Chapter 8 of
Voting Theory for Democracy

Using The Economics Pack
Applications of Mathematica
for Direct Single Seat Elections

Thomas Cool, February 2001

http://www.dataweb.nl/~cool

Summary: The following contains Chapter 8 of the book "Voting Theory for Democracy" by Thomas Cool, ISBN 90-804774-3-5

Published by Thomas Cool *Consultancy & Econometrics*

Rotterdamsestraat 69, NL-2586 GH Scheveningen, The Netherlands

http://www.dataweb.nl/~cool

8. Measuring utility

8.1 Introduction

8.1.1 Introduction

The situation thus is as follows:

- Since there are no obvious objective measures for cardinal utility, we have to ask people for their opinion; since people can cheat, we try to limit their options; and then we end up with the paradoxes of voting.
- Above chapters have shown that the paradoxes of voting only *seem* contraditions but are no real contraditions. Which means that we can live with them.

Yet, it could be fruitful to work into the other direction, and to see whether we could develop more acceptable measures for cardinal utility.

The economic literature contains the suggestion, in some important places, that such a measure might be found by experiments in which probability plays a role. This brings us to the subjects of Prospects and Certainty Equivalence. Above, we already looked at games and matches, but we have somewhat neglected the question of the Prize of the match. It was just win or lose, without much utility attached to it. The following deals better with that by including the utilities attached to Profit and Loss.

The discussion below will show, unfortunately, that recovering cardinal utility is still no easy feat. Sometimes people risk their lives to save others, but economic theory still cannot say that there is cardinal comparison in this. So the main conclusion of the following chapters is negative. It does not seem possible, yet, to determine cardinal utility, free of cheating. Above schemes of voting hence cannot be replaced by simply adding (or Nash multiplying) of such utilities. The positive side of this conclusion is that above chapters are not useless.

```
ResetAll
```

```
Economics[Probability, Risk]
```

Cool`Probability`

Bayes	NumberOfElements
Bordered2DPrSolve	NumberOfFailures
Bordered2DPrToConditionals	Odds
Bordered2DPrToEquations	Options$SequentialDraws
ConditionalPrForm	OrRule
CPrToFunction	PrDomain
CreateProspect	Prob
Draw	Prospect
FromConditional	ProspectApply
FromConditionalRule	ProspectEV
FromOdds	ProspectInnerSort
FromProb	ProspectQ
IDProspectsPrMatrix	ProspectReList
JoinIDProspects	ProspectUtility
JointProspect	PutIn
JointProspectEV	SequentialDraws
JointProspectPrValue	TakeOut
JointToMarginalProspects	ToConditional
JointToProspects	ToConditionalRule
Laplace	ToExpectedUtility
LevelToPr	ToProb
LevelToSum	ToUtility
NormalisePr	UnitPr

Cool`Risk`

AllaisParadox	RiskAversion
BinaryRiskSimplify	RiskEquations
CertaintyEq	Risket
ConditionalRisk	RiskFactors
IfNegativeToMaxRule	RiskModel
IfNegativeToMinRule	RiskPr
Options$CertaintyEq	RiskPremium
Prospect2DSimplify	RiskRatio
Prospect3DPrTriangle	RiskRatioSplit
ProspectInverse	RiskRatioSplitPlot
ProspectPlot	Risks
ProspectProjection	RiskyQ
ProspectPrValue	SpreadAversion
ProspectSort	SpreadPremium
ProspectSplit	ToProspect
ProspectSplitPlot	ToRisket
ProspectStatistics	Uncertainty
ProspectTo3D	UncertaintyAversion
RelativeRisk	UncertaintyDefinitionsPlot
RelativeRiskEquations	UncertaintyPremium
Risk	

8.1.2 Structure of the discussion

We have to develop notions of probability and risk before we can deal with certainty equivalence. The definition of risk is mine, and I also develop a new (non-standard) model for certainty equivalence, using this risk measure. The *Mathematica* routines provide an accessible way to verify that these new approaches are sound. We will take examples from Luenberger (1998) and Mas-Colell c.s. (1995:171) so that it is also quickly verified that the routines are sound.

If you are interested in extending on the issues in the following chapters, then you should be aware that Cool (1999, 2001), "The Economics Pack", contains a large number of additional routines on probability and risk that we will not discuss here. You are advised to first look into that Pack before you start programming, since this can save you a lot of work.

8.2 Development of probability

This package implements basic probability theory. The Prospect object allows easy handling of discrete probability situations, and Draw helps you to find a proper playing strategy.

```
Economics[Probability]
```

8.2.1 The Pr object

With Pr[A] the probability of event A, we find for two events A and B:

- the joint probability Pr[A, B] = Pr[A & B] = Pr[$A \cap B$]
- Pr[A or B] = Pr[$A \cup B$] = Pr[A] + Pr[B] - Pr[$A \cap B$]
- the conditional probability of A given B: Pr[$A \mid B$] = Pr[A, B] / Pr[B]
- the probability of the ordered event of first A and then B is Pr[{A, B}]

Since *Mathematica* has different uses for semicolon ";" and the vertical line "|", we represent conditional probability as ConditionalPr[A][B]. We can use Prob[$A \mid B$] as an output printing facility however.

Note that some textbooks interprete "either A or B" as inclusive-or. Here we will use normal English where it is exclusive-or.

`Pr[x__]`	gives the joint probability of events x
`ConditionalPr[x__][y__]`	gives the conditional probability of events x given events y
`Prob[x__]`	is a format for probability that humans can read better, but it is less tractible for *Mathematica*
`ToProb`	rules to turn Pr[…] into Prob[…] format
`FromProb`	rules to turn Prob[…] into Pr[…] format
`ConditionalPrForm[x]`	control TraditionalForm printing. x = On, Off or Blank

Users may, alternatively, opt to use Prob as the basic function, then use FromProb to get to the structural form that the routines recognise, and then apply ToProb again.

8.2.2 Bayes

`Bayes[A, B][x___Rule]`	is a SolveFrom application, using the equations Pr[A , B] == Pr[B] ConditionalPr[A][B] Pr[A , B] == Pr[A] ConditionalPr[B][A]

```
Bayes[A, B][Pr[A, B] → .15, Pr[A] → .2, Pr[B] → .3]  // Last
```

{{ConditionalPr[A][B] → 0.5, ConditionalPr[B][A] → 0.75}}

```
% /. ToProb
```

{{Prob(A | B) → 0.5, Prob(B | A) → 0.75}}

```
Bayes[A, B][ConditionalPr[A][B] → .5, Pr[A] → .2, Pr[B] → .3]  // Last
```

{{Pr(A, B) → 0.15, ConditionalPr[B][A] → 0.75}}

```
% /. ToProb
```

{{Prob(A, B) → 0.15, Prob(B | A) → 0.75}}

Using matrices is often more instructive. A bordered 2D probability matrix is a {n, m} matrix of which the last row (column) is the sum of the preceding rows (columns).

- Give the minimal information, and let *Mathematica* find the rest. Let A be the first column, ¬A the second column, B the first row, ¬B the second row.

```
         .15  □  .3
pmat =    □   □  □  ;
          .2  □  □
```

```
Bordered2DPrSolve[pmat]
```

$$\begin{pmatrix} 0.15 & 0.15 & 0.3 \\ 0.05 & 0.65 & 0.7 \\ 0.2 & 0.8 & 1 \end{pmatrix}$$

```
Bordered2DPrToConditionals[%]
```

$$\left\{\text{Row} \to \begin{pmatrix} 0.75 & 0.1875 & 0.3 \\ 0.25 & 0.8125 & 0.7 \\ 1. & 1. & 1 \end{pmatrix}, \text{Column} \to \begin{pmatrix} 0.5 & 0.5 & 1. \\ 0.0714286 & 0.928571 & 1. \\ 0.2 & 0.8 & 1 \end{pmatrix}\right\}$$

`Bordered2DPrSolve[x_?MatrixQ, X_Symbol: XYXZX]`	solves □
`Bordered2DPrToConditionals[x_?MatrixQ]`	
`Bordered2DPrToEquations[x_?MatrixQ]`	
subroutines, for inner cell conditional probabilities and equations	

8.2.3 Pr is orderless

Pr has the Attribute Orderless.

- This is as it should be.

```
Pr[A, B] === Pr[B, A]
```

True

Ordered probabilities are denoted with lists, so that Pr[{A, B}] gives the probability of the ordered sequence {A, B}. (This is much better than ClearAttributes[Pr, Orderless].)

Be aware of the subtleties of order. Traditional notation is awkward here. The probability of first a black card and then a red card Pr[{B, R}] = Pr[B] Pr[R | {B}] differs from the probability of just a black and red card Pr[{B, R} or {R, B}] = P[{B, R}] + Pr[{R, B}]. And the latter differs from the probability Pr[B, R], since the latter need not have anything to do with sequential draws. By using the {} in the conditional part, as in Pr[R | {B}], we can express that sequential draws are at hand.

An example can help. Let the universe have n elements, with r elements in R, b elements in B and m elements in the intersection of R and B. Then:

$$\Pr[R] = r/n, \quad \Pr[B] = b/n, \quad \Pr[R, B] = m/n, \quad \Pr[R \mid B] = m/b$$

If the events are independent then m = 0 (as would happen with black and red cards):

$$\Pr[\{B, R\}] = \Pr[B]\,\Pr[R \mid \{B\}] = \frac{b}{n}\,\frac{r}{n-1}$$

$$\Pr[\{R, B\}] = \Pr[R]\,\Pr[B \mid \{R\}] = \frac{r}{n}\,\frac{b}{n-1}$$

The sum P[{B, R}] + Pr[{R, B}] clearly may differ from Pr[R, B] = 0.

- In the special case that $n = b + r$, this sum is:

```
   b r          r b
————————— + ————————— /. b → n - r // Simplify
n (n - 1)   n (n - 1)
```

$$\frac{2(n-r)r}{(n-1)n}$$

- This is actually from the hypergeometric distribution.

```
Binomial[n - r, 1] Binomial[r, 1] / Binomial[n, 2]
```

$$\frac{2(n-r)r}{(n-1)n}$$

If the events are dependent then $m \neq 0$ (as would happen with black cards and picture cards):

$$\Pr[\{B, R\}] = \Pr[B]\ \Pr[R \mid \{B\}] = \tfrac{b}{n}\left(\tfrac{m}{b}\ \tfrac{r-1}{n-1} + \tfrac{b-m}{b}\ \tfrac{r}{n-1}\right)$$

$$\Pr[\{R, B\}] = \Pr[R]\ \Pr[B \mid \{R\}] = \tfrac{r}{n}\left(\tfrac{m}{r}\ \tfrac{b-1}{n-1} + \tfrac{r-m}{r}\ \tfrac{b}{n-1}\right)$$

Taking that example for B = black cards, and R = picture cards indeed, then n = 52, b = 26, r = 12, and m = 6 (black picture cards).

$$\Pr[\{B, R\}] = \Pr[B]\ \Pr[R \mid \{B\}] = \tfrac{26}{52}\left(\tfrac{6}{26}\ \tfrac{11}{51} + \tfrac{20}{26}\ \tfrac{12}{51}\right)$$

$$\Pr[\{R, B\}] = \Pr[R]\ \Pr[B \mid \{R\}] = \tfrac{12}{52}\left(\tfrac{6}{12}\ \tfrac{25}{51} + \tfrac{6}{12}\ \tfrac{26}{51}\right)$$

- Adding these.

```
b  m  r-1   b-m   r        r  m  b-1   r-m   b
- (- ——— + ——— ———)  +  - (- ——— + ——— ———)  //Simplify
n  b  n-1    b   n-1       n  r  n-1    r   n-1
```

$$\frac{2m-2br}{n-n^2}$$

8.2.4 Prospects

A prospect is an object that collects both the events that can occur and the probabilities that they occur. A binary prospect recognises only two events; if the probability of the first is p, then the probability of the other is (1 - p). The multidimensional prospect generalises from this. Only lists have been implemented here, not continuous variables.

`Prospect[`*event1, event2*`, Pr[`*event1*`]]`	a binary prospect
`Prospect[`*x_List*`, `*p_List*`]`	is an object with values x and associated probabilities p
`ProspectQ[`*q*`]`	tests whether q is a prospect
`ProspectEV[`*x_Prospect*`]`	gives the expected value of prospect x

The function Spread recognises Prospects too.

```
ex1 = Prospect[a, b, p];
```

```
ProspectEV[ex1]
```

$b(1-p)+a\,p$

```
ex2 = Prospect[{c, d, e}, {.3, .3, .4}];
```

```
ProspectEV[%]
```

$0.3\,c+0.3\,d+0.4\,e$

8.2.5 Valueing prospects

If we want to be able to compare states of the world or add them, then they must have the same dimensions or be valued to a same dimension. If the dimensions are not the same then we use money or utility.

We use Utility to stand for non-stochastic utility that evaluates certain states of the world, and ProspectUtility for stochastic utility that for example weighs expected value and spread.

`CreateProspect[`*n_Integer*`]`	creates a propect with n states and probabilities
`CreateProspect[Price, `*n_Integer*`]`	idem with n priced states and probabilities
`CreateProspect[Utility, `*n_Integer*`]`	idem with n utility states and probabilities
`ToUtility[`*q_Prospect*`]`	makes a Von Neumann – Morgenstern prospect with utilities of the outcomes
`ToExpectedUtility[`*q_Prospect*`]`	combines ToUtility and ProspectEV
`ProspectUtility[`*x_Prospect, crit__*`]`	gives the utility of prospect x based the criteria.

If the Risk package is loaded, then default criteria for ProspectUtility are ProspectEV, Spread and Risk.

- Though states can have different dimensions, evaluation requires that they must be valued by money or utility - or they must have the same dimension (like dimensionless rates of return).

```
sameDims = CreateProspect[2]
```

Prospect({State(1), State(2)}, {Pr(1), Pr(2)})

```
withMoney = CreateProspect[Price, 2]
```

Prospect({Price(1) State(1), Price(2) State(2)}, {Pr(1), Pr(2)})

```
withUtility = CreateProspect[Utility, 2]
```

Prospect({Utility(State(1)), Utility(State(2))}, {Pr(1), Pr(2)})

- The following might be conceptually dangerous since it means that we take the utilities of sums of money. This could be alright, though, if we were to use economic indices like "national income" - since those are constructed as (money) (chain) indices.

```
ToExpectedUtility[withMoney]
```

Pr(1) Utility(Price(1) State(1)) + Pr(2) Utility(Price(2) State(2))

- This is the Von Neumann - Morgenstern criterion without the latter money complexity.

```
ProspectEV[withUtility]
```

Pr(1) Utility(State(1)) + Pr(2) Utility(State(2))

- The following would be the utility evaluation of above example ex1 in the $\{\sigma, \mu\}$ space as is common in finance. If we limit our attention to dimensionless rates of return, then the addition is no problem.

```
ProspectUtility[ex1, Spread, ProspectEV] //Simplify
```

$\text{Utility}\left(\sqrt{-(a-b)^2 (p-1) p},\ -p\, b + b + a\, p\right)$

8.2.6 Subroutines

Before we continue with the interesting section 8.2.7, we should look at some of the routines that manipulate prospects. We will use these manipulations.

`ProspectReList[q]`	changes a binary prospect q into a prospect with lists

```
ProspectReList[ex1]
```

$\text{Prospect}(\{a, b\}, \{p, 1-p\})$

`ProspectInnerSort[`*q_Prospect*`(, p)]`
sorts the states and keeps the probabilities right, with p an ordering function

```
ProspectInnerSort[Prospect[{x, q, a}, {.5, .1, .4}]]
```

$\text{Prospect}(\{a, q, x\}, \{0.4, 0.1, 0.5\})$

`ProspectApply[f, x]`	applies f to Prospects in x

- Take an example expression.

```
expr = {{g}, func[σ → Prospect[1, 2, .3]]}
```

$\{\{g\}, \text{func}(\sigma \to \text{Prospect}(1, 2, 0.3))\}$

- Apply a function f to the prospects in the expression.

```
ProspectApply[f, expr]
```

$\{\{g\}, \text{func}(\sigma \to f(\text{Prospect}(1, 2, 0.3)))\}$

- Take the Spread of prospects.

```
ProspectApply[Spread, expr]
```

{{g}, func($\sigma \to 0.458258$)}

In later discussions it appears useful to 'put in' values or 'take out' values.

`PutIn[`x+ ... +q_*Prospect*+ ... +y`]`	for a single q includes the additions into the prospect
`PutIn[`a_*List*, q_*Prospect*`]`	takes in the strategy for playing the prospect
`TakeOut[`q_*Prospect*, f`]`	takes out a f[profit, loss]; f = Max takes out profits, f = Min takes out losses.
`TakeOut[`q, *Profit* || *Loss*`]`	for simple prospects: uses positions

```
Prospect[10] + 56 + test
```

test + Prospect(10, 0, 1) + 56

```
PutIn[%]
```

Prospect(test + 66, test + 56, 1)

```
TakeOut[%, Profit]
```

test + Prospect(0, −10, 1) + 66

```
Prospect[{1, 2, 3, 4}, Laplace[4]]
```

$\text{Prospect}\left(\{1, 2, 3, 4\}, \left\{\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right\}\right)$

```
TakeOut[%, Max]
```

$\text{Prospect}\left(\{-3, -2, -1, 0\}, \left\{\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right\}\right) + 4$

```
PutIn[%]
```

$\text{Prospect}\left(\{1, 2, 3, 4\}, \left\{\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right\}\right)$

8.2.7 Random drawing from Prospects

8.2.7.1 The routine Draw

`Draw[]` is a small but powerful routine to actually draw randomly from a prospect. You can also define a strategy how much to bet at each turn.

- This is a single draw for a "50% win 10, lose 10" proposition. The result is random, and can change with each evaluation.

```
Draw[Prospect[10, -10, 1/2]]
```

-10

- This draws three times in a row.

```
Draw[3, Prospect[10, -10, 1/2]]
```

$\{-10, 10, 10\}$

The following example is taken from Luenberger (1998). Suppose that there is a *Wheel of Fortune* with areas of size {1/2, 1/3, 1/6} that pay out 3, 2, or 6 times the bet on the respective area. A player puts amounts $\{a, b, c\}$ on the respective areas, and thus has a Prospect[{3*a*, 2*b*, 6*c*}, {1/2, 1/3, 1/6}].

- Let us set a on area 1/2, b on area 1/3 and c on area 1/6. Let us draw 10 times.

```
prp = Prospect[{3a, 2b, 6c}, {1/2, 1/3, 1/6}];
```

```
Draw[10, prp]
```

$\{3\,a, 3\,a, 3\,a, 3\,a, 3\,a, 6\,c, 3\,a, 2\,b, 2\,b, 2\,b\}$

- Here we draw 100 times. We then determine the frequencies (in this case percentages).

```
res = Draw[100, prp]; Frequencies[res]
```

$$\begin{pmatrix} 57 & 3\,a \\ 31 & 2\,b \\ 12 & 6\,c \end{pmatrix}$$

- For the expected return we must substract the bet $a + b + c$.

```
ProspectEV[prp] - (a+b+c) // Simplify
```

$$\frac{1}{6}(3\,a - 2\,b)$$

From this, a maximiser of 'expected value' would erroneously conclude that he or she should play the game, and then bet only a on area 1/2. You bet a, expect to win $3a/2$, and thus your expected winning is $a/2$ per turn. A goldmine ! Or do you see the catch ?

`Draw[`*q_Prospect*`]`	draws from the probability density and returns the drawn outcome.
`Draw[`*a_List*`, Prospect[`*v*`, `*p*`]]`	draws with strategy a. Part of wealth (1 – Add[a]) is retained, and a * v are the adjusted rewards with probabilities p. Multiply the outcomes to get the wealth after so many tries
`Draw[]`	draws from the prospect or strategy given earlier
`Draw[`*n_Integer*`, `*x___*`]`	draws n times

8.2.7.2 Strategy

There is something seriously wrong with 'expected value theory' as taught in many less advanced textbooks. Sometimes there is made a distinction between single games and repeated games, but a strong case can be made that a single game should better be regarded (then) as a 'repeated game for one step'. Whatever that be, Luenberger (1998) ch. 15 is essential for your ticket to wealth here. The problem with the 'expected value' approach is that it does not take account of the fact that you may lose all your money, and cannot play the game any more. The House, the owner of the 'wheel of fortune', here has the advantage.

Drawing can be done with a strategy however. Let the player decide on a budget for gambling, and then bet *proportions* $\{a, b, c\}$, while retaining $1 - a - b - c$ on the side. The prospect then becomes Prospect[$\{3a, 2b, 6c\}$, $\{1/2, 1/3, 1/6\}$] + 1 - a - b - c. The proportion is on the current wealth. The trick is that retaining a proportion of winnings can cause a bias towards growth. Part of the error in above approach is to assume that there is an unlimited budget such that after any string of losses there still would be $a + b + c$ available. If we define a, b and c as proportions, and take $a + b + c < 1$, then we make sure that something is available indeed. (Though, in practice, runs can go to millionths of cents, and thus the strategy is not always that practical.)

- Let us play the wheel with *proportions* $\{a, b, c\}$. The routine now returns the winnings plus the proportion that was kept on the side. This output can be interpreted as a proportion of the budget again. (Note: each evaluation can give another result.)

```
prop = Prospect[{3, 2, 6}, {1/2, 1/3, 1/6}];
```

```
Draw[{a, b, c}, prop]
```

$2a - b - c + 1$

- Let us play the wheel for 3 times. Each outcome is a proportion of the budget created by the former outcome.

```
Draw[3, {a, b, c}, prop]
```

$\{2a - b - c + 1, 2a - b - c + 1, 2a - b - c + 1\}$

- We can check that only three outcomes are possible.

```
res = Draw[100, {a, b, c}, prop]; Frequencies[res]
```

$$\begin{pmatrix} 51 & 2a - b - c + 1 \\ 30 & -a + b - c + 1 \\ 19 & -a - b + 5c + 1 \end{pmatrix}$$

If we subsequently *multiply* the outcomes, then we get the proportion in terms of the original budget. We then can also make a plot of the evolution of wealth over the turns. To do this, we need numerical values.

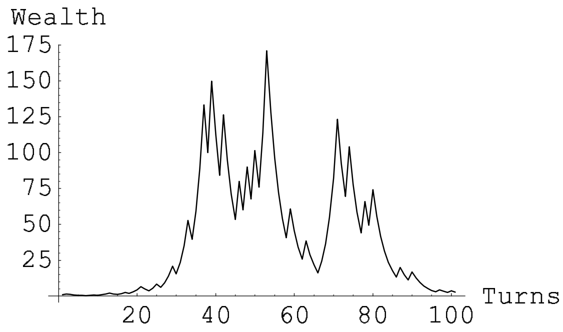
- Let us use a strategy $\{1/4, 0, 0\}$.

```
res = Draw[100, {1/4, 0, 0}, prop];
```

```
wealth = FoldList[Times, 1., res];
```

- This gives the evolution of wealth over time as the wheel turns. A string of wins causes wealth to increase, but a string of losses is possible too. Once the random walk gets in the low wealth range, it may be difficult to get out again because of the implied growth rate of the strategy (see below).

```
PlotLine[wealth, AxesLabel → {"Turns", "Wealth"}];
```



- We can take the $1/n$ root to determine the average growth rate.

```
N[Times @@ res]^(1/100)
```

1.01043

8.2.7.3 Optimal strategy

Let us find the optimal strategy. (Again I follow Luenberger (1998).)

- First define the formal prospect situation.

```
fp = PutIn[{a, b, c}, prop]
```

$$-a - b - c + \text{Prospect}\left(\{3\,a, 2\,b, 6\,c\}, \left\{\frac{1}{2}, \frac{1}{3}, \frac{1}{6}\right\}\right) + 1$$

```
properfp = PutIn[fp]
```

$$\text{Prospect}\left(\{2\,a - b - c + 1, -a + b - c + 1, -a - b + 5\,c + 1\}, \left\{\frac{1}{2}, \frac{1}{3}, \frac{1}{6}\right\}\right)$$

- Take logarithmic utility to warrant the highest rate of growth

```
eu = ToExpectedUtility[properfp] /. Utility → Log
```

$$\frac{1}{2}\log(2\,a - b - c + 1) + \frac{1}{3}\log(-a + b - c + 1) + \frac{1}{6}\log(-a - b + 5\,c + 1)$$

- Setting up the first order conditions for a maximum.

```
Economics[Calculus, Print → False]
```

```
foc = Foc[eu, {a, b, c}]
```

$$\left\{-\frac{1}{3(-a+b-c+1)}-\frac{1}{6(-a-b+5c+1)}+\frac{1}{2a-b-c+1}==0,\right.$$
$$\frac{1}{3(-a+b-c+1)}-\frac{1}{6(-a-b+5c+1)}-\frac{1}{2(2a-b-c+1)}==0,$$
$$\left.-\frac{1}{3(-a+b-c+1)}+\frac{5}{6(-a-b+5c+1)}-\frac{1}{2(2a-b-c+1)}==0\right\}$$

- We can show already that {1/4, 0, 0} is not optimal.

```
foc /. Thread[{a, b, c} → {1/4, 0, 0}]
```

{True, False, False}

- Let us make sure that we are dealing with a probability measure.

```
pm = foc /. c → 1 - a - b;
```

- And this gives the optimal strategy.

```
Solve[pm, {a, b}]
```

$$\left\{\left\{a\to\frac{1}{2},\ b\to\frac{1}{3}\right\}\right\}$$

- We can compare the single period expected values of the prospects. These also give the long run expectation for the average outcome.

```
ProspectEV[properfp] /. Thread[{a, b, c} → {1/4, 0, 0}]//N
```

1.125

```
ProspectEV[properfp] /. Thread[{a, b, c} → {1/2, 1/3, 1/6}]//N
```

1.13889

- Let us try this new strategy again. Well, bad luck ! (Growth was not as high as it could have been.)

```
res = Draw[100, {1/2, 1/3, 1/6}, prop];
N[Times @@ res]^(1/100)
```

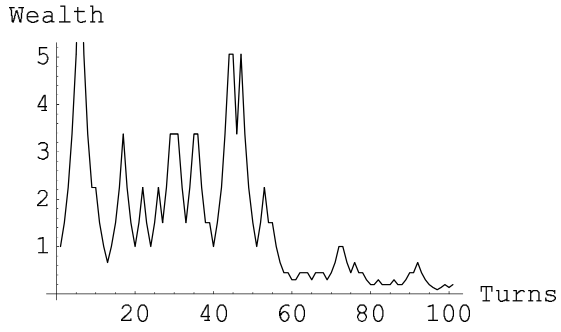
0.983912

Note that also the optimal strategy may result in long states of low wealth. There is no guarantee that one will have more than the original budget in one's lifetime, or, if it has grown to the sky, that it will remain there.

■ Show the evolution of wealth for the above.

```
wealth = FoldList[Times, 1., res];
```

```
PlotLine[wealth, AxesLabel → {"Turns", "Wealth"}];
```



8.3 Risk

8.3.1 Prospects and risk

Prospects are perfect to continue the discussion on risk. The typical binary risky prospect recognises only win or lose situations. Let *profit* ≥ 0 stand for the positive return of a prospect, and - *loss* ≤ 0 for the negative return of a prospect, where loss is the absolute value of that negative return. The probability of a profit is p, the probability of a loss is (1 - p). The multidimensional prospect generalises from this.

`Prospect[`*profit*`, −`*loss*`, Pr[Profit]]`	a binary Prospect convention for risky situations

Note that loss is an absolute value, and that a real loss must be entered as a negative value.

- An example risky prospect.

```
eg = Prospect[Profit, -Loss, p];
```

```
ProspectEV[eg]
```

p Profit – Loss (1 – p)

It is important to see that there is nothing in the concept of a Prospect that requires that the second position is reserved for a loss. A binary prospect may well give two profits or two losses. Therefor, the proper definition of Risk requires a formal test on the values of the entries. For formal discussions, however, such a test reads awkward, and we better resort to the positional convention.

- For formal analysis we resort to the convention that the second position is the loss. The Position option works only for binary risk (in non-list-format).

```
SetOptions[Risk, Position → True];
```

- The risk is the probable loss (the loss weighed by its probability).

```
Risk[eg]
```

Loss (1 – p)

```
RiskyQ[eg]
```

True

- The default option setting is Position → False, since it is not obvious that you would be using Prospects formally.

```
SetOptions[Risk, Position → False];
```

Risk[*q*]	gives the risk, i.e. the expected absolute loss
RiskPr[*q*]	gives the cumulative probability of a loss in prospect q
RiskyQ[*q*]	gives False if not risky, True if q is risky (at least one negative possible outcome with nonzero probability)

All defined on prospects q. These routines use an If[Negative...] construction since its derivative is defined. For reading, use IfNegativeToMinRule or IfNegativeToMaxRule. On the binary Prospect, if Options[Risk] have Position → True, the second position is taken as the loss.

8.3.2 Theoretical definition

We better understand our subject when we first have a foundation for uncertainty and probability:

(1) First there is the distinction between *certainty* and *uncertainty*.

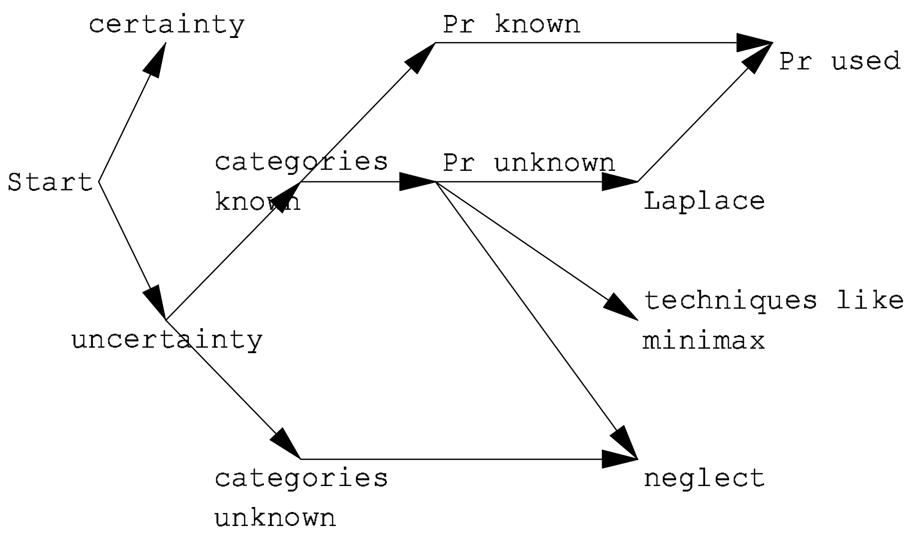
(2) Uncertainty forks into *known categories* and *unknown categories*.

(4) Known categories forks into *known* and *unknown probabilities*.

(3) Unknown probabilities forks into *assuming a uniform distribution* (Laplace) or use non-probabilistic *techniques like minimax* or *neglect*.

These definitions can be clarified by the following plot.

```
UncertaintyDefinitionsPlot[];
```



What do we mean by 'risk' ? A.S. Hornby's (1985) "Oxford Advanced Learner's Dictionary of Current English" defines 'risk' as: "(instance of) possibility or chance of meeting danger, suffering loss, injury, etc." Also: "at the ~ risk of / at ~ of, with the possibility of (loss etc.)".

Thus, if there are possible outcomes $O = \{o_1, o_2, ..., o_n\}$, then the situation is risky if at least one of the o's represents a loss. The risks themselves are the o_i that are those losses. The risks factors are the dimensions or positions of the risky outcomes, the i's (or the causes that make such positions to be filled).

We will use the term 'valued risk' when a risk is valued with money or utility. When all risks have been made comparable by valuing them, then we can add them, and we will use the term *expected risk value* for the *expected value of the 'valued risks'*. Then, crucially, once these definitions are well understood, then we may also use *'the risk'* for the expected risk value.

With such understanding, risk will be $\rho = -E[x < 0]$. Note that the term 'risk' has not been used in the 4 points above, so that an independent definition is possible.

Relative risk is defined as $\rho(t) = t - E[x < t]$ for some target level t. Risk (or *absolute* risk) takes $t = 0$, and relative risk would allow for a different target level. An interesting application is when x is a stochastic rate of return and r the certain rate (rate of interest), so that there is relative risk $\rho(r) = r - E[x < r]$.

This relative risk answers the question: What is the probable loss with respect to a target return of r ? Here, r - $\rho(r)$ = E[x < r] gives the weight of underperformance in the total target return (which weight has to be compensated by probable profits to achieve the target).

Conditional (relative) risk is defined as $\kappa(t) = t$ - E[x | $x < t$] for some target level t. With respect to rates of return, conditional risk $\kappa(r)$ answers the question: What would one expect to lose with respect to r, if earnings actually underperform and fall below r. Indeed, r - $\kappa(r)$ would give your expected return when actually underperforming.

Conditional risk is related to relative risk by the property that E[x | $x < t$] = E[x < t] / Pr[$x < t$]. The probable loss thus is corrected for the probability of a loss. Or, the probability measure in the expectation is corrected so that a density is taken that sums to 1.

Note that above definitions are proper in the sense that they conform to every day parlance and the definitions provided by Hornby's dictionary op. cit.. The definitions provided here however differ from other definitions within the economics literature. First there are the definitions of Knight (1921) that have been adopted widely in economics, as for example in The New Palgrave (1998:III:358). Or it has become custom in finance to associate risk with the standard deviation. Cool (1999), "Proper definitions of risk and uncertainty" (available in the Pack and on the internet or as chapter 35 in Cool (2000)) further discusses why such alternatives generate conceptual problems and why the current definitions are preferable.

Below we will develop these notions somewhat further. The Economics Pack (Cool (1999, 2001)) has a much more extensive development that would lead too far here.

8.3.3 Tests in Mathematica

Since prospects need not satisfy the formal convention for risky prospects, *Mathematica* needs a test on what are the negative values.

- This is the proper risk definition.

```
Risk[eg]
```

$-(1-p)\,\mathrm{If}[\mathrm{Negative}[-\mathrm{Loss}], -\mathrm{Loss}, 0] - p\,\mathrm{If}[\mathrm{Negative}[\mathrm{Profit}], \mathrm{Profit}, 0]$

```
RiskyQ[eg]
```

Negative[−Loss] ⇒ True

- For formal analysis, however, we may resort to the convention that the second position is the loss. The Position option works only for binary risk (in non-list-format).

```
SetOptions[Risk, Position → True];
Risk[eg]
```

Loss $(1 - p)$

```
RiskyQ[eg]
```

True

- If you set the option to True, you can still have formal testing by using the list format. Also, routines like ProspectPrValue internally call ProspectReList, and thus are not affected by the Position option.

```
Risk[ProspectReList[eg]]
```

$-(1 - p)$ If[Negative[−Loss], −Loss, 0] − p If[Negative[Profit], Profit, 0]

- This gives the risk probabilities.

```
RiskPr[eg]
```

$1 - p$

```
eglis = Prospect[{-1, 2, -2, 3}, {0.2, 0.4, 0.3, 0.1}];
```

```
RiskPr[eglis]
```

0.5

Since risk selects the negative values from the states of the world, and since some of the prospects are algebraic rather than numeric, the Risk function basically uses a conditional statement. Where the choice was between Negative[ ], Min[ ] or Max[ ], the use of Negative[ ] has been chosen, since the derivative D[ ] still applies. Since an If[Negative[ ], ..] statements reads difficult at times, there is the possibility to replace it by the following.

`IfNegativeToMinRule`	A rule that changes an If[Negative…] statement into a Min condition. The derivative of If[Negative…] is defined, but Min reads better
`IfNegativeToMaxRule`	A rule that changes an If[Negative…] statement into a Max condition. The derivative of If[Negative…] is defined, but Max reads better

8.3.4 Risk model

This model presumes binary risk, and therefor is a good introduction to the subject.

`RiskModel[{`*rules*`}]`	is a SolveFrom application, default with RiskEquations
`RiskEquations`	used in RiskModel. Profit, Loss and Risk are nonnegative values, and the expected value is Prob Profit – (1–Prob) Loss

```
RiskEquations
```

{Risk == Loss (1 – Prob), ExpectedValue == Prob Profit – Risk}

```
RiskModel[{Profit → 0.6, Prob → 0.5, Risk → 0.2}] // Last
```

{{ExpectedValue → 0.1, Loss → 0.4}}

8.3.5 Risket

Given the relevance of expected value, spread, risk and loss probability, we define the Risket object.

`Risket[`μ, σ, ρ, 1 -p`]`	is an object with expected value μ, spread σ, risk ρ and probability p of profit
`ToRisket[`q`]`	turns a Prospect object or *pdf* or data vector into a Risket object
`ToProspect[`x`]`	turns a Risket object or *pdf* or data vector into a Prospect object

ToRisket has default option Spread → StandardDeviation (division by n-1). Use Spread → Spread to divide by n, and Spread → False if you want the spread implied by applying ToProspect again.

Importantly, the Risket object can contain more information than a Prospect. Going from a Risket to a Prospect in generally is a projection, where information about the true spread can be lost.

- This gives a clear formal result since we now work with Position → True in Options[Risk].

```
ToRisket[eg]
```

Risket(p Profit − Loss (1 − p),
$\sqrt{((1-p)\,((1-p)\,\text{Loss} - \text{Loss} - p\,\text{Profit})^2 + p\,(\text{Loss}\,(1-p) - p\,\text{Profit} + \text{Profit})^2)}$, Loss (1 − p), 1 − p)

- This function neglects the spread, since for true binary prospects all information is in the expected value, risk and risk probability. If the original is not truely binary, then information is lost.

```
ToProspect[%]
```

Prospect(Profit, −Loss, p)

If one wishes to determine a prospect by using the spread instead, the following would be useful.

ProspectInverse[*mean*, *spread*, *risk*]	gives a binary prospect with these properties

Note that there can be more solutions.

- With these values of mean, standard deviation and risk, there are two binary prospects that satisfy those properties.

```
ProspectInverse[0.5, 0.6, 0.1]
```

{Prospect(0.626795, −2.33923, 0.957251), Prospect(0.973205, −0.26077, 0.61652)}

8.3.6 Prospect plotting

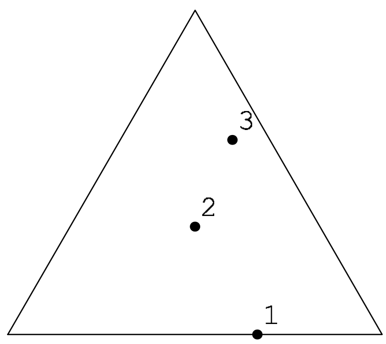
A useful feature is that the probabilities of 3D prospects can be plotted in a 2D triangle, that essentially is a transform of the 3D unit simplex. If the dimension is less than 3 then the 3rd dimension is set to 0. If the dimension is larger than 3 then the higher dimensions can be summed. The triangle has sides $2 / \sqrt{3}$, and the corners are at $c1 = \{0, 0\}$, $c2 = \{2 / \sqrt{3}, 0\}$ and $c3 = \{1 / \sqrt{3}, 1\}$. A point in the triangle has the property that the distances to the sides add up to one. The prospect probabilities $\{p1, p2, p3\}$ are plotted such that the distance from the plotted point to the side opposite to ci gives the probability pi of outcome i.

`Prospect3DPrTriangle[`*lis_List*`,` *opts*`]`	plots probabilities of prospects in *lis* in a triangle

Note: The points are in Results[ProspectPr3DTriangle]. Note: Options are: Point → True (default) plots points, PointSize→ size (default .025) gives the size of these points, Label → Automatic (default) gives labels. The latter can also be a list, or None. If Point → False, then the labels are printed right on the co-ordinates. Note: Subroutines are: ProspectPr3DTriangle[] gives the graphics primitive of the triangle; while ProspectPr3DTriangle[Point, lis] gives those of the points.

- Point 1 of the simple prospect plots on the bottom side, since the distance from the bottom side is zero. The distance to the right side is 1/3 and to the left side 2/3. Point 2 plots in the middle, and lies on a line through point 1 parallel to the right side. Point 3 then is clear.

```
Prospect3DPrTriangle[{Prospect[3, -2, 1/3],
   Prospect[{a, b, c}, {1, 1, 1}/3],
   Prospect[{a, b, c}, {0.1, 0.3, 0.6}]}];
```



Above plots just the probabilities. The following include the values.

`ProspectPlot[`*x_List*`]`	plots the prospects x in the Expected Value, Risk and Spread space. Subroutines are:
`ProspectPlot[Set,` *x*`]`	sets the data to be plotted (⇔ SetOptions[ProspectPlot, Data → ProspectStatistics[x]])
`ProspectPlot[`*x_Symbol*`,` *y_Symbol*`,` *opts___Rule*`]`	
	plots the keys x and y of these
`ProspectPlot[All]`	plots for the mentioned three keys
`ProspectPlot[`*q_Prospect*`,` *a_Symbol*`]`	
	plots Expected Value, Risk and Spread values of a prospect that is a function of a (in the domain [0, 1])

8.4 Certainty equivalence

This section discusses how one could try to recover an ordinal utility scale from experiments with prospects. The current assumption is that there is no difference between the buying or selling price of a prospect (lottery ticket), although in practice there can be this difference (since people tend to overcharge for what they have, even when they think that the grass of the neighbour is greener).

8.4.1 The price of a lottery ticket

Suppose that there is a lottery with a probability p on a prize, called Profit. We use simple expected values (with the assumption that the proportion to total wealth is very small).

- If you buy a lottery ticket then you expect a positive return. Your selling price of the lottery ticket should generally be larger than your expected return.

```
ProspectEV[Prospect[Profit, 0, p]] - Price_Buy ≥ 0
```

$p \text{ Profit} - \text{Price}_{\text{Buy}} \geq 0$

```
Price_Sell - ProspectEV[Prospect[Profit, 0, p]] ≥ 0
```

$\text{Price}_{\text{Sell}} - p \text{ Profit} \geq 0$

- In equilibrium, you can sell the ticket only at the price that you pay yourself, and you can only buy it at the price for which you would be willing to sell it.

```
Price == ProspectEV[Prospect[Profit, 0, p]]
```

$\text{Price} == p \text{ Profit}$

The existence of lotteries, where there also are overhead costs, is a mystery that economic theory tries to explain, e.g. with risk preference and risk aversion.

8.4.2 A standard approach

A way to recover utility functions is to precisely use such lotteries.

Luenberger (1998:234): "(...) select two fixed wealth values *A* and *B* as reference points. A lottery is then proposed that has outcome *A* with probability *p* and outcome *B* with probability 1 - *p*. For various values of p the investor is asked how much certain wealth *C* he or she would accept in place of the lottery. *C* will vary as *p* changes. Note that the values *A*, *B* and *C* are values for total wealth, not just increments based on a bet."

Here we use *A* for "above" and *B* for "below". We will reproduce Luenbergers example - but with reversed values for *A* and *B* because of this interpretation. Because of the word 'accepting', the certainty equivalent *C* = *C(p)* and lottery apparently already are part of the investor's wealth. The investor may win *A* - *C*(*p*) or lose *C*(*p*) - *B* (as absolute loss) compared to the certainty equivalent wealth level *C(p)* of doing nothing.

- If we look at the budget only (choosing *A*, *B* and *C* as wealth levels).

```
CertaintyEq[Prospect[A, B, p], C, "Budget"] /. Wealth → 0
```

$C + \text{Prospect}(A - C, B - C, p)$

- However, we should consider utility.

```
cond = CertaintyEq[Prospect[A, B, p], C, None]
```

$\text{Utility}(C) == p\,\text{Utility}(A) + (1 - p)\,\text{Utility}(B)$

`CertaintyEq[q_Prospect, c]`	gives the condition so that *c* is the certainty equivalent of the prospect
`CertaintyEq[q_Prospect, c, None]`	gives the same but excluding wealth
`CertaintyEq[Prospect[A, B, p], C, ''Budget'']`	

gives the budget for who reasons as follows: certain is Wealth + C, and then there is the prospect of winning A – C or losing (absolute) C – B.

- The certainty equivalence condition is invariant for a linear transformation. This means that we can recover only ordinal utility.

```
cond /. Utility[x_] :→ a U[x] + b // Simplify
```

$a\,(-p\,U(A) + (p - 1)\,U(B) + U(C)) == 0$

Because of this property of ordinal utility functions, we can normalise $A = U(A)$ and $B = U(B)$. This means that $U(C)$ collapses to $U(C) = p\,A + (1 - p)\,B = EV$. In other words, taking money prospects causes that the utility can be fully represented by the expected value, even though we first argued that we should not take the expected value but should take utility. Next, having a set of points $\{C, EV\}$ allows us to interpolate values (which avoids estimation of parameters), and subsequently to interprete this interpolation as the U.

- We can normalise our findings by setting the utility levels of the extremes of the prospect equal to those extremes, in this case Utility[Max] = Max = A and Utility[Min] = Min = B. This uses Luenberger's example with $A = 9$ and $B = 1$ (million dollars).

```
CertaintyEq[Equations]
```

$\{c_2 + c_1 \text{ Utility}(1) == 1,\ c_2 + c_1 \text{ Utility}(9) == 9\}$

CertaintyEq[*Equations*]	gives the equations for the linear transform of utility

Note that we can also determine the utility as a function of the probabilities, since:

$$U(C(p)) = p\,A + (1\text{-}p)\,B = U_p(p \mid \{A, B\})$$

It is unclear why we should try to recover values for C. The utility function is wholly determined now, and in practice the investor should concentrate on finding the correct values for A, B and p. Yet, let us see how the standard approach proceeds.

- This example is taken from Luenberger (1998:236). He uses million dollars, for a 'moderately successful venture capitalist'.

```
below = 1; above = 9;
```

```
probpoints = Range[0, 10]/8.
```

{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.}

- This sets the options for the maximum *A* and minimum *B* and the probability points. It calculates the expected values - which might be communicated to the investor, so that he or she is actually informed about his or her utility function as depending upon the probabilities.

```
CertaintyEq[Set, above, below, probpoints]
```

{1, 1.8, 2.6, 3.4, 4.2, 5., 5.8, 6.6, 7.4, 8.2, 9.}

`CertaintyEq[Set, max, min, p_List]`	sets the options for the maximum and minimum points, the probabilities p that are being considered, and the implied expected values

- Luenberger's investor gives these certainty equivalent values *C(p)* for the various probability points (taking the expected values into account or not).

```
cedata = {1, 1.44, 1.96, 2.56, 3.24, 4, 4.84, 5.76, 6.76, 7.84, 9};
```

- These *C(p)* data now can be used to derive an interpolated utility function (the goal of the whole exercise).

```
func = CertaintyEq[Data, cedata]
```

InterpolatingFunction[(1. 9.), <>]

- For example, at a certainty equivalent value *C* = 2 million dollar (not given in the *C(p)* data), the utility level would be:

```
func[2]
```

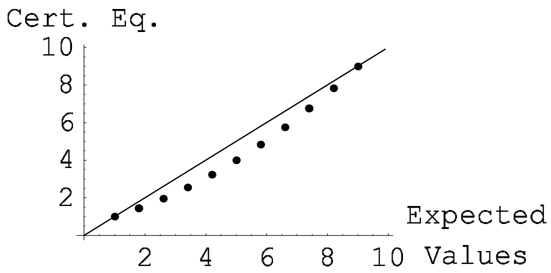
2.65701

`CertaintyEq[Data, CE_List]`	sets CertaintyEq[Data] and creates in interpolation function for the certainty equivalent values CE
`CertaintyEq[Data]`	contains pairs {expected value, certainty equivalent}

Note: The data are recorded in Options[CertaintyEq].

- This plots the {*EV*, *C*} points, using CertaintyEq[Data].

```
CertaintyEq[Plot];
```



The reason for the exercise can be that people better understand Utility as something that depends upon income rather than as something that depends upon probability - even though the income depends upon the probability. The Arrow - Pratt measure for 'risk aversion' also requires a normal utility function.

ArrowPratt[u[*x*], *x*]	gives the Arrow–Pratt measure –u''/u' of utility function u[x]. This is a measure of concavity that is independent of linear transformations of u[x]. Normally u[x] is rising, so that u' > 0

```
ArrowPratt[a Utility[x] + b, x]
```

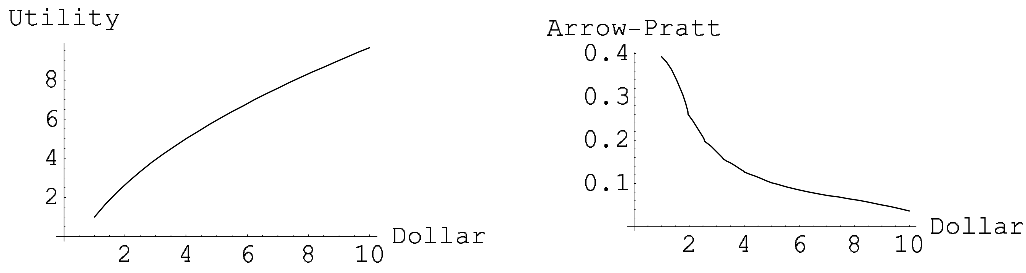
$$-\frac{\text{Utility}''(x)}{\text{Utility}'(x)}$$

- It appears possible to differentiate the interpolated utility function. The Arrow-Pratt measure for 'risk aversion' becomes:

```
ap[x_] = ArrowPratt[func[x], x];
```

- The following plots the estimated utility function as well as the Arrow-Pratt measure.

```
Show[GraphicsArray[{
  Plot[func[x], {x, 1, 10}, AxesLabel → {Dollar, Utility},
           DisplayFunction → Identity],
  Plot[ap[x], {x, 1, 10}, AxesLabel → {Dollar, "Arrow-Pratt"},
          DisplayFunction → Identity]}],
      DisplayFunction → $DisplayFunction];
```



`CertaintyEq[ListPlot, `*opts*`]`	plots CertaintyEq[Data]
`CertaintyEq[Plot, `*opts*`]`	adds a diagonal line for reference

8.4.3 A non-standard approach

The above (standard) approach can be criticised for using nonstochastic utility for a stochastic situation. The curvature of the utility function, that applies for *certain* changes, now is applied to *stochastic* changes. The standard approach also does not explicitly state the risk which is exemplified by the budget line. An alternative (non-standard) approach is to regard a utility function that includes stochastic data and that uses that explicit risk.

- First recall the budget above.

```
CertaintyEq[Prospect[A, B, p], C, "Budget"] /. Wealth → 0
```

$C + \text{Prospect}(A - C, B - C, p)$

There is more information in this than is used by above standard approach. By accepting a certainty equivalent, the subject forgoes some expected Income = ΔWealth which is given by ProspectEV. The reason to let this money go is the risk in the prospect. While doing so, the subject remains on the same utility contour *with C just by itself.* We may thus infer, first, that utility depends upon

the wealth level *and* stochastic data as summarised by the expected value and the risk, so that $U(wealth, \mu, \rho)$. It is not strange to let expected money and risky money into the utility function, where we already had 'certainty equivalent' money in it. And we may infer, secondly, that an indifference contour goes through $U(C, 0, 0) = U(C, \mu, \rho)$, and that this is '*proper* certainty equivalence'.

- There is indifference when the expected value is balanced by the risk.

```
cond2 = Utility[C, 0, 0] ==
  ProspectUtility[Prospect[A - C, B - C, p], C &, ProspectEV, Risk]
```

$\mathrm{Utility}(C, 0, 0) == \mathrm{Utility}(C, (B - C)(1 - p) + (A - C)\, p, -(B - C)(1 - p))$

To tackle this, we need to have access to the prospects for the various probability points.

- These are the prospects per considered probability point. For example, when $p = 0$, then the investor can gain 8 with p and lose 0 with $(1 - p)$. This sets all Options[CertaintyEq].

```
CertaintyEq[Set, Prospect]
```

{Prospect(8, 0, 0), Prospect(7.56, −0.44, 0.1), Prospect(7.04, −0.96, 0.2), Prospect(6.44, −1.56, 0.3), Prospect(5.76, −2.24, 0.4), Prospect(5, −3, 0.5), Prospect(4.16, −3.84, 0.6), Prospect(3.24, −4.76, 0.7), Prospect(2.24, −5.76, 0.8), Prospect(1.16, −6.84, 0.9), Prospect(0, −8, 1.)}

- The following determines the ProspectEV and Risk implied by the current Options[CertaintyEq], that are to be used for above ProspectUtility.

```
lisev = CertaintyEq[Prospect, ProspectEV]
```

{0, 0.36, 0.64, 0.84, 0.96, 1., 0.96, 0.84, 0.64, 0.36, 0.}

```
lisr = CertaintyEq[Prospect, Risk]
```

{0, 0.396, 0.768, 1.092, 1.344, 1.5, 1.536, 1.428, 1.152, 0.684, 0.}

- Above, we called the following *cedata*, but if you have not given a specific name, the data are still is available in a structural manner.

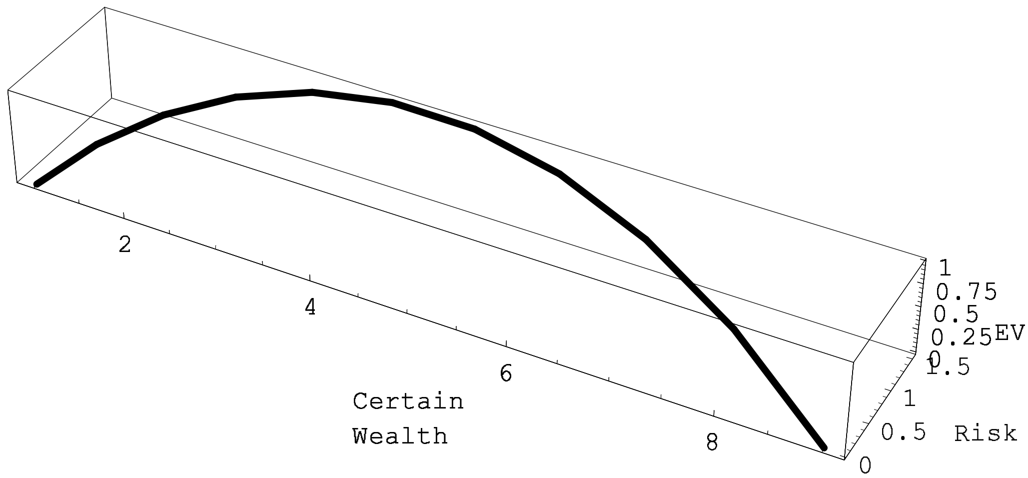
```
lisc = CertaintyEq /. Options[CertaintyEq]
```

{1, 1.44, 1.96, 2.56, 3.24, 4, 4.84, 5.76, 6.76, 7.84, 9}

`CertaintyEq[Set, Prospect]`	creates the prospects implied by the options
`CertaintyEq[Prospect, f]`	applies function f to the prospects in the options

- We can plot the data in 3D space, joining up the observed points by a line. A higher risk needs compensation with higher expected addition to wealth. Alternatively, a higher expected addition to wealth allows taking more risk on it.

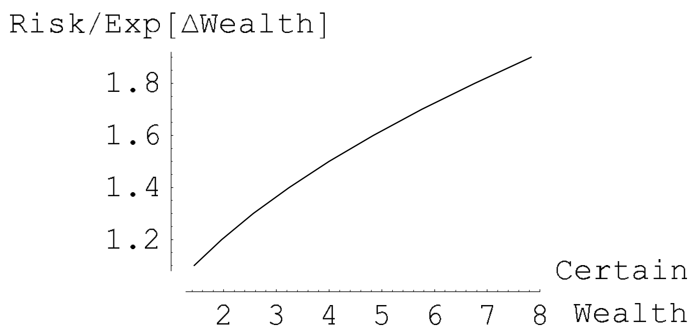
```
Needs["Graphics`Graphics3D`"]

ScatterPlot3D[Transpose[{lisc, lisr, lisev}],
   PlotJoined → True, PlotStyle → Thickness[0.008],
   AxesLabel → {"Certain\nWealth", Risk, "EV"}];
```



With rising wealth, this particular experimental person apparently is willing to accept more risk for the same amount of expected income = ΔWealth. The investor can use this relationship between $C(p)$ and ρ / μ for additional decisions and predictions.

- Plotting the relationship between $C(p)$ and ρ / μ = Risk / ProspectEV.

```
PlotLine[lisc, lisr/lisev, AxesLabel →
    {"Certain\n Wealth", "Risk/Exp[ΔWealth]"}, AxesOrigin → {1, 1}];
```



PM. Since we have ρ / μ = $f(C(p))$ then above $U(C(p))$ would imply $U_f(\rho$ / $\mu)$.

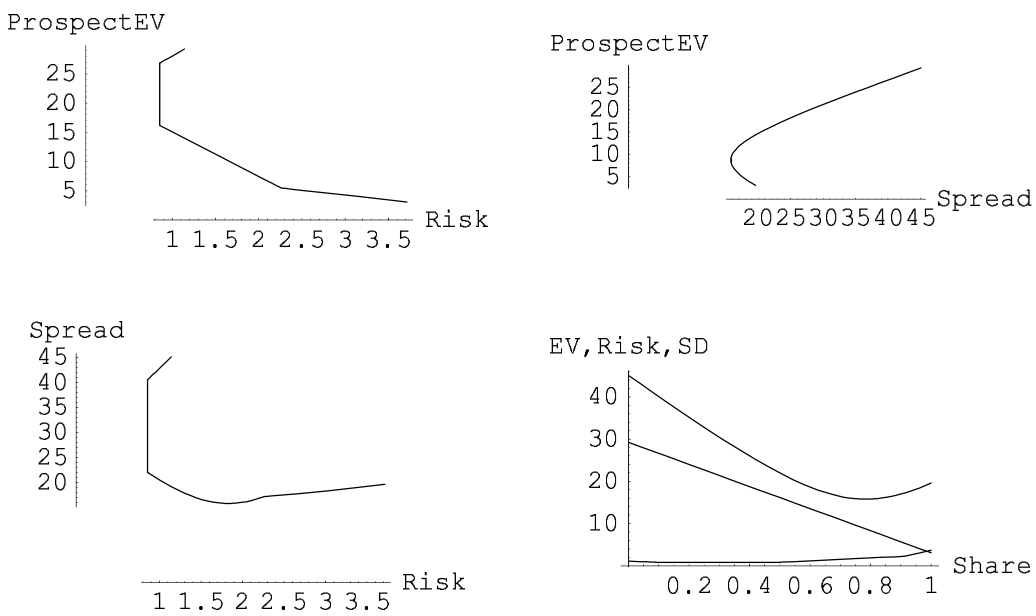
8.4.4 Justification for the non-standard approach

We can better understand the non-standard approach by taking an example from finance. In finance, individual assets (bonds, shares and property) are characterised by their expected values and spreads. For portfolio's of assets, risks can cancel, and then there arises the Markowitz efficiency frontier in the $\{\sigma, \mu\}$ space. It then is assumed that investors have utilities $U(\sigma, \mu)$ such that, with equal spread the higher expected values will be taken, and with equal expected values the lower spreads. Maximising utility then allows the selection of the best mix of assets in the portfolio. For us, it is more appropriate to take $U(\rho, \mu)$ since a high spread is less relevant if it would concern only positive values.

The following example is taken from The Economics Pack An investor will allocate a budget over two prospects, and will be interested in the optimal mix, allocating share S to one prospect and 1 - S to the other. (The calculations are not given here.)

- The following plots for S in [0,1]. The finance community is familiar with the upper right hand plot, the other plots are novel.

```
ProspectPlot[prs, S];
```



8.4.5 Comparing the methods

The final question however is whether we could still accept the oldstyle condition for 'certainty equivalence' as a realistic assumption.

- The 'oldstyle certainty equivalence' condition now becomes:

```
cond3 = cond /. Utility[x_] :> Utility[x, 0, 0]
```

$$\text{Utility}(C, 0, 0) == p\,\text{Utility}(A, 0, 0) + (1 - p)\,\text{Utility}(B, 0, 0)$$

We concluded earlier that $C(p)$, by being 'accepted', apparently was part of wealth already. If a choice is being offered, then the experimental person should place himself or herself into the position, even when it is a thought experiment, that it is a real choice, and hence the value of the choice is part of wealth, in this case the $C(p)$ as the certainty equivalent (properly defined) of the choice. This conclusion allowed us to define proper values for both the expected increase in wealth and the risk, both based upon the Prospect[A - $C(p)$, B - $C(p)$, p].

But the conclusion that $U(C, 0, 0) = U(C, \mu, \rho)$ is a quite different conclusion than *cond3*. In principle these are independent conditions, and they do not have to be true at the same time The alternative notion of 'expected utility' in *cond,* the weighing (forecasted) utilities of prospective different worlds by the probabilities of those worlds, is not self-evident. It is also possible that people balance μ with ρ.

Another way to understand this is to go back to the original experimental setup (the Luenberger quote). If a choice has probability p, then $C = C(p)$ can be derived from the contour of *cond2* by which the subject balances μ with ρ. Then it does not follow yet that at the same time, additionally, *cond3* will hold. The conditions are clearly different, and thus they do not need to hold at the same time. This means that the 'oldstyle certainty equivalent condition' would not be our first hypothesis, but only a secondary possibility.

Hence, it does not seem feasible to estimate the utility function as originally thought. The case is not settled, and real world experiments are required to determine what applies.

Note: The oldstyle notion of certainty equivalence combines with the Arrow - Pratt measure of 'risk aversion', and together they form a stong team. However, the Arrow - Pratt measure is basically related to decreasing marginal utility (events with increases cannot be excluded), and it clearly is not a direct function of probability p (only indirectly, via $C(p)$, which depends upon *cond*), while ρ is not present. Arrow - Pratt runs into problems when marginal utility would behave one way and ρ would behave in the opposite way. My suggestion is that a measure for risk aversion that directly relies on ρ is more convincing. And if the oldstyle notion of certainty equivalence loses its Arrow - Pratt measure, it also becomes deficient for dealing with risk. So a key question, on the point of theory is: can marginal utility of deterministic income serve two purposes ?

8.4.6 Application to voting

There is a suggestion in the literature that prospects can be used to recover utility functions, and sometimes it is suggested that this would be cardinal utility. The first disappointment is that these utility functions are only ordinal, the second disappointment is that the 'utility' basically is expected income, and the third disappointment is that the standard derivation uses utility functions from a deterministic realm for a stochastic realm which does not seem adequate. The approach also neglects cheating, which would be a key issue for voting.

Nevertheless, we have identified some tools that can be used to better deal with voting situations. Indeed, people have to vote on risky prospects, and since the notions of risk are often confusing, this discussion has given some clarity.

8.4.7 A note on independence

Above difference with the standard approach can also be clarified by reference to Mas-Colell c.s. (1995:171) and their theorem 6.B.4 on the "independence" axiom of preferences on prospects. The discussion on prospects there is a bit sterile, since it concentrates on the probabilities (or is in danger of confusion with that) while economically we are rather interested in the commodity space. The authors overstate it when they write that the independence theorm is "at the heart of the theory of choice under uncertainty". Similarly, the discussion there on p 179-180 on the Allais paradox leaves much to be said.

Let us regard three situations: Good, Normal and Bad. Use $\supseteq$ for the preference relation 'at least as good as'. (We do this in this subsubsection, since we want to use P for a prospect.) The axiom states that the preference between prospects P and P' is independent of any third P'', or, for Pref[P', P]:

$$P \supseteq P' \iff (\alpha\, P + (1 - \alpha)\, P'' \supseteq \alpha\, P' + (1 - \alpha)\, P'' \text{ for } \alpha \in (0, 1))$$

This sounds seductively true if we look at probabilities only. But economically, we cannot neglect the wealth effect. Clearly my preference generally is Good $\supseteq$ Normal $\supseteq$ Bad, and when I am in a bad situation then clearly I am willing to gamble on getting better. But when I am in a Normal situation, then I will

hesitate on a gamble with the Bad risk. The independence axiom would force me to gamble though !

In formula's: When at Bad, there can be a β so that

- β Good + (1 - β) Bad $\supseteq$ Normal

Let us take a α and check independence from Normal itself (which the axiom allows). Then:

- α (β Good+ (1 - β) Bad) + (1 - α) Normal $\supseteq$ Normal

But when I am at Normal, I may not wish to gamble when Bad is a possible outcome !

Let us consider the Allais paradox. If we include the wealth effect and risk aversion (with the proper definition of risk), then we find that the situation need not be irrational.

`AllaisParadox[]`	contains the four prospects for the Allais paradox, discussed by Mas–Colell c.s. (1995), p179–180.

People are offered two choices, one between 1 and 2, and one between 3 and 4. They tend to prefer 1 $\supseteq$ 2 and 4 $\supseteq$ 3, though this violates the independence axiom. Below shows that their choice is not really irrational.

Prospect 1 appears to have a certain outcome, and Prospect 2 has a risk element. For the choice between 3 and 4, we can use certainty equivalence (properly defined).

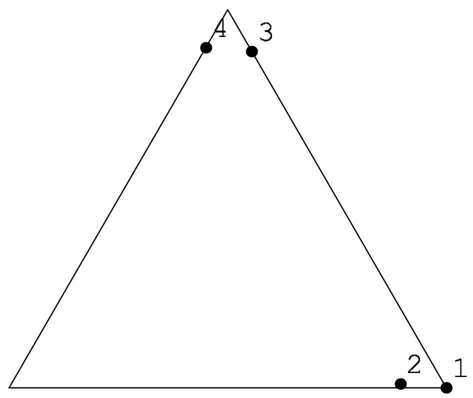
- This is available in the package.

```
alpar = AllaisParadox[]
```

{Prospect({2.5×10^6, 500000., 0}, {0, 1, 0}), Prospect({2.5×10^6, 500000., 0}, {0.1, 0.89, 0.01}),
Prospect({2.5×10^6, 500000., 0}, {0, 0.11, 0.89}), Prospect({2.5×10^6, 500000., 0}, {0.1, 0, 0.9})}

- See above for the explanation of a Prospect Plot.

```
Prospect3DPrTriangle[AllaisParadox[]];
```



- The choice between 1 and 2 is also the choice between certainty and risk. The certain value becomes our default wealth, and we can find the Risket around it.

```
PutIn[alpar[[2]] - 500000]
```

Prospect({$2. \times 10^6$, 0., −500000}, {0.1, 0.89, 0.01})

- This gives Risket[μ, σ, ρ, 1 - p], with μ the expected value, σ the standard deviation, ρ the risk (properly defined) and 1 - p the cumulated probability of a loss.

```
ToRisket[%]
```

Risket(195000., 603718., 5000., 0.01)

- For the choice between 3 and 4, we can take a reference point in the certainty equivalence (properly defined) of the least attractive option. What this is, depends upon the agent. Let us here take the minimal expected value as the reference point. First determine the expected values.

```
ProspectEV /@ alpar
```

{500000., 695000., 55000., 250000.}

- It turns out that 3 has the minimum expected value. Taking this as the addition to wealth, we can determine the riskets, and find that option 4 clearly is better. The standard deviation does not tell much, what are important are the ρ and 1 - p. These are quite comparable, so that the expected value decides.

```
{r3, r4} = ToRisket /@ PutIn /@ (Take[alpar, -2] - 55000)
```

{Risket(0., 156445., 48950., 0.89), Risket(195000., 750000., 49500., 0.9)}

Literature

EWP references are to the Economics Working Papers Archive at the Washington University at St. Louis: http://econwpa.wustl.edu. See also http://www.dataweb.nl/~cool.

Ancot & Hughes Hallet (1984), “Establishing public preferences for coalition government: an empirical study in economic planning behavior”, De Economist, p153-170

Aronson, E. (1972, 1992a), “The social animal”, Freeman, sixth edition 1992

Aronson, E. (ed) (1973, 1992b), Readings about The Social Animal”, Freeman, sixth edition 1992

Arrow, K, & T. Scitovsky eds. (1969), “Readings in welfare economics”, Allen & Unwin

Arrow, K. & F. Hahn (1971), “General competitive analysis”, North Holland

Arrow, K. (1950), “A difficulty in the concept of social welfare”, J. of political economy, pp328-346, reprinted in Arrow & Scitovsky (1969) pp147-168

Arrow, K. (1951, 1963), “Social choice and individual values”, J. Wiley

Barrow, J.D. (1998), “Impossibility. The limits of science and the science of limits”, Vintage

Bernstein (1992), “Capital ideas”, The Free Press

Bernstein (1996), “Against the gods”, Wiley

Blanchard, O. & S. Fischer (1989), “Lectures on macroeconomics”, MIT

Bochenski (1956, 1970), “A history of formal logic”, 2nd edition, Chelsea

Csikszentmihalyi (1997), “Finding flow”, Basic Books

Cool (1990c), “Why a social welfare (meta) function *does* exit: The Arrow Impossibility Theorem for Social Choice resolved, A better analysis suggested,” internal note Central Planning Bureau, The Hague, reprinted in Cool (1992b) pp57-104

Cool (1992a), “On the reasonableness and the moral desirability of the conditions of Arrow’s Theorem”, published in Cool (1992b) pp273-285

Cool (1992b), “Definition and Reality in the general theory of political economy; Some background papers 1989-1992”, ISBN 905518-207-9, Magnana Mu Publishing and Research, Rotterdam

Cool (1996b), “A constitutional amendment for an Economic Supreme Court”, ewp-get/9604003

Cool (1997b) “The solution to Arrow’s difficulty in social welfare”, ewp-get/9707001

Cool (1999, 2001), “The Economics Pack, Applications for *Mathematica*”, Scheveningen, JEL-99-0820, ISBN 90-804774-1-9

Cool (1999a), “Proper definitions for risk and uncertainty”, EconWPA ewp-get/9902002

Cool (2000), "Definition and Reality in the General Theory of Political Economy", Samuel van Houten Genootschap, ISBN 90-802263-2-7, JEL 2000-1325

Cool (2000a), "Aspects of testing (matching, ranking and rating)", ewp-get/0004004

Cool (2000b), "The choice on sustainability: information or the meta-SWF approach to a shift of preferences", ewp-othr/0004004

Cool (2000c), "The seminal contribution of Roefie Hueting to economic science: Theory and measurement of Sustainable National Income", November 16 2000, see the internet

Cullis, J. & P. Jones (1992), "Public finance & public choice", McGraw-Hill

Dahl & Lindblom (1976), "Politics, economics and welfare", Chicago

Dahl, R (1956), "A preface to democratic theory", Chicago

Dasgupta, A.K. & D.W. Pearce (1980), "Cost-benefit analysis", Macmillan

DeLong, H. (1971), "A profile of mathematical logic", Addison Wesley

Doel, J. van den, and B.C.J. van Velthoven (1990), "Democratie en welvaartstheorie", Samson

Dopp (1969), "Formale Logik", Benziger Verlag, Einsiedeln

Eatwell c.s. (1998), "The New Palgrave", Macmillan (entries risk, risk aversion, Arrow' Theorem, Social Welfare)

Elo, A.E. (1978), "The Rating of Chess Players, Past and Present", Arco Publishing, Inc., New York.

Ferguson, Th. (1967), "Mathematical statistics", Academic Press

Freeman, J. (1994), "Simulating neural networks with Mathematica", Addison-Wesley

Good, A. & F. Graybill (1950, 1963), "Introduction to the theory of statistics", McGraw-Hill

Hart (1961, 1997), "The concept of law", Oxford

Hicks (1981), "Collected essays on economic theory, volume I; Wealth and welfare", Blackwell

Hornby (1985), "Oxford advanced learner's dictionary of current English", Oxford

Huang, C. & P. Crooke (1997), "Mathematics and Mathematica for economists", Blackwell

Johansen (1969), "An examination of the relevance of Kenneth Arrow's general possibility theorem for economic planning", Economics of planning 9, pp5-41

Jorgenson, D. (1990), "Aggregate consumer behavior and the measurement of social welfare", Econometrica 58 no 5 pp1007-1040

Luce & Raiffa (1957), "Games & decisions", J. Wiley

Luenberger, D. (1998), "Investment science", Oxford

Maeder, R. (1991), "Programming in Mathematica", Addison-Wesley

Mas-colell, A., M. Whinston and J. Green (1995), "Microeconomic theory", Oxford

Mueller (1989), "Public Choice II", Cambridge

Noguchi, Asahi (1993), "General Equilibrium models", in Varian (ed), "Economic and financial modeling with Mathematica", Springer Telos 1993 pp 104-123

Pikkemaat, A. (1990), "Arrow's significance for economic science in the Netherlands", De Economist 138 No 3 p235-255

Rafiei, D. & A. Mendelzon (2000), "What is this Page Known for? Computing Web Page Reputations", http://www9.org/w9cdrom/index.html

Rijken van Olst, H. (1969), "Inleiding tot de statistiek", Part I-II-III, Van Gorcum

Saari, D, (internet), "The symmetry and complexity of elections" (S&C)

Saari, D, (internet), "Explaining all three alternative voting outcomes" (E3)

Saari, D, (internet), "Connecting and resolving Sen's and Arrow's theorems" (C&R)

Schrijver (1987), "Analyse van macro-economische beleidspreferenties met behulp van optimale besturingstheorie", Research memorandum 35, CPB

Sen, A. (1970), "Collective choice and social welfare", North Holland

Sen, A. (1986), "Social choice theory," p1073-1181 in Arrow & Intrilligator eds. (1986), "Handbook of mathematical economics, Volume III", North Holland

Sen, A. (1999a), "Development as freedom", Knopf

Sen, A. (1999b), "The possibility of social choice", AER p349-366 (excluding literature)

Shone, R. (1998), "Economic dynamics", Cambridge (uses *Mathematica*)

Sudgen, R. (1999), "Book review: On voting. A public choice approach. By Gordon Tullock", JEL 1715-1716 (December)

Swank, O. (1989). "The objectives of Democrats and Republicans In the United States, 1956—1984", Rotterdam

Takayama, A. (1974), "Mathematical economics", The Dryden Press

The Economist (2000),

The New Palgrave, Macmillan 1988 - see Eatwell

Theil, H. (1971), "Principles of econometrics", John Wiley

Tinbergen, J. (1956), "Economic policy: principles and design", North-Holland

Tinbergen, J. (1985), "Production, income and welfare: the search for an optimal social order", Weatsheaf books, Harvester press

Tobin, J. (1990), "On the theory of macroeconomic policy", Tinbergen Lecture, De economist 138 pp1-14

Varian, H. (ed) (1993), "Economic and financial modeling with Mathematica", Springer Telos

Varian, H. (ed) (1996), "Computational economics and finance", Springer Telos

Wilson, J.Q. (1993), "The moral sense", The Free Press

Wolfram, S. (1992), "Mathematica", Addison-Wesley

Wolfram, S. (1996), "Mathematica 3.0", Cambridge