Sample size determination

Yegor Tkachenko – June 2023

Marketing researchers often need to figure out the right *sample size*, that is, how many observations it is necessary to collect for a specific task, such as determining some quantity of interest. Here are examples of possible sample size questions:

- What is the right number of individuals to poll to learn an accurate proportion of a politician's supporters in the population?
- How many ads of type A and B do we need to show online to accurately assess the differences in their quality and select the best one?

Answering such questions accurately is critical. Get the sample size too low, and the collected data will be insufficient to answer to the posed question. Get the sample size too large, and you will waste money on collection of unnecessary observations.

In this chapter, I review several approaches to figuring out the right sample size that are broadly applicable across a variety of practical marketing situations. Python code implementation is provided.

Review of key statistical concepts

Before we begin, let us recall several key statistical concepts here. For random variables x_i:

1. *Population mean* is $\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$, where N is a size of the population. For example, μ could indicate the average height of all people in a particular country of interest.
2. *Sample mean* is $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$, where $n \leq N$ is a size of the sample from the population. For example, $\bar{x}$ could indicate the average height of random set of residents in a country of interest.
3. *Population standard deviation* is $\sigma = \sqrt{var(x_i)} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$, where μ is the true (unknown) population mean. It captures how much random variables x_i vary or scatter around the population mean.

4. *Sample standard deviation* is $s = \sqrt{\frac{1}{(n-1)} \sum_{i=1}^{n} (x_i - \bar{x})^2}$, as an approximation of the population standard deviation σ, using sample mean $\bar{x}$ in place of μ. Division by $n-1$ instead of n allows us to correct for an underestimation of variance due to re-use of the sample observations to both estimate the sample mean $\bar{x}$ and then to compute the deviation from this sample mean – it is also known as Bessel's correction.[1]

5. *Standard deviation of the mean* or *standard error* is $SE = \sqrt{var(\bar{x})} = \sigma/\sqrt{n} \approx s/\sqrt{n}$. *Note:* While standard deviation captures how individual random variables vary around the true mean μ, standard error captures how the sample mean of n random variables varies around the true mean μ. In other words, standard error is just a standard deviation applied to sample means, treated as random variables, instead of the individual x_i. *Derivation:*

$$SE = \sqrt{var(\bar{x})} = \sqrt{var\left(\frac{1}{n}\sum_{i=1}^{n} x_i\right)} = \sqrt{\frac{1}{n^2} var\left(\sum_{i=1}^{n} x_i\right)} = \sqrt{\frac{n}{n^2} var(x_i)} = \sqrt{\frac{\sigma^2}{n}} = \frac{\sigma}{\sqrt{n}},$$

based on the properties of the variance of a sum of independent random variables.

6. *Central Limit theorem (CLT)* is a foundational theorem in probability theory – it roughly says that for a large sample size ($n \geq 30$), a sample mean of n random variables $\bar{x}_n = \frac{1}{n} \sum_{i=1}^{n} x_i$ is distributed approximately Normal with mean μ and standard deviation $\sigma/\sqrt{n}$, where μ is population mean, σ is population standard deviation, and n is the size of the sample. In more compact notation, $\bar{x}_n \rightarrow_d N(\mu, \sigma^2/n)$.

Sampling error and the Central Limit theorem

Consider an upcoming presidential election. Two candidates – Alice (A) and Bob (B) – are running, and we need to estimate the proportion of eligible voters for each candidate in a particular region. So we pose the following question to a random sample of eligible voters in the region:

> Q.: Select the candidate are you planning to vote for: (a) Alice; (b) Bob.

As a toy example, let us first consider estimating the proportion of votes in a small company of 100 people. The 100 people whose opinion we are interested in are called the *population*. We will represent the population as a 10 by 10 grid, where each individual is represented by a cell with their vote, coded 1 for Alice and 0 for Bob. Here, 53 individuals are pro-Alice and 47 are pro-Bob (that is, 53%=53/100 of votes for Alice). Of course, we do not know this before conducting research.

[1] `https://math.oxford.emory.edu/site/math117/besselCorrection/`

1	0	1	1	1	0	0	1	0	1
1	0	0	1	1	1	0	1	1	1
0	1	1	0	1	1	0	1	0	0
1	1	0	1	0	0	1	1	0	1
1	1	1	1	0	1	1	0	1	0
1	0	0	0	1	1	0	1	0	0
0	1	0	0	1	0	0	0	0	1
0	0	1	1	0	0	1	1	1	1
1	0	1	1	0	0	0	0	0	0
1	0	1	0	1	0	1	0	1	1

(1)

Assume we can only survey 5 people out of 100. The 5 people we survey constitute a *sample* we draw from the population. How should we select the sample from the population? The most straightforward and common approach is to select the sample randomly. The highlighting below indicates one possible random sample of size $n = 5$.

1	0	**1**	1	1	0	0	1	0	1
1	0	0	1	1	1	0	1	1	**1**
0	1	1	0	1	1	0	1	0	0
1	1	0	1	0	0	1	1	0	1
1	1	1	1	0	1	1	0	1	0
1	**0**	0	0	1	1	0	1	0	0
0	1	0	0	1	0	0	**0**	0	1
0	0	1	1	0	0	1	1	1	1
1	0	1	1	0	0	0	0	0	0
1	0	1	**0**	1	0	1	0	1	1

(2)

What is the problem if we do not do random sampling? For example, if we survey only women, or only people that leave work after 21:00, their opinions may be special, not representative of the full population, so we would get the wrong idea about what everyone thinks by asking just this subset of people. Random sampling breaks this dependency between the criterion for selection of the sample and that thing we are measuring based on the sample. Truly random selection of the sample ensures that we can expect the sample to be representative, that is, to reflect the opinions of the overall population.[2]

[2]In practice, even if the list of potential respondents is generated truly randomly, whether specific respondents agree to be surveyed and whether they complete the questionnaire, depends on their motivation, which, in turn, depends on their compensation for participation in the survey and on their personal characteristics. Correlations between survey non-completes and personal characteristics can create biased results even if the respondents are randomly selected. Proper compensation may mitigate such issues to a degree. Depending on the specific situation, it may also be possible to account for such bias during the modeling stage.

Now notice that if we compute the proportion of votes for Alice in the $n = 5$ sample above, we get $2/5 = 40\%$, which is distinct from the population vote proportion of 53%. We have encountered the *sampling error*. As we draw samples from the population, the measurements we get will differ from the population-level truth.

In practice, marketers deal with much larger populations than 100 people. Let us scale up the example above, so that we sample instead from an abstract large country-level population. In other words, we will assume the population is too large for us to survey even 1% of it, to say nothing of surveying all of it like in a census.[3]

Let the true proportion of Alice supporters in the population be $\mu = 53\%$, as before. (In practice, we do not know this number – but want to estimate it.) Let us simulate a set of 50 binary Alice vs. Bob responses randomly drawn from the population and examine the proportion of Alice votes in the resulting sample. We will simulate the votes using draws from Bernoulli distribution, which will output 1 with probability 0.53, and 0 otherwise.

```
import numpy as np       # provides random number and array support
np.random.seed(1234)     # setting seed for replicable random numbers

# vote distribution - Alice is designated by 1, Bob - by 0;
# 50 votes sampled; (n=1 means we perform 1 coin roll per person,
# so we are sampling from Bernoulli distribution)

votes = np.random.binomial(n=1, p=0.53, size=50)
x_avg = np.mean(votes)  # sample proportion of votes for Alice
print(x_avg)
```

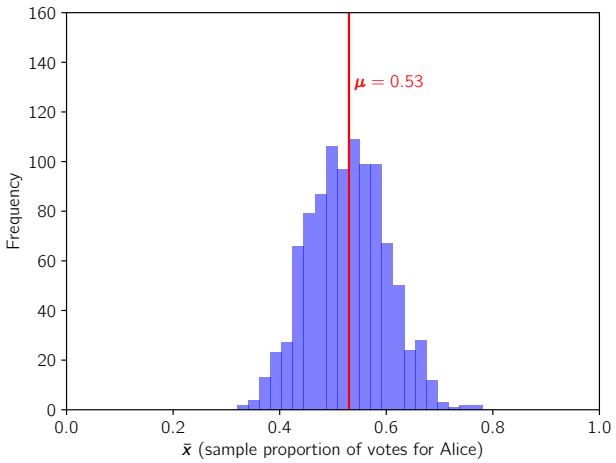
```
0.48
```

We find sample mean $\bar{x} = 0.48$, which is the proportion of Alice votes in the sample of $n = 50$ individuals. We see here too that the sample proportion 0.48 is not identical to the population proportion of $\mu = 0.53$ – a demonstration of the sampling error. As we collect new surveys with the same sample size, we will be observing ever changing proportions. The hope, however, is that they will not be far from the population proportion.

[3]There are special considerations when one is sampling from a small population (also called finite), so a sample may constitute a large proportion of the population. This is because even though a sample may be small, so it should yield a large sample error, if the population itself is small too, it may be almost perfectly approximated by that small sample (for example, when the population is 5 people, the relatively small sample, in absolute terms, of those 5 people will yield zero sampling error). For this reason, there is a special technique to analyze sample size in the finite population settings, called the finite population correction. However, it is rarely applicable in the majority of cases relevant to marketers due to large population, so we do not cover it here. There are two specific reasons the correction can typically be safely ignored. First, when the population is not extremely small, the effect of the finite population correction is negligible. Second, the finite population correction technique can only *reduce* the sample size requirements compared to the classical infinite population analysis. Thus, if one applies the classical analysis, one is erring on the conservative side, selecting larger sample size, which can be considered prudent in practical situations.

Let us examine the distribution of vote proportions we get as we collect 1,000 surveys, each survey based on 50 random respondents. We visualize the histogram of sample Alice vote proportions below.

```
np.random.seed(1234)
votes = np.random.binomial(n=1, p=0.53, size=(50,1000))
x_avg = np.mean(votes, axis=0)

import matplotlib.pyplot as plt
plt.rcParams['text.usetex'] = True
plt.rcParams.update({'font.size': 12})
plt.rc('text.latex', preamble=r'\usepackage{amsmath,cmbright}')

n, bins, patches = plt.hist(x_avg, bins=22, facecolor='blue', alpha=0.5)
plt.xlim(0,1)
plt.ylim(0,160)
plt.xlabel(r'$\boldsymbol{\bar x}$ (sample proportion of votes for Alice)')
plt.ylabel('Frequency')
plt.axvline(x=0.53, color='red')
plt.text(0.54, 130, r"$\boldsymbol{\mu}=0.53$", rotation=0, color='red')
plt.savefig('./figs/hist.pdf', dpi=300, bbox_inches='tight', pad_inches=0)
```



The visualization carries grim news to those who would wish to rely on a sample of just 50 respondents when calling an election – numbers are all over the place, so the prediction would likely be off.

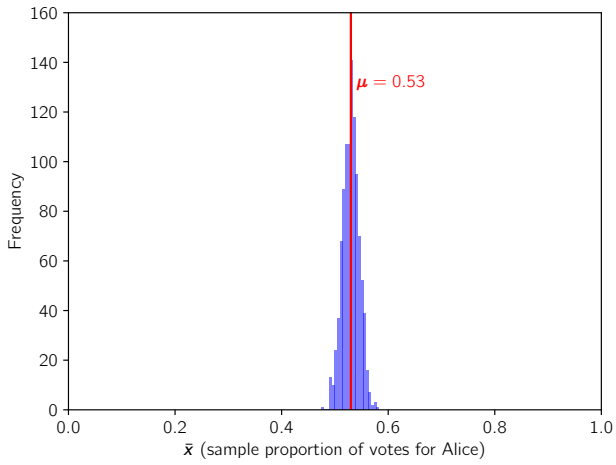
```
np.round(np.sum(x_avg<0.5)/x_avg.shape[0],3)
```

```
0.301
```

In fact, 30% of simulated polls have sample means less than 0.5 majority cutoff, wrongly awarding the victory to Bob rather than Alice. The average of the survey means (proportions) is close to the true population proportion. However, the standard deviation of the survey means (also called a standard error of the mean or SE) is quite high!

Let us now repeat the computation, but this time each survey will be collected from a much larger sample of 1,000 individuals.

```
np.random.seed(1234)
votes = np.random.binomial(n=1, p=0.53, size=(1000,1000))
x_avg = np.mean(votes, axis=0)

n, bins, patches = plt.hist(x_avg, bins=22, facecolor='blue', alpha=0.5)
plt.xlim(0,1)
plt.ylim(0,160)
plt.xlabel(r'$\boldsymbol{\bar x}$ (sample proportion of votes for Alice)')
plt.ylabel('Frequency')
plt.axvline(x=0.53, color='red')
plt.text(0.54, 130, r"$\boldsymbol{\mu}=0.53$", rotation=0, color='red')
plt.savefig('./figs/hist_large_sample.pdf', dpi=300, bbox_inches='tight',
                                        pad_inches=0)
```



We can see now that survey responses are very concentrated around the right value. Notice how much the standard deviation of the mean – the standard error – has dropped, compared to the case where the sample was only 50 people.

```
np.round(np.sum(x_avg>0.5)/x_avg.shape[0],3)
```

```
0.97
```

Now, 97% of surveys would correctly give victory to the winning candidate (i.e., judge their vote share to be above 50%). Larger sample size seems to have helped – good news for survey companies that get paid by the number of responses.

What these results show is that as the sample size grows, the means of the samples $\bar{x}$ will be closer to the population mean μ. The spread of sample means around the population mean grows narrower. Luckily for us, statisticians have figured out a precise way to quantify how sample size relates to the spread of the sample means around the population means. The key result in this area is known as the *Central Limit theorem (CLT)*.

CLT is a critical foundational theorem in probability theory – it roughly says that for a large sample size ($n \geq 30$), an average of n random variables $\bar{x}_n = \frac{1}{n}\sum_{i=1}^{n} x_i$ is distributed approximately Normal with mean μ and standard deviation $\sigma/\sqrt{n}$, where μ is population mean, σ is population standard deviation across responses, and n is the size of the sample. In more compact notation, $\bar{x}_n \rightarrow_d N(\mu, \sigma^2/n)$.

What the result says is that the standard error of the sample mean (standard deviation across means) around the population mean decreases as a square root of the sample size. We can take advantage of this relationship to decide what n is required for the purpose of a given research project, as we will see below.

Sample size based on a confidence interval

One approach to set the sample size is to specify the size of the acceptable confidence interval within which we can expect the sample mean $\bar{x}_n$ to fall around the population mean μ. For the Normal distribution, we can write down what the size of the confidence interval should be so that a normally distributed random variable falls outside this confidence interval with probability α. By the Central Limit theorem, we know that sample mean $\bar{x}_n$ should be Normally distributed for large enough n.

Then the confidence interval for the sample mean is $\bar{x} \in (\mu - z_{1-\alpha/2}\sigma/\sqrt{n}, \mu + z_{1-\alpha/2}\sigma/\sqrt{n})$. α is the chance that sample mean $\bar{x}_n$ falls outside the confidence interval around the population mean μ. $z_{1-\alpha/2}$ is called a z-score – it is a factor that depends on α and defines the size of the of confidence interval in terms of the standard deviation of sample mean $\bar{x}_n$ (standard error): $\sigma/\sqrt{n}$.

Alternatively, we can express the dependency between the confidence interval size and the probability of a sample mean falling outside of such confidence interval using the the following relationship:

$$P(d = |\bar{x}_n - \mu| \geq z_{1-\alpha/2}\sigma/\sqrt{n}) = \alpha$$

The equation reads as follows: With probability α, the distance between true mean μ and the drawn sample mean $\bar{x}_n$ exceeds $z_{1-\alpha/2}$ standard deviations of the mean (standard errors, $\sigma/\sqrt{n}$).[4] For example, for $\alpha = 0.05$, $z_{1-\alpha/2} = 1.96$ – that is, in 95% of experiments/studies that we run, sample mean will fall within approximately two standard errors ($\sigma/\sqrt{n}$)

[4] *Derivation:* z-score is indexed by $1-\alpha/2$ to indicate that for a standard normal random variable $Z_i \sim_d$

around the population mean. In other words, $\bar{x} \in (\mu - 1.96\sigma/\sqrt{n}, \mu + 1.96\sigma/\sqrt{n})$ in 95% of random samples.

Note that the distribution is symmetric and the sample mean $\bar{x}$ may fall outside the confidence interval on the right and on the left of μ. Thus, the probability that $\bar{x}_n$ exceeds μ by $z_{1-\alpha/2} = 1.96$ standard deviations is $\alpha/2 = 0.025$ – and it is the same as the probability that $\bar{x}_n$ is lower than μ by 1.96 standard deviations; combining these two possibilities, we get $\alpha = 0.05$. Another way to express this is to write that $P(\bar{x}_n \leq \mu - 1.96\sigma/\sqrt{n}) = P(\bar{x}_n \geq \mu + 1.96\sigma/\sqrt{n}) = 0.025$ and thus $P(|\bar{x}_n - \mu| \geq 1.96\sigma/\sqrt{n}) = 0.05$.

We can get z-scores corresponding to desired α by applying a quantile function (also called inverse CDF) for Normal distribution to $1 - \alpha/2$ quantity as follows:

```
# z-scores for different alpha
from scipy.stats import norm
alpha = np.array([0.1, 0.05, 0.01])
z = np.round(norm.ppf(1 - alpha / 2), 2) # applying inverse CDF
print(z)
```

```
[1.64 1.96 2.58]
```

We can use the inequality

$$d = |\bar{x}_n - \mu| \geq z_{1-\alpha/2}\sigma/\sqrt{n}$$

to directly set the sample size. After algebraic manipulation, we get the lower bound for the sample size

$$n \geq \frac{z^2_{1-\alpha/2}\sigma^2}{d^2}$$

that ensures that in $100 \cdot (1-\alpha)\%$ of experiments the sample mean will fall within the confidence interval around the population mean with the specified margin of error $d = |\bar{x}_n - \mu|$. Optimal sample size n^* is then the smallest integer greater or equal to this quantity: $n^* = \textbf{ceil}\left(\frac{z^2_{1-\alpha/2}\sigma^2}{d^2}\right)$. For 95% confidence interval, we get $n^* = \textbf{ceil}\left(\frac{1.96^2\sigma^2}{d^2}\right)$. Consider the case, where $\bar{x}_n$ is a proportion between 0 and 1, as in the earlier examples with the proportions of votes. Let us set the margin of error $d = 0.01$, or 1 percentage point difference between sample and population means. Then $n^* = \textbf{ceil}\left(\frac{1.96^2\sigma^2}{0.01^2}\right)$.

The last challenge we face is that we do not *a priori* know the population standard deviation σ across observations (binary preference statements by the respondents for Alice vs. Bob). Luckily, it turns out that, for binary random variables, there is a closed form

$N(0,1)$, $P(Z_i \leq z_{1-\alpha/2}) = 1 - \alpha/2$. By symmetry, $P(z_{1-\alpha/2} \leq Z_i \leq z_{1-\alpha/2}) = 1 - \alpha$. Multiplying Z_i by $\sigma/\sqrt{n}$ and adding μ, we get $\mu + Z_i\sigma/\sqrt{n} \sim_d N(\mu, \sigma^2/n)$ and $P(\mu - z_{1-\alpha/2}\sigma/\sqrt{n} \leq \bar{x}_n \leq \mu + z_{1-\alpha/2}\sigma/\sqrt{n}) = 1-\alpha$, where $\bar{x}_n = \mu + Z_i\sigma/\sqrt{n}$. Subtracting μ and taking absolute value, we get $P(|\bar{x}_n - \mu| \leq z_{1-\alpha/2}\sigma/\sqrt{n}) = 1 - \alpha$. We can also write it as $P(|\bar{x}_n - \mu| \geq z_{1-\alpha/2}\sigma/\sqrt{n}) = \alpha$. Note that we can write $\geq$ instead of $>$ because equality $|\bar{x}_n - \mu| = z_{1-\alpha/2}\sigma/\sqrt{n}$ is a zero-probability event.

formula for the standard deviation based on the population proportion (see variance of Bernoulli distribution[5]): $\sigma = \sqrt{\mu(1-\mu)}$. For $0 \leq \mu \leq 1$, this quantity is maximized at $\mu = 0.5$: $0.5 = \arg\max_\mu \sqrt{\mu(1-\mu)}$, which can be obtained via differentiation. Then $\sigma_{max} = \sqrt{0.5 \cdot 0.5} = 0.5$. We can then assume this largest possible standard deviation, which will, conservatively, give us the necessary sample size under the worst possible conditions:

$$n^* = \textbf{ceil}\left(\frac{1.96^2 0.5^2}{0.01^2}\right) = 9,604$$

(Alternatively, appropriate standard deviation could be gleaned from past studies.)

Thus, if we want to measure proportion to ± 0.01 error in 95% of measurements under the worst possible variance (when true proportion is close to 0.5), the sample size has to be 9,604 – almost ten thousand responses. That is a large number! What if we are fine with a wider margin of error of ± 0.04? Then

$$n^* = \textbf{ceil}\left(\frac{1.96^2 0.5^2}{0.04^2}\right) = 601.$$

This sounds much more feasible, but such a survey would not be very useful if the true proportion is within 0.5 ± 0.04 range.

You should now have a good idea of how such analysis goes. If you are dealing not with proportion, but with continuous variables, you would only need to substitute in a reasonable guess of σ, possibly, based on prior research, as well as the desired margin of error on the same scale. However, luckily, most critical questions in marketing surveys can be cast as binary questions, and the sample size based on proportions is, in practice, a good enough guide.

As a side note, the optimal sample size is closely bounded above as follows:

$$n^* = \textbf{ceil}\left(\frac{1.96^2 0.5^2}{d^2}\right) \leq \textbf{ceil}\left(\frac{2^2 0.5^2}{d^2}\right) = \textbf{ceil}\left(\frac{1}{d^2}\right),$$

which gives a great rule of thumb for a quick conservative sample size estimate.

Sample size based on Type 1 and 2 errors

We can use this type of $z\sigma/\sqrt{n}$ confidence interval as a way to test hypotheses. For example, H_0: $\mu = 0.5$ vs. H_1: $\mu \neq 0.5$. To reject $\mu = 0.5$, we need our sample mean $\bar{x}$ to lie outside the 95% confidence interval around the 0.5 value, which can be expressed as the following familiar test: $|\bar{x} - 0.5| \geq z_{1-\alpha/2}\sigma/\sqrt{n}$, with $z_{1-\alpha/2} = 1.96$ for $\alpha = 0.05$. If we use this rule, there is only 5% chance that we are wrong when we reject H_0 – that is, we

[5] `https://en.wikipedia.org/wiki/Bernoulli_distribution`

conclude $\mu \neq 0.5$, whereas in reality $\mu = 0.5$. In other words, the confidence interval size controls the *false positive rate*, also known as the *type 1 error*.

However, this may not be enough of an assurance. We may further be interested in limiting the *false negative rate*, also known as the *type 2 error*, of declaring $\mu = 0.5$ when $\mu \neq 0.5$. This would happen, for example, if for some $\mu > 0.5$, $|\bar{x} - \mu| < z\sigma/\sqrt{n}$. It is customary to limit this probability to be less than $\beta = 0.2$.

It turns out that we can achieve this if we compute the sample size using an inflated z-score $z = z_{1-\alpha/2} + z_{1-\beta} = 1.96 + 0.84 = 2.8$ and then perform hypothesis test as usual using $z_{1-\alpha/2}$ z-score. This hypothesis test is expected to have at least 100%-20% = 80% *power* under the 95% confidence interval. That is, H_0 will be rejected when H_1 is actually true in 80% of cases. See a chapter on sample size by Andrew Gelman for the explanation and algebra behind this [GH06].

A common challenge marketers face is to evaluate a set of ads and to determine which one is more effective. Here, the largest detectable effect conditional on the study size is the quantity we worry about. The already presented analytical machinery can be readily extended to these settings. In particular, A/B test can be cast in the hypothesis testing settings with $H_0 : \mu_A - \mu_B = 0$ and $H_1 : \mu_A - \mu_B \neq 0$.

Here μ_A, μ_B are the group means (for example, click through rates – proportion of ad displays resulting in clicks) and d denotes a difference between two means. Let n_A, n_B be sample sizes for the two groups. $N = n_A + n_B$. We will assume equal sample sizes between groups $n = n_A = n_B$ for simplicity, so $N = 2n$ (unequal sample size and variance formulas are also available [GH06]). We will also assume equal variance within groups at σ^2. We reject the hypothesis of group equality when $|\mu_A - \mu_B| \geq z_{1-\alpha/2}\sigma\sqrt{2/n}$ (factor $\sqrt{2}$ arises to account for pooled variance between two compared groups). Then, following our previous discussion and using the inflated z-score, per-group sample size should be

$$n \geq \frac{(z_{1-\alpha/2} + z_{1-\beta})^2 2\sigma^2}{d^2}$$

Here $z_{1-\alpha/2} = 1.96$ is, as earlier, the z-score for 95% confidence interval – it ensures that the probability of a false positive – detected difference when $d = 0$ – is under 5%. $z_{1-\beta} = 0.84$ ensures that the probability of a false negative (no difference when $d \neq 0$) is under 20%, that is, power $1 - \beta$ is over 80%. Let us apply this formula for the optimal sample size to detect a 0.001 difference in click-through rates (CTRs), under $\sigma^2 = 0.01 \cdot 0.99$ within-group variance, corresponding to CTRs of around 1%.

Then we get

$$n^* = \mathbf{ceil}\left(\frac{(1.96 + 0.84)^2 \cdot 2 \cdot 0.01 \cdot 0.99}{0.001^2}\right) = 155,232.$$

What we find is that a statistical A/B test requires over 310 thousand ad impressions (displays) across two arms combined, which, in and of itself, is a substantial advertising

campaign and may be unacceptable at early testing stages. In the next section, we will see what alternative options we have when the sample sizes dictated by the hypothesis testing framework are infeasible.

Let us run a simple simulation to check if we are indeed achieving the desired power. We will simulate comparisons in CTRs between two ads with true CTR of 1.1% and 1.2% ($d = 0.001$), with $\alpha = 0.05$ and $\beta = 0.2$, as in the example above. We will compare CTRs between two ads on samples of size n^* as dictated by the formula above and record how frequently we can correctly reject H_0 of identical CTRs between the two ads.

```
def power_simulation():
    # true CTR for ads for simulation
    pA = 0.011
    pB = 0.012
    # desired detectable difference between ads
    d = 0.001
    # z scores for desired confidence interval (1-alpha) and power (1-beta)
    alpha = 0.05
    beta = 0.2
    z_alpha, z_beta = np.round(norm.ppf(
        np.array([1 - alpha / 2,  1 - beta])), 2)
    z = z_alpha + z_beta

    # sample size using z = 2.8 score to ensure 80% power
    n = int(np.ceil(2 * 0.01 * 0.99 * z**2 / d**2))

    # array to store zero difference hypothesis rejection
    reject = np.zeros(1000)

    # what power do we observe empirically?
    for i in range(1000):
        # sampling CTR for two ads
        theta_A = np.mean(np.random.binomial(n=1, p=pA, size=n))
        theta_B = np.mean(np.random.binomial(n=1, p=pB, size=n))
        # simulated hypothesis test using z = 1.96 score
        reject[i] = 1.0 * (np.abs(theta_A-theta_B) > z_alpha * np.sqrt(0.01
                                                * 0.99 * 2 / n))

    power = np.mean(reject)
    return power

np.random.seed(1234)
print(power_simulation())
```

```
0.79
```

We obtain power of $\sim$0.8 as required.

Sample size based on a profit maximization principle

When selecting the sample size so far, we have used as our target metrics different statistical concepts such as confidence intervals and the probabilities of false positives and false negatives (type 1 and 2 errors). We set acceptable cutoffs for probabilities of type 1 error (α) and type 2 error (β or 1-power) to get a minimally acceptable sample size n.

However, there are some issues with this approach. First, as we have seen, such analysis can result in infeasibly large sample sizes. Second, metrics like type 1 and 2 errors do not directly translate into management priorities. In particular, the focus on type 1 error – the error of concluding ads are different when they are actually similarly effective – does not make much sense profit-wise, as we would be fine picking the wrong ad when the profits from the two are almost identical. Type 1 and 2 errors also do not take into account exploration-exploitation trade-off due to use of a potentially limited population for testing suboptimal ads. Application of hypothesis testing also leads to an awkward philosophy. Even if H_0 cannot be rejected and two ads are considered statistically indistinguishable based on adopted statistical standards, we would still want to pick the ad with higher observed reward – but under hypothesis testing such thinking is considered faulty.

Luckily, there is an alternative formal decision-theoretic framework due to Feit and Berman [FB19] based on profit maximization rather than classical statistical hypothesis testing philosophy that we can adopt to address this problem. The setup is as follows: $Y_j \sim N(\mu_j, s^2)$ is random "profit" from an ad j (for example, the ad's click through rate). Two group means follow an identical prior distribution: $\mu_A, \mu_B \sim N(\mu, \sigma^2)$. Parameters μ, σ, s are assumed known. Let N be the total population on which we can run the ad. The decision framework considers two stages, where in the first stage we test ads A and B on samples of size n_A and n_B. We then deploy the best ad on the remaining sample $N - n_A - n_B$. Customers that get shown the ads are randomly assigned between these groups. Notice that the expected profit during the first stage is just $\Pi[1] = \mu(n_A + n_B)$. It can be shown that the the profit during the second stage is $\Pi[2] = (N - n_A - n_B)\left[\mu + \frac{\sqrt{2}\sigma^2}{\sqrt{\pi}\sqrt{2\sigma^2 + s^2\frac{n_A+n_B}{n_A n_B}}}\right]$. Finally, we can obtain an analytical formula that gives the optimal sample size that maximizes the total profit $\Pi[1] + \Pi[2]$:

$$n_A = n_B = \sqrt{\frac{N}{4}\left(\frac{s}{\sigma}\right)^2 + \left(\frac{3}{4}\left(\frac{s}{\sigma}\right)^2\right)^2} - \frac{3}{4}\left(\frac{s}{\sigma}\right)^2.$$

Notice that $n_A = n_B$ need not be an integer – we will err on the conservative side of the larger sample size and set $n^* = \textbf{ceil}\,(n_A)$.

Let us apply this formula to compare the sample sizes we can get vs. statistical hypothesis testing. We will follow the setup from before, where the standard deviation around the ad mean μ_j is $s = \sqrt{0.01 \cdot 0.99}$ and we would like to detect a difference of $d = 0.001$

between the ads. We now need to specify standard deviation of the prior distribution σ. We can set it so that it gives $E[|\mu_A - \mu_B|] = d = 0.001$ to make the setup comparable to hypothesis testing. After some manipulation, we get $\sigma = 0.00089$, which corresponds to expected absolute difference of 0.1% between means.[6] We will also set $N = 1,000,000$ as the total potential audience for simplicity.

```
N = 1000000
s = np.sqrt(0.01*0.99)
sigma = 0.00089  # d=0.001
r = (s/sigma)**2
n = int(np.ceil(np.sqrt((N/4)*r + (0.75*r)**2) - 0.75*r))
print(n)
```

```
47,305
```

As a result, we get $n^* = 47,305$, which is ~ 3 times less than the $155,232$ required sample size under the classical hypothesis testing framework. If this sample size is still inappropriately large, we could reduce the available population N, which will result in a smaller sample size still. This flexibility makes profit maximizing framework attractive in practice and we expect to see its growing use in ad testing and other settings.

As a bonus, there is also an expression to compute the probability of the wrong ad being chosen as a result:

$$E[Pr(\textit{Wrong choice})] = \frac{1}{4} - \frac{1}{2\pi}\arctan\left(\frac{\sqrt{2}\sigma}{s}\sqrt{\frac{n_A n_B}{n_A + n_B}}\right).$$

(Formula simplifies when $n_A = n_B = n$.)

```
p_wrong = np.round(0.25-(1 / (2 * np.pi)) * np.arctan(np.sqrt(n) * sigma /
                                                      s),3)
print(p_wrong)
```

```
0.076
```

Applying the formula for the probability of the wrong choice, we get 7.6%, which is not bad at all.

[6] $\mu_A, \mu_B \sim N(\mu, \sigma^2) \to \mu_A - \mu_B \sim N(0, 2\sigma^2) \to E[|\mu_A - \mu_B|] = 2\sigma/\sqrt{\pi} := 0.001 \to \sigma = 0.001\sqrt{\pi}/2 \approx 0.00089$.

A/B test without a sample size: Online learning approach

In the ad testing example, the distinction between the ad testing and the ad deployment stages is somewhat artificial and, in fact, harmful. For example, by fixing the ad after the ad testing stage we are missing the opportunity to learn more and to adjust ad displays throughout the testing stage. It turns out, there is an even more profit-maximizing way to perform an ad test, which does not require worrying about the sample size at all. The approach is based on *Thompson sampling*, which belongs to a class of *online learning* algorithms and is designed to solve the *multi-armed bandit* problem.[7] The idea is to continuously adjust the proportion in which the ads are shown – in proportion to their observed rewards. Specifically, the heuristic but powerful principle behind Thompson sampling is to show an ad with the probability that it is better than any other considered ad.

For binary outcomes (e.g., clicks), it is easy to keep track of the performance of the ads using $Beta(\alpha, \beta)$ distribution, which is a flexible distribution of the random variable in the $(0, 1)$ range. In particular, for each ad, let θ_A and θ_B denote the corresponding click-through rates. At the beginning, we assume $\theta_A, \theta_B \sim Beta(\alpha, \beta)$, which is our prior distribution of the CTR for both ads. Its expectation is $E[\theta] = \alpha/(\alpha+\beta)$. Let k_A denote the number of clicks observed on ad A during n_A ad displays (also called impressions), which is distributed Binomial. Then for ad A, an updated posterior distribution of the CTR is also $Beta(\alpha + k_A, \beta + (n_A - k_A))$, with the expectation $E[\theta_A^{post}] = (\alpha + k_A)/(\alpha + \beta + n_A)$. The results are the same for ad B (substitute the index). (We are able to obtain such a nice closed form expression as Beta distribution is a conjugate prior for the Binomial distribution.[8])

How do we decide which ad to show? We simply take draws from the distributions for two tracked ads and show the ad that has the highest drawn value. That is, for $\theta_A \sim Beta(\alpha + k_A, \beta + (n_A - k_A))$ and $\theta_B \sim Beta(\alpha + k_B, \beta + (n_B - k_B))$, if $\theta_A > \theta_B$, we show ad A; we show ad B otherwise ($\theta_A = \theta_B$ is a zero-probability event in case of random draws). (This approach easily extends to multiple ads as well – we just show the ad that has the highest drawn θ from across all ads.) Note that the event $\theta_A > \theta_B$, so ad A gets shown, will occur with probability $E[\mathbf{1}\{\theta_A > \theta_B\}] = P(\theta_A > \theta_B)$, which is the probability that ad A is better than ad B, as desired under Thompson sampling heuristic (we are essentially performing a single step of a Monte Carlo simulation to decide which ad to show). After showing the winning ad, say A, we record whether we observed a click, updating k_A and n_A parameters appropriately.

Here is a simple example in code. Consider ads A and B with true CTR $\mu_A = 0.011$ and $\mu_B = 0.012$. How fast will Thompson sampling figure out the better ad?

[7] A problem of maximizing rewards from pulling different arms in a slot machine, one arm at a time, where each arm is associated with an unknown expected reward.

[8] `https://ocw.mit.edu/courses/18-05-introduction-to-probability-and-statistics-spring-2014/d6d722fef1c4f36cf2525bfe0b4f905a_MIT18_05S14_Reading14a.pdf`

```
def simulate(N):
    # parameters of Beta prior for two ads
    k_A = 0
    n_A = 0
    k_B = 0
    n_B = 0
    alpha = 1
    beta = 1

    # true CTR for ads - customer simulation
    muA = 0.011
    muB = 0.012

    # simulation for N ad displays decided by Thompson sampling
    for i in range(N):
        # sampling CTR for two ads
        # using updated posterior distributions
        # to reflect current knowledge about the ads
        theta_A = np.random.beta(alpha + k_A, beta + n_A - k_A, size=1)
        theta_B = np.random.beta(alpha + k_B, beta + n_B - k_B, size=1)

        # deciding which ad to show and updating running statistics
        # for the displayed ad
        if theta_A > theta_B:
            click_A = np.random.binomial(n=1, p=muA, size=1)[0]
            k_A += click_A
            n_A += 1
        else:
            click_B = np.random.binomial(n=1, p=muB, size=1)[0]
            k_B += click_B
            n_B += 1

    # expected CTR under posterior
    theta_A_est = (alpha + k_A)/(alpha + beta + n_A)
    theta_B_est = (alpha + k_B)/(alpha + beta + n_B)

    return theta_A_est, theta_B_est, k_A, k_B, n_A, n_B

np.random.seed(1234)
theta_A_est, theta_B_est, k_A, k_B, n_A, n_B = simulate(1000000)
theta_A_est = np.round(theta_A_est, 4) # Ad A estimated CTR
theta_B_est = np.round(theta_B_est, 4) # Ad B estimated CTR
prop_A = np.round(n_A / (n_A + n_B), 3) # Proportion of time (suboptimal)
                                        ad A shown
print(theta_A_est, theta_B_est, prop_A)
```

```
0.011 0.012 0.075
```

We can see that, when using Thompson sampling, the wrong ad gets shown during 7.5% of total 1 million ad displays. With the two-stage approach from previous section, we deterministically show $n_A/N \approx 4.7\%$ of wrong ads during test phase and then have 7.6% chance of having selected the wrong ad on top of that, which yields $\sim$11.6% proportion of wrong ads displayed in expectation,[9] so Thompson sampling "wastes" fewer ad displays on the wrong ads according to this simulation.

Thompson sampling approach obviates the need for knowing a sample size a priori (beforehand), as we can stop at any point (e.g., when ad budget runs out), while being sure that we spent money in the best possible way. Empirically, this Thompson sampling approach has been found to outperform profit-wise the two-stage approach from the prior section [FB19]. Thompson sampling approach can also easily incorporate more than two ads simultaneously, whereas closed-form solutions of this kind are not available in the case of results from the previous section. It is also easy to do personalization, where the expected ad performance can differ across encountered individuals based on their personal data, allowing for the more effective ad for a given individual to be shown with higher likelihood. Finally, the sampling approach can handle continuous outcomes as easily – for example, by using the normal distribution as a way to keep track of the ad performance.

The main drawback of the sampling approach is that it may be complex to implement from the technological standpoint, as online tracking of ad performance and fast data aggregation are necessary for its use.

Advanced: Brute force simulation approach to sample sizes

The discussed profit-maximizing framework does not yield closed-form analytical formulas for A/B/C tests (tests with more than two compared ads). Either profit maximization or hypothesis testing frameworks may fail to yield nice closed form solutions under different distributional assumptions. We may also want to extend the framework to non-trivial statistical tests, for example, involving predictive models and bootstrap confidence interval estimation.

If we nevertheless want to know a number for a sample size and do not want to use Thompson sapling method, we can resort to brute-force simulation to evaluate, for example, type 1 and 2 errors, profit, or any other measure of interest from different sample sizes and problem setups. While such simulations can be time-consuming, they are typically quite simple to implement and are affordable on the modern computers.

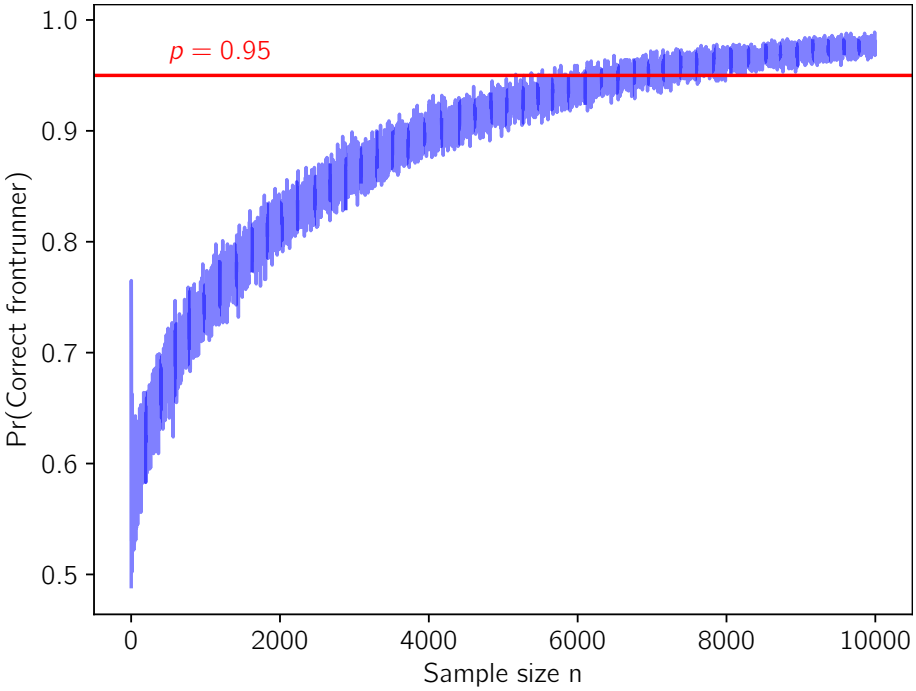
The core idea behind the simulations is to imagine what a true scenario might look like (e.g., how large is the difference between ad CTRs or between voter proportions for different political candidates) and to assess what errors/profit we would be getting depending on

[9] $P(\textit{wrong ad shown}) = P(\textit{test chose wrong ad})P(\textit{wrong ad shown} \mid \textit{test chose wrong ad}) + P(\textit{test chose correct ad})P(\textit{wrong ad shown} \mid \textit{test chose correct ad}) = 0.076 \cdot (1-0.047) + (1-0.076) \cdot 0.047 = 0.076 + 0.047 - 2 \cdot 0.076 \cdot 0.047 = 0.116$.

the sample size if the reality were like we imagine it. If we are uncertain about what the reality might be, it is better to be conservative, assuming the worst case scenario (e.g., little difference between CTRs or political candidates), or to evaluate multiple scenarios and make a decision holistically.

In fact, we have already conducted a simple simulation at the beginning of this chapter when we looked at the histogram spread under two sample sizes for a political survey. Let us now extend that example to find the optimal sample size. Consider we suspect two candidates are neck to neck and we suspect one of the front-runners has around 51% of votes. We want to find out what should be the sample size to correctly judge the front-runner as victorious (having $> 50\%$ share of votes).

```
from tqdm.notebook import tqdm # tracks progress

def simulate(N):
    # front-runner's true proportion of votes
    mu = 0.51

    # array to store the probability of correctly selecting the winner
    p_correct = np.zeros(N)

    # simulation for sample sizes up to 10,000
    for n in tqdm(range(N)):
        # simulating collection of 1,000 surveys with sample size n
        votes = np.random.binomial(n=1, p=mu, size=(n+1, 1000))

        # proportion of votes for front-runner in each survey
        x_avg = np.mean(votes, axis=0)

        # proportion of surveys where the true front-runner
        # is correctly identified as a winner
        p_correct[n] = np.sum(x_avg >= 0.5) / x_avg.shape[0]

    plt.plot(range(1,N+1), p_correct, color='blue', alpha=0.5)
    plt.xlabel('Sample size n')
    plt.ylabel('Pr(Correct frontrunner)')
    plt.axhline(y=0.95, color='red')
    plt.text(500, 0.965, r"$p=0.95$", rotation=0, color='red')
    plt.savefig('./figs/simulation.pdf', dpi=300, bbox_inches='tight',
                                         pad_inches=0)
    return None

np.random.seed(1234)
simulate(10000) # takes some time
```



Boom! Even with $n = 1,000$ individuals taking a survey, only in 75% of situations do we correctly call the front-runner, when the front-runner has 51% of votes in the population! Only at $n = 5,000$ are we solidly above 90% accuracy, and only at around $n = 9,000$ are we solidly above 95% mark. Remember this results when you see the next TV poll with $n < 1,000$ – especially where the race is even tighter than 51% for the front-runner. (We could do better by pooling different polls and thus getting a larger sample size, of course.)

Note that the setup in this voting simulation is somewhat different than in the statistical hypothesis testing from before. We only care here about the error of getting less than 50% votes for the victor with de-facto 51% votes (which is, arguably, what really matters to us as pollsters – we really care about calling the right victor, and probably care less about wrongly predicting the victor's support to be higher than it really is). In contrast, in the statistical analysis from before we cared about minimizing the symmetric error of our guess around 0.51 proportion. Also, we use the exact Bernoulli distribution in this simulation, whereas the statistical confidence interval from before was based on the normal distribution approximation to the distribution of the voting proportions, motivated by the Central Limit theorem. So, while these simulation results may be directionally similar to the analytical (formula-based) results from earlier, they are distinct. Arguably, they are also somewhat easier to explain to an average person.

This is an example of how simulation can help us answer precise questions without much hard math, but at the expense of extra computing time and resources. Importantly, given increasing computing power, this drawback becomes less important. Even when

an analytical formula is available, simulation can be a great sanity check to verify the correctness of the analytical computation.

Following this example, we can use simulations to answer a variety of questions. For instance, we could do power simulations, determining the necessary sample size to achieve a specific power of the experiment. In particular, we could measure how likely one is to get a significant coefficient in a linear regression when the data is randomly generated from a specific true model and the corresponding coefficient is non-zero in this true model. This is left to the reader as an exercise.

Advanced: Challenging cases and empirical scaling laws

There are types of problems where the right sample size can be hard to determine. Consider, for example, the problem of training a complex predictive model – specifically, a neural network – to predict personal information (such as demographics) from consumer facial images. How many observations do we need to train the model?

In case of a simple regression, we could assume how the dependency between inputs and outputs might look like, sample data from this model, run the simulation as in the previous section, and choose sample size that allows for a satisfactory approximation of the true model from the sampled data. However, it is not clear how to build a high-fidelity hypothetical "true" model for such complex data as facial images, where dependencies are highly non-linear, so we cannot straightforwardly apply this approach.

One feasible approach, if there is some existing data, is to estimate the model on existing data subsets of different sizes – tracking how the metric of interest, for example, holdout accuracy of prediction, changes with the sample size used for training. This can give one an idea of how the predictive power empirically scales with the sample size. One can then use this empirical scaling law to select the sample size for the new data collection.

Unfortunately, this approach can be problematic. First, if one is collecting data of very different nature (different types of predictions to be made form facial images, facial images from individuals of different demographic distribution, etc.), it is very hard to say how relevant the historical data would be to determining the sufficiency of the sample size for the problem at hand. Second, if the historical data only contained 1,000 observations, it may shed little light on the possible predictive power of the model trained on 1 million observations, even if the data were drawn from similar population, simply because the larger scale of data can non-linearly increase the model quality [Kap+20] and it may be hard to infer the predictive power achievable on a large sample from such a relatively small sample.

In these cases, while historical data may be of some guidance, one really has to resort to the old but good approach of "let's try and see." In other words, one may have to sequentially accumulate data, continuously assessing the resulting model quality, stopping once it achieves a satisfactory level, which is similar to the continuous experimentation

philosophy of the Thompson sampling, reviewed earlier. While this conclusion may not be the most satisfactory one, it is good to be transparent about the fact that there are instances where the sample size cannot be reliably determined a priori but needs to be figured out on the go.

Further reading

This chapter covers the most fundamental and critical sample size concepts for any marketing analyst, however, it is not all-encompassing. Sample size literature has been developed in a variety of fields and can get rather specialized. The references at the end of this chapter will be a good place to go for more information. Van Belle [VB11] provides guidance on different rules of thumb for sample size determination. A curious reader may also perform an online search for keywords like finite population correction; minimum detectable effect; Bayesian sample size determination; group sequential methods and the predictive probability of success from the clinical trial literature; and simulation for power analysis. There are also many online software packages that yield sample size quantities for the desired settings if one does not want to do the independent work.

References

[FB19] Elea McDonnell Feit and Ron Berman. "Test & roll: Profit-maximizing A/B tests". *Marketing Science* 38.6 (2019), pp. 1038–1058.

[GH06] Andrew Gelman and Jennifer Hill. *Data analysis using regression and multilevel/hierarchical models*. Cambridge University Press, 2006.

[Kap+20] Jared Kaplan et al. "Scaling laws for neural language models". *arXiv preprint arXiv:2001.08361* (2020).

[VB11] Gerald Van Belle. *Statistical rules of thumb*. John Wiley & Sons, 2011.