

First Lutheran Church of Venice
Endowment Distribution Policy
As approved by the Voters' Assembly on July 14, 2002

Objective: To distribute an annual $4.5\% \pm 1\%$ of the church's endowment fund to designated endowment projects with relatively stable inflation-adjusted distribution amounts from year to year and to preserve the principal of the endowment fund in real dollars.

Background: In order to provide high levels of funding for church projects while preserving the endowment principal in real dollars, a large portion of the endowment fund has to be invested in stocks, which historically have produced significantly higher returns than bonds. However, while stocks have been performing well in the long run, they fluctuate significantly more than bonds.

This endowment distribution policy strives to reconcile two goals: (1) to provide a relatively high and steady flow of distributions to designated endowment projects, with a growth that on average is slightly above inflation and (2) to keep the distribution within a percentage window around a target rate (4.5%) that in the long run will preserve the endowment principal in inflation-adjusted dollars.

General idea: 4.5% of the 'new' endowment money (stemming from donations given over the past year) is distributed to fund projects. But instead of also distributing 4.5% of the 'old' endowment money, we compute the raw distribution by taking the previous year's endowment distribution and increasing it by the inflation rate plus a small 'kick-up' rate of 0.5%. We then compute what percentage of the old endowment money that represents and adjust that percentage towards 4.5% using the Gaussian bell curve. As a result, raw distribution rates near 4.5% are adjusted relatively little, while raw distribution rates further away are adjusted more such that the adjusted rate will always be between 3.5% and 5.5%. The total endowment distribution then is the adjusted distribution percentage of the old endowment money plus the 4.5% of the new endowment money. For a newly established endowment, all endowment money is 'new', so that in the first year, the endowment distribution rate is a plain 4.5%.

The small 'kick-up' rate is added on top of inflation because we have to expect program expenses to grow slightly above inflation. One reason for this is that education and ministry are particularly labor-intensive (and salaries tend to grow a bit faster than inflation).

The detailed procedure below and the formula on the following page might first appear slightly intimidating, but using a computer program (as listed in appendix C), one can simply enter (1) the endowment fund total, (2) the amount thereof given in donations during the past year (3) last year's endowment fund distribution amount and (4) inflation during last year. The program then computes and outputs the new endowment fund distribution. See appendix D for two examples.

Procedure:

(a) In order to determine the endowment distribution amount for a given year, the endowment fund total is divided into two parts: (1) the amount stemming from new donations to the endowment fund and (2) the remaining amount, which corresponds to 'old' endowment money.

(b) Of the amount stemming from new donations, 4.5% are distributed. For a new endowment fund, this will be 4.5% of the entire endowment fund.

Example: if new donations to the endowment fund during the last year amounted to \$8,900 and investment of these \$8,900 already yielded another \$100, the endowment distribution for these new donations will be \$405 (4.5% of \$9,000).

(c) For the 'old' endowment money, the raw distribution amount is the amount distributed the previous year, adjusted by inflation and an additional 'kick-up' of 0.5%.

Example: if the endowment distributions were \$10,000 the previous year, and the inflation rate was 3%, the raw distribution amount is \$10,350 (\$10,000 plus 3% for inflation plus 0.5% 'kick-up').

(d) The raw distribution rate is computed by dividing the raw distribution amount by the ‘old’ endowment money.

Example: if the raw distribution amount is \$10,350 and the ‘old’ endowment money (total endowment amount minus new donations) is \$250,000, the raw distribution rate is 4.14% (\$10,350 divided by \$250,000).

(e) The raw distribution rate is adjusted using the Gaussian bell curve. The adjustment will move the distribution rate towards the target rate of 4.5% and will make sure that the distribution rate will be in the distribution interval of [3.5% .. 5.5%]. This adjustment can be done (i) by using the formula below, or, (ii) more easily, by using the adjustment table in appendix B, or, (iii) yet more easily, by using the program in appendix C.

Example: a raw distribution rate of 4.14% is adjusted to 4.21885%.

(f) The adjusted distribution amount is computed by multiplying the ‘old’ endowment money amount by the adjusted distribution rate.

Example: following up on the examples above, the adjusted distribution amount is \$10,547.13 (4.21885% of \$250,000).

(g) The total new distribution amount is the distribution amount based on new donations from step (b) plus the adjusted distribution amount from step (f).

Example: following up on the examples above, the total new distribution amount is \$10,952.13 (\$10,547.13 + \$405). Note that the \$10,952.13 are less than the \$11655 which correspond to 4.5% of the total \$259,000 endowment fund of this example. In a year when endowment fund investment yield is above average this is reasonable so that in years with below average investment results the distribution can still grow at a relatively stable rate.

The formula:

$$D = (T - R + 2R \int_{-\infty}^{\frac{S}{R}(\frac{P*(1+K+I)}{E-C}) - T} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx) * (E - C) + T * C$$

with all the constants plugged in:

$$D = (0.035 + 0.02 \int_{-\infty}^{100(\frac{P*(1.005+I)}{E-C}) - 0.045} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx) * (E - C) + 0.045 * C$$

where

- D: new endowment distribution
- T: distribution target rate: constant 0.045 (4.5%)
- R: distribution target radius: constant 0.01 (1.0%): maximum deviation of distribution rate from target rate
- K: kick-up rate: constant 0.005 (0.5%): rate beyond inflation by which previous distribution amount is increased (before target interval adjustment)
- S: scaling factor: constant 1.0: the lower the scaling factor, the stronger the adjustment towards the target-rate
- E: total amount in endowment fund
- C: amount in endowment fund stemming from new donations in previous year
- I: inflation rate (where for example 3% would be expressed as 0.03)
- P: endowment distribution of previous year

Distribution dates: During the first two months of each calendar year, the new endowment distribution is computed using December 31 of the preceding year as a reference date for all computations. The actual endowment distribution is then made in July or whenever else the endowment committee decides.

Appendix A: Ten-year example with and without adjustment

All three scenarios assume a new initial endowment of \$200,000, no subsequent donations to the endowment, an inflation rate of 3% every year and the same market performance profile (“perf.”), with 9% return on investment in the first year, very bullish 25% and 26% and the second and third year, a crash with -15% in the fourth year, and mostly mediocre returns after that.

Scenario 1: without smoothing adjustment

In all years, exactly 4.5% of the endowment fund are distributed. The endowment distributions jump up and down quite a bit.

Year	perf.	endowment	grant	growth
0		\$200000.00	\$ 9000.00	
1	9%	\$208190.00	\$ 9368.55	4.1%
2	25%	\$248526.81	\$11183.71	19.4%
3	26%	\$299052.31	\$13457.35	20.3%
4	-15%	\$242755.72	\$10924.01	-18.8%
5	5%	\$243423.29	\$10954.05	0.3%
6	10%	\$255716.17	\$11507.23	5.1%
7	-2%	\$239324.76	\$10769.61	-6.4%
8	14%	\$260552.87	\$11724.88	8.9%
9	5%	\$261269.39	\$11757.12	0.3%
10	9%	\$271968.37	\$12238.58	4.1%

Scenario 2: with smoothing adjustment

Distribution rates are adjusted within [3.5% ... 5.5%] as proposed. The endowment distributions grow much more evenly than in scenario 1.

Year	perf.	endowment	grant	growth
0		\$200000.00	\$ 9000.00	
1	9%	\$208190.00	\$ 9325.82	3.6%
2	25%	\$248580.22	\$10035.68	7.6%
3	26%	\$300566.13	\$11410.92	13.7%
4	-15%	\$245781.93	\$11649.53	2.1%
5	5%	\$245839.02	\$11835.13	1.6%
6	10%	\$257404.27	\$12108.85	2.3%
7	-2%	\$240389.51	\$12078.25	-0.3%
8	14%	\$260274.85	\$12332.10	2.1%
9	5%	\$260339.88	\$12529.75	1.6%
10	9%	\$270113.05	\$12794.25	2.1%

Scenario 3: with double smoothing adjustment

Distribution rates are adjusted within [2.5% ... 6.5%], which is double the distribution target radius of scenario 2. The endowment distributions grow even more evenly.

Year	perf.	endowment	grant	growth
0		\$200000.00	\$ 9000.00	
1	9%	\$208190.00	\$ 9325.81	3.6%
2	25%	\$248580.24	\$ 9981.37	7.0%
3	26%	\$300634.57	\$11092.41	11.1%
4	-15%	\$246110.84	\$11398.30	2.8%
5	5%	\$246448.17	\$11652.41	2.2%
6	10%	\$258275.33	\$11971.33	2.7%
7	-2%	\$241377.93	\$12061.37	0.8%
8	14%	\$261420.87	\$12336.26	2.3%
9	5%	\$261538.84	\$12561.35	1.8%
10	9%	\$271385.46	\$12839.41	2.2%

Appendix B: Distribution Rate Adjustment Table

raw rate	adjusted rate	raw rate	adjusted rate	raw rate	adjusted rate	raw rate	adjusted rate	raw rate	adjusted rate	raw rate	adjusted rate
0.10%	3.50001%	1.02%	3.50050%	1.57%	3.50349%	2.12%	3.51741%	2.67%	3.56734%	3.22%	3.70064%
0.20%	3.50002%	1.03%	3.50052%	1.58%	3.50360%	2.13%	3.51789%	2.68%	3.56885%	3.23%	3.70418%
0.30%	3.50003%	1.04%	3.50054%	1.59%	3.50371%	2.14%	3.51837%	2.69%	3.57039%	3.24%	3.70776%
0.40%	3.50004%	1.05%	3.50056%	1.60%	3.50383%	2.15%	3.51887%	2.70%	3.57195%	3.25%	3.71139%
0.50%	3.50006%	1.06%	3.50058%	1.61%	3.50395%	2.16%	3.51938%	2.71%	3.57354%	3.26%	3.71507%
0.51%	3.50007%	1.07%	3.50060%	1.62%	3.50408%	2.17%	3.51991%	2.72%	3.57517%	3.27%	3.71879%
...	3.50007%	1.08%	3.50063%	1.63%	3.50420%	2.18%	3.52044%	2.73%	3.57682%	3.28%	3.72256%
0.54%	3.50007%	1.09%	3.50065%	1.64%	3.50434%	2.19%	3.52099%	2.74%	3.57850%	3.29%	3.72637%
0.55%	3.50008%	1.10%	3.50067%	1.65%	3.50447%	2.20%	3.52155%	2.75%	3.58021%	3.30%	3.73023%
0.56%	3.50008%	1.11%	3.50070%	1.66%	3.50461%	2.21%	3.52212%	2.76%	3.58195%	3.31%	3.73414%
0.57%	3.50008%	1.12%	3.50072%	1.67%	3.50475%	2.22%	3.52271%	2.77%	3.58372%	3.32%	3.73809%
0.58%	3.50009%	1.13%	3.50075%	1.68%	3.50490%	2.23%	3.52331%	2.78%	3.58552%	3.33%	3.74209%
0.59%	3.50009%	1.14%	3.50078%	1.69%	3.50505%	2.24%	3.52392%	2.79%	3.58736%	3.34%	3.74614%
0.60%	3.50010%	1.15%	3.50081%	1.70%	3.50521%	2.25%	3.52455%	2.80%	3.58922%	3.35%	3.75024%
0.61%	3.50010%	1.16%	3.50084%	1.71%	3.50537%	2.26%	3.52519%	2.81%	3.59112%	3.36%	3.75438%
0.62%	3.50010%	1.17%	3.50087%	1.72%	3.50554%	2.27%	3.52585%	2.82%	3.59305%	3.37%	3.75857%
0.63%	3.50011%	1.18%	3.50090%	1.73%	3.50571%	2.28%	3.52652%	2.83%	3.59501%	3.38%	3.76281%
0.64%	3.50011%	1.19%	3.50093%	1.74%	3.50588%	2.29%	3.52721%	2.84%	3.59700%	3.39%	3.76709%
0.65%	3.50012%	1.20%	3.50097%	1.75%	3.50606%	2.30%	3.52791%	2.85%	3.59903%	3.40%	3.77142%
0.66%	3.50012%	1.21%	3.50100%	1.76%	3.50624%	2.31%	3.52862%	2.86%	3.60110%	3.41%	3.77580%
0.67%	3.50013%	1.22%	3.50104%	1.77%	3.50643%	2.32%	3.52936%	2.87%	3.60319%	3.42%	3.78023%
0.68%	3.50013%	1.23%	3.50108%	1.78%	3.50663%	2.33%	3.53011%	2.88%	3.60532%	3.43%	3.78471%
0.69%	3.50014%	1.24%	3.50111%	1.79%	3.50683%	2.34%	3.53087%	2.89%	3.60749%	3.44%	3.78924%
0.70%	3.50014%	1.25%	3.50115%	1.80%	3.50703%	2.35%	3.53166%	2.90%	3.60969%	3.45%	3.79381%
0.71%	3.50015%	1.26%	3.50120%	1.81%	3.50725%	2.36%	3.53245%	2.91%	3.61193%	3.46%	3.79843%
0.72%	3.50016%	1.27%	3.50124%	1.82%	3.50746%	2.37%	3.53327%	2.92%	3.61420%	3.47%	3.80310%
0.73%	3.50016%	1.28%	3.50128%	1.83%	3.50768%	2.38%	3.53411%	2.93%	3.61651%	3.48%	3.80782%
0.74%	3.50017%	1.29%	3.50133%	1.84%	3.50791%	2.39%	3.53496%	2.94%	3.61885%	3.49%	3.81259%
0.75%	3.50018%	1.30%	3.50137%	1.85%	3.50815%	2.40%	3.53583%	2.95%	3.62123%	3.50%	3.81740%
0.76%	3.50018%	1.31%	3.50142%	1.86%	3.50839%	2.41%	3.53672%	2.96%	3.62365%	3.51%	3.82217%
0.77%	3.50019%	1.32%	3.50147%	1.87%	3.50864%	2.42%	3.53763%	2.97%	3.62611%	3.52%	3.82708%
0.78%	3.50020%	1.33%	3.50152%	1.88%	3.50889%	2.43%	3.53855%	2.98%	3.62860%	3.53%	3.83204%
0.79%	3.50021%	1.34%	3.50158%	1.89%	3.50915%	2.44%	3.53950%	2.99%	3.63113%	3.54%	3.83705%
0.80%	3.50022%	1.35%	3.50163%	1.90%	3.50942%	2.45%	3.54046%	3.00%	3.63371%	3.55%	3.84211%
0.81%	3.50022%	1.36%	3.50169%	1.91%	3.50970%	2.46%	3.54145%	3.01%	3.63632%	3.56%	3.84721%
0.82%	3.50023%	1.37%	3.50175%	1.92%	3.50998%	2.47%	3.54246%	3.02%	3.63896%	3.57%	3.85237%
0.83%	3.50024%	1.38%	3.50181%	1.93%	3.51027%	2.48%	3.54348%	3.03%	3.64165%	3.58%	3.85757%
0.84%	3.50025%	1.39%	3.50187%	1.94%	3.51057%	2.49%	3.54453%	3.04%	3.64438%	3.59%	3.86282%
0.85%	3.50026%	1.40%	3.50194%	1.95%	3.51087%	2.50%	3.54560%	3.05%	3.64715%	3.60%	3.86812%
0.86%	3.50027%	1.41%	3.50200%	1.96%	3.51119%	2.51%	3.54668%	3.06%	3.64996%	3.61%	3.87346%
0.87%	3.50028%	1.42%	3.50207%	1.97%	3.51151%	2.52%	3.54779%	3.07%	3.65281%	3.62%	3.87886%
0.88%	3.50029%	1.43%	3.50214%	1.98%	3.51184%	2.53%	3.54893%	3.08%	3.65570%	3.63%	3.88430%
0.89%	3.50031%	1.44%	3.50221%	1.99%	3.51217%	2.54%	3.55009%	3.09%	3.65863%	3.64%	3.88979%
0.90%	3.50032%	1.45%	3.50229%	2.00%	3.51252%	2.55%	3.55127%	3.10%	3.66160%	3.65%	3.89532%
0.91%	3.50033%	1.46%	3.50237%	2.01%	3.51287%	2.56%	3.55247%	3.11%	3.66462%	3.66%	3.90091%
0.92%	3.50034%	1.47%	3.50245%	2.02%	3.51324%	2.57%	3.55370%	3.12%	3.66768%	3.67%	3.90654%
0.93%	3.50036%	1.48%	3.50253%	2.03%	3.51361%	2.58%	3.55495%	3.13%	3.67078%	3.68%	3.91221%
0.94%	3.50037%	1.49%	3.50261%	2.04%	3.51399%	2.59%	3.55622%	3.14%	3.67392%	3.69%	3.91794%
0.95%	3.50039%	1.50%	3.50270%	2.05%	3.51439%	2.60%	3.55752%	3.15%	3.67711%	3.70%	3.92371%
0.96%	3.50040%	1.51%	3.50289%	2.06%	3.51479%	2.61%	3.55885%	3.16%	3.68034%	3.71%	3.92952%
0.97%	3.50042%	1.52%	3.50298%	2.07%	3.51520%	2.62%	3.56020%	3.17%	3.68361%	3.72%	3.93539%
0.98%	3.50043%	1.53%	3.50308%	2.08%	3.51562%	2.63%	3.56157%	3.18%	3.68693%	3.73%	3.94130%
0.99%	3.50045%	1.54%	3.50318%	2.09%	3.51605%	2.64%	3.56298%	3.19%	3.69029%	3.74%	3.94725%
1.00%	3.50047%	1.55%	3.50328%	2.10%	3.51650%	2.65%	3.56440%	3.20%	3.69369%	3.75%	3.95325%
1.01%	3.50048%	1.56%	3.50338%	2.11%	3.51695%	2.66%	3.56586%	3.21%	3.69714%	3.76%	3.95930%

raw rate	adjusted rate	raw rate	adjusted rate	raw rate	adjusted rate	raw rate	adjusted rate	raw rate	adjusted rate	raw rate	adjusted rate
3.77%	3.96539%	4.37%	4.39657%	4.97%	4.86164%	5.57%	5.21529%	6.17%	5.40499%	6.77%	5.47669%
3.78%	3.97152%	4.38%	4.40449%	4.98%	4.86877%	5.58%	5.21977%	6.18%	5.40695%	6.78%	5.47729%
3.79%	3.97770%	4.39%	4.41241%	4.99%	4.87586%	5.59%	5.22420%	6.19%	5.40888%	6.79%	5.47788%
3.80%	3.98392%	4.40%	4.42035%	5.00%	4.88292%	5.60%	5.22858%	6.20%	5.41078%	6.80%	5.47845%
3.81%	3.99019%	4.41%	4.42829%	5.01%	4.88995%	5.61%	5.23291%	6.21%	5.41264%	6.81%	5.47901%
3.82%	3.99650%	4.42%	4.43624%	5.02%	4.89693%	5.62%	5.23719%	6.22%	5.41448%	6.82%	5.47956%
3.83%	4.00285%	4.43%	4.44420%	5.03%	4.90388%	5.63%	5.24143%	6.23%	5.41628%	6.83%	5.48009%
3.84%	4.00925%	4.44%	4.45216%	5.04%	4.91080%	5.64%	5.24562%	6.24%	5.41805%	6.84%	5.48062%
3.85%	4.01569%	4.45%	4.46013%	5.05%	4.91768%	5.65%	5.24976%	6.25%	5.41979%	6.85%	5.48113%
3.86%	4.02217%	4.46%	4.46810%	5.06%	4.92452%	5.66%	5.25386%	6.26%	5.42150%	6.86%	5.48163%
3.87%	4.02869%	4.47%	4.47607%	5.07%	4.93133%	5.67%	5.25791%	6.27%	5.42318%	6.87%	5.48211%
3.88%	4.03525%	4.48%	4.48405%	5.08%	4.93809%	5.68%	5.26191%	6.28%	5.42483%	6.88%	5.48259%
3.89%	4.04186%	4.49%	4.49203%	5.09%	4.94481%	5.69%	5.26586%	6.29%	5.42646%	6.89%	5.48305%
3.90%	4.04850%	4.50%	4.50000%	5.10%	4.95150%	5.70%	5.26977%	6.30%	5.42805%	6.90%	5.48350%
3.91%	4.05519%	4.51%	4.50798%	5.11%	4.95814%	5.71%	5.27363%	6.31%	5.42961%	6.91%	5.48395%
3.92%	4.06192%	4.52%	4.51596%	5.12%	4.96475%	5.72%	5.27744%	6.32%	5.43115%	6.92%	5.48438%
3.93%	4.06868%	4.53%	4.52394%	5.13%	4.97131%	5.73%	5.28121%	6.33%	5.43266%	6.93%	5.48480%
3.94%	4.07548%	4.54%	4.53191%	5.14%	4.97783%	5.74%	5.28493%	6.34%	5.43414%	6.94%	5.48521%
3.95%	4.08232%	4.55%	4.53988%	5.15%	4.98431%	5.75%	5.28861%	6.35%	5.43560%	6.95%	5.48561%
3.96%	4.08920%	4.56%	4.54785%	5.16%	4.99075%	5.76%	5.29224%	6.36%	5.43702%	6.96%	5.48601%
3.97%	4.09612%	4.57%	4.55581%	5.17%	4.99715%	5.77%	5.29582%	6.37%	5.43843%	6.97%	5.48639%
3.98%	4.10307%	4.58%	4.56377%	5.18%	5.00350%	5.78%	5.29936%	6.38%	5.43980%	6.98%	5.48676%
3.99%	4.11005%	4.59%	4.57172%	5.19%	5.00981%	5.79%	5.30286%	6.39%	5.44115%	6.99%	5.48713%
4.00%	4.11708%	4.60%	4.57966%	5.20%	5.01608%	5.80%	5.30631%	6.40%	5.44248%	7.00%	5.48748%
4.01%	4.12414%	4.61%	4.58759%	5.21%	5.02230%	5.81%	5.30971%	6.41%	5.44378%	7.01%	5.48783%
4.02%	4.13123%	4.62%	4.59552%	5.22%	5.02848%	5.82%	5.31307%	6.42%	5.44505%	7.02%	5.48816%
4.03%	4.13836%	4.63%	4.60344%	5.23%	5.03461%	5.83%	5.31639%	6.43%	5.44630%	7.03%	5.48849%
4.04%	4.14552%	4.64%	4.61134%	5.24%	5.04070%	5.84%	5.31966%	6.44%	5.44753%	7.04%	5.48881%
4.05%	4.15271%	4.65%	4.61924%	5.25%	5.04675%	5.85%	5.32289%	6.45%	5.44873%	7.05%	5.48913%
4.06%	4.15994%	4.66%	4.62712%	5.26%	5.05275%	5.86%	5.32608%	6.46%	5.44991%	7.06%	5.48943%
4.07%	4.16720%	4.67%	4.63499%	5.27%	5.05870%	5.87%	5.32922%	6.47%	5.45107%	7.07%	5.48973%
4.08%	4.17449%	4.68%	4.64284%	5.28%	5.06461%	5.88%	5.33232%	6.48%	5.45221%	7.08%	5.49002%
4.09%	4.18181%	4.69%	4.65069%	5.29%	5.07048%	5.89%	5.33538%	6.49%	5.45332%	7.09%	5.49030%
4.10%	4.18916%	4.70%	4.65852%	5.30%	5.07629%	5.90%	5.33840%	6.50%	5.45440%	7.10%	5.49058%
4.11%	4.19654%	4.71%	4.66633%	5.31%	5.08206%	5.91%	5.34137%	6.51%	5.45547%	7.11%	5.49085%
4.12%	4.20395%	4.72%	4.67412%	5.32%	5.08779%	5.92%	5.34430%	6.52%	5.45652%	7.12%	5.49111%
4.13%	4.21139%	4.73%	4.68190%	5.33%	5.09346%	5.93%	5.34719%	6.53%	5.45754%	7.13%	5.49136%
4.14%	4.21885%	4.74%	4.68967%	5.34%	5.09909%	5.94%	5.35004%	6.54%	5.45855%	7.14%	5.49161%
4.15%	4.22634%	4.75%	4.69741%	5.35%	5.10468%	5.95%	5.35285%	6.55%	5.45954%	7.15%	5.49185%
4.16%	4.23386%	4.76%	4.70513%	5.36%	5.11021%	5.96%	5.35562%	6.56%	5.46050%	7.16%	5.49209%
4.17%	4.24140%	4.77%	4.71284%	5.37%	5.11570%	5.97%	5.35835%	6.57%	5.46145%	7.17%	5.49232%
4.18%	4.24897%	4.78%	4.72052%	5.38%	5.12114%	5.98%	5.36104%	6.58%	5.46237%	7.18%	5.49254%
4.19%	4.25656%	4.79%	4.72818%	5.39%	5.12654%	5.99%	5.36368%	6.59%	5.46328%	7.19%	5.49275%
4.20%	4.26418%	4.80%	4.73582%	5.40%	5.13188%	6.00%	5.36629%	6.60%	5.46417%	7.20%	5.49297%
4.21%	4.27182%	4.81%	4.74344%	5.41%	5.13718%	6.01%	5.36887%	6.61%	5.46504%	7.21%	5.49317%
4.22%	4.27948%	4.82%	4.75103%	5.42%	5.14243%	6.02%	5.37140%	6.62%	5.46589%	7.22%	5.49337%
4.23%	4.28716%	4.83%	4.75860%	5.43%	5.14763%	6.03%	5.37389%	6.63%	5.46673%	7.23%	5.49357%
4.24%	4.29487%	4.84%	4.76614%	5.44%	5.15279%	6.04%	5.37635%	6.64%	5.46755%	7.24%	5.49376%
4.25%	4.30259%	4.85%	4.77366%	5.45%	5.15789%	6.05%	5.37877%	6.65%	5.46834%	7.25%	5.49394%
4.26%	4.31033%	4.86%	4.78115%	5.46%	5.16295%	6.06%	5.38115%	6.66%	5.46913%	7.26%	5.49412%
4.27%	4.31810%	4.87%	4.78861%	5.47%	5.16796%	6.07%	5.38349%	6.67%	5.46989%	7.27%	5.49429%
4.28%	4.32588%	4.88%	4.79605%	5.48%	5.17292%	6.08%	5.38580%	6.68%	5.47064%	7.28%	5.49446%
4.29%	4.33367%	4.89%	4.80346%	5.49%	5.17783%	6.09%	5.38807%	6.69%	5.47138%	7.29%	5.49463%
4.30%	4.34148%	4.90%	4.81084%	5.50%	5.18260%	6.10%	5.39031%	6.70%	5.47209%	7.30%	5.49479%
4.31%	4.34931%	4.91%	4.81819%	5.51%	5.18741%	6.11%	5.39251%	6.71%	5.47279%	7.31%	5.49495%
4.32%	4.35716%	4.92%	4.82551%	5.52%	5.19218%	6.12%	5.39468%	6.72%	5.47348%	7.32%	5.49510%
4.33%	4.36501%	4.93%	4.83280%	5.53%	5.19690%	6.13%	5.39681%	6.73%	5.47415%	7.33%	5.49525%
4.34%	4.37289%	4.94%	4.84006%	5.54%	5.20157%	6.14%	5.39890%	6.74%	5.47481%	7.34%	5.49539%
4.35%	4.38077%	4.95%	4.84729%	5.55%	5.20619%	6.15%	5.40097%	6.75%	5.47545%	7.35%	5.49553%
4.36%	4.38866%	4.96%	4.85448%	5.56%	5.21076%	6.16%	5.40300%	6.76%	5.47608%	7.36%	5.49566%

raw rate	adjusted rate	raw rate	adjusted rate	raw rate	adjusted rate	raw rate	adjusted rate	raw rate	adjusted rate	raw rate	adjusted rate
7.37%	5.49580%	7.59%	5.49800%	7.81%	5.49907%	8.03%	5.49958%	8.25%	5.49982%	...	5.49993%
7.38%	5.49592%	7.60%	5.49806%	7.82%	5.49910%	8.04%	5.49960%	8.26%	5.49983%	8.49%	5.49993%
7.39%	5.49605%	7.61%	5.49813%	7.83%	5.49913%	8.05%	5.49961%	8.27%	5.49984%	8.50%	5.49994%
7.40%	5.49617%	7.62%	5.49819%	7.84%	5.49916%	8.06%	5.49963%	8.28%	5.49984%	...	5.49994%
7.41%	5.49629%	7.63%	5.49825%	7.85%	5.49919%	8.07%	5.49964%	8.29%	5.49985%	8.53%	5.49994%
7.42%	5.49640%	7.64%	5.49831%	7.86%	5.49922%	8.08%	5.49966%	8.30%	5.49986%	8.54%	5.49995%
7.43%	5.49651%	7.65%	5.49837%	7.87%	5.49925%	8.09%	5.49967%	8.31%	5.49986%	...	5.49995%
7.44%	5.49662%	7.66%	5.49842%	7.88%	5.49928%	8.10%	5.49968%	8.32%	5.49987%	8.58%	5.49995%
7.45%	5.49672%	7.67%	5.49848%	7.89%	5.49930%	8.11%	5.49969%	8.33%	5.49987%	8.59%	5.49996%
7.46%	5.49682%	7.68%	5.49853%	7.90%	5.49933%	8.12%	5.49971%	8.34%	5.49988%	...	5.49996%
7.47%	5.49692%	7.69%	5.49858%	7.91%	5.49935%	8.13%	5.49972%	8.35%	5.49988%	8.63%	5.49996%
7.48%	5.49702%	7.70%	5.49863%	7.92%	5.49937%	8.14%	5.49973%	8.36%	5.49989%	8.64%	5.49997%
7.49%	5.49711%	7.71%	5.49867%	7.93%	5.49940%	8.15%	5.49974%	8.37%	5.49989%	...	5.49997%
7.50%	5.49730%	7.72%	5.49872%	7.94%	5.49942%	8.16%	5.49975%	8.38%	5.49990%	8.71%	5.49997%
7.51%	5.49739%	7.73%	5.49876%	7.95%	5.49944%	8.17%	5.49976%	8.39%	5.49990%	8.72%	5.49998%
7.52%	5.49747%	7.74%	5.49880%	7.96%	5.49946%	8.18%	5.49977%	8.40%	5.49990%	...	5.49998%
7.53%	5.49755%	7.75%	5.49885%	7.97%	5.49948%	8.19%	5.49978%	8.41%	5.49991%	8.82%	5.49998%
7.54%	5.49763%	7.76%	5.49889%	7.98%	5.49950%	8.20%	5.49978%	8.42%	5.49991%	8.83%	5.49999%
7.55%	5.49771%	7.77%	5.49892%	7.99%	5.49952%	8.21%	5.49979%	8.43%	5.49992%	...	5.49999%
7.56%	5.49779%	7.78%	5.49896%	8.00%	5.49953%	8.22%	5.49980%	8.44%	5.49992%	9.06%	5.49999%
7.57%	5.49786%	7.79%	5.49900%	8.01%	5.49955%	8.23%	5.49981%	8.45%	5.49992%	9.07%	5.50000%
7.58%	5.49793%	7.80%	5.49903%	8.02%	5.49957%	8.24%	5.49982%	8.46%	5.49993%	above	5.50000%

Appendix C: C program to compute endowment distribution amount

```
/* this file:                endowment.c
 * author:                   Ulf Hermjakob
 * to compile this file (in UNIX): gcc -lm endowment.c
 * to move compiled file into
   proper location:         mv a.out endowment
 * to call program:         endowment
 */

#include <stdio.h>
#include <math.h>
#define one_over_sqrt_2_pi 0.3989422804014327

double gauss(double x)
{
    return(one_over_sqrt_2_pi * exp(-0.5 * x * x));
}

double core_gauss_integral(double from, double to)
{
    double s = 0.0;
    double granularity = 0.00001;
    double x;

    for (x=from; x<to; x=x+granularity)
        s += granularity * gauss(x);
    if (to > from)
        s -= 0.5 * granularity * gauss(from);
    return (s);
}

double gauss_integral(double x)
{
    if (x >= 5) return (1.0);
    if (x >= 4) return (0.9999683 + core_gauss_integral(4.0, x));
    if (x >= 3) return (0.9986501 + core_gauss_integral(3.0, x));
    if (x >= 2) return (0.9772 + core_gauss_integral(2.0, x));
    if (x >= 1) return (0.8413 + core_gauss_integral(1.0, x));
    if (x >= 0) return (0.5 + core_gauss_integral(0.0, x));
    return (1.0 - gauss_integral(-1 * x));
}

double gauss_smoothing (double raw_rate, double target_rate,
double deviation_radius_rate, double s)
{
    return (target_rate
- deviation_radius_rate
+ (2 * deviation_radius_rate
* gauss_integral(s * (raw_rate - target_rate))));
}

double endowment_grant (double endowment_total, double last_endowment_distr,
double inflation_rate, double new_endowment_money,
int verbose_type, double deviation_radius_rate)
{
    double target_rate = 0.045;
    double kick_up_rate = 0.005;
    double old_endowment_money = 0.0;
    double raw_endowment_distr = 0.0;
    double raw_endowment_rate = 0.0;
    double adj_endowment_rate = 0.0;
    double adj_endowment_distr = 0.0;
}
```

```

old_endowment_money = endowment_total - new_endowment_money;
raw_endowment_distr =
    last_endowment_distr * (1.0 + inflation_rate + kick_up_rate);
if (old_endowment_money > 0.0)
    raw_endowment_rate = raw_endowment_distr / old_endowment_money;
if (deviation_radius_rate <= 0.0)
    adj_endowment_rate = raw_endowment_rate;
else
    adj_endowment_rate = gauss_smoothing(raw_endowment_rate, target_rate,
        deviation_radius_rate, 1.0 / deviation_radius_rate);
adj_endowment_distr = adj_endowment_rate * old_endowment_money
    + target_rate * new_endowment_money;

if (verbose_type == 2)
{
    printf("\n");
    printf("Endowment total:           $%9.2f\n", endowment_total);
    printf("Last endowment distribution: $%9.2f\n", last_endowment_distr);
    printf("New endowment donations:       $%9.2f\n", new_endowment_money);
    printf("Endowment w/o new donations: $%9.2f\n", old_endowment_money);
    printf("Inflation rate:                 %6.2f%%\n", 100 * inflation_rate);
    printf("Kick-up rate:                   %6.2f%%\n", 100 * kick_up_rate);
    printf("Raw endowment distribution: $%9.2f (which is $%.2f plus %%.2f%% plus %%.2f%%)\n",
        raw_endowment_distr, last_endowment_distr,
        100 * inflation_rate, 100 * kick_up_rate);
    printf("Raw endowment rate:             %6.2f%% (which is $%.2f out of $%.2f)\n",
        100 * raw_endowment_rate,
        raw_endowment_distr, old_endowment_money);
    printf("Adjusted endowment rate:        %6.2f%% ", 100 * adj_endowment_rate);
    printf("(centering inside interval [%3.1f%% ... %3.1f%%])\n",
        100 * (target_rate - deviation_radius_rate),
        100 * (target_rate + deviation_radius_rate));

    printf("New endowment distribution: $%9.2f (which is %%.2f%% of $%.2f",
        adj_endowment_distr, 100 * adj_endowment_rate,
        old_endowment_money);
    if (new_endowment_money > 0.0)
        printf(" plus %%.2f%% of $%.2f", 100 * target_rate, new_endowment_money);
    printf(")\n");
}
return (adj_endowment_distr);
}

void print_gauss_smoothing_table()
{
    double x;
    for (x=0.0; x<=0.10001; x=x+0.001)
        printf ("%4.1f%% %6.4f%%\n", 100 * x, 100 * gauss_smoothing(x, 0.045, 0.01, 100));
}

/* static int market_perf[11] = {0, 9, -15, 25, 26, 5, 10, -2, 14, 5, 8}; */
static int market_perf[11] = {0, 9, 25, 26, -15, 5, 10, -2, 14, 5, 9};

void endowment_scenario(double init_endowment, double infl_rate, int n_years,
int *end_perf, int grant_proc, double deviation_radius_rate)
{
    int i;
    double endowment_total = init_endowment;
    double new_endowment = init_endowment;
    double last_grant = 0.0;
    double grant, grant_growth;

```



```

printf("Year perf. endowment grant growth\n");
for (i=0; i<=n_years; i++)
{
    endowment_total *= 1.0 + (end_perf[i] / 100.0);
    if (grant_proc == 1)
        grant = endowment_grant(endowment_total, last_grant, infl_rate,
new_endowment, 0, deviation_radius_rate);
    else
        grant = endowment_total * 0.045;
    if (last_grant > 0.0)
grant_growth = (grant - last_grant) / last_grant * 100;
    else
grant_growth = 999.9;
    printf("%3d %4d%% $%9.2f $%8.2f %5.1f%%\n",
i, end_perf[i], endowment_total, grant, grant_growth);
    new_endowment = 0.0;
    last_grant = grant;
    endowment_total -= grant;
}
}

void single_comp()
{
    double endowment_total, last_endowment_distr,
inflation_percentage, inflation_rate,
new_endowment_money, new_endowment_distr;

    printf("Please enter endowment total: ");
    scanf("%lf", &endowment_total);

    printf("Please enter last year's endowment distribution: ");
    scanf("%lf", &last_endowment_distr);

    printf("Please enter last year's inflation (in percent): ");
    scanf("%lf", &inflation_percentage);
    inflation_rate = inflation_percentage / 100;

    printf("Please enter last year's new endowment donations: ");
    scanf("%lf", &new_endowment_money);

    endowment_grant(endowment_total, last_endowment_distr,
inflation_rate, new_endowment_money, 2, 0.01);
}

main()
{
    single_comp();
    /*
    printf("Both scenarios assume a new initial endowment of $200,000 and inflation of 3% ev-
every year.\n\n");
    printf("\nScenario 1: without smoothing adjustment\n");
    endowment_scenario(200000.0, 0.03, 10, market_perf, 0, 0.01);
    printf("Scenario 2: with smoothing adjustment\n");
    endowment_scenario(200000.0, 0.03, 10, market_perf, 1, 0.01);
    printf("Scenario 3: with double smoothing adjustment\n");
    endowment_scenario(200000.0, 0.03, 10, market_perf, 1, 0.02);
    */
    /*
    print_gauss_smoothing_table();
    */
}

```

Appendix D: sample usage of program in appendix C

Example 1: A new endowment fund.

Let's assume that the endowment starts with \$200,000. The user is prompted to supply four numbers. Then the computer prints intermediate steps and the new endowment distribution.

```
Please enter endowment total: 200000
Please enter last year's endowment distribution: 0
Please enter last year's inflation (in percent): 1.8
Please enter last year's new endowment donations: 200000

Endowment total:                $200000.00
Last endowment distribution: $    0.00
New endowment donations:        $200000.00
Endowment w/o new donations: $    0.00
Inflation rate:                  1.80%
Kick-up rate:                    0.50%
Raw endowment distribution: $    0.00 (which is $0.00 plus 1.80% plus 0.50%)
Raw endowment rate:              0.00% (which is $0.00 out of $0.00)
Adjusted endowment rate:         3.50% (centering inside interval [3.5% ... 5.5%])
New endowment distribution: $  9000.00 (which is 3.50% of $0.00 plus 4.50% of $200000.00)
```

Example 2: The following year.

\$9,000 were distributed in the first year. Let's assume that there have been \$10,000 in new donations and that the endowment fund has appreciated in value to a total of \$218,190, including first year's distribution and new donations:

```
Please enter endowment total: 218190
Please enter last year's endowment distribution: 9000
Please enter last year's inflation (in percent): 3.0
Please enter last year's new endowment donations: 10000

Endowment total:                $218190.00
Last endowment distribution: $  9000.00
New endowment donations:        $ 10000.00
Endowment w/o new donations: $208190.00
Inflation rate:                  3.00%
Kick-up rate:                    0.50%
Raw endowment distribution: $  9315.00 (which is $9000.00 plus 3.00% plus 0.50%)
Raw endowment rate:              4.47% (which is $9315.00 out of $208190.00)
Adjusted endowment rate:         4.48% (centering inside interval [3.5% ... 5.5%])
New endowment distribution: $  9775.82 (which is 4.48% of $208190.00 plus 4.50% of $10000.00)
```