Sample Management - Sample Weighting (Markets & Timezones)

Understanding Market & Timezone Weights in Survox

Market weights control how Survox distributes dialing across the markets built for a project. Timezone weights follow the same logic and are covered briefly at the end of this article. Markets themselves are "groupings" of similar sample records, for example gender combined with age ranges, or counties crossed by political party. They are used as a means to target specific sample types for dialing, or focusing on, by adjusting their weighted value.

The Basics

Each market in a project can be assigned a weight from 0 to 9.

A weight of **0** means the market will not be dialed at all. Survox will skip it entirely until the weight is changed.

Weights **1 through 9** are not true ratios. Think of them more as multipliers that influence what percentage of numbers Survox pulls from each market every time the dialer requests numbers, or when an agent requests a number in 1:1 mode.

How Weights Work When Sample Sizes Are Equal

When all markets have roughly the same number of records, the math is straightforward.

Five markets all set to weight 1 with similar record counts will each receive about 20% of every pull. Over 100 calls, you can expect roughly 20 numbers drawn from each market.

If you raise one of those markets to weight 2, Survox treats that market as if it has twice as many records in the pool. The result is that market receives approximately 40% of every pull, while the remaining four markets each drop to roughly 15%.

The table below shows a two-market equal-sample scenario where Market A stays at weight 1 and Market B is increased step by step:

Market A Weight	Market B Weight	Market A calls per 100	Market B calls per 100
1	**1**	50	50
1	**2**	33	67
1	**3**	25	75
1	**4**	20	80
1	**5**	17	83
1	**6**	14	86
1	**7**	13	87
1	**8**	11	89
1	**9**	10	90

How Weights Work When Sample Sizes Are Not Equal

This is where it gets more nuanced. When markets have significantly different record counts, the starting percentages are already unequal before weights even come into play.

For example, three markets with no special weighting and record counts of 1,000 / 500 / 100 will naturally pull at roughly 63 / 31 / 6 out of every 100 calls respectively. That is simply because the larger markets represent a larger share of the total pool.

When you apply a weight to one of those markets, you are multiplying its effective record count in the calculation. Increasing the smallest market from weight 1 to weight 2 doubles its effective count from 100 to 200, which shifts the percentages across all three markets. The larger markets do not stay fixed -- they all recalculate together.

This means that in unequal-sample projects, predicting exact pull percentages requires knowing both the record counts and the weights across all markets simultaneously.

The table below shows this same three-market scenario with A and B staying at weight 1 while Market C is increased:

Market A Weight	Market B Weight	Market C Weight	Market A calls per 100	Market B calls per 100	Market C calls per 100
1	1	**1**	63	31	6
1	1	**2**	59	29	12
1	1	**3**	56	28	17
1	1	**4**	53	26	21
1	1	**5**	50	25	25
1	1	**6**	48	24	29
1	1	**7**	45	23	32
1	1	**8**	44	22	35
1	1	**9**	42	21	38

Notice that even at weight 9, Market C is still trailing both other markets because the 10:1 record count gap is too large for any weight to fully overcome. Weights adjust the ratio, but they cannot manufacture sample that does not exist.

Practical Ceiling: Weights Above 4

Because of the percentage-based math, raising a weight above 4 rarely produces a meaningful change in how aggressively that market is dialed relative to the others. The gains compress as the multiplier grows. In most real-world projects, weights between 1 and 4 are where the useful adjustments happen.

Timezone Weights

Timezone weights follow the exact same logic described above. The weight value acts as a multiplier on the effective record count for each timezone segment, and all the same rules around proportionality and diminishing returns above 4 apply.

There is one important distinction with timezone weights -- they are evaluated first, before market weights. When the dialer requests numbers, Survox checks timezone eligibility and weighting before it ever looks at market weighting.

This creates a scenario worth being aware of: if you weight the East Coast timezone heavier than Central, but then also weight a market that is primarily Central numbers higher, those two settings can partially cancel each other out. The timezone weight pulls the system toward East Coast numbers, while the market weight is trying to push it toward Central -- and the result may not be what was intended.

In practice this does not come up often, because geographic regions that are separated by timezone are usually not also separated into different markets on the same project. But it is always worth a quick check. If you are unsure how your timezone and market weights may be interacting, review the CfMC Info Sheet for the project or speak with the Project Director or Programmer before making changes.

Real World Example

The following example uses an actual project sample load to illustrate how weights behave in practice. The numbers shown represent expected calls out of every 10,000 dialed, which roughly mirrors what you might see over a single busy shift hour.

With all markets at their default weight of 1, Survox distributes calls purely based on the proportion of available records in each market. Females Over 65 dominates simply because it has the most records loaded, while Males Under 30 receives the fewest calls for the same reason.

Scenario 1 -- All weights at 1 (189,711 total records)

Code	Market	Weight	Records	Calls per 10,000
11	Males - Under 30	1	10,603	559
12	Males - Under 50	1	19,279	1,016
13	Males - Over 50	1	17,261	910
14	Males - Over 65	1	27,470	1,448
21	Females - Under 30	1	16,994	896
22	Females - Under 50	1	29,043	1,531
23	Females - Over 50	1	26,389	1,391
24	Females - Over 65	1	42,672	2,249

Scenario 2 -- Adjusted weights (295,512 effective records)

Now suppose the project needs to push harder on overall younger and more specifically male respondents. Males Under 30 and Males Under 50 are raised to weight 3, and Females Under 30 and Females Under 50 are raised to weight 2. Everything else stays at 1.

Survox now multiplies those markets' record counts by their respective weights to calculate a new effective pool of 295,512 records. This is NOT the number of records you have, just the "effective" values after applying the weights, to calculate the calls per market... The call distribution shifts significantly, but notice that the markets left at weight 1 all drop -- even though nothing was done to them. That is the nature of percentage-based weighting. Boosting some markets always comes at the cost of the others.

Code	Market	Records	Weight	Effective Records	Calls per 10,000
11	Males - Under 30	10,603	3	31,809	1,076
12	Males - Under 50	19,279	3	57,837	1,957
13	Males - Over 50	17,261	1	17,261	584
14	Males - Over 65	27,470	1	27,470	930
21	Females - Under 30	16,994	2	33,988	1,150
22	Females - Under 50	29,043	2	58,086	1,966
23	Females - Over 50	26,389	1	26,389	893
24	Females - Over 65	42,672	1	42,672	1,444

Males Under 30 nearly doubled from 559 to 1,076 calls per 10,000, and Males Under 50 went from 1,016 to 1,957. The tradeoff is visible across every market left at weight 1 -- Males Over 50 dropped from 910 to 584, and Females Over 65 dropped from 2,249 to 1,444. No records were added or removed. The weights simply changed how the existing pool was divided.

A Note on Sample Availability

All of the calculations above are based solely on numbers that are available at the exact moment the dialer requests them. The following are excluded from the math entirely: hidden sample, scheduled callbacks, numbers that have not yet aged to the minimum required call interval, and any numbers flagged as a special type. If a number cannot be dialed right now, it does not count toward the pool, and therefore does not influence the weighting ratios.

Revision #7
Created 2026-02-26 20:30:58 UTC by Admin
Updated 2026-02-26 21:27:45 UTC by Admin