

# Uebungen ■ Exercices

## 12. Anonyme Funktionen ■ Fonctions anonymes

Die Gliederung dieses Kurses folgt in groben Zügen dem Buch von Nancy Blachman: A Practical Approach.... Hinweis: Kapitel 12 lesen!

- L'articulation de ce cours correspond à peu près à celle du livre de Nancy Blachman: A Practical Approach....  
Indication: Lire le chapitre 12.

Run mit WIN+*Mathematica* Version 5.2

- Testé avec *Mathematica* version 5.2+WIN

WIR94/98/99/2000/2007 // Copyright Rolf Wirz

### Aufgabe 1 ■ Problème 1

**Leseübung: Schreibe benannte Funktionen, die dasselbe ausgeben wie die folgenden anonymen Funktionen:**

- **Exercice de lecture: Ecris des fonctions dénommées qui sortent la même chose que les fonctions anonymes suivantes:**

(a) Anonym:  $(\#^3)\&$

- Anonyme:  $(\#^3)\&$

```
In[1]:= {(\#^3)\&[0],(\#^3)\&[1],(\#^3)\&[2],(\#^3)\&[3],(\#^3)\&[4]}
```

```
Out[1]= {0, 1, 8, 27, 64}
```

Benannt: ■ Dénommé:

```
In[2]:= f[x_]:=x^3;
        {f[0],f[1],f[2],f[3],f[4]}
```

```
Out[3]= {0, 1, 8, 27, 64}
```

(b) Anonym:  $(\#^{\#})\&$

- Anonyme:  $(\#^{\#})\&$

```
In[4]:= {(#^#)&[0],(#^#)&[1],(#^#)&[2],(#^#)&[3],(#^#)&[4],
        (#^#)&[5]}
```

Power::indet : Indeterminate expression 0<sup>0</sup> encountered. Mehr...

```
Out[4]= {Indeterminate, 1, 4, 27, 256, 3125}
```

Benannt: ■ Dénommmé:

```
In[5]:= Remove[f];
        f[x_]:=x^x;
        {f[0],f[1],f[2],f[3],f[4],f[5]}
```

Power::indet : Indeterminate expression 0<sup>0</sup> encountered. Mehr...

```
Out[7]= {Indeterminate, 1, 4, 27, 256, 3125}
```

Benannt: ■ Dénommmé:

```
In[8]:= f[x_]:=x^3;
        {f[0],f[1],f[2],f[3],f[4]}
```

```
Out[9]= {0, 1, 8, 27, 64}
```

```
In[10]:=
```

(c) Anonym: {#,#^2}&

■ Anonyme: (#^#^2)&

```
In[10]:= {{#,#^2}&[0],{#,#^2}&[1],{#,#^2}&[2],{#,#^2}&[3],
          {#,#^2}&[4],{#,#^2}&[5]}
```

```
Out[10]= {{0, 0}, {1, 1}, {2, 4}, {3, 9}, {4, 16}, {5, 25}}
```

Benannt: ■ Dénommmé:

```
In[11]:= Remove[f];
        f[x_]:=x,x^2;
        {f[0],f[1],f[2],f[3],f[4],f[5]}
```

```
Out[13]= {{0, 0}, {1, 1}, {2, 4}, {3, 9}, {4, 16}, {5, 25}}
```

(d) Anonym: If[#>0,#,-#]&

■ Anonyme: If[#>0,#,-#]&

```
In[14]:= {If[#>0,#,-#]&[-2],If[#>0,#,-#]&[-1],If[#>0,#,
          -#]&[0], If[#>0,#,-#]&[1],If[#>0,#,-#]&[2]}
```

```
Out[14]= {2, 1, 0, 1, 2}
```

Benannt: ■ Dénommmé:

```
In[15]:= Remove[f];
        f[x_]:=Abs[x];
        {f[-2],f[-1],f[0],f[1],f[2],f[3]}
```

```
Out[17]= {2, 1, 0, 1, 2, 3}
```

(e) Anonym:  $(\#/x \rightarrow y) \&$   
 ■ Anonyme:  $(\#/x \rightarrow y) \&$

```
In[18]:= {(#/.x->y)&[-2],(#/.x->y)&[-1],(#/.x->y)&[0],
          (#/.x->y)&[1],(#/.x->y)&[2],(#/.x->y)&[3],
          (#/.x->y)&[x],(#/.x->y)&[y],(#/.x->y)&[z]}
```

```
Out[18]= {-2, -1, 0, 1, 2, 3, y, y, z}
```

Benannt: ■ Dénomonné:

```
In[19]:= Remove[f];
          f[z_]:=z /. x->y;
          {f[-2],f[-1],f[0],f[1],f[2],f[3],f[x],f[y],f[z]}
```

```
Out[21]= {-2, -1, 0, 1, 2, 3, y, y, z}
```

## Aufgabe 2 ■ Problème 2

**Schreibe eine anonyme Funktion, die die 3. Potenz des Arguments berechnet.**  
**■ Ecris une fonction anonyme qui calcule la puissance 3 des arguments.**

Zum Beispiel  $(\#^3) \&$

■ Par exemple  $(\#^3) \&$

```
In[22]:= {(#^3)&[-3],(#^3)&[-2],(#^3)&[-1],(#^3)&[0],
          (#^3)&[1],(#^3)&[2],(#^3)&[3]}
```

```
Out[22]= {-27, -8, -1, 0, 1, 8, 27}
```

## Aufgabe 3 ■ Problème 3

**Verwende "Select" sowie eine anonyme Funktion, um aus einer Liste von Paaren, diejenigen Paare herauszusuchen, bei denen die erste Zahl grösser ist als die zweite.**

**■ Utilise "Select" ainsi qu'une fonction anonyme pour choisir dans une liste de paires les paires dont le premier nombre est plus grand que le deuxième.**

Ein Beispiel:

■ Un exemple:

```
In[23]:= newList = {{1,2},{2,1},{3,1},{2,3},{6,4},{4,7}};
          Select[newList,#[[1]] > #[[2]]&]
```

```
Out[24]= {{2, 1}, {3, 1}, {6, 4}}
```

## Aufgabe 4 ■ Problème 4

Definiere eine anonyme Funktion, um die Option "PlotLabel" bei den plotgenerierenden Kommandos zu setzen. Bilde die Funktion ab auf ContourPlot, DensityPlot, ParametricPlot, Plot, Plot3D.

■ Définis une fonction anonyme pour joindre l'option "PlotLabel" aux commandements qui génèrent des plots. "Applique" la fonction à ContourPlot, DensityPlot, ParametricPlot, Plot, Plot3D.

Was ist "PlotLabel"?

■ Qu'est "PlotLabel"?

```
In[25]:= ??PlotLabel
```

```
PlotLabel is an option for graphics functions that specifies an overall label for a plot.
Mehr...
```

```
Attributes[PlotLabel] = {Protected}
```

Wie wird es eingesetzt?

■ Comment l'applique-t-on?

```
In[26]:= ??Plot
```

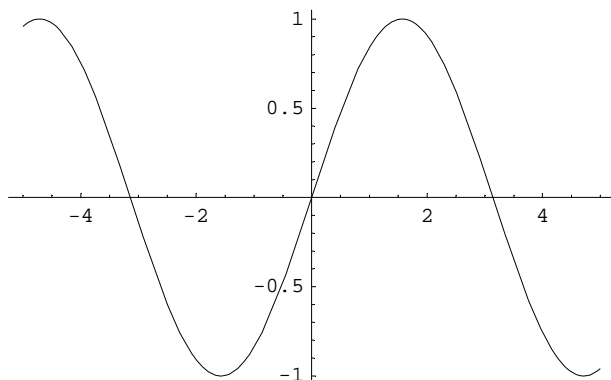
```
Plot[f, {x, xmin, xmax}] generates a plot of f as a function of x from xmin to
xmax. Plot[{f1, f2, ... }, {x, xmin, xmax}] plots several functions fi. Mehr...
```

```
Attributes[Plot] = {HoldAll, Protected}
```

```
Options[Plot] = {AspectRatio ->  $\frac{1}{\text{GoldenRatio}}$ , Axes -> Automatic, AxesLabel -> None,
  AxesOrigin -> Automatic, AxesStyle -> Automatic, Background -> Automatic, ColorOutput -> Automatic,
  Compiled -> True, DefaultColor -> Automatic, DefaultFont -> $DefaultFont,
  DisplayFunction -> $DisplayFunction, Epilog -> {}, FormatType -> $FormatType, Frame -> False,
  FrameLabel -> None, FrameStyle -> Automatic, FrameTicks -> Automatic, GridLines -> None,
  ImageSize -> Automatic, MaxBend -> 10., PlotDivision -> 30., PlotLabel -> None,
  PlotPoints -> 25, PlotRange -> Automatic, PlotRegion -> Automatic, PlotStyle -> Automatic,
  Prolog -> {}, RotateLabel -> True, TextStyle -> $TextStyle, Ticks -> Automatic}
```

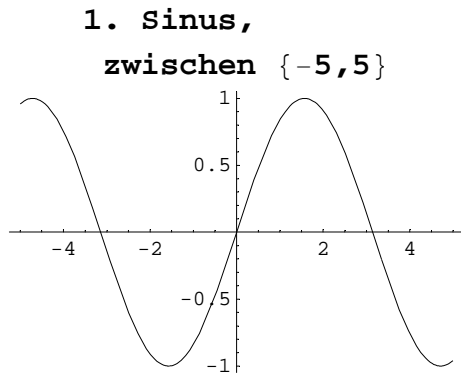
Ein Plot: ■ Un plot:

```
In[27]:= Plot[Sin[x],{x,-5,5}];
```



Mit Label: ■ Avec Label:

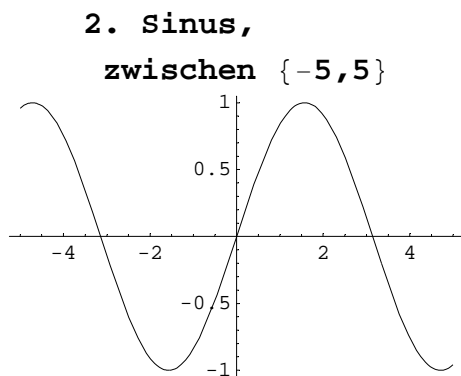
```
In[28]:= Plot[Sin[x],{x,-5,5},PlotLabel->FontForm[
  "1. Sinus,\n zwischen {-5,5}",
  {"Courier-Bold",15}]];
```



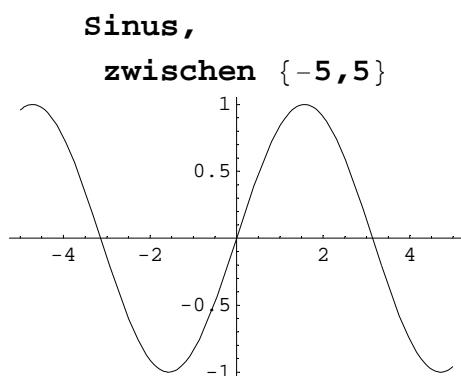
Mit Hilfe einer anonymen Funktion:

■ A l'aide d'une fonction anonyme:

```
In[29]:= (Plot[Sin[x],{x,-5,5},PlotLabel->#])&[FontForm[
  "2. Sinus,\n zwischen {-5,5}",
  {"Courier-Bold",15}]];
```



```
In[30]:= (Plot[Sin[x],{x,-5,5},PlotLabel->#])&[FontForm[
  "Sinus,\n zwischen {-5,5}",{"Courier-Bold",15}]];
```

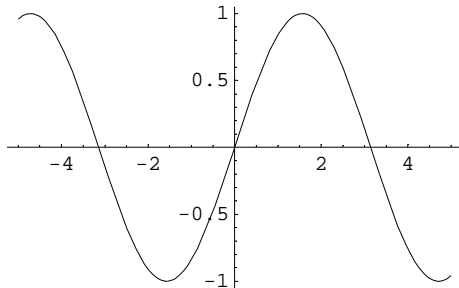


Etwas geschraubter:

■ Un peu plus sophistiqué:

```
In[31]:= Apply[Plot, {#[[1]],#[[2]],#[[3]]}&[{Sin[x],
{x,-5,5},(PlotLabel->#)}&[FontForm["3.
Sinus,\n zwischen {-5,5}","Courier-Bold",15]]];
```

### 3. Sinus, zwischen {-5,5}

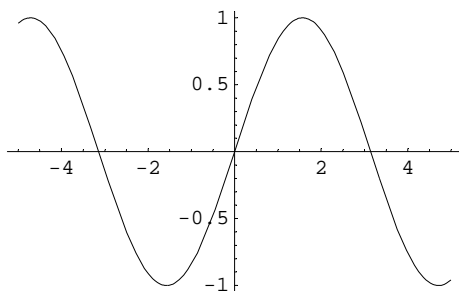


Noch etwas geschraubter:

■ Ecore plus sophistiqué:

```
In[32]:= Plot @@ {#[[1]],#[[2]],#[[3]]}&[{Sin[x],{x,-5,5},
(PlotLabel->#)}&[FontForm["4.
Sinus,\n zwischen {-5,5}","Courier-Bold",15]]];
```

### 4. Sinus, zwischen {-5,5}



Ein Versuch, die Options zu ändern:

■ Un essai de changer les options:

```
In[33]:= Options[Plot]
```

```
Out[33]= {AspectRatio ->  $\frac{1}{\text{GoldenRatio}}$ , Axes -> Automatic, AxesLabel -> None,
AxesOrigin -> Automatic, AxesStyle -> Automatic, Background -> Automatic,
ColorOutput -> Automatic, Compiled -> True, DefaultColor -> Automatic,
DefaultFont -> $DefaultFont, DisplayFunction -> $DisplayFunction, Epilog -> {},
FormatType -> $FormatType, Frame -> False, FrameLabel -> None, FrameStyle -> Automatic,
FrameTicks -> Automatic, GridLines -> None, ImageSize -> Automatic,
MaxBend -> 10., PlotDivision -> 30., PlotLabel -> None, PlotPoints -> 25,
PlotRange -> Automatic, PlotRegion -> Automatic, PlotStyle -> Automatic,
Prolog -> {}, RotateLabel -> True, TextStyle -> $TextStyle, Ticks -> Automatic}
```

```
In[34]:= Unprotect[Options[Plot]];
Drop[Options[Plot],{11}];
Options[Plot]=Union[Options[Plot],{PlotLabel->
FontForm["5. Neues Bild","Courier-Bold",15]},
Frame->True}];
(*Protect[Plot]*)
```

Protect::pspl : Options[Plot] is not a string, symbol, or list of strings and symbols. Mehr...

Set::write : Tag Plot in Options[Plot] is Protected. Mehr...

```
In[38]:= Attributes[Plot]
```

```
Out[38]= {HoldAll, Protected}
```

```
In[39]:= Options[Plot]
```

```
Out[39]= {AspectRatio ->  $\frac{1}{\text{GoldenRatio}}$ , Axes -> Automatic, AxesLabel -> None,
AxesOrigin -> Automatic, AxesStyle -> Automatic, Background -> Automatic,
ColorOutput -> Automatic, Compiled -> True, DefaultColor -> Automatic,
DefaultFont -> $DefaultFont, DisplayFunction -> $DisplayFunction, Epilog -> {},
FormatType -> $FormatType, Frame -> False, FrameLabel -> None, FrameStyle -> Automatic,
FrameTicks -> Automatic, GridLines -> None, ImageSize -> Automatic,
MaxBend -> 10., PlotDivision -> 30., PlotLabel -> None, PlotPoints -> 25,
PlotRange -> Automatic, PlotRegion -> Automatic, PlotStyle -> Automatic,
Prolog -> {}, RotateLabel -> True, TextStyle -> $TextStyle, Ticks -> Automatic}
```

```
In[40]:= Unprotect[Plot];
Drop[Options[Plot],{11}];
Options[Plot]=Union[Options[Plot],{PlotLabel->
FontForm["5. Neues Bild","Courier-Bold",15]},
Frame->True}];
(*Protect[Plot]*)
```

```
In[44]:= Attributes[Plot]
```

```
Out[44]= {HoldAll}
```

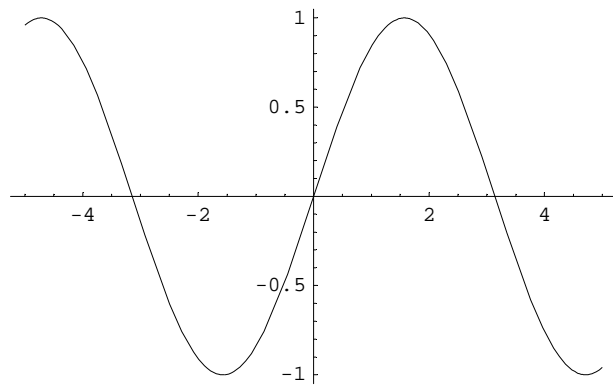
```
In[45]:= Options[Plot]
```

```
Out[45]= {AspectRatio ->  $\frac{1}{\text{GoldenRatio}}$ , Axes -> Automatic, AxesLabel -> None,
AxesOrigin -> Automatic, AxesStyle -> Automatic, Background -> Automatic,
ColorOutput -> Automatic, Compiled -> True, DefaultColor -> Automatic,
Epilog -> {}, Frame -> False, Frame -> True, FrameLabel -> None,
FrameStyle -> Automatic, FrameTicks -> Automatic, GridLines -> None,
ImageSize -> Automatic, MaxBend -> 10., PlotDivision -> 30., PlotLabel -> None,
PlotLabel -> FontForm[5. Neues Bild, {Courier-Bold, 15}],
PlotPoints -> 25, PlotRange -> Automatic, PlotRegion -> Automatic,
PlotStyle -> Automatic, Prolog -> {}, RotateLabel -> True, Ticks -> Automatic,
DefaultFont -> $DefaultFont, DisplayFunction -> $DisplayFunction,
FormatType -> $FormatType, TextStyle -> $TextStyle}
```

Was hat er nicht getan?

■ Que n'a-t-il pas fait?

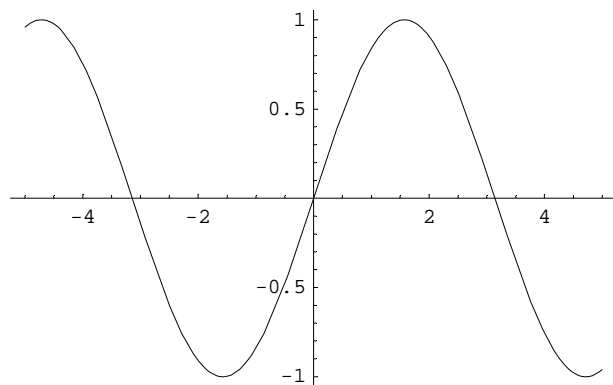
```
In[46]:= Plot[Sin[x],{x,-5,5}];
```



Wieso kam der Titel nicht?

■ Pourquoi le titre n'est-il pas apparu?

```
In[47]:= Show[%];
```



Uebungsfeld: ■ Terrain d'exercice:

```
In[48]:=
```

---

## Aufgabe 5 ■ Problème 5

**Schreibe nochmals eine Funktion, die die Häufigkeit eines Elements in einer Liste zählt.**

■ **Ecris encore une fonction qui compte la fréquence d'un élément dans une liste.**

Verwende Union, Count und Map.

■ Utilise Union, Count et Map.

Diese Funktion zählt die Häufigkeit von verschiedenen Elementen einer Liste:

■ Cette fonction compte la fréquence de différents éléments d'une liste:



```
In[48]:= frequenz[x_List]:= Module[{elemente=Union[x]},
      Map[{-#,Count[x,#]}&,elemente]];
frequenz::usage="frequenz[x_List] gibt eine Liste \
von verschiedenen Elementen zusammen \
mit ihrer Häufigkeit ---- \
donne une liste avec différents \
éléments avec leurs fréquences.";
testListe = Table[Random[Integer,{1,20}],{18}];
Sort[testListe]
```

Out[51]= {2, 3, 3, 6, 7, 8, 8, 9, 10, 15, 15, 16, 16, 18, 18, 19, 19}

```
In[52]:= frequenz[testListe]
```

```
Out[52]= {{2, 1}, {3, 2}, {6, 1}, {7, 1}, {8, 2},
          {9, 1}, {10, 1}, {15, 2}, {16, 2}, {18, 2}, {19, 3}}
```

```
In[53]:= ??frequenz
```

```
frequenz[x_List] gibt eine Liste von verschiedenen Elementen zusammen mit ihrer
Häufigkeit ---- donne une liste avec différents éléments avec leurs fréquences.

frequenz[x_List] := Module[{elemente = Union[x]}, ({#1, Count[x, #1]} &) /@elemente]
```

Probieren eigene Listen!

■ Essaie des listes à toi!

```
In[54]:=
```

## Aufgabe 6 ■ Problème 6

### Graphisches

#### ■ Graphiques

Eine Aufgabe in mehreren Schritten

■ Un problème en plusieurs étapes

**(a) Generiere eine Liste von 10 Pseudo-Zufallspunkten im Raum!**

■ Génère une liste de 10 points pseudo-aléatoires dans l'espace!

```
In[54]:= punkte = Partition[
      Table[Random[Integer,{1,20}],{30}],3]
```

```
Out[54]= {{9, 6, 10}, {11, 17, 18}, {7, 11, 12}, {6, 4, 12}, {7, 11, 9},
          {9, 1, 10}, {16, 11, 15}, {6, 15, 11}, {9, 1, 16}, {17, 9, 3}}
```

**(b) Generiere eine Liste von 10 Pseudo-Zufallszahlen für das Grau-Niveau!**

**■ Génère une liste de 10 points pseudo-aléatoires pour le niveau gris!**

```
In[55]:= grau = Table[Random[Real,{0,1}],{10}]
```

```
Out[55]= {0.292196, 0.647111, 0.279979, 0.029356,
          0.842525, 0.343356, 0.155026, 0.0516282, 0.709496, 0.358394}
```

**(c)**

```
In[56]:= graul=Partition[grau,1]
```

```
Out[56]= {{0.292196}, {0.647111}, {0.279979}, {0.029356}, {0.842525},
          {0.343356}, {0.155026}, {0.0516282}, {0.709496}, {0.358394}}
```

```
In[57]:= trans=Transpose[{punkte,graul}];trans
```

```
Out[57]= {{{9, 6, 10}, {0.292196}}, {{11, 17, 18}, {0.647111}},
          {{7, 11, 12}, {0.279979}}, {{6, 4, 12}, {0.029356}}, {{7, 11, 9}, {0.842525}},
          {{9, 1, 10}, {0.343356}}, {{16, 11, 15}, {0.155026}},
          {{6, 15, 11}, {0.0516282}}, {{9, 1, 16}, {0.709496}}, {{17, 9, 3}, {0.358394}}}
```

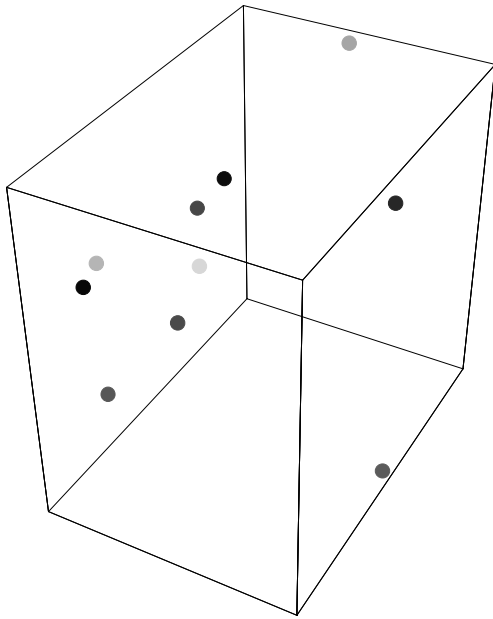
```
In[58]:= RemoveAll[mix];
mix=Map[Join#[[1]],#[[2]]&,trans]
```

```
Out[59]= {{9, 6, 10, 0.292196}, {11, 17, 18, 0.647111},
          {7, 11, 12, 0.279979}, {6, 4, 12, 0.029356}, {7, 11, 9, 0.842525},
          {9, 1, 10, 0.343356}, {16, 11, 15, 0.155026},
          {6, 15, 11, 0.0516282}, {9, 1, 16, 0.709496}, {17, 9, 3, 0.358394}}
```

```
In[60]:= menge=Map[({PointSize[0.03],GrayLevel#[[4]],
                    Point[#[[1]],#[[2]],#[[3]]]
                    })&,
                    mix];
menge
```

```
Out[61]= {{PointSize[0.03], GrayLevel[0.292196], Point[{9, 6, 10}]},
          {PointSize[0.03], GrayLevel[0.647111], Point[{11, 17, 18}]},
          {PointSize[0.03], GrayLevel[0.279979], Point[{7, 11, 12}]},
          {PointSize[0.03], GrayLevel[0.029356], Point[{6, 4, 12}]},
          {PointSize[0.03], GrayLevel[0.842525], Point[{7, 11, 9}]},
          {PointSize[0.03], GrayLevel[0.343356], Point[{9, 1, 10}]},
          {PointSize[0.03], GrayLevel[0.155026], Point[{16, 11, 15}]},
          {PointSize[0.03], GrayLevel[0.0516282], Point[{6, 15, 11}]},
          {PointSize[0.03], GrayLevel[0.709496], Point[{9, 1, 16}]},
          {PointSize[0.03], GrayLevel[0.358394], Point[{17, 9, 3]}}}
```

```
In[62]:= Show[Graphics3D[menge]];
```



Einige Erklärungen:

■ Quelques explications:

```
In[63]:= ??Show
```

Show[graphics, options] displays two- and three-dimensional graphics, using the options specified. Show[g1, g2, ... ] shows several plots combined. Mehr...

```
Attributes[Show] = {Protected}
```

```
In[64]:= ??Graphics3D
```

Graphics3D[primitives, options] represents a three-dimensional graphical image. Mehr...

```
Attributes[Graphics3D] = {Protected, ReadProtected}
```

```
Options[Graphics3D] =
```

```
{AmbientLight → GrayLevel[0], AspectRatio → Automatic, Axes → False, AxesEdge → Automatic,
AxesLabel → None, AxesStyle → Automatic, Background → Automatic, Boxed → True,
BoxRatios → Automatic, BoxStyle → Automatic, ColorOutput → Automatic, DefaultColor → Automatic,
DefaultFont → $DefaultFont, DisplayFunction → $DisplayFunction, Epilog → {},
FaceGrids → None, FormatType → $FormatType, ImageSize → Automatic, Lighting → True,
LightSources → {{1., 0., 1.}, RGBColor[1, 0, 0]}, {{1., 1., 1.}, RGBColor[0, 1, 0]},
{{0., 1., 1.}, RGBColor[0, 0, 1]}}, Plot3Matrix → Automatic, PlotLabel → None,
PlotRange → Automatic, PlotRegion → Automatic, PolygonIntersections → True,
Prolog → {}, RenderAll → True, Shading → True, SphericalRegion → False,
TextStyle → $TextStyle, Ticks → Automatic, ViewCenter → Automatic,
ViewPoint → {1.3, -2.4, 2.}, ViewVertical → {0., 0., 1.}}
```

```
In[65]:= ??DefaultColor
```

DefaultColor is an option for graphics functions which specifies the default color to use for lines, points, etc. Mehr...

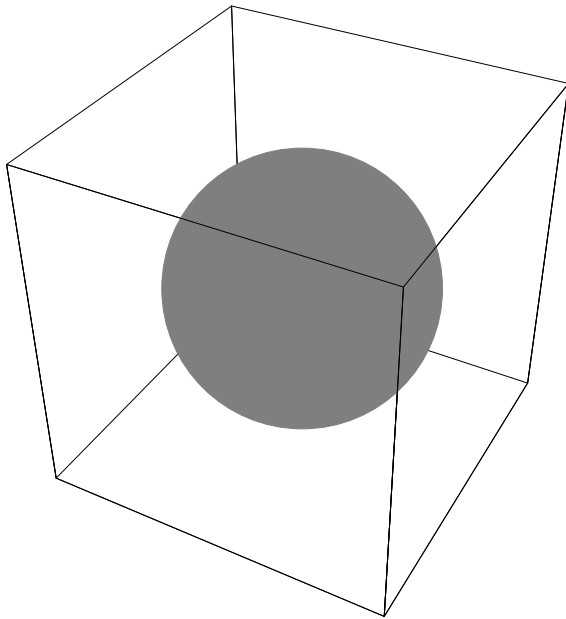
```
Attributes[DefaultColor] = {Protected}
```

```
In[66]:= ??GrayLevel
```

GrayLevel[level] is a graphics directive which specifies the gray-level intensity with which objects that follow should be displayed. Mehr...

```
Attributes[GrayLevel] = {Protected}
```

```
In[67]:= {PlotStyle->GrayLevel[1]};  
Show[Graphics3D[{PointSize[0.5],GrayLevel[0.5],  
Point[{1,1,1]}  
}]];
```



```
In[69]:= ??#
```

# represents the first argument supplied to a pure function. #n represents the nth argument. Mehr...

```
Attributes[Slot] = {NHoldAll, Protected}
```

## Aufgabe 7 ■ Problème 7

### Das Problem deines "signifikanten Partners"

#### ■ Le problème de ton "partenaire significatif"

Hier geht es um die Analyse einer besonderen Strategie zum Auffinden Deines "signifikanten Partners". Angenommen, da seinen  $n$  Personen des andern Geschlechts, die Du in Betracht ziehen willst. Du selbst bist noch ledig, jedoch heiratswillig. Jeder der  $n$  Personen ist ein unterscheidbarer Wert zugewiesen, den Du nicht kennst, bis Du das Individuum triffst. Jede Person kannst Du nur einmal treffen. Nach jedem Treffen musst Du entscheiden, ob das eben getroffene Individuum Dein "signifikanter Partner" ist und ihm dann einen Heiratsantrag stellen. Im Falle der Ablehnung gehst Du zum nächsten Treffen, wo dieselben Regeln gelten. Eine abgelehnte Person kannst Du nicht nochmals treffen, entsprechend den hier geltenden Gepflogenheiten.

Du sollst jetzt folgende Strategie analysieren: Du triffst  $(n/e)$  Partner als Testmenge,  $e =$  Eulersche Zahl. Nachdem Du jetzt einen Eindruck gewonnen hast, wählst Du den ersten Partner Deines Wunsches, der Dir besser gefällt als alle bisherigen Partner.

Wähle  $n = 30$  und lasse eine Simulation 50 mal laufen. Bezeichne im voraus einen besten Partner. Ein Lauf sei positiv, wenn Du den vorher bezeichneten besten Partner durch das Verfahren auch findest.

■

Il s'agit ici d'une stratégie spéciale pour trouver ton "partenaire significatif". Supposons  $n$  personnes de l'autre sexe que tu veux considérer. Toi-même tu es encore célibataire, mais décidé de te marier. A chacune des  $n$  personnes est attribuée une valeur qu'on peut distinguer, que tu ne connais pas, jusqu'à ce que tu rencontres l'individu. Tu ne peux rencontrer chaque personne qu'une seule fois. après chaque rencontre tu dois décider si l'individu que tu viens de rencontrer est ton "partenaire significatif" et lui faire ensuite une proposition de mariage. En cas de refus tu te rends au prochain rendez-vous, où valent les mêmes règles. Une personne refusée tu ne peux plus la rencontrer, selon les us et coutumes en vigueur ici.

Analyse la stratégie suivante: Tu rencontres  $(n/e)$  partenaires comme ensemble de test,  $e =$  nombre d'Euler. Après cette première impression tu choisis le partenaire qui te plaît le mieux que tous les précédents. Choisis  $n = 30$  et fais marcher la simulation 50 fois. Désigne à l'avance le meilleur partenaire. Un parcours soit positif si tu trouves le partenaire désigné à l'avance par cette méthode.

#### (a) Berechne den prozentualen Anteil der positiven Läufe.

##### ■ Calcule la partie proportionnelle des parcours positifs:

Definiere zuerst eine Funktion "zufall", die eine Liste von Leuten in zufälliger Reihenfolge wieder ausgibt.

■ Définis d'abord une fonction "hasard", qui sort une liste de personnes en ordre aléatoire.

Definiere zuerst die Funktion "zufall1". Diese Funktion stellt die Liste rekursiv zufällig um.

■ Définis d'abord la fonction "zufall1" ("hasard"). Cette fonction redonne la liste de façon récursive aléatoire.

```
In[70]:= zufall1[{}]:={};
          zufall1[personen_List]:=
            Block[{n=Random[Integer,{1,Length[personen]}]},
              Prepend[zufall1[Drop[personen,{n}]],
                personen[[n]]];
```

Einige Details zum Test:

■ Quelques détails quant au test:

```
In[72]:= personen={a,b,c,d};
        zufall1[personen]
```

```
Out[73]= {a, b, d, c}
```

```
In[74]:= n=Random[Integer,{1,Length[personen]}]
```

```
Out[74]= 2
```

```
In[75]:= n=Random[Integer,{1,Length[personen]}];
        Drop[personen,{n}]
```

```
Out[76]= {a, b, c}
```

```
In[77]:= zufall1[Drop[{a,b,c,d},{2}]]
```

```
Out[77]= {a, c, d}
```

Oder definiere dann die Funktion "zufall2". Diese Funktion tut dasselbe wie "zufall1":

■ Ou bien définis la fonction "zufall2". Cette fonction fait la même chose que "zufall1":

```
In[78]:= zufall2[{}]:={};
        zufall2[personen_List]:=
        Block[{n=Random[Integer,{0,Length[personen]-1}],
              shifted},
              shifted=RotateLeft[personen,n];
              Prepend[zufall2[Drop[shifted,1]],
                      shifted[[1]]];
```

Einige Details zum Test:

■ Quelques détails quant au test:

```
In[80]:= personen={a,b,c,d};
        zufall2[personen]
```

```
Out[81]= {d, a, c, b}
```

```
In[82]:= n=Random[Integer,{0,Length[personen]-1}]
```

```
Out[82]= 3
```

```
In[83]:= n=Random[Integer,{0,Length[personen]-1}];
        shifted=RotateLeft[personen,n]
```

```
Out[84]= {b, c, d, a}
```

```
In[85]:= n=Random[Integer,{0,Length[personen]-1}];
        shifted=RotateLeft[personen,n];
        Drop[shifted,1]
```

```
Out[87]= {a, b, c}
```

Oder definiere dann die Funktion "zufall" mit Hilfe der Funktion "Sort".

Hier werden Paare {a,b} benutzt, wobei a eine Zufallszahl und b die

Ordnungszahl der Person ist. Die Paare werden dann nach a geordnet. Diese Funktion tut dasselbe wie "zufall1":

■ Ou bien définis la fonction "zufall" à l'aide de la fonction "Sort". Ici on emploie des paires {a,b}, a étant le nombre aléatoire, b le nombre ordinal de la personne. Les paires sont rangées selon a. Cette fonction fait la même chose que "zufall1":

```
In[88]:= zufall[{}]:={};
          zufall[personen_List]:=
            Map[Last,
              Sort[Map[{Random[], #}&, personen],
                (#1[[1]] > #2[[1]])&
              ]];
          zufall::usage="zufall[list] gibt eine zufällige \
            Permutation der Elemente der Liste \
            --- donne une permutation aléatoire \
            des éléments de la liste."
```

```
Out[90]= zufall[list] gibt eine zufällige Permutation der Elemente der
          Liste --- donne une permutation aléatoire des éléments de la liste.
```

Einige Details zum Test:

■ Quelques détails quant au test:

```
In[91]:= personen={a,b,c,d};
          zufall[personen]
```

```
Out[92]= {c, d, a, b}
```

```
In[93]:= {Random[], #}&[personen]
```

```
Out[93]= {0.0532293, {a, b, c, d}}
```

```
In[94]:= Map[{Random[], #}&, personen]
```

```
Out[94]= {{0.0669755, a}, {0.648717, b}, {0.148611, c}, {0.232326, d}}
```

```
In[95]:= Sort[Map[{Random[], #}&, personen],
            (#1[[1]] > #2[[1]])&]
```

```
Out[95]= {{0.494212, c}, {0.463157, a}, {0.397076, d}, {0.113987, b}}
```

```
In[96]:= Map[Last, Sort[Map[{Random[], #}&, personen],
                        (#1[[1]] > #2[[1]])& ]]
```

```
Out[96]= {a, c, b, d}
```

Gib nun den Personen statt Namen Ordnungszahlen von 1 bis n. 1 bedeutet "am meisten bevorzugt" etc..

■ Donne aux personnes au lieu de noms des nombres ordinaux de 1 jusqu'à n. 1 signifie "préféré le plus" etc..

```
In[98]:= n = 30;
          personen = zufall[Range[n]]
```

```
Out[99]= {24, 2, 18, 27, 4, 11, 1, 30, 5, 9, 15, 8, 13, 10,
          12, 29, 3, 16, 23, 25, 21, 19, 17, 6, 26, 22, 7, 20, 14, 28}
```

Die Funktion "wahlVorzug" soll nun n Personen nehmen und diese in zufällige Reihenfolge bringen. Sie soll die ersten (n/e) (gerundet) Personen nehmen und den Wert der meist bevorzugten Person in dieser Menge in die Variable mmin speichern. Der Wert der ersten noch mehr bevorzugten Person, die dann folgt, soll in die Variable wahl gespeichert werden. Wenn wahl eine Zahl grösser gleich 1 enthält, soll wahlVorzug diese Zahl als Element einer Menge ausgeben. Andernfalls soll die leere Menge kommen.

■ La fonction "wahlVorzug" prene maintenant n personnes et les mette dans un ordre aléatoire. Qu'elle prene les premières (n/e) (arrondi) personnes et qu'elle mémorise dans la variable mmin la valeur de la personne préférée de cet ensemble. La valeur de la personne préférée encore davantage, qui suit, doit être mémorisée dans la variable wahl. Si

wahl contient un nombre plus grand ou égal 1, wahl doit sortir ce nombre comme élément d'un ensemble. Autrement doit apparaître l'ensemble vide.

```
In[100]:=
  wahlVorzug[n_Integer]:=
    Module[{personen=zufall[Range[n]],
            m=Round[N[n/E]],
            mmin, wahl},
            mmin=Min[Take[personen,m]];
            wahl=Select[personen, (#<mmin)&];
            If[Length[wahl]>0,First[wahl],{}]];
  anzahlPersonen=30;
  anzahlLauf=50;
  resultat=Table[wahlVorzug[anzahlPersonen],
                {anzahlLauf}]
```

```
Out[103]=
  {{}, 1, {}, {}, 1, 1, 2, 1, 1, {}, {}, {}, 3, {}, {}, 1, 2, 1, {}, 4, {}, {}, {}, {}, {},
  6, 1, {}, {}, 3, 2, {}, 1, 1, {}, 1, 2, 3, {}, {}, 2, 1, {}, {}, 1, 4, 1, {}, 1, 1}
```

Anzahl positive Simulationen mit Resultat 1, 2 oder 3:

■ Nombre de simulations avec le résultat 1, 2 ou 3:

```
In[104]:=
  posSim1={Map[(N[Count[resultat,#]/anzahlLauf])&,
              {1,2,3}]}
```

```
Out[104]=
  {{0.32, 0.1, 0.06}}
```

Kürzer mit Vergleich mit (1/e):

■ Plus brièvement avec la comparaison avec (1/e):

```
In[105]:=
  posSim1={ (N[Count[resultat,#]/anzahlLauf
             1])&/@{1,2,3},N[1/E]}
```

```
Out[105]=
  {{0.32, 0.1, 0.06}, 0.367879}
```

### (b) Prozentualer Anteil der Läufe, die mit einem signifikanten Partner enden:

#### ■ Partie propositionnelle des parcours qui terminent par un partenaire significatif:

Studiert man den Output, so sieht man, dass meistens die Anzahl der Fälle, in denen die leere Menge herauskommt, sehr gering ist. Dann hat die Strategie keinen Erfolg. Falls die Strategie eine Zahl ergibt, so wollen wir von Erfolg insofern reden, dass dann eine kleine Nummer erwartet werden kann. Wir sagen dann, es hätte sich ein "signifikanter Partner" finden lassen. Wie oft kommt das prozentual vor?

■ Si on étudie l'output, on voit que pour la plupart le nombre des cas où sort l'ensemble vide et très petit. Dans ces cas la stratégie n'a pas de succès. Si la stratégie donne un nombre, nous pouvons nous attendre à un nombre bas. Nous dirons qu'un "partenaire significatif" a pu être trouvé. Dans quel pourcentage cela arrive-t-il?



```
In[106]:=
    signifikant = 1- N[Count[resultat,{}]/50]
```

```
Out[106]=
    0.54
```

### c) Wie gut oder schlecht war die Auswahl nun?

#### ■ Combien le choix était donc bon ou mauvais?

Wir berechnen, in wievielen Läufen prozentual die Simulation mit einer Person endete, die unter "den favorisierten 10%" einzureihen ist.

■ Nous calculons dans combien de parcours la simulation s'est terminée par une personne qu'on a pu placer parmi les "10 % favorisés".

```
In[107]:=
    signifikant10Prozent =
        N[Length[Select[resultat,
            (#<=Round[N[anzahlPersonen 0.1]])&]]/50]
```

```
Out[107]=
    0.48
```

### (d) Bei einer grösseren Laufzahl:

#### ■ Lors d'un nombre supérieur de parcours:

```
In[108]:=
    anzahlLauf=50;
    resultat=Table[wahlVorzug[anzahlPersonen],
        {anzahlLauf}];
    posSim1={Map[(N[Count[resultat,#]/anzahlLauf])&,
        {1,2,3}]}];
    signifikant = 1- N[Count[resultat,{}]/50];
    signifikant10Prozent =
        N[Length[Select[resultat,
            (#<=Round[N[anzahlPersonen 0.1]])&]]/50];
    Print["resultat = ", resultat];
    Print["posSim1      = ", posSim1];
    Print["signifikant   = ", signifikant];
    Print["signifikant10Prozent = ",
        signifikant10Prozent]

    resultat = {2, 1, {}, 1, {}, {}, {}, 5, 2, {}, 1, 1, {}, {}, 1, {}, 1, 1, {}, {}, 1, 8, {},
        {}, 1, 1, 3, {}, 1, 1, 1, {}, 1, 6, 1, 2, {}, 2, {}, 1, {}, {}, 1, 3, 2, {}, {}, 1, 1, 2}

    posSim1      = {{0.38, 0.12, 0.04}}

    signifikant   = 0.6

    signifikant10Prozent = 0.54
```

## "Putzmaschine" einsetzen

### ■ Employer la "machine de nettoyage"

```
In[117]:=
    (* Old Form: Remove["Global`@*"] *)
```

```
In[118]:=  
  Remove["Global`*"]
```