Mathematik mit Mathematica

Praktikum im Wintersemester 2021/22 an der TU Braunschweig
betreut von Prof. Dr. Michael Herrmann

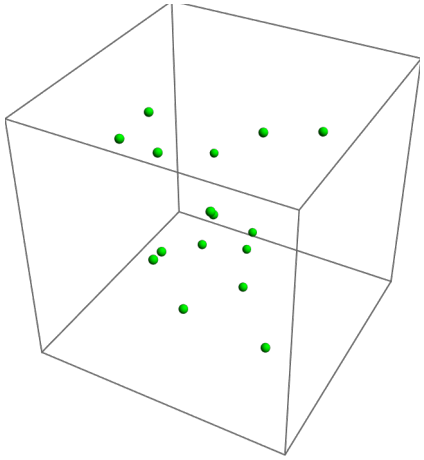
Lösungen Serie 06

Zufallsdatensatz

In[•]:=
```
numPoints = 16;
datPoints = RandomReal[{-1, 1}, {numPoints, 3}];
datCenter = Sum[datPoints[[i]], {i, 1, numPoints}] / numPoints;

(* plotte die Punkte *)
Graphics3D[
 {
  Green,
  Table[Sphere[datPoints[[i]], 0.05], {i, 1, numPoints}]
 },
 PlotRange → {{-1.5, 1.5}, {-1.5, 1.5}, {-1.5, 1.5}},
 ImageSize → 72 × 3
]
```

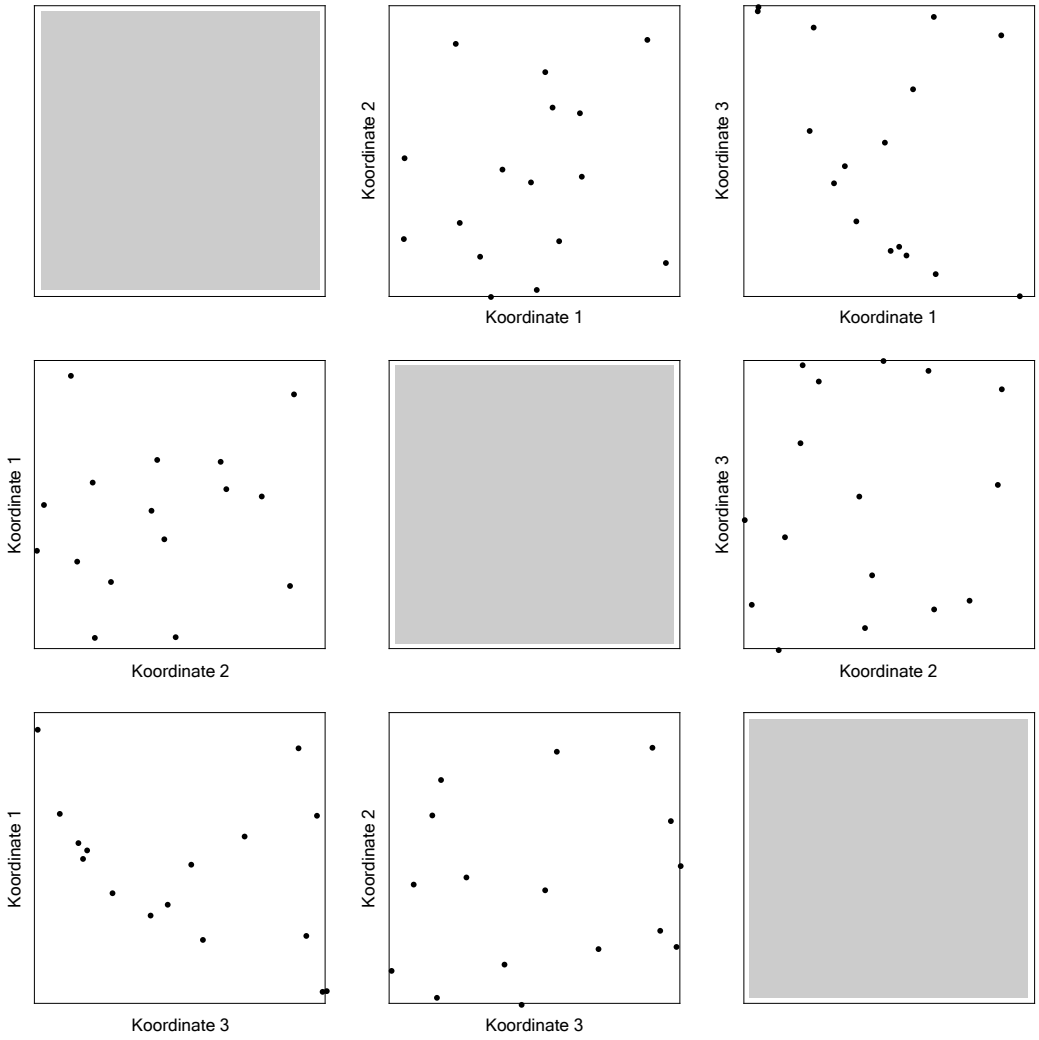
Out[•]=



Streudiagramm

In[•]:=
```
minVal = Min[Flatten[datPoints]]; (* minimaler Wert *)
maxVal = Max[Flatten[datPoints]]; (* maximaler Wert *)
pic1 =
 Show[
  GraphicsGrid[
   Table[
    If[
     k1 == k2,
     (* graue Box auf der Diagonalen *)
     Graphics[
      {
       GrayLevel[0.8],
       Rectangle[{minVal, minVal}, {maxVal, maxVal}],
       AspectRatio → 1
      },
      FrameTicks → None,
      Frame → True, FrameLabel → {{" ", None}, {" ", None}}
     ],
     (* projizierte Punktwolke auf den Nebendiagonalen *)
     Graphics[
      {
       PointSize[0.02],
       Table[
        {
         Point[{datPoints[[i, k1]], datPoints[[i, k2]]}]
        },
        {i, 1, numPoints}
       ]
      },
      (* einige Optionen *)
      AspectRatio → 1,
      ImageSize → Small,
      PlotRange → {{minVal, maxVal}, {minVal, maxVal}},
      BaseStyle → {FontFamily → "Microsoft Sans Serif", 9},
      Frame → True,
      FrameTicks → None,
      FrameLabel → {{"Koordinate " <> ToString[k2], None},
        {"Koordinate " <> ToString[k1], None}}
      ]
     ],
     {k1, 1, 3}, {k2, 1, 3}
    ]
   ]
  ]
```

Out[ ]=


Koordinate 2
Koordinate 1
Koordinate 3
Koordinate 1
Koordinate 1
Koordinate 2
Koordinate 3
Koordinate 2
Koordinate 1
Koordinate 3
Koordinate 2
Koordinate 3

Kovarianz-Matrix

In[•]:=
```
(* Berechnung der Matrix *)
covMatrix =
  Sum[
    y = datPoints[[i]] - datCenter;
    {{y1 * y1, y1 * y2, y1 * y3},
     {y1 * y2, y2 * y2, y2 * y3},
     {y1 * y3, y2 * y3, y3 * y3}} /. {y1 → y[[1]], y2 → y[[2]], y3 → y[[3]]},
    {i, 1, numPoints}
  ] / numPoints;
MatrixForm[covMatrix]
```

Out[•]//MatrixForm=

$$\begin{pmatrix} 0.245153 & 0.0435489 & -0.143106 \\ 0.0435489 & 0.308863 & 0.0627695 \\ -0.143106 & 0.0627695 & 0.468494 \end{pmatrix}$$

In[•]:=
```
(* alternative Berechnung *)
covMatrix =
  Sum[
    Dot[Transpose[{datPoints[[i]] - datCenter}], {datPoints[[i]] - datCenter}],
    {i, 1, numPoints}
   ] / numPoints;
MatrixForm[covMatrix]
```

Out[•]//MatrixForm=

$$\begin{pmatrix} 0.245153 & 0.0435489 & -0.143106 \\ 0.0435489 & 0.308863 & 0.0627695 \\ -0.143106 & 0.0627695 & 0.468494 \end{pmatrix}$$

In[•]:=
```
(* Eigenwerte und Eigenvektoren *)
covEVectors = Eigenvectors[covMatrix];
covEValues = Eigenvalues[covMatrix];
covPValues = Sqrt[covEValues];
```

Einhüllender Ellipsoid mit internem Befehl

In[•]:=
```
pic2 =
 Graphics3D[
  {
   (* plotte den Ellipsoiden *)
   {
    Opacity[0.3], (* Durchsichtigkeit *)
    BoundingRegion[datPoints, "FastEllipsoid"] (* das ist der Ellipsoid *)
   },
   (* plotte die Punkte *)
   {
    Green,
    Table[Sphere[datPoints[[i]], 0.05], {i, 1, numPoints}]
   },
   (* plotte die Hauptachsen,
   die mittels der Kovarianzmatrix berechnet wurden *)
   {
    Cyan,
    Tube[{datCenter - 2 covPValues[[1]] * covEVectors[[1]],
      datCenter + 2 covPValues[[1]] * covEVectors[[1]]}, 0.04],
    Magenta,
    Tube[{datCenter - 2 covPValues[[2]] * covEVectors[[2]],
      datCenter + 2 covPValues[[2]] * covEVectors[[2]]}, 0.04],
    Yellow,
    Tube[{datCenter - 2 covPValues[[3]] * covEVectors[[3]],
      datCenter + 2 covPValues[[3]] * covEVectors[[3]]}, 0.04]
   }
  },
  PlotRange → {{-1.75, 1.75}, {-1.75, 1.75}, {-1.75, 1.75}},
  ImageSize → 72 × 3
 ]
```

Out[•]=