgroup("(", atop("0.20  0.00", " 0.00  0.20"), ")")),
             expression(bgroup("(", atop("0.20 -0.15", "-0.15  0.20"), ")")))

# Kovarianzparametervariantenschleife
i = 1
for (Sigma in S){

  # 200 Zufallsvektorrealisierungen
  samples = rmvnorm(n = 200, mu, Sigma)

  # Visualisierung
  plot(
  samples,
  xlim  = c(0,2),
  ylim  = c(0,2),
  xlab  = TeX("$x_1$"),
  ylab  = TeX("$x_2$"),
  pch   = 21,
  col   = "white",
  bg    = "gray60",
  cex   = 1.5)
  text(1,2.2, L[[i]], xpd = TRUE)
  i = i + 1
}
dev.off()
```

Wir erhalten Abbildung 2, wobei sich die Matrix im Titel der Unterabbildungen wieder auf den jeweiligen Kovarianzmatrixparameter bezieht.

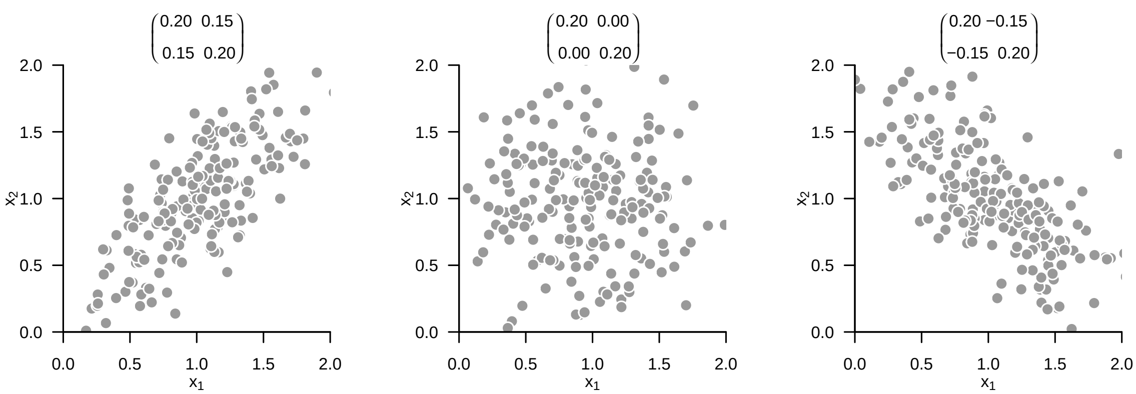

Abbildung 2. Realisierungen bivariater normalverteilter Zufallsvektoren

Marginale, gemeinsame und bedingte Normalverteilungen

Schließlich wollen wir uns die Theoreme zu marginalen, gemeinsamen und bedingten Normalverteilungen für den bivariaten Fall auf der Ebene der entsprechenden WDFen verdeutlichen.

Wir greifen hierzu auf die in der Vorlesung behandelten bivariaten Beispiel zurück.

Marginale Normalverteilungen

Um uns das Theorem zu marginalen Normalverteilungen zu verdeutlichen, studieren wir folgenden **R** Code, der Abbildung 3 erzeugt.

```
# R Pakete
library(mvtnorm)
library(latex2exp)

# Abbildungsparameter
pdf(
file       = "./3_Abbildungen_S/mv_3_marginale_mvnorm.pdf",
width       = 8,
height      = 4)
par(
family      = "sans",
mfcol       = c(1,3),
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)

# Ergebnisraumdefintion
x_min  = 0                                                                  # x_i Minimum
x_max  = 3                                                                  # x_i Maximumum
x_res  = 1e3                                                                # x_i Auflösung
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

# Parameterdefinition
mu     = c(1,2)                                                             # \mu \in \mathbb{R}^2
Sigma  = matrix(c(0.10, 0.08,
                  0.08, 0.15),
                nrow = 2,
                byrow = TRUE)                                               # \Sigma in \mathbb{R}^{2 \times 2}


# Visualisierung gemeinsame Verteilung
p      = matrix(                                                            # Matrixkonversion des von
                dmvnorm(as.matrix(X), mu, Sigma),                           # dmvnorm() ausgegebenen Vektors
                nrow = x_res)
contour(
x_1,
x_2,
p,
xlim      =  c(x_min,x_max),
ylim      =  c(x_min,x_max),
xlab      = TeX("$y$"),
ylab      = TeX("$z$"),
main      = TeX("$N(x;\\mu,\\Sigma)$"),
nlevels   = 10)

# Visualisierung Marginalverteilungen
p_marg   = list(dnorm(x_1, mu[1],Sigma[1,1]), dnorm(x_1, mu[2],Sigma[2,2]))
l_marg   = c(TeX("$y$"), TeX("$z$"))
L_marg   = c(TeX("$N(y;\\mu_\\upsilon,\\Sigma_{\\upsilon\\upsilon})$"), TeX("$N(z; \\mu_\\zeta,\\Sigma_{\\zeta\\zeta})$"))
i        = 1
for(i in 1:length(p_marg)){
  plot(
  x_1,
  p_marg[[i]],
  type = "l",
  xlab = l_marg[[i]],
  ylim = c(0,5),
  ylab = "",
```

```
  main = L_marg[[i]])
}
dev.off()
```

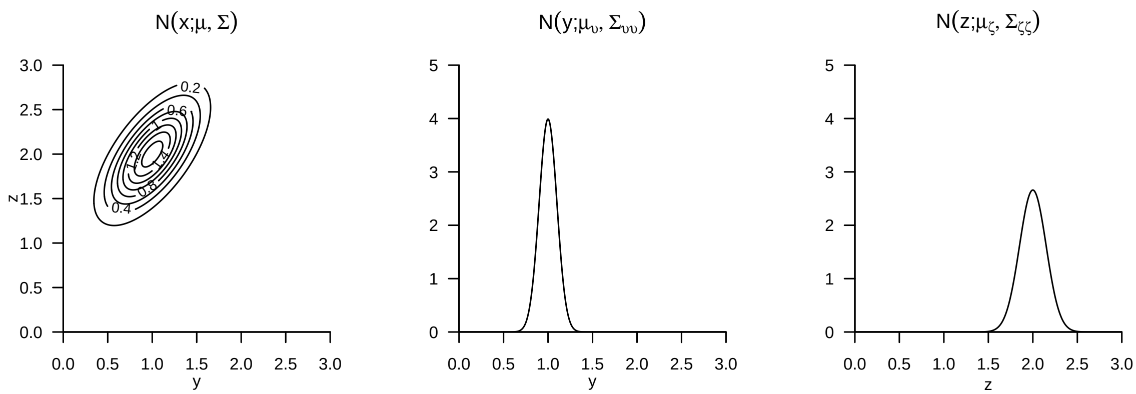

Abbildung 3. Marginale WDFen bivariater Normalverteilungen

Gemeinsame Normalverteilungen

Um uns das Theorem zu gemeinsamen Normalverteilungen zu verdeutlichen, studieren wir folgenden **R** Code, der Abbildung 4 erzeugt.

```
# Abbildungsparameter
library(mvtnorm)
library(latex2exp)
pdf(
file        = "./3_Abbildungen_S/mv_3_gemeinsame_mvnorm.pdf",
width       = 8,
height      = 4)
par(
family      = "sans",
mfcol       = c(1,3),
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)

# Ergebnisraumdefintion
x_min  = -1                                         # x_i Minimum
x_max  = 4                                          # x_i Maximum
x_res  = 1e3                                        # x_i Auflösung
x      = seq(x_min, x_max, length.out = x_res)      # x_1 Raum
y      = seq(x_min, x_max, length.out = x_res)      # x_2 Raum
xy     = expand.grid(x,y)                       # X = (x_1,x_2)^T Raum
```

```
# Parameterdefinition
mu_x      = 1                                          # \mu \in \mathbb{R}
Sigma_xx  = 0.2                                        # \Sigma_xx in \mathbb{R}
A         = 1                                          # A
b         = 1                                          # b
Sigma_yy  = 0.1                                        # \Sigma_yy

mu_xy     = c(mu_x, A*mu_x + b)
Sigma_xy  = matrix(c(Sigma_xx  , Sigma_xx*t(A),
                     A*Sigma_xx, Sigma_yy + A*Sigma_xx*t(A)),
                   nrow = 2,
                   byrow = TRUE)

# Visualisierung marginale Verteilung
plot(
x,
dnorm(x, mu_x, sqrt(Sigma_xx)),
type = "l",
xlab = TeX("$x$"),
ylim = c(0,2),
ylab = "",
main = TeX("$p(x) = N(x;\\mu_\\xi,\\Sigma_{\\xi\\xi})$"))

# Visualisierung bedingte Verteilung
plot(
y,
dnorm(y, A*1 + b, sqrt(Sigma_yy)),
type = "l",
xlab = TeX("$y$"),
ylim = c(0,2),
ylab = "",
main = TeX("$p(y|x) = N(y; Ax + b, \\Sigma_{\\upsilon\\upsilon})$"))
text(0,1.75, TeX("$x = 1$"), cex = 1.2)

# Visualisierung gemeinsame Verteilung
p_xy    = matrix(                                                       # Matrixkonversion des von
                dmvnorm(as.matrix(xy), mu_xy, Sigma_xy),                # dmvnorm() ausgegebenen Vektors
                nrow = x_res)
contour(
x,
y,
p_xy,
xlim      = c(x_min,x_max),
ylim      = c(x_min,x_max),
xlab      = TeX("$x$"),
ylab      = TeX("$y$"),
main      = TeX("$p((x,y)^T) = N((x,y)^T; \\mu_{\\xi,\\upsilon},\\Sigma_{\\xi,\\upsilon})$"),
nlevels   = 5)
dev.off()
```

Bedingte Normalverteilungen

Um uns das Theorem zu bedingten Normalverteilungen zu verdeutlichen, studieren wir folgenden **R** Code, der Abbildung 5 erzeugt.

```
# Abbildungsparameter
library(mvtnorm)
library(latex2exp)
pdf(
file        = "./3_Abbildungen_S/mv_3_bedingte_mvnorm.pdf",
width       = 8,
height      = 4)
par(
family      = "sans",
mfcol       = c(1,3),
pty         = "s",
bty         = "l",
lwd         = 1,
las         = 1,
mgp         = c(2,1,0),
xaxs        = "i",
yaxs        = "i",
font.main   = 1,
```

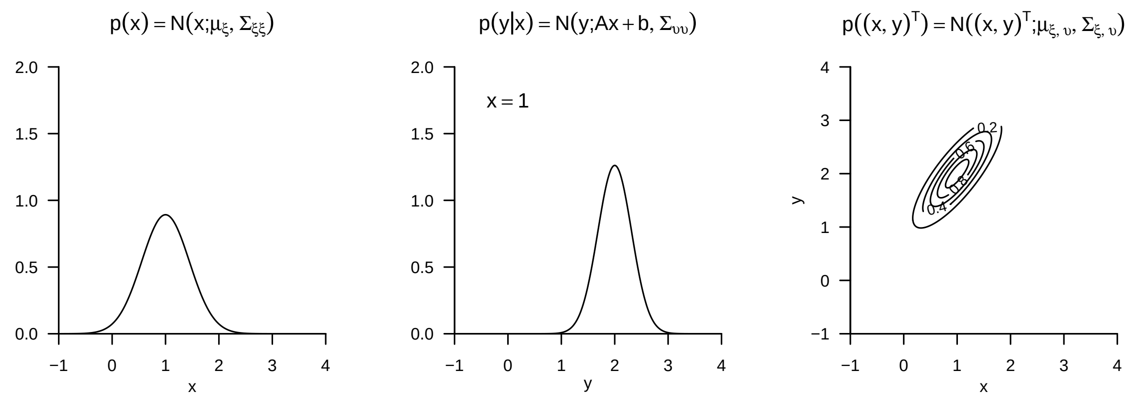

Abbildung 4. Gemeinsame WDFen univariater Normalverteilungen

```
cex         = .7,
cex.main    = 1.2)


# Ergebnisraumdefintion
x_min  = 0                                                            # x_i Minimum
x_max  = 4                                                            # x_i Maximum
x_res  = 1e3                                                          # x_i Auflösung
x      = seq(x_min, x_max, length.out = x_res)                        # x Raum
y      = seq(x_min, x_max, length.out = x_res)                        # y Raum
X      = expand.grid(x,y)                                             # (x,y)^T Raum

# Parameterdefinition
mu     = c(1,2)                                                       # \mu \in \mathbb{R}^2
Sigma  = matrix(c(0.12, 0.09,
                  0.09, 0.12),
                nrow = 2,
                byrow = TRUE)                                         # \Sigma in \mathbb{R}^{2 \times 2}


# Visualisierung gemeinsame Verteilung
p      = matrix(                                                      # Matrixkonversion des von
                dmvnorm(as.matrix(X), mu, Sigma),                     # dmvnorm() ausgegebenen Vektors
                nrow = x_res)
contour(
x,
y,
p,
xlim      = c(x_min,x_max),
ylim      = c(x_min,x_max),
xlab      = TeX("$x$"),
ylab      = TeX("$y$"),
main      = TeX("$p(x,y) = N((x,y)^T; \\mu_{\\xi,\\upsilon},\\Sigma_{\\xi,\\upsilon})$"),
nlevels   = 8)
abline(1.5,0)
abline(2.5,0)
text(3.5,1, TeX("$y = 1.5$"))
text(3.5,3, TeX("$y = 2.5$"))

# Visualsierung bedingter Verteilung
plot(
x,
dnorm(x, mu[1] + Sigma[1,2]*(1/Sigma[2,2])*(1.5 - mu[2]),sqrt(Sigma[1,1] - Sigma[1,2]*(1/Sigma[2,2]*Sigma[2,1]))),
```

```
type = "l",
xlim = c(x_min,x_max),
ylim = c(0,2),
xlab = TeX("$x$"),
ylab = "",
main = TeX("$p(x|y) = N(x; \\mu_{\\xi|\\upsilon},\\Sigma_{\\xi|\\upsilon})$"))
text(3.5,1.8, TeX("$y = 1.5$"))

# Visualsierung bedingter Verteilung
plot(
x,
dnorm(x, mu[1]+ Sigma[1,2]*(1/Sigma[2,2])*(2.5 - mu[2]),sqrt(Sigma[1,1] - Sigma[1,2]*(1/Sigma[2,2]*Sigma[2,1]))),
type = "l",
xlim = c(x_min,x_max),
ylim = c(0,2),
xlab = TeX("$x$"),
ylab = "",
main = TeX("$p(x|y) = N(x; \\mu_{\\xi|\\upsilon},\\Sigma_{\\xi|\\upsilon})$"))
text(3.5,1.8, TeX("$y = 2.5$"))
dev.off()
```

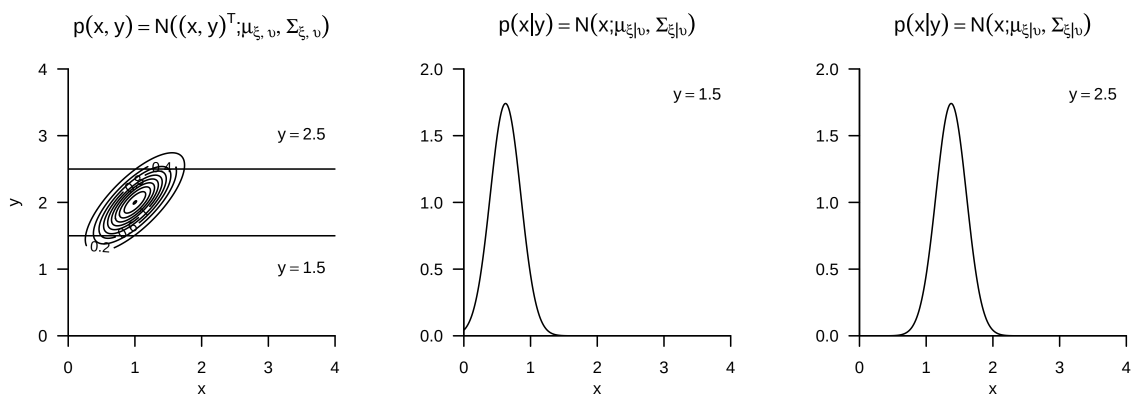

Abbildung 5. Bedingte WDFen bivariater Normalverteilungen