

WinCross's Sample Balancing Program

WinCross's Sample Balancing Program determines weights to be applied to each respondent in a sample in order that the weighted marginal distributions on each of v variables match preset distributions on those v variables. The program is an adaptation of the procedure known today as "iterative proportional fitting" and referred to by some as "rim weighting." The original iterative proportional fitting procedure was devised by W. Edwards Deming and Frederick F. Stephan, first published in their December, 1940 paper, "On a Least Squares Adjustment of a Sampled Frequency Table when the Expected Marginal Totals are Known," in Volume 11 of *The Annals of Mathematical Statistics*, pages 427-444, and further explicated in Chapter 7 of Deming's book, Statistical Adjustment of Data (New York: John Wiley & Sons, 1943). WinCross's adaptation was developed by J. Stephens Stock, a colleague of Deming, in the 1960s.

To describe the procedure used by the program, we must first establish some notation. Let v be the number of variables to be considered in the balancing. Let c_i denote the number of levels (sometimes referred to as "breaks") of the i -th variable, $i=1, \dots, v$. Let $p_{j_1 \dots j_v}$ denote the proportion of respondents in the sample in level j_1 on variable 1, j_2 on variable 2, \dots , j_v on variable v , where $j_i = 1, \dots, c_i$. Let $f_{j_i}^i$ denote the marginal proportion in the sample of level j_i of variable i ($j_i=1, \dots, c_i$, $i=1, \dots, v$).

To make things concrete, let $v=3$, with the three variables being income ($i=1$), age ($i=2$), and region ($i=3$). Suppose there are 5 income breaks ($c_1=5$), 10 age breaks ($c_2=10$), and 9 region breaks ($c_3=9$). Then, in our notation, if for example $j_1=2$, $j_2=1$, and $j_3=4$, then $p_{j_1 j_2 j_3} = p_{214}$ is the proportion of the sample that are of income level 2, age level 1, and region level 4. And, as another example of the interpretation of this notation, if $i=3$ then $f_{j_i}^i = f_{j_3}^3 = f_2^3$ is the proportion of the sample that are in region level 2 (the superscript "3" indicates that we are looking at variable 3, region, and the subscript "2" indicates that we are looking at level 2 of that variable).

Clearly $f_{j_i}^i$ can be determined by adding up all the $p_{j_1 \dots j_v}$ across all the values of each of the $v-1$ j_k for which $k \neq i$. For example, to obtain f_2^3 one adds up all the proportions $p_{j_1 j_2 2}$ across $j_1=1,2$ and $j_2=1,2,3$. We express this relation symbolically as

$$f_{j_i}^i = \sum_{j_k, k \neq i} p_{j_1 \dots j_v}$$

These $f_{j_i}^i$ are called sample rim percents.

Suppose that the preset distributions on the v variables are given by the set of target proportions $g_{j_i}^i$. The object of the sample balancing program is to find a set of weights $w_{j_1 \dots j_v}$ such that if, when looking at the j_i -th break, instead of adding up the

$p_{j_1 \dots j_v}$ across all but the i -th category, we add up the $w_{j_1 \dots j_v} p_{j_1 \dots j_v}$ across all but the i -th category, we will obtain the $g_{j_i}^i$. That is,

$$g_{j_i}^i = \sum_{j_k, k \neq i} w_{j_1 \dots j_v} p_{j_1 \dots j_v}$$

To illustrate the WinCross sample balancing program, we will use a data set of 1000 respondents, whose income, age, and region distribution are given in the table below, along with the target percentages input into the sample balancing program.

		sample	target
Income	n	f	g
i=1		percents	percents
	1	124	12.40
	2	150	15.00
	3	305	30.50
	4	221	22.10
	5	200	20.00
Total		1000	100.00
Age			
i=2			
	1	6	0.60
	2	34	3.40
	3	185	18.50
	4	276	27.60
	5	248	24.80
	6	92	9.20
	7	64	6.40
	8	65	6.50
	9	26	2.60
	10	4	0.40
Total		1000	100.00
Region			
i=3			
	1	55	5.50
	2	79	7.90
	3	187	18.70
	4	129	12.90
	5	169	16.90
	6	56	5.60
	7	103	10.30
	8	82	8.20
	9	140	14.00
Total		1000	100.00

In this example there are three *variables* (income, age, and region) to be used in determining the weights, with 5, 10, and 9 *levels*, respectively. There are potentially $5 \times 10 \times 9 = 450$ cells in the three-way crosstabulation of these data. In actuality, there are only 275 cells for which there is at least one respondent among the 1000 in the sample. The frequency distribution across these 275 cells is given in Appendix I.

The procedure for determining the weights is iterative. Each iterative "round" consists of v "passes," one "pass" through each of the v variables. We begin at "round 0" by setting all weights $w_{j_1 \dots j_v}(0, i)$ equal to 1, i.e., we begin with the unweighted data.

Suppose we are on the i -th "pass" in "round $t+1$." Let $w_{j_1 \dots j_v}(t, i)$ denote the weights at this point in the iterative process. Let $g_{j_i}^i(t)$ denote the results of the computation

$$g_{j_i}^i(t) = \sum_{j_k, k \neq i} w_{j_1 \dots j_v}(t, i) p_{j_1 \dots j_v}$$

These $g_{j_i}^i(t)$ are called estimated target rim percents.

At the first pass ($i=1$) of the t -th round of the iterative procedure the program calculates a set of increments $d_{j_1 \dots j_v}(t, 1)$ to add to the $w_{j_1 \dots j_v}(t-1, v)$, producing $w_{j_1 \dots j_v}(t, 1) = w_{j_1 \dots j_v}(t-1, v) + d_{j_1 \dots j_v}(t, 1)$. At the i -th pass ($i > 1$) of the t -th round of the iterative procedure the program calculates a set of increments $d_{j_1 \dots j_v}(t, i)$ to add to the $w_{j_1 \dots j_v}(t, i-1)$, producing $w_{j_1 \dots j_v}(t, i) = w_{j_1 \dots j_v}(t, i-1) + d_{j_1 \dots j_v}(t, i)$.

These increments are given by the formula

$$d_{j_1 j_2 \dots j_v} = [g_{j_i}^i(t) - g_{j_i}^i] / f_{j_i}^i$$

That is, we compare the ratio of the estimated target rim percent to the sample rim percent to the ratio of the target rim percent to the sample rim percent, and increment or decrement by the difference between these two ratios.

Let us illustrate this on our sample data. The first pass of round 1 considers the first variable, income. Suppose one wanted to only balance to the marginal on *income*. The way to determine those weights is a simple process: weight each respondent by a factor such that the sum of the weights equals the target for each level. The appropriate weights that accomplish this are simply the ratio of the target rim percents to the sample rim percents for each level of income. This is illustrated in the following table (since the total count is 1000, we multiply our sample and target percents by 10 I what follows):

income	sample	target	target/sample
1	124	179.5	1.44758
2	150	232.0	1.54667
3	305	272.8	0.89443
4	221	143.4	0.64887
5	200	172.3	0.86150
	1000	1000.0	

When one applies the income weight (in this case, the initial weight 1.0) to each respondent and looks at the weighted marginals by income, one obtains the following table:

income	sample	est target	est target/sample
1	124	124	1.0
2	150	150	1.0
3	305	305	1.0
4	221	221	1.0
5	200	200	1.0
	1000	1000	

Our goal is to achieve the target, and so we compare the ratio of estimated target rim percent to sample rim percent with the ratio of target rim percent to sample rim percent, and calculate the additive or subtractive adjustment to the initial weight of 1.0, labeled "delta", necessary to make these two ratios equal. The following table presents this comparison and adjustment.

income	target/sample	est target / sample	delta
1	1.44758	1.0	0.44758
2	1.54667	1.0	0.54667
3	0.89443	1.0	-0.10557
4	0.64887	1.0	-0.35113
5	0.86150	1.0	-0.13850

So, for example, the new weight for income group 1 is $1.0 + 0.44758 = 1.44758$.

We now move to pass 2 of round 1, where we look at the same initial table for the next variable, *age*.

age	sample	target	target/sample
1	6	38.7	6.45000
2	34	90.7	2.66765
3	185	190.9	1.03189
4	276	216.0	0.78261
5	248	180.2	0.72661
6	92	64.4	0.70000
7	64	51.7	0.80781
8	65	88.0	1.35385
9	26	59.1	2.27308
10	4	20.3	5.07500
	1000	1000.0	

When one applies the income weight to each respondent and looks at the weighted marginals by age, one obtains the following table:

age	sample	est target	est target/sample
1	6	5.58	0.92923
2	34	41.17	1.21103
3	185	182.08	0.98421
4	276	277.32	1.00478
5	248	237.35	0.95708
6	92	91.18	0.99111
7	64	63.03	0.98482
8	65	67.97	1.04564
9	26	30.12	1.15848
10	4	4.20	1.04925
	1000	1000.00	

As our goal is to achieve the target, we compare the ratio of weighted data to data with the ratio of target to data, and calculate the additive or subtractive adjustment to the current respondent weight, labeled "delta", necessary to make these two ratios equal. The following table presents this comparison and adjustment.

age	target/sample	est target / sample	delta
1	6.45000	0.92923	5.52077
2	2.66765	1.21103	1.45662
3	1.03189	0.98421	0.04768
4	0.78261	1.00478	-0.22218
5	0.72661	0.95708	-0.23046
6	0.70000	0.99111	-0.29111
7	0.80781	0.98482	-0.17701
8	1.35385	1.04564	0.30821
9	2.27308	1.15848	1.11460
10	5.07500	1.04925	4.02575

How are these deltas used? Let's look at a few examples. Consider a respondent whose income group was 2. After the first pass of round 1 he would be given a weight of 1.54667. Now suppose his age group were 10. After the second pass of round 1 his weight would be $1.54667+4.02575=5.57242$. And if his age group were 4 his weight would now be $1.54667-0.22218=1.32449$.

We now move to pass 3 of round 1, where we consider the next variable, *region*. We first develop our goal, the "target/sample" ratio, given in the following table.

region	sample	target	target/sample
1	55	51.4	0.93455
2	79	142.1	1.79873
3	187	164.4	0.87914
4	129	71.4	0.55349
5	169	190.5	1.12722
6	56	63.0	1.12500
7	103	109.1	1.05922
8	82	64.0	0.78049
9	140	144.1	1.02929
	1000	1000.0	

When one applies the weight based on both income and age to each respondent and looks at the weighted marginals by region, one obtains the following table:

region	sample	est target	est target/sample
1	55	49.27	0.89579
2	79	81.62	1.03317
3	187	184.37	0.98593
4	129	143.46	1.11207
5	169	168.87	0.99920
6	56	58.81	1.05024
7	103	104.50	1.01457
8	82	74.85	0.91278
9	140	134.26	0.95899
	1000	1000.00	

Once again our goal is to achieve the target, and so we compare the ratio of weighted data to data with the ratio of target to data, and calculate the additive or subtractive adjustment to each respondent's weight, labeled "delta", necessary to make these two ratios equal. The following table presents this comparison and adjustment.

region	target/sample	est target / sample	delta
1	0.93455	0.89579	0.03876
2	1.79873	1.03317	0.76557
3	0.87914	0.98593	-0.10679
4	0.55349	1.11207	-0.55858
5	1.12722	0.99920	0.12802
6	1.12500	1.05024	0.07476
7	1.05922	1.01457	0.04465
8	0.78049	0.91278	-0.13229
9	1.02929	0.95899	0.07030

Continuing with our examples, consider a respondent whose income group was 2 and his age group was 10. After pass 2 of round 1 he would be given a weight of 5.57242. If his region were 2 his new weight after pass 3 of round 1 would be $5.57242 + 0.76557 = 6.33799$. And if his region were 4 his weight would now be $5.57242 - 0.55858 = 5.01383$.

The WinCross sample balancing program now applies these new weights to the respondents and begins round 2, once again in pass 1 looking at the income marginals. The principle in each step is the same: adjust the weights so that the ratio of estimated target rim percents data to sample rim percents equals the ratio of target rim percents to sample rim percents.

The program continues iterating until a criterion of goodness of fit has been met. WinCross uses the measure

$$\sqrt{\sum_{i=1}^v \sum_{j=1}^{c_i} [(g_{j_i}^i(t) - g_{j_i}^i) / f_{j_i}^i]^2}$$

This measure is the sum of squares of deviations across levels and variables between the ratio of estimated target rim percents to sample rim percents and the ratio of actual target rim percents to sample rim percents. The program iterates until this measure is less than some preset value (with default set at 0.0005). This is in contrast to iterative proportional fitting, which has no overall criterion and iterates until each $g_{j_i}^i(t)$ is within some preset distance from $g_{j_i}^i$, that is, until each estimated target rim percent is within some preset distance from the actual target rim percent.

Appendix I also contains the results of the application of the WinCross sample balancing program to these data. The program iterations ended after the fourth pass. Appendix I contains the resulting weights for each of the 275 combinations of responses found in the data, as well as the total weight for all the respondents with that given combination. For example, the 11 respondents who were in income group 5, age group 4, and region 5 each are weighted by the factor 0.76018.

Here is the buildup of that weight from the four rounds (with three passes per round) of the WinCross sample balancing. The "weight" is the one derived above for the appropriate variable level and the "cum weight" is the sum of the "delta" and the previous "cum weight" (except for the first step, where the previous "cum weight" is implicitly 1, so that the first "delta" is implicitly -0.13850.)

	delta	cum weight
income	-0.13850	0.86150
age	-0.22218	0.63932
region	0.12802	0.76734
income	-0.01178	0.75556
age	-0.00391	0.75165
region	0.00694	0.75859
income	0.00123	0.75982
age	-0.00024	0.75958
region	0.00044	0.76002
income	0.00016	0.76018
age	-0.00003	0.76015
region	0.00002	0.76017

One situation that might occur in using the WinCross sample balancing program is that it will produce "negative" weights. The following table presents the buildup of the weight for the respondents with in (income group 4, age group 5, region 4), (income group 4, age group 4, region 4), and (income group 5, age group 6, region 4).

	454		444		564	
	delta	cum weight	delta	cum weight	delta	cum weight
income	-0.35113	0.64887	-0.35113	0.64887	-0.13850	0.86150
age	-0.23046	0.41841	-0.22218	0.42669	-0.29111	0.57039
region	-0.55858	-0.14018	-0.55858	-0.13189	-0.55858	0.01181
income	0.06625	-0.07392	0.06625	-0.06564	-0.01178	0.00003
age	-0.02374	-0.09767	-0.00391	-0.06954	-0.01621	-0.01618
region	0.00000	-0.09767	0.00000	-0.06955	0.00000	-0.01619
income	0.00393	-0.09374	0.00393	-0.06562	0.00122	-0.01496
age	-0.00084	-0.09458	-0.00024	-0.06586	-0.00055	-0.01551
region	0.00041	-0.09417	0.00041	-0.06545	0.00041	-0.01510
income	0.00020	-0.09397	0.00020	-0.06525	0.00016	-0.01494
age	-0.00007	-0.09404	-0.00003	-0.06528	-0.00001	-0.01495
region	0.00003	-0.09401	0.00003	-0.06525	0.00003	-0.01492

Note that, in particular due to the first round adjustment for region 4, the cumulative weights become negative. This is due to the extreme alteration that is needed for region 4, where there were 129 respondents and the target is 71.4. The WinCross sample balancing algorithm makes a major reduction in the weight in the first round, so much so that it may produce negative weights. One way to avoid this is to replace those weights with a small number, such as 0.001, as is done in Appendix I.

How does the WinCross sample balancing program compare with iterative proportional fitting? Following are the estimated target rim percents produced by both programs (we used the Quantum implementation of iterative proportional fitting as our comparison). The rightmost two columns contain the elements of the criterion measure used in the WinCross sample balancing program, i.e., the values of the squares of the

$$d_{j_1 j_2 \dots j_v} = [g_{j_i}^i(t) - g_{j_i}^i] / f_{j_i}^i$$

for each level of each variable.

	data	target	WinCross	IPF	WinCross	IPF
income						
1	124	179.5	179.4892	179.5003	0.000000007586	0.000000000006
2	150	232.0	231.9841	231.9992	0.000000011236	0.000000000028
3	305	272.8	272.7745	272.7999	0.000000006990	0.000000000000
4	221	143.4	143.3749	143.3990	0.000000012899	0.000000000020
5	200	172.3	172.3763	172.2998	0.000000145542	0.000000000001
Total	1000	1000.0	999.9990	999.9982	0.000000000001	0.000000000003
age						
1	6	38.7	38.6998	38.6668	0.000000001111	0.000030617778
2	34	90.7	90.6999	90.7535	0.000000000009	0.000002475995
3	185	190.9	190.8987	190.9080	0.000000000049	0.000000001870
4	276	216.0	216.0004	215.9743	0.000000000002	0.000000008671
5	248	180.2	180.2011	180.1599	0.000000000020	0.000000026145
6	92	64.4	64.3999	64.3959	0.000000000001	0.000000001986
7	64	51.7	51.6993	51.6988	0.000000000120	0.000000000352
8	65	88.0	88.0001	88.0201	0.000000000002	0.000000095624
9	26	59.1	59.0998	59.1316	0.000000000059	0.000001477160
10	4	20.3	20.2999	20.2893	0.000000000625	0.000007155625
Total	1000	1000.0	999.9989	999.9982	0.000000000001	0.000000000003
region						
1	55	51.4	51.3999	51.3596	0.000000000003	0.000000539557
2	79	142.1	142.0999	142.2019	0.000000000002	0.000001663773
3	187	164.4	164.3999	164.4212	0.000000000000	0.000000012853
4	129	71.4	71.3999	71.4617	0.000000000001	0.000000228766
5	169	190.5	190.4999	190.5803	0.000000000000	0.000000225766
6	56	63.0	62.9999	62.9725	0.000000000003	0.000000241151
7	103	109.1	109.0999	109.0077	0.000000000001	0.000000803025
8	82	64.0	63.9999	63.9870	0.000000000001	0.000000025134
9	140	144.1	144.0999	144.0063	0.000000000001	0.000000447943
Total	1000	1000.0	999.9991	999.9982		

The criterion measure used by WinCross is the square root of the sum of the values in the right hand columns of the above tables. For WinCross this value is 0.000431586; for iterative proportional fitting this value is 0.006785959, over 15 times as large.

APPENDIX I

combination	income	age	region	count	weight	wtd count
1	1	1	1	5	1 7.03868	7.03868
2	1	2	2	3	5 2.64602	13.23010
3	1	2	2	5	1 2.89112	2.89112
4	1	2	2	6	1 2.83734	2.83734
5	1	2	2	7	1 2.80156	2.80156
6	1	3	3	1	1 1.44292	1.44292
7	1	3	3	3	3 1.29764	3.89292
8	1	3	3	4	4 0.84916	3.39664
9	1	3	3	5	1 1.54273	1.54273
10	1	3	3	6	2 1.48895	2.97790
11	1	3	3	7	3 1.45317	4.35951
12	1	3	3	8	2 1.27067	2.54134
13	1	3	3	9	1 1.46797	1.46797
14	1	4	4	1	2 1.13520	2.27040
15	1	4	4	2	2 1.87272	3.74544
16	1	4	4	3	3 0.98992	2.96976
17	1	4	4	4	7 0.54144	3.79008
18	1	4	4	5	11 1.23501	13.58511
19	1	4	4	6	2 1.18123	2.36246
20	1	4	4	7	4 1.14545	4.58180
21	1	4	4	8	4 0.96296	3.85184
22	1	4	4	9	5 1.16026	5.80130
23	1	5	5	1	2 1.10644	2.21288
24	1	5	5	2	3 1.84395	5.53185
25	1	5	5	3	4 0.96116	3.84464
26	1	5	5	4	2 0.51268	1.02536
27	1	5	5	5	5 1.20625	6.03125
28	1	5	5	6	1 1.15247	1.15247
29	1	5	5	8	2 0.93420	1.86840
30	1	5	5	9	1 1.13149	1.13149
31	1	6	6	1	1 1.05367	1.05367
32	1	6	6	2	1 1.79119	1.79119
33	1	6	6	3	1 0.90839	0.90839
34	1	6	6	4	2 0.45992	0.91984
35	1	6	6	5	2 1.15349	2.30698
36	1	6	6	6	1 1.09971	1.09971
37	1	6	6	8	1 0.88143	0.88143
38	1	6	6	9	1 1.07873	1.07873
39	1	7	7	1	1 1.20926	1.20926
40	1	7	7	2	1 1.94677	1.94677
41	1	7	7	5	2 1.30907	2.61814

42	1	7	6	1	1.25529	1.25529
43	1	7	7	1	1.21951	1.21951
44	1	7	9	1	1.23431	1.23431
45	1	8	2	4	2.38434	9.53736
46	1	8	3	2	1.50154	3.00308
47	1	8	4	3	1.05307	3.15921
48	1	8	5	1	1.74664	1.74664
49	1	8	9	1	1.67188	1.67188
50	1	9	3	3	2.45789	7.37367
51	1	9	4	2	2.00941	4.01882
52	1	9	5	2	2.70298	5.40596
53	1	9	9	3	2.62822	7.88466
54	2	2	3	1	2.84731	2.84731
55	2	2	4	2	2.39883	4.79766
56	2	2	5	3	3.09240	9.27720
57	2	2	7	4	3.00284	12.01136
58	2	2	8	1	2.82035	2.82035
59	2	3	3	4	1.49892	5.99568
60	2	3	4	4	1.05045	4.20180
61	2	3	5	4	1.74402	6.97608
62	2	3	6	4	1.69024	6.76096
63	2	3	7	3	1.65446	4.96338
64	2	3	8	4	1.47196	5.88784
65	2	3	9	5	1.66926	8.34630
66	2	4	1	3	1.33649	4.00947
67	2	4	2	6	2.07401	12.44406
68	2	4	3	8	1.19121	9.52968
69	2	4	4	6	0.74273	4.45638
70	2	4	5	6	1.43630	8.61780
71	2	4	6	2	1.38252	2.76504
72	2	4	7	4	1.34674	5.38696
73	2	4	8	2	1.16425	2.32850
74	2	4	9	3	1.36154	4.08462
75	2	5	1	1	1.30773	1.30773
76	2	5	2	1	2.04524	2.04524
77	2	5	3	6	1.16245	6.97470
78	2	5	4	5	0.71397	3.56985
79	2	5	5	6	1.40754	8.44524
80	2	5	6	2	1.35376	2.70752
81	2	5	7	3	1.31798	3.95394
82	2	5	8	2	1.13548	2.27096
83	2	5	9	6	1.33278	7.99668
84	2	6	1	1	1.25496	1.25496
85	2	6	3	3	1.10968	3.32904
86	2	6	4	1	0.66120	0.66120
87	2	6	5	2	1.35477	2.70954

88	2	6	6	2	1.30099	2.60198
89	2	6	7	2	1.26521	2.53042
90	2	6	8	2	1.08272	2.16544
91	2	6	9	2	1.28002	2.56004
92	2	7	8	1	1.23830	1.23830
93	2	7	9	5	1.43560	7.17800
94	2	8	2	1	2.58563	2.58563
95	2	8	3	4	1.70283	6.81132
96	2	8	4	1	1.25435	1.25435
97	2	8	5	1	1.94792	1.94792
98	2	8	6	2	1.89414	3.78828
99	2	8	7	2	1.85836	3.71672
100	2	8	8	1	1.67587	1.67587
101	2	8	9	2	1.87317	3.74634
102	2	9	3	1	2.65918	2.65918
103	2	9	4	2	2.21070	4.42140
104	2	10	3	1	5.38320	5.38320
105	3	1	4	1	5.92843	5.92843
106	3	2	1	1	2.37462	2.37462
107	3	2	2	3	3.11213	9.33639
108	3	2	5	2	2.47443	4.94886
109	3	2	6	1	2.42065	2.42065
110	3	2	8	1	2.20238	2.20238
111	3	2	9	1	2.39967	2.39967
112	3	3	1	3	1.02623	3.07869
113	3	3	2	5	1.76375	8.81875
114	3	3	3	16	0.88095	14.09520
115	3	3	4	7	0.43247	3.02729
116	3	3	5	9	1.12604	10.13436
117	3	3	6	5	1.07226	5.36130
118	3	3	7	7	1.03648	7.25536
119	3	3	8	6	0.85399	5.12394
120	3	3	9	11	1.05128	11.56408
121	3	4	1	3	0.71852	2.15556
122	3	4	2	5	1.45603	7.28015
123	3	4	3	15	0.57324	8.59860
124	3	4	4	13	0.12476	1.62188
125	3	4	5	14	0.81833	11.45662
126	3	4	6	3	0.76455	2.29365
127	3	4	7	4	0.72877	2.91508
128	3	4	8	8	0.54628	4.37024
129	3	4	9	5	0.74357	3.71785
130	3	5	1	2	0.68975	1.37950
131	3	5	2	12	1.42727	17.12724
132	3	5	3	16	0.54447	8.71152
133	3	5	4	12	0.09600	1.15200

134	3	5	5	14	0.78956	11.05384
135	3	5	6	3	0.73578	2.20734
136	3	5	7	9	0.70001	6.30009
137	3	5	8	3	0.51751	1.55253
138	3	5	9	8	0.71481	5.71848
139	3	6	1	1	0.63699	0.63699
140	3	6	2	2	1.37450	2.74900
141	3	6	3	5	0.49171	2.45855
142	3	6	4	4	0.04323	0.17292
143	3	6	5	7	0.73680	5.15760
144	3	6	6	2	0.68302	1.36604
145	3	6	7	2	0.64724	1.29448
146	3	6	8	2	0.46475	0.92950
147	3	6	9	2	0.66204	1.32408
148	3	7	1	2	0.79257	1.58514
149	3	7	2	3	1.53009	4.59027
150	3	7	3	9	0.64729	5.82561
151	3	7	4	5	0.19881	0.99405
152	3	7	5	2	0.89238	1.78476
153	3	7	6	2	0.83860	1.67720
154	3	7	8	4	0.62033	2.48132
155	3	7	9	2	0.81763	1.63526
156	3	8	1	1	1.23014	1.23014
157	3	8	2	1	1.96765	1.96765
158	3	8	3	2	1.08486	2.16972
159	3	8	4	3	0.63638	1.90914
160	3	8	5	1	1.32995	1.32995
161	3	8	8	2	1.05790	2.11580
162	3	8	9	2	1.25519	2.51038
163	3	9	2	1	2.92400	2.92400
164	3	9	3	1	2.04120	2.04120
165	3	9	5	1	2.28630	2.28630
166	3	9	8	1	2.01424	2.01424
167	3	9	9	3	2.21154	6.63462
168	3	10	4	1	4.31675	4.31675
169	3	10	5	1	5.01032	5.01032
170	4	1	2	1	7.06970	7.06970
171	4	2	3	1	2.03933	2.03933
172	4	2	5	1	2.28442	2.28442
173	4	2	7	1	2.19486	2.19486
174	4	3	1	5	0.83622	4.18110
175	4	3	2	2	1.57374	3.14748
176	4	3	3	4	0.69094	2.76376
177	4	3	4	7	0.24247	1.69729
178	4	3	5	7	0.93604	6.55228
179	4	3	6	1	0.88226	0.88226

180	4	3	7	3	0.84648	2.53944
181	4	3	8	2	0.66398	1.32796
182	4	3	9	10	0.86128	8.61280
183	4	4	1	3	0.52851	1.58553
184	4	4	2	8	1.26603	10.12824
185	4	4	3	14	0.38323	5.36522
186	4	4	4	5	0.00100	0.00500
187	4	4	5	8	0.62832	5.02656
188	4	4	6	2	0.57454	1.14908
189	4	4	7	7	0.53876	3.77132
190	4	4	8	8	0.35627	2.85016
191	4	4	9	9	0.55357	4.98213
192	4	5	1	3	0.49975	1.49925
193	4	5	2	4	1.23726	4.94904
194	4	5	3	17	0.35447	6.02599
195	4	5	4	7	0.00100	0.00700
196	4	5	5	11	0.59956	6.59516
197	4	5	6	2	0.54578	1.09156
198	4	5	7	6	0.51000	3.06000
199	4	5	8	3	0.32751	0.98253
200	4	5	9	6	0.52480	3.14880
201	4	6	1	2	0.44698	0.89396
202	4	6	3	7	0.30170	2.11190
203	4	6	5	3	0.54679	1.64037
204	4	6	7	4	0.45723	1.82892
205	4	6	8	4	0.27474	1.09896
206	4	6	9	4	0.47204	1.88816
207	4	7	3	2	0.45729	0.91458
208	4	7	4	2	0.00881	0.01762
209	4	7	5	1	0.70238	0.70238
210	4	7	9	1	0.62762	0.62762
211	4	8	1	1	1.04013	1.04013
212	4	8	2	1	1.77765	1.77765
213	4	8	3	2	0.89485	1.78970
214	4	8	4	2	0.44637	0.89274
215	4	8	5	8	1.13994	9.11952
216	4	8	6	2	1.08616	2.17232
217	4	8	7	1	1.05038	1.05038
218	4	8	8	2	0.86789	1.73578
219	4	8	9	2	1.06519	2.13038
220	4	9	4	1	1.40272	1.40272
221	4	9	6	1	2.04251	2.04251
222	5	1	3	1	6.31875	6.31875
223	5	1	4	1	5.87028	5.87028
224	5	1	7	1	6.47429	6.47429
225	5	2	1	1	2.31647	2.31647

226	5	2	7	1	2.32672	2.32672
227	5	2	9	1	2.34152	2.34152
228	5	3	1	2	0.96808	1.93616
229	5	3	3	5	0.82280	4.11400
230	5	3	4	4	0.37432	1.49728
231	5	3	5	3	1.06789	3.20367
232	5	3	6	1	1.01411	1.01411
233	5	3	7	6	0.97833	5.86998
234	5	3	8	3	0.79584	2.38752
235	5	3	9	6	0.99313	5.95878
236	5	4	1	3	0.66037	1.98111
237	5	4	2	4	1.39788	5.59152
238	5	4	3	5	0.51509	2.57545
239	5	4	4	6	0.06661	0.39966
240	5	4	5	11	0.76018	8.36198
241	5	4	6	4	0.70640	2.82560
242	5	4	7	10	0.67062	6.70620
243	5	4	8	5	0.48812	2.44060
244	5	4	9	14	0.68542	9.59588
245	5	5	1	4	0.63160	2.52640
246	5	5	2	4	1.36912	5.47648
247	5	5	3	15	0.48632	7.29480
248	5	5	4	4	0.03784	0.15136
249	5	5	5	11	0.73141	8.04551
250	5	5	6	5	0.67763	3.38815
251	5	5	7	7	0.64185	4.49295
252	5	5	8	2	0.45936	0.91872
253	5	5	9	6	0.65666	3.93996
254	5	6	2	3	1.31635	3.94905
255	5	6	3	1	0.43356	0.43356
256	5	6	4	1	0.00100	0.00100
257	5	6	5	5	0.67865	3.39325
258	5	6	6	1	0.62487	0.62487
259	5	6	7	1	0.58909	0.58909
260	5	6	8	2	0.40660	0.81320
261	5	6	9	2	0.60389	1.20778
262	5	7	1	4	0.73442	2.93768
263	5	7	4	1	0.14066	0.14066
264	5	7	7	5	0.74467	3.72335
265	5	7	8	2	0.56218	1.12436
266	5	7	9	4	0.75948	3.03792
267	5	8	1	1	1.17199	1.17199
268	5	8	5	1	1.27180	1.27180
269	5	8	7	1	1.18224	1.18224
270	5	8	9	4	1.19704	4.78816
271	5	9	1	1	2.12833	2.12833

272	5	9	4	1	1.53458	1.53458
273	5	9	6	1	2.17437	2.17437
274	5	9	9	1	2.15339	2.15339
275	5	10	2	1	5.58987	5.58987
				1000		1001.01