DecisionModels
Erläuternde Beispiele zum Paket

Das Paket *DecisionModels* laden:

```
Needs["DecisionModels`DecisionModels`"]
```

1. Datenaufbereitung

Entscheidungsmatrix

A sei die Menge der möglichen Auflagen für ein zu verlegendes Buch, B die Menge der Absatzzahlen, g die Gewinnfunktion des Verlegers:

```
A = {5000, 7000, 9000};
B = {4000, 5000, 6000, 7000, 8000, 9000};
g[x_, y_] := Min[x, y] * 15 - 10 x - 10000;
```

Berechnung der Konsequenzen-Matrix (hier auch: Entscheidungsmatrix):

```
cmat = DecisionMatrix[g, A, B];
cmat // MatrixForm
```

$$\begin{pmatrix} 0 & 15000 & 15000 & 15000 & 15000 & 15000 \\ -20000 & -5000 & 10000 & 25000 & 25000 & 25000 \\ -40000 & -25000 & -10000 & 5000 & 20000 & 35000 \end{pmatrix}$$

Normierung

Die Skalentransformation bildet das Intervall [3;7] (monoton wachsend) auf das Intervall [1;17] ab:

```
ScaleTransform[{3, 7}, {1, 17}][x]
```

$-11 + 4\,x$

Die Skalennormierte bildet das Intervall [3;7] (monoton wachsend) auf das Einheitsintervall [0;1] ab:

```
ScaleNorm[3, 7][x]
```

$-\frac{3}{4} + \frac{x}{4}$

Der Einzelwert 5 wird der eben berechneten Skalennormierten unterworfen:

```
NormalizeValue[5, {3, 7}]
```

$\frac{1}{2}$

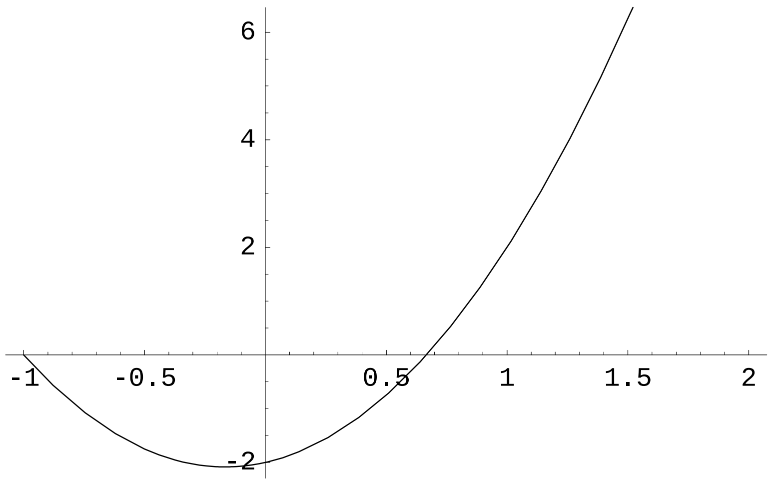
Eine Menge von Werten wird in Bezug auf das von Minimum und Maximum gebildete Intervall normiert:

```
cmat1 = NormalizeRange[cmat];
cmat1 // MatrixForm
```

$$\begin{pmatrix} \frac{8}{15} & \frac{11}{15} & \frac{11}{15} & \frac{11}{15} & \frac{11}{15} & \frac{11}{15} \\ \frac{4}{15} & \frac{7}{15} & \frac{2}{3} & \frac{13}{15} & \frac{13}{15} & \frac{13}{15} \\ 0 & \frac{1}{5} & \frac{2}{5} & \frac{3}{5} & \frac{4}{5} & 1 \end{pmatrix}$$

Definition einer Funktion und Darstellung ihres Graphen:

```
f[x_] := -2 + x + 3 x^2;
Plot[f[x], {x, -1, 2}];
```

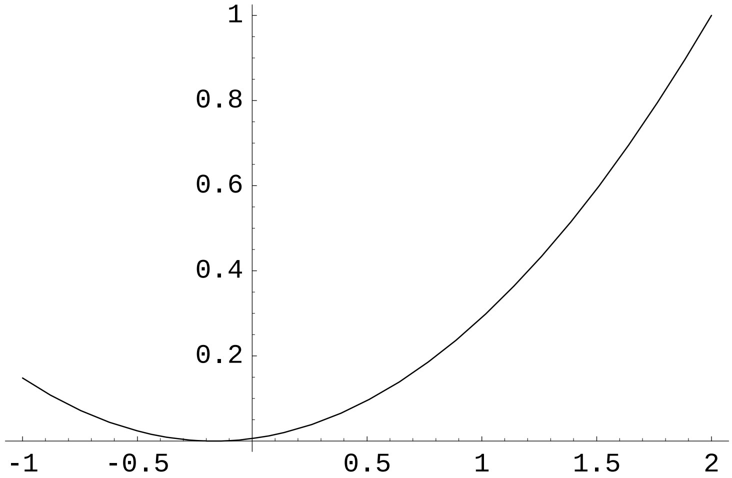


(Angenäherte) Transformation in eine Funktion mit Wertbereich [0;1]:

```
fnorm = NormalizeFunction[f, {-1, 2}];
```

```
fnorm[x]
Plot[fnorm[x], {x, -1, 2}, PlotRange → All];
```

$$\frac{2500000000\left(\frac{208333333}{2500000000}+x+3x^2\right)}{35208333333}$$



2. Nutzenfunktionen

Erzeugung diverser Funktionstypen

Erzeugung einer linearen Nutzenfunktion, die über dem Intervall [5;10] normiert ist:

```
LinUtility[2, -5, Normalize → {5, 10}][x]
```

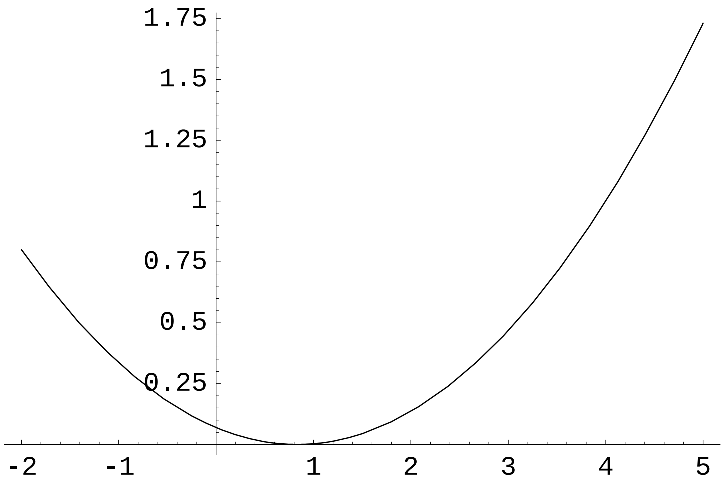
$$\frac{10-x}{5}$$

Erzeugung einet quadratischen Nutzenfunktion, die über dem Intervall [-1;4] normiert ist:

```
QuadUtility[2, -5, 3, Normalize → {-1, 4}][x]
```

$$\frac{12}{361}\left(\frac{25}{12} - 5\,x + 3\,x^2\right)$$

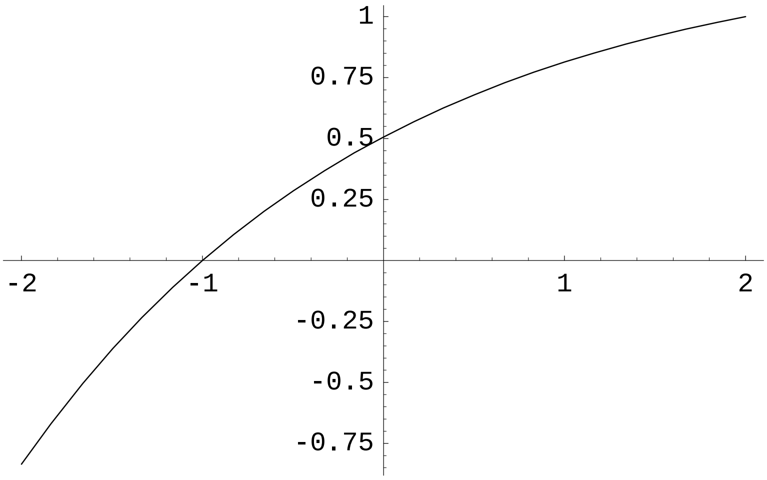
```
Plot[%, {x, -2, 5}];
```



Generierung und Darstellung einer über [-1;2] normierten exponentiellen Nutzenfunktion:

```
ExpUtility[1, -2, -0.5][x]
Plot[ExpUtility[1, -2, -0.5, Normalize → {-1, 2}][x], {x, -2, 2}];
```

$$1 - 2\,e^{-0.5\,x}$$



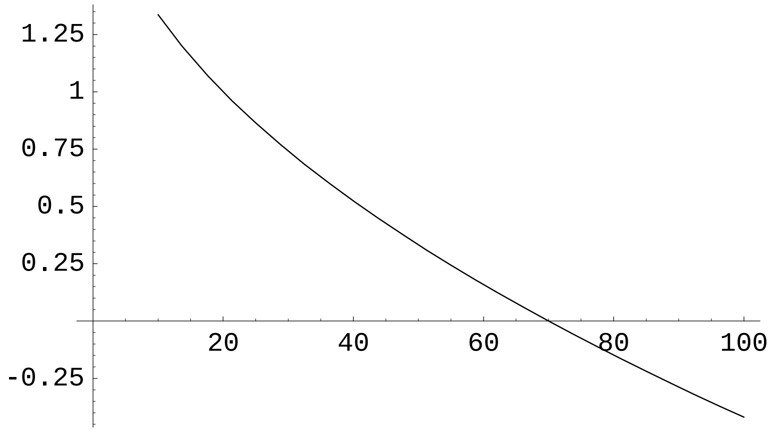
Eine nicht-normierte Nutzenfunktion vom Quadratwurzeltyp:

```
SqrtUtility[2, -0.3][x]
```

```
2 - 0.3 √x
```

Diesselbe Funktion in über [20;70] normierter Gestalt:

```
Plot[SqrtUtility[2, -0.3, Normalize → {20, 70}][x], {x, 10, 100}];
```



Bernoullis logarithmischer Nutzen (normiert für Geldbeträge von 1 bis 100):

```
LogUtility[1, 1, Normalize → {1, 100}][x]
```

$$\frac{\mathrm{Log}[x]}{\mathrm{Log}[100]}$$

Mischung von Nutzenwerten

Eine Reihe von Nutzenwerten (utab) wird auf den Maximalwert umax bezogen und mit dem Gewichtsvektor (gvek) gemischt:

```
utab = {5, 7, 4, 8, 8, 7, 5, 10, 8};
umax = 10;
gvek = {5, 3, 5, 3, 2, 2, 2, 4, 3};
```

```
CompoundUtility[utab, umax, gvek]
```

$$\frac{97}{145}$$

3. Zahlungsreihen

Barwert

Zahlungsreihe: c, Vektor der Diskontierungsfaktoren: df

```
c = {-300, 105, 105, 150};
df = {1.025, 1.035, 1.045};
```

```
PresentValue[c, df]
```

```
36.718
```

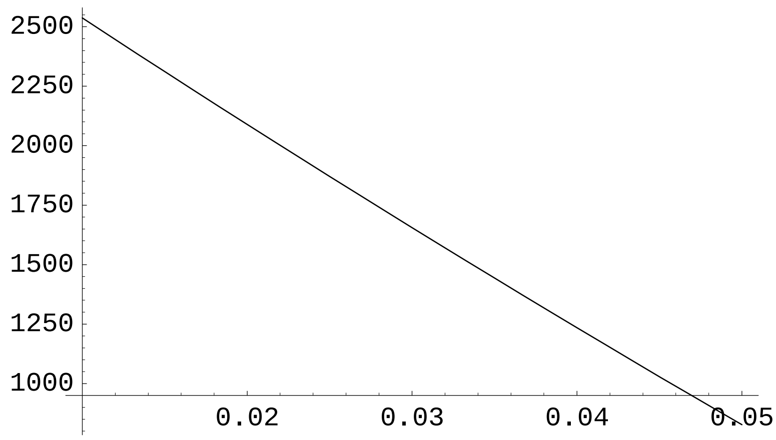
Barwert der Zahlungsreihe c bei kalkulatorischem Zinssatz r:

```
c = {-20000, 4000, 14000, 5000};
r = 0.0035;
PresentValueAtRate[c, r]
```

```
2836.43
```

Verlauf des Barwerts im Zinsintervall von 1 % bis 5 %:

```
PresentValueGraph[c, 0.01, 0.05]
```



Rendite-Polynom und Rendite

Das Rendite-Polynom der Zahlungsreihe:

```
InterestRatePoly[c][x]
```

```
-1000 (-3 + 38 x + 56 x^2 + 20 x^3)
```

Die Menge seiner reellen Nullstellen:

```
InterestRateRealZeros[c]
```

```
{0.0712711}
```

Interne Zinsrate (Rendite) als Nullstelle des Rendite-Polynoms:

```
InternalRate[c]
```

```
0.0712711
```

Eindeutige Existenz des internen Zinssatzes

Eine Zahlungsreihe $c_0, c_1, \ldots, c_n$ mit positiver Nettosumme $c_0 + c_1 + \ldots + c_n$ besitzt eine eindeutig bestimmte Rendite, wenn der "transformierte Zahlungsstrom" $\tilde{c}_0, \tilde{c}_1, \ldots, \tilde{c}_n$ mit $\tilde{c}_k := \sum_{j=0}^{k} \binom{n-j}{k-j} c_j$ genau einen Vorzeichenwechsel aufweist (hinreichendes Kriterium; vgl. P. Bender: Der interne Zinssatz bei beliebigen Investitionen, Preprint Universität Paderborn, 1990):

```
PositiveInvestmentQ[{-100, 20, -50, 70, 80}]
```

```
True
```

```
PositiveInvestmentQ[{1, 4, -7, 6, 1, -1}]
```

```
False
```

4. Entscheidungsmethoden

Vier Hilfsfunktionen

Optimismus-Parameter $\lambda \in [0; 1]$ für das Hurwicz-Verfahren:

```
lambda = 0.5;
```

Die Wertfunktion nach Hurwicz, angewandt auf einen Vektor von Nutzenwerten:

```
HurwiczValue[{-3, 0, 7, 4, 2}, lambda]
```

```
2.
```

Gewöhnlicher Erwartungswert (Summe 1 der Wahrscheinlichkeitsverteilung wird überprüft):

```
Expectation[{1, 2, 3, 4, 5, 6}, {0.1, 0.1, 0.2, 0.1, 0.2, 0.3}]
```

```
4.1
```

Nutzenerwartungswert bei gegebener Nutzenfunktion (Übergabe als reine Funktion) sowie Wahrscheinlichkeitsverteilung prob (Summe 1 wird überprüft):

```
prob = {0.1, 0.15, 0.15, 0.3, 0.2, 0.1};
ExpectedUtility[LogUtility[1, 1, Normalize → {1, 100}], {1, 2, 3, 4, 5, 6}, prob]
```

```
0.257475
```

Noch einmal das Beispiel aus Abschnitt 1:

```
cmat // MatrixForm
```

$$\begin{pmatrix} 0 & 15000 & 15000 & 15000 & 15000 & 15000 \\ -20000 & -5000 & 10000 & 25000 & 25000 & 25000 \\ -40000 & -25000 & -10000 & 5000 & 20000 & 35000 \end{pmatrix}$$

Die resultierende Matrix der "Bedauernswerte" (nach L. Savage):

```
RegretMatrix[cmat] // MatrixForm
```

$$\begin{pmatrix} 0 & 0 & 0 & -10000 & -10000 & -20000 \\ -20000 & -20000 & -5000 & 0 & 0 & -10000 \\ -40000 & -40000 & -25000 & -20000 & -5000 & 0 \end{pmatrix}$$

Entscheidungen bei "völligem Nichtwissen"

Die Lösungsmenge nach dem Maximin-Verfahren (entspricht Hurwicz mit $\lambda = 0$). Ausgegeben werden die Indizes der Alternativen (bzw. Strategien):

```
MaximinSolve[cmat]
```

```
{1}
```

Die Lösungsmenge nach dem Maximax-Verfahren (entspricht Hurwicz mit $\lambda = 1$):

```
MaximaxSolve[cmat]
```

```
{3}
```

Die Lösungsmenge nach der Hurwicz-Methode mit $\lambda = 0.7$:

```
HurwiczSolve[cmat, 0.7]
```

```
{3}
```

Die Lösungsmenge nach der Savage-Methode:

```
SavageSolve[cmat]
```

```
{1, 2}
```

Die Lösungsmenge nach der Laplace-Methode (implizite Annahme von Gleichwahrscheinlichkeit der zur Auswahl stehenden Alternativen):

```
LaplaceSolve[cmat]
```

```
{1}
```

Entscheidungen bei Risiko

Die Lösungsmenge nach dem Bernoulli-Verfahren (explizite Einbeziehung von Wahrscheinlichkeiten für die Umweltbedingungen, siehe oben):

```
BernoulliSolve[cmat, prob]
```

```
{2}
```

5. Risikoanalyse

Ausgangsdaten einer Lotterie (Beispiel)

Die Wahrscheinlichkeitsverteilung der Lotterie:

```
prob = {1 / 10^5, 1 / 10^4, 1 / 10^3, 1 - (1 / 10^5 + 1 / 10^4 + 1 / 10^3)} // N
```

```
{0.00001, 0.0001, 0.001, 0.99889}
```

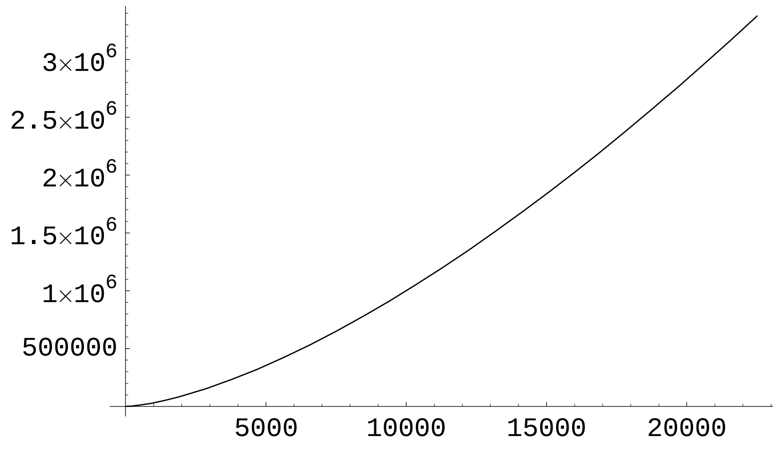
Die möglichen Reingewinne beim 1., 2., 3. Preis und Nicht-Gewinn (Lospreis 5 Euro):

```
c = {22500 - 5, 6000 - 5, 100 - 5, 0 - 5};
```

Die Nutzenfunktion des Lotterie-Spielers Timo:

```
uTimo[x_] := (x + 5) ^ (3 / 2)
```

```
Plot[uTimo[x], {x, -5, 22500}];
```



Analyse

Timo's Sicherheitsäquivalent für die Teilnahme an der Lotterie:

```
CertaintyEquivalent[uTimo, {-5, 22500}, c, prob]
```

```
13.7555
```

Die zugehörige Risikoprämie, die zeigt, dass Timo risikofreudig ist:

```
RiskPremium[uTimo, {-5, 22500}, c, prob]
```

```
-17.8305
```

Das absolute und das relative Risikoeinstellungsmaß (als Funktion nach Arrow und Pratt):

```
AbsoluteRiskFunction[uTimo][x]
```

$$-\frac{1}{2\,(5+x)}$$

```
RelativeRiskFunction[uTimo][x]
```

$$-\frac{x}{2\,(5+x)}$$

6. Zweipersonenspiele

Spielmatrix

Die Matrix-Normalform g (Auszahlungsmatrix) eines Zweipersonen-Nullsummenspiels:

```
Clear[g]
```

```
g = {{1, 1, -2, -2}, {-1, 2, -1, 2}, {0, 2, 0, 2}}
```

```
{{1, 1, -2, -2}, {-1, 2, -1, 2}, {0, 2, 0, 2}}
```

```
% // MatrixForm
```

$$\begin{pmatrix} 1 & 1 & -2 & -2 \\ -1 & 2 & -1 & 2 \\ 0 & 2 & 0 & 2 \end{pmatrix}$$

Dominanz-Beziehungen

Strategie A2 wird dominiert (von Strategie A3):

```
ReduceLines[g]
```

```
{{1, 1, -2, -2}, {0, 2, 0, 2}}
```

```
% // MatrixForm
```

$$\begin{pmatrix} 1 & 1 & -2 & -2 \\ 0 & 2 & 0 & 2 \end{pmatrix}$$

Strategie B3 ist dominant (dominiert alle übrigen Strategien von Spieler B):

```
ReduceColumns[g]
```

```
{{-2}, {-1}, {0}}
```

```
% // MatrixForm
```

$$\begin{pmatrix} -2 \\ -1 \\ 0 \end{pmatrix}$$

Wert des Spiels, Sattelpunktlösung

Kleinster Wert des Spiels und Maximin-Strategien (für Spieler A):

```
MaximinSet[g]
```

```
{0, {3}}
```

Größter Wert und Minimax-Strategien (für Spieler B):

```
MinimaxSet[g]
```

```
{0, {3}}
```

Der Wert des Spiels und das Paar (hier existierender) optimaler Gleichgewichtsstrategien für Spieler A und B:

```
SaddlepointSolution[g]
```

```
{0, {{3, 3}}}
```

Lösung in gemischten Strategien

Ein einfaches Beispiel aus S. Vajda: Theorie der Spiele und Linearprogrammierung (Walter de Gruyter: Berlin 1962, S. 128):

```
gvajda = {{4, 1, 3}, {2, 3, 4}};
```

```
% // MatrixForm
```

$$\begin{pmatrix} 4 & 1 & 3 \\ 2 & 3 & 4 \end{pmatrix}$$

Ausgabe von: Wert / optimale Strategie von A / optimale Strategie von B:

```
MixedStrategySolution[gvajda]
```

$$\left\{\frac{5}{2}, \left\{\frac{1}{4}, \frac{3}{4}\right\}, \left\{\frac{1}{2}, \frac{1}{2}, 0\right\}\right\}$$