university of groningen / faculty of science and engineering / mathematics and applied mathematics

Requesting a referendum: estimating the number of valid requests

Bachelor's Project Mathematics

November 2017

Student: C. Kool

First supervisor: Prof. dr. E.C. Wit

Second assessor: Dr. M.A. Grzegorczyk

Requesting a referendum: estimating the number of valid requests

Chantal Kool[1]

November 2017

Abstract

When there is the desire to organize a consulting referendum in The Netherlands, a certain number of requests needs to be collected to make sure there is enough support to organize one. History tells us that there are people who submit more then one request, which makes all the requests of that person invalid. Therefore, these requests need to be filtered out of the total number of valid requests. With the use of a sample from the total number of requests, the number of valid requests is estimated. This thesis will provide a detailed explanation of the estimation procedure: what sample size needs to be chosen and how the estimator is constructed. The results will be compared to the method used before and to other known methods using simulations. Finally, there will be a look at possibilities to extend/improve the procedure.

Keywords: referendum, method of moments, hypergeometric distribution, approximation to normal distribution, sample size determination

[1]University of Groningen, c.kool@student.rug.nl

Contents

Introduction

It is possible for every citizen of The Netherlands to request a consulting referendum about all laws accepted by parliament, this is made possible by the "Wet raadgevend referendum", [Wrr] (Advisory referendum act). When there is the desire to organize a referendum, the initiator can start collecting requests in the form of signatures. In order to get an *introductory request*, 10,000 requests have to be collected within a term of four weeks and for checking their validity, the electoral council then has one week to conclude whether a sufficient number of requests is indeed valid. If the introductory request is accepted, another 300,000 requests are needed within a term of six weeks for a *definitive request*. Here the electoral council has two weeks to decide whether the number of requests is sufficient. If this definitive request is accepted, the referendum committee picks a date, on which the referendum will be held.

The question that now arises is: why do you even need to check the requests while the total number of requests is known? Within all the requests there appear to be invalid requests. Invalid here means that the request lacks information, is submitted by someone who is not entitled to vote or contains wrong information. However, these requests can be taken into account by including an error term in the number of valid requests. The biggest problem is that the experience teaches us that there are people who submit more than one request, which makes all the requests this person submitted invalid. In order to check whether there are truly enough valid *single* requests, the government uses a sample and a formula recorded in the "Besluit raadgevend referendum", [Brr] (Decision advisory referendum). However, it appears that that formula does not produce reliable results, which will be shown in Chapter 2[1].

A better method has been developed in [Wit, 2016] [2]. This method is able to confirm that the number of requests is sufficient, but is not able to reject the request; instead the conclusion is that the verdict is undecided. If the latter is the case, all requests need to be counted manually. This thesis will discuss this method: how it is constructed, what assumptions are made, the results it produces and how it can be improved.

Chapter 1 contains some background information, assumptions and some derivations that will be used later on in this thesis. In Chapter 2, there will be a thorough explanation of the method including determining the optimal sample size and defining the estimator for the number of valid single requests.

The social relevance of this topic is very high, especially at the moment of writing. In Spain there is a referendum on the independence of Catalonia and in the Netherlands they are in the process of trying to organize a referendum about updating the "Wet op de inlichtingen- en veiligheidsdiensten" (Act on Information and Security Services). One place from which we know how they determine the number of valid requests in the process of requesting a referendum is the State of Washington, they use the method stated in the Legislature. In Chapter 3 there will be a description of this method and it will be compared to the results from the new and old method from the Netherlands. These results are obtained by programming the methods in R and simulating different situations.

After contacting the government about the new method, they decided that they wanted a more specific answer to this problem. The request is formulated as follows: can we determine the number of valid requests in such a way that we allow ourselves a one percent chance on a wrong decision with a error margin of two percent? In Chapter 4 there will be a look at solving this problem, which will also result in an improvement of the existing method.

[1] This method will be called the "old method" from now onwards
[2] This method will be called the "new method" from now onwards

1 Background

To know whether there are enough valid requests in the process of requesting a referendum, the main variable of interest is the number of people who submitted a *single* valid request. This because it does happen that one person submits more than one request, because that person may have forgotten that he or she already submitted a request before, or that he or she is just so keen on the referendum getting organized that he or she decides to submit for example 50 requests. If someone submits more than one request, all the requests of that single person will become invalid. A referendum only gets organized if the number of single, valid requests is more than the threshold stated in the Wrr.
In order estimate the number of people who submitted one single valid request, we define:

$$\eta_i = \text{number of people who submitted } i \text{ requests.}$$

The total population is given by: $(\eta_1, \eta_2, ..., \eta_N)$ where we set the total number of requests by: $N = \sum_{i=1}^{N} i\,\eta_i$. We now want to define an estimator $\hat{\eta_1}$, this represents the number of people who submitted only one request and therefore, tells us whether there are enough requests. As stated before, a sample will be used, because it is highly undesirable to count all the requests by hand (since we are talking about more than 10,000 or 300,000 requests). In this sample all the requests will be checked on validity and from the information of this sample and the upcoming method, the request as a whole will be accepted or rejected.[3]
The information of the sample can be summarized as $X = (X_1, X_2, ..., X_n)$, where:

$$X_i = \text{the number of people who submitted } i \text{ requests according to the sample of size } n.$$

The sample is here defined as: $n = \sum_{i=1}^{n} i\,X_i$.

According to article 33 of the Wrr, there can also still be requests present in the population which lack information, contain wrong information or contain data from a person who is not entitled to vote. These requests are summarized in the following variable:

$$Y = \text{All the requests that lack information, contain wrong information or contain data from a person who is not entitled to vote.}$$

The $Var(Y)$ is however, very small compared to $Var(X_i)$ and will therefore not be taken into a count in the rest of this thesis.

1.1 Why does the old method cause trouble?

Now that all the variables are defined, let us first have a look at why the old method does not produce reliable results.

Example 1.1. Consider the case that we are trying to pass the definitive request, i.e. we need 300,000 valid single requests. If we now think of the situation where: N = 400,000, η_1 = 0 and η_2 = 200,000. This means that there are more than enough requests collected, but not a single one of

[3] A request can be accepted or rejected, with this we mean a request in the sense of an introductory or definitive request. Such a request is supported by people who submitted requests, note that the word "request" has two different meanings here.

them is valid. If we now use the method described in Brr, the sample size becomes:

$$\begin{aligned} n &= \frac{N \cdot z^2 \cdot p(1-p)}{z^2 \cdot p(1-p) + (N-1) \cdot F^2} \\ &= \frac{400,000 \cdot 2.576^2 \cdot 0.5(1-0.5)}{2.576 \cdot 0.5(1-0.5) + (400,000-1) \cdot 0.02^2} \\ &= 4104. \end{aligned}$$

Here, z, p and F are variables with fixed values. The probability that for a randomly chosen request in the sample the duplicate of that request occurs *in the sample* is $\frac{4103}{399999} \approx 0.0103$. So we expect that there are $0.0103 \cdot 4104 \approx 42$ double requests in the sample. This means that we expect to have approximately 4061 valid requests in the sample ($X_1 \approx 4061$). If we now use the given formula for the estimator, we get:

$$\begin{aligned} \hat{\eta_1} &= X_1 \cdot \frac{N}{n} \\ &\approx 4061 \cdot \frac{400,000}{4104} \\ &\approx 395,897 \end{aligned}$$

This would imply that there are more than enough valid requests, and therefore the request would be accepted, while in reality there is not a single valid request. This is without a doubt a very undesirable situation and therefore the reason to think about a different way to tackle this problem. In order to do this, we first need to look at some choices and assumptions that need to be made.

1.2 A conservative estimator

In order to make a good estimation of η_1, we need to look at all the possible ways that the requests were submitted. But as can be imagined, this can be done in an enormous number of ways, since there is an enormous amount of ways that a total of, for example 10100 requests can be permuted among the population $(\eta_1, \eta_2, ..., \eta_N)$. Using an algorithm on a computer can still solve this problem, but the main problem that arises here is that this leads to an estimator with an enormous variance. We conclude that without further assumptions the sample size needs to be very big, which is not desirable.

Instead of trying the above, we are going to look at a "worst case scenario", which means that we assume that there are people who submit their requests in such a way that they try to avoid detection. This makes the procedure easier and produces a smaller variance. The consequence of this assumption is that we take into a count that there are more multiple requests than estimated (this will be explained more thoroughly in Subsection 2.1). However, if this assumption fails to hold it means that there are more single submissions than estimated. We call the estimator that relies on the above assumption a *conservative estimator*.

The implication is now that we can conclude that there are enough requests if the procedure says so, but we can *not* guarantee that there are *not* enough valid requests if the procedure concludes there are not enough valid requests. This means that *if* the procedure concludes that there are not enough valid requests, all the requests need to be counted separately by hand.

In the method that will be explained in Chapter 2, an estimator for the number of valid requests will be created. In the process of describing the estimator, we need to define two more variables:

E_i = someone who appears in the sample and submitted i requests according to the sample

and

$$F_j = \text{someone in the sample who submitted } j \text{ requests in real life.}$$

We need those variables, since we want to find an expression for p_i, which is the probability that E_i happens. In order to understand how to calculate p_i we first need a little more information on the so-called hypergeometric distribution.

1.3 The hypergeometric distribution

The hypergeometric distribution is a discrete distribution which describes the probability of x successes in n draws *without* replacement from a population of size N which contains K successes. It differs from the binomial distribution in the sense that the binomial distribution describes the probability of x successes in n draws *with* replacement.

If a random variable X follows the following pmf:

$$\mathbb{P}(X = x) = \frac{\binom{K}{x}\binom{N-K}{n-x}}{\binom{N}{n}}$$

$$\text{where : } x = 0, 1, 2, ..., K; \qquad K - (N - n) \leq x \leq K; \qquad N, K, n \geq 0,$$

then X is hypergeometrically distributed.[Casella and Berger, 2002]

Now we can start calculating p_i. In order to get an expression which we can work with, we first need to use Bayes theorem:

$$\begin{aligned} p_i = \mathbb{P}(E_i) &= \frac{\frac{\mathbb{P}(E_i \cap F_j)\mathbb{P}(F_j)}{\mathbb{P}(F_j)}}{\frac{\mathbb{P}(E_i \cap F_j)}{\mathbb{P}(E_i)}} \\ &= \frac{\mathbb{P}(E_i|F_j)\mathbb{P}(F_j)}{\mathbb{P}(F_j|E_i)} \\ &= \sum_{j=i}^{N} \mathbb{P}(E_i|F_j)\mathbb{P}(F_j). \end{aligned} \tag{1.1}$$

Now we want to determine the preceding probabilities, the easiest is the probability of F_j: this is the number of requests submitted by person j divided by the total number of requests N, this leads to:

$$\mathbb{P}(F_j) = \frac{j\,\eta_j}{N}.$$

If we now look at the conditioning probability, we want to know the probability that we have i successes out of n draws in a "population" of size N with j successes. There is, however one thing we need to take into a count, which is that the probability that someone submitted 0 requests needs to be excluded, since it is not possible that this happens. Combining all this information gives the following formula:

$$\mathbb{P}(E_i|F_j) = \frac{\binom{j}{i}\binom{N-j}{n-i}}{\binom{N}{n}} \cdot \frac{1}{1 - \frac{\binom{j}{0}\binom{N-j}{n}}{\binom{N}{n}}}.$$

Which, combined with Equation (1.1) finally gives:

$$p_i = \mathbb{P}(E_i) = \sum_{j=i}^{N} \frac{\binom{j}{i}\binom{N-j}{n-i}}{\binom{N}{n}} \cdot \frac{1}{1 - \frac{\binom{j}{0}\binom{N-j}{n}}{\binom{N}{n}}} \cdot \frac{j\,\eta_j}{N}. \tag{1.2}$$

Now that we know why the old method doesn't produce reliable results, which assumptions we made, and what the value of p_i is, we can start with defining the estimator and optimal sample size.

2 Defining the estimator and the optimal sample size

In this chapter we try to find an expression for the estimator $\hat{\eta}_i$, we will do this by using the method of moments. Let's start by noticing that: $X_i \sim Bin(n, p_i)$. This implies that:

$$\mathbb{E}(X_i) = n \cdot p_i \qquad i = 1, ..., n.$$

Now we are going to make the following assumption:

$$\eta_{n+1} = ... = \eta_N = 0 \tag{2.1}$$

which states that we assume that there are no people who submitted more requests than the chosen sample size. This is the assumption that makes our estimator conservative. But why do we need to make this assumption?

2.1 Why an assumption is needed

It is known that the method of moments has a similar asymptotic efficiency (up to a constant) as the maximum likelihood estimator. Because our estimator is created using the method of moments and the fact that the maximum likelihood estimator is efficient, we can write:

$$Var(\hat{\eta}) \approx \mathcal{I}(\eta)^{-1}.$$

Recall that $\mathcal{I}(\eta)$ is the *Fischer information matrix*, which is given by:

$$\mathcal{I}(\eta) = -\mathbb{E}\left(\frac{\partial^2 \ell}{\partial \eta^2}\right)$$

where ℓ is the full likelihood. Note that $\frac{\partial \ell}{\partial \eta}$ is an $N \times 1$ vector, and therefore, $\frac{\partial^2 \ell}{\partial \eta^2}$ is an $N \times N$ matrix (the Hessian). Since we only have n observations from the sample, $rank\left(\frac{\partial^2 \ell}{\partial \eta^2}\right) = n$. Meaning that $\frac{\partial^2 \ell}{\partial \eta^2}$ is not invertible, since it has no full rank. This in turn implies that: $Var(\hat{\eta}) \approx \infty$. An estimator with a very large variance does not provide much information, so this is why we make the assumption stated in Equation (2.1).

2.2 Defining the estimator

Now we want to use the method of moments to define the estimator. To do this, it is convenient to start at $i = n$ and work from the back to the front and end at $i = 1$. This is convenient since when $i = n$, all requests in the sample are submitted by one person only. The lower the value of i becomes, the more different people submitted requests. The method of moments therefore gives:

$$\begin{aligned} X_i &= n \cdot p_i. \\ &= n \cdot \sum_{j=i}^{n} \frac{\binom{j}{i}\binom{N-j}{n-i}}{\binom{N}{n}} \cdot \frac{1}{1 - \frac{\binom{j}{0}\binom{N-j}{n}}{\binom{N}{n}}} \cdot \frac{j\,\eta_j}{N} \\ &= \sum_{j=i}^{n} \eta_j \cdot \frac{\binom{j}{i}\binom{N-j}{n-i}}{\binom{N}{n}} \cdot \frac{1}{1 - \frac{\binom{j}{0}\binom{N-j}{n}}{\binom{N}{n}}} \cdot \frac{j}{N} \cdot n \end{aligned}$$

Because we want to find an estimator for η, we define the following expression for convenience:

$$\pi_{i,j} = \frac{\binom{j}{i}\binom{N-j}{n-i}}{\binom{N}{n}} \cdot \frac{1}{1 - \frac{\binom{j}{0}\binom{N-j}{n}}{\binom{N}{n}}} \cdot \frac{j}{N} \cdot n.$$

Now we can write:

$$\begin{aligned}
X_n &= \sum_{j=n}^{n} \eta_j \cdot \pi_{n,j} \\
&= \pi_{n,n} \cdot \eta_n \\
&\Rightarrow \hat{\eta}_n = \frac{X_n}{\pi_{n,n}} \\
X_{n-1} &= \sum_{j=n-1}^{n} \eta_j \cdot \pi_{n-1,j} \\
&= \pi_{n-1,n-1} \cdot \eta_{n-1} + \pi_{n-1,n} \cdot \eta_n \\
&\Rightarrow \hat{\eta}_{n-1} = \frac{X_{n-1} - \pi_{n-1,n} \cdot \hat{\eta}_n}{\pi_{n-1,n-1}} \\
X_{n-2} &= \sum_{j=n-2}^{n} \eta_j \cdot \pi_{n-2,j} \\
&= \pi_{n-2,n-2}\eta_{n-2} + \pi_{n-2,n-1}\eta_{n-1} + \pi_{n-2,n}\eta_n \\
&\Rightarrow \hat{\eta}_{n-2} = \frac{X_{n-2} - \sum_{j=n-1}^{n} \pi_{n-2,j}\hat{\eta}_j}{\pi_{n-2,n-2}}. \\
&\vdots
\end{aligned}$$

We can proceed this way until we arrive at $n = 1$. This can now be generalized to the following formula for the estimator:

$$\hat{\eta}_i = \frac{X_i - \sum_{j=i+1}^{n} \pi_{i,j}\hat{\eta}_j}{\pi_{i,i}}. \tag{2.2}$$

For simplifying reasons we make one more assumption, namely that there are only people who submitted 1 or 2 requests. This means that $\eta_3 = \cdots = \eta_N = 0$ and that our population now only consists of η_1 and η_2. This assumption is still in line with the conservative nature of the estimator. Given this, the fact that we now only observe X_1 and X_2 and using Equation (2.2), we can write:

$$\begin{aligned}
\hat{\eta}_2 &= \frac{X_2 - \sum_{j=3}^{n} \pi_{2,j}\hat{\eta}_j}{\pi_{2,2}} = \frac{X_2 - 0}{\pi_{2,2}} = \frac{X_2}{\pi_{2,2}} \\
\hat{\eta}_1 &= \frac{X_1 - \sum_{j=2}^{n} \pi_{1,j}\hat{\eta}_j}{\pi_{1,1}} = \frac{X_1 - \frac{\pi_{1,2}}{\pi_{2,2}}X_2}{\pi_{1,1}}.
\end{aligned}$$

2.3 Defining the optimal sample size

Now that we have an expression for η_1 and η_2, we can proceed to defining the optimal sample size.

The assumption we made above also implies that:

$$X_2 = \frac{n - X_1}{2}$$

where we recall that n is the sample size. Substituting the above into the equation for $\hat{\eta}_1$ we get:

$$\begin{aligned}
\hat{\eta}_1 &= \frac{X_1 - \frac{\pi_{1,2}}{\pi_{2,2}} X_2}{\pi_{1,1}} \\
&= \frac{X_1 - \frac{\pi_{1,2}}{\pi_{2,2}} \cdot (\frac{n - X_1}{2})}{\pi_{1,1}} \\
&= \frac{X_1 \cdot (1 + \frac{\pi_{1,2}}{2 \cdot \pi_{2,2}}) - \frac{\cdot n \pi_{1,2}}{2 \cdot \pi_{2,2}}}{\pi_{1,1}} \\
&= X_1 \cdot \underbrace{\frac{2 \cdot \pi_{2,2} + \pi_{1,2}}{2 \cdot \pi_{1,1} \cdot \pi_{2,2}}}_{a} - n \cdot \underbrace{\frac{\pi_{1,2}}{2 \cdot \pi_{1,1} \cdot \pi_{2,2}}}_{b} \\
&= X_1 \cdot a - n \cdot b.
\end{aligned}$$

In proceeding in the determination of the optimal sample size, the hypothesis we would like to test is the following:

$$\begin{aligned}
H_0 &: \text{we have } N_0 - k \text{ valid requests, i.e., we don't have enough requests} \\
H_1 &: \text{we have more than } N_0 \text{ valid requests, i.e., we have enough requests}
\end{aligned}$$

where N_0 is the number of requests needed. Under H_0 we now have that: $\eta_1 = N_0 - k$ and $\eta_2 = \frac{N - (N_0 - k)}{2}$. Having X_1 being approximately binomially distributed, we can make use of a normal approximation for $\hat{\eta}_1$. Recall that this means that:

$$\begin{aligned}
& \hat{\eta}_1 \sim \mathcal{N}(\mathbb{E}(\hat{\eta}_1), Var(\hat{\eta}_1)) \\
\Rightarrow \quad & \hat{\eta}_1 \sim \mathcal{N}(anp_1 - bn, a^2 n(1 - p_1)p_1).
\end{aligned}$$

where p_1 can be derived using Equation (1.2):

$$\begin{aligned}
p_1 &= \frac{\binom{N-1}{n-1}}{\binom{N}{n}} \cdot \frac{1}{1 - \frac{\binom{N-1}{n}}{\binom{N}{n}}} \cdot \frac{N_0 - k}{N} + \frac{\binom{2}{1}\binom{N-2}{n-1}}{\binom{N}{n}} \cdot \frac{1}{1 - \frac{\binom{N-2}{n}}{\binom{N}{n}}} \cdot \frac{N - (N_0 - k)}{N} \\
&= \frac{\binom{N-1}{n-1}}{\binom{N}{n} - \binom{N-1}{n}} \cdot \frac{N_0 - k}{N} + \frac{2 \cdot \binom{N-2}{n-1}}{\binom{N}{n} - \binom{N-2}{n}} \cdot \frac{N - (N_0 - k)}{N}
\end{aligned}$$

note that $\eta_3, ..., \eta_N = 0$, so we have only two nonzero terms in the summation that represents p_1. If we take a step back and look at the original problem, the most desirable situation is that the probability of having enough valid requests while there are in reality not enough valid requests is as small as possible. Therefore, we want that:

$$\mathbb{P}(\text{having enough valid requests} \,|\, \text{there are not enough valid requests}) \leq \alpha$$

i.e.

$$\mathbb{P}(\hat{\eta}_1 > (1 + m)N_0 \,|\, H_0 \text{ is true}) \leq \alpha. \tag{2.3}$$

Where $(1+m)N_0$ is the decision margin, which will be explained more in Chapter 3. Shortly, it is a correction which depends on the allowed error margin m, which can be chosen by the user of this method. It makes sure that the chance of accepting the request as a whole, while in reality there are not enough requests, is as small as the user desires.
It is, however undesirable that Equation (2.3) involves an inequality, the Karlin-Rubin Theorem helps with creating an equality we can work with.

Theorem 1 (Karlin-Rubin)**.** *Consider testing $H_0 : \theta \leq \theta_0$ versus $H_1 : \theta > \theta_0$. Suppose that T is a sufficient statistic for θ and the family of pdfs or pmfs $\{g(t|\theta) : \theta \in \Theta\}$ of T has an MLR (monotone likelihood ratio). Then for any t_0, the test that rejects H_0 if and only if $T > t_0$ is a UMP (uniformly most powerful) level α test, where $\alpha = \mathbb{P}(T > t_0 \,|\, H_0$ is rejected$)$.[Casella and Berger, 2002]*

Since $\hat{\eta}_1$ is normally distributed, we know that it has an MLR. Now using Karlin-Rubin and Equation (2.3) we arrive at the following expression:

$$\mathbb{P}(\hat{\eta}_1 > (1+m)N_0 \,|\, N_0 \text{ valid requests}) = \alpha.$$

Using the normal approximation of $\hat{\eta}_1$, the preceding equation can be solved for sample size n:

$$\begin{aligned}\alpha &= \mathbb{P}(\hat{\eta}_1 > (1+m)N_0 \,|\, N_0 \text{valid requests}) \\ &= \mathbb{P}\left(\frac{\hat{\eta}_1 - (anp_1 - bn)}{\sqrt{a^2 n(1-p_1)p_1}} > \frac{(1+m)N_0 - (anp_1 - bn)}{\sqrt{a^2 n(1-p_1)p_1}} | N_0 \text{ valid requests}\right).\end{aligned}$$

Using that $\frac{\hat{\eta}_1 - (anp_1 - bn)}{\sqrt{a^2 n(1-p_1)p_1}} \sim \mathcal{N}(0,1)$:

$$\begin{aligned}&\alpha = \mathbb{P}(Z > Z_{1-\alpha/2} | (1+m)N_0 \text{ valid requests}) \\ &\Rightarrow Z_{1-\alpha/2} = \frac{N_0 - n(ap_1 - b)}{\sqrt{a^2 n(1-p_1)p_1}} \\ &\Rightarrow Z_{1-\alpha/2} a^2 n p_1(1-p_1) = N_0^2 + n^2(ap_1 - b)^2 - 2 \cdot nN_0(ap_1 - b) \\ &\Rightarrow n^2 \underbrace{(ap_1 - b)^2}_{\beta} + n \underbrace{(2N_0(b - ap_1) - Z^2_{1-\alpha/2} a^2 p_1(1-p_1))}_{\gamma} + \underbrace{N_0^2}_{\delta} = 0.\end{aligned}$$

The last expression is just a quadratic equation with regard to n, solving this results in the following expression for n:

$$\begin{aligned}n &= \frac{-\gamma \pm \sqrt{\gamma^2 - 4\beta\delta}}{2\beta} \\ &= \frac{-\gamma + \sqrt{\gamma^2 - 4\beta\delta}}{2\beta} \qquad (n > 0) \\ &= \frac{-Z^2_{1-\alpha/2}(N - N_0)^2 + \sqrt{Z^4_{1-\alpha/2}(N-N_0)^4 + 4m^2 N_0^2 N^2 Z^2_{1-\alpha/2}(N - N_0)}}{2m^2 N_0^2}.\end{aligned} \tag{2.4}$$

3 Different methods and their corresponding results

As mentioned in the introduction, the State of Washington also developed a method to tackle the same problem, explained in the Legislature. Their method, together with the method present in the Wrr and the newly developed method are implemented in the software program R. In this chapter, the different methods will be compared by seeing what happens if we all apply them to the same situation.

3.1 The Washington procedure

In this subsection, there will be a brief discussion on how the Washington method works. The method consists of 9 points, these points are listed below. We first start by noticing that the assumption has been made that the number of pairs in the sample (X_2) is distributed as a Poisson random variable. This is needed to understand where the formula at 7. comes from.

1. Start by taking a random sample (of size n) of at least three percent of the total number of requests (N).

2. Check each signature in the sample on validity and check the number of duplicate requests. A request is invalid if it is not in proper from or if the individual who signed did not have the right to do so. The number of valid requests in the sample is denoted by n_v, where the number of invalid requests is denoted by n_i.

3. Start by determining the error allowance, this is defined as follows:

$$allowance = \sqrt{n_i} \cdot 1.5.$$

4. Now, an upper limit is determined for the number of invalid signatures in the population, which is:

$$upper\ limit\ = \frac{n_i + allowance}{\frac{n}{N}}.$$

5. Determine the maximum allowable number of pairs of signatures in the sample as follows:

$$maximum\ allowable\ number\ of\ pairs\ = N - N_0 - upper\ limit.$$

   Where N_0 is the number of signatures required by Article II, Section 1 of the Washington state Constitution. The latter can, of course, also be replaced by the number of signatures required in the Netherlands (which is 10,000 or 300,000 for respectively an introductory request or definitive request).

6. The expected number of pairs of signatures in the sample is given by:

$$\begin{aligned}\mathbb{E}(X_2) &= \left(\frac{n}{N}\right)^2 \cdot maximum\ allowable\ number\ of\ pairs \\ &= \frac{n^2(1 - \frac{N_0}{N}) - \sqrt{n_i}(\sqrt{n_i} + 1.5)}{N}.\end{aligned}$$

   Recall that it is assumed that X_2 is distributed as a Poisson random variable, this means that the value calculated above is equal to the parameter λ which is then also equal to $Var(X_2)$.

However, it appears that stating that $Var(X_2) = \lambda$ is an assumption that is not reasonable, and also the reason that this method does not produce the most reliable results. From the simulations that will be done in this chapter it appears that this method works good when η_2 is small, but when η_2 is big, it does not estimate the valid number of requests very well.

7. Finally, the acceptable number of pairs of signatures in the sample (a) is is given by:

$$a = \mathbb{E}(X_2) - 1.65 \cdot \sqrt{\mathbb{E}(X_2)}.$$

To see where the formula above comes from, we derive it below.
Start by stating that:

$$\begin{aligned}
\alpha &= \mathbb{P}(a > \mathbb{E}(X_2)) \\
&= \mathbb{P}(a > \lambda) \\
&= \mathbb{P}(2 \cdot a > 2 \cdot \lambda) \\
&= \mathbb{P}(a - \lambda > \lambda - a) \\
&= \mathbb{P}\left(\frac{a-\lambda}{\sqrt{\lambda}} > \frac{\lambda - a}{\sqrt{\lambda}}\right).
\end{aligned}$$

Using that $\frac{a-\lambda}{\sqrt{\lambda}} \sim \mathcal{N}(0,1)$:

$$\begin{aligned}
&\alpha = \mathbb{P}(Z > Z_{1-\alpha/2}) \\
&\Rightarrow Z_{1-\alpha/2} = \frac{\lambda - a}{\sqrt{\lambda}} \\
&\Rightarrow a = \lambda - Z_{1-\alpha/2}\sqrt{\lambda}.
\end{aligned}$$

Now, setting α equal to 0.10 means that $Z_{1-\alpha/2} = 1.65$. This altogether now results in:

$$\begin{aligned}
a &= \lambda - 1.65 \cdot \sqrt{\lambda} \\
&= \mathbb{E}(X_2) - 1.65 \cdot \sqrt{\mathbb{E}(X_2)}.
\end{aligned}$$

8. If the situation arises that:

   number of pairs of signatures in the sample $\geq$ *acceptable number of pairs in the sample*

   it means that there there is a possibility that there are too many invalid requests, all requests need to be checked separately on their validity to determine the exact number of valid signatures.

9. If the situation arises that:

   number of pairs of signatures in the sample $<$ *acceptable number of pairs in the sample*

   it means that there is a sufficient amount of valid requests. The conclusion that will now be drawn is that there is enough support to organize a (consulting) referendum.

3.2 Decision margin

Since the estimator is normally distributed, it has the property that half of the time its estimation is too high, and therefore the other half of the time its estimation is too low. This causes some trouble if we look at situations in which the true value lies very close to N_0, in this case it will conclude half the time that there are enough valid requests and half the time that there are not enough valid requests.

To make sure that "acceptance while there are in reality not enough valid requests" will occur as rarely as possible, we introduce a decision margin $(1+m)N_0$, where m can be chosen by the user of this method. The margin will be a number higher than N_0 and ensures that the situation above happens as rarely as desired.

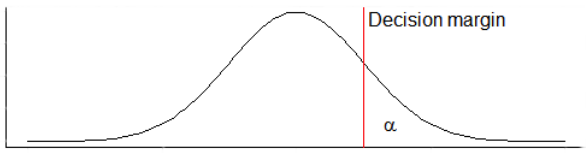


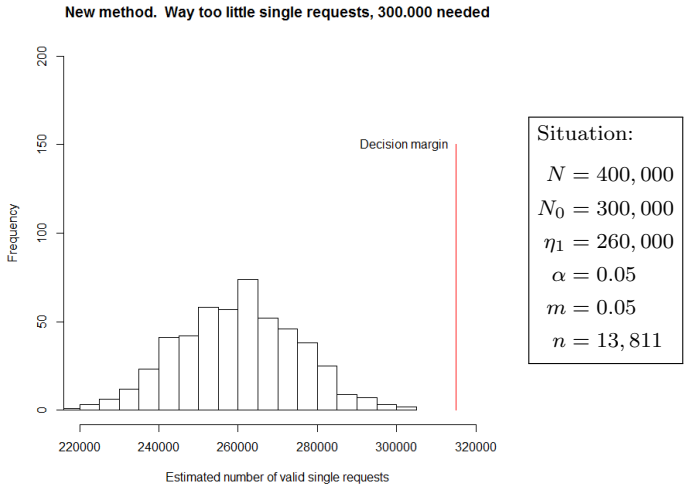
Figure 1: Visual representation of the decision margin

3.3 Results

All the preceding methods, which are: the old method, the new method and the Washington method have been implemented in the statistical programing language R (Appendix A-D). Different situations are simulated by creating a population and then using one of the methods to calculate a sample size and estimate the number of single valid requests. In this section we will look at different situations and the differences in output they produce among the different methods. The way this is done is by running the same simulation 500 times with a different random sample out of the total population. The 500 estimators are then plotted in a histogram. Here we make the side note that there is no estimator for the Washington method, for the number of single valid requests there is only an accept or reject. Therefore, for the Washington method, the percentage of accepted requests will be displayed.

3.3.1 An easy situation

Let's start with the following situation to visualize the result of a simulation:



Situation:

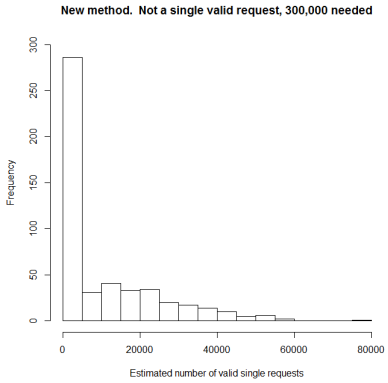
$N = 400,000$
$N_0 = 300,000$
$\eta_1 = 260,000$
$\alpha = 0.05$
$m = 0.05$
$n = 13,811$

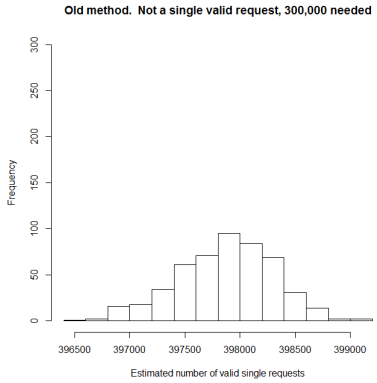
We see that indeed the method approximates the true number of single requests very accurately (in this case: 260,000) and we clearly see the bell-shaped normal curve. The latter because we know from Chapter 2 that $\hat{\eta_1}$ is normally distributed. In this case it is know that there are not enough requests and as we see, all 500 times the the estimator lies left of the decision margin, and therefore, the request would be rejected. In this situation, the Washington method would accept 100 percent

of requests, this is in line with the statement made earlier in this chapter: that the Washington method works very badly when η_2 is big.

3.3.2 The case where the old method breaks down

As discussed in the beginning of this article, there is a particular situation which very clearly indicates that there is something wrong with the old method. A simulation of this situation resulted in the following plots:



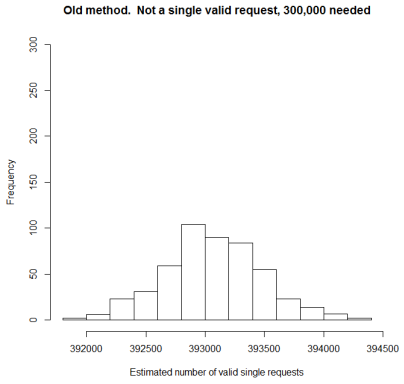


Situation:

$$N = 400,000$$
$$N_0 = 300,000$$
$$\eta_1 = 0$$
$$\eta_2 = 200,000$$
$$\alpha = 0.05$$
$$m = 0.05$$
$$n_{new} = 13,811$$
$$n_{old} = 4,105$$

These two plots clearly visualize the problem of the old method, we see that it *always* estimates the number of single submissions around approximately 398,000, where the new method would always conclude that there are way to less single requests. Notable is that the Washington method accepts 0 percent of the requests here, so in this case it works perfectly.

Observe that the new method makes use of a much bigger sample size than the old method; however, changing the sample size to 13,811 just as the new method does, results in:



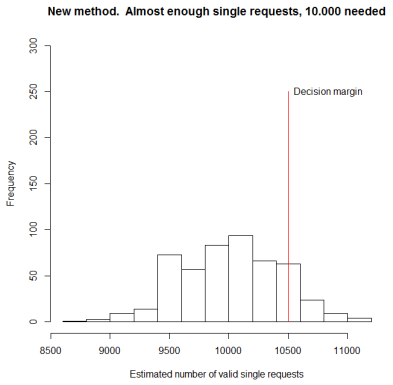
Situation:

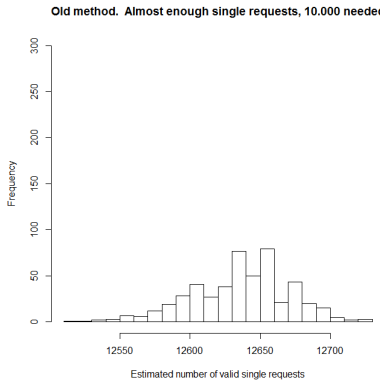
$$N = 400,000$$
$$N_0 = 300,000$$
$$\eta_1 = 0$$
$$\eta_2 = 200,000$$
$$\alpha = 0.05$$
$$m = 0.05$$
$$n = 13,811$$

Not a better result than with a sample size of 4,105.

3.3.3 A difficult situation

Next, we are going to look at the situation where we need 1 more single request to make the request valid:





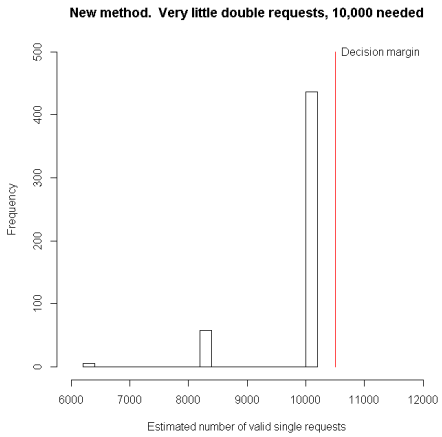
Situation:

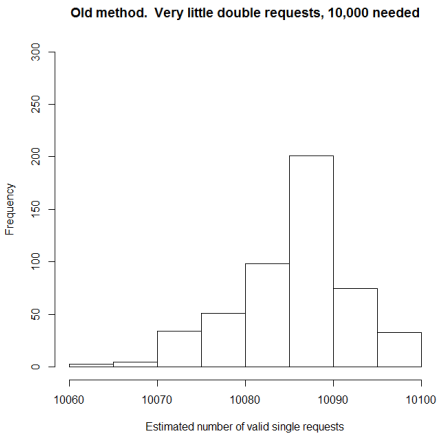
$$N = 13,000$$
$$N_0 = 10,000$$
$$\eta_1 = 9999$$
$$\alpha = 0.05$$
$$m = 0.05$$
$$n_{new} = 2,294$$
$$n_{old} = 3,144$$

This case very clearly indicates why we need to include the decision margin. As we see, without the decision margin, 50 percent of the times the method would accept the request, while in reality there are not enough requests. The latter is of course a very undesirable situation since the whole point of doing this estimation is to prevent invalid requests from passing. If we look at the old method we see that it estimates the number of single requests way too high. The Washington method rejects around 50 percent of the times the request, also not a very desirable result.

3.3.4 Very few double requests

In the example of Section 3.3.3 there were still a lot of double requests (namely: $\frac{13000-9999}{2} = 1500$). If we now look at the situation where there are only a couple of double requests, we get to the following results:





Situation:

$$N = 10,100$$
$$N_0 = 10,000$$
$$\eta_1 = 9800$$
$$\alpha = 0.05$$
$$m = 0.05$$
$$n_{new} = 332$$
$$n_{old} = 2,940$$

We see now that the number of bars in the plot is very low, only three bars. We also note that we

have a sample size of 332 requests, this explains why there are only three bars. The sample probably includes not a single double request most of the times, which results in the high right bar. About 70 times the sample includes only 1 double request, which results in the middle bar. Almost all the times there will be an estimated number of valid requests of 10,100, which would imply that there are enough valid requests, which there are not (there are only 9999 valid requests). This example again shows the importance of the decision margin, as long ad this is chosen big enough, the request will be rejected as it is supposed to be.

The old method estimates too high, and because of the lack of a decision margin, the request would have been accepted. Furthermore, it is also important to note that the old method uses a way bigger sample size than the new one, but does not produce a much better result. The Washington procedure has an acceptance percentage of 0 percent, a good result which is in line with the statement made before that when η_2 is small, the Washington procedure works good.

3.4 The impact of the sample size

Untill now, all the examples used the sample size determined by Equation (2.4). But what happens if we increase or decrease the sample size? How much does that influence the estimator?
To answer this question we are going to look at the situation from Section 3.3.3. We see that in this case, $n = 2294$. To see how many of the 500 simulated requests get accepted we are going to look at the following percentage:

$$\text{percentage of accepted requests} = \sum_{i=1}^{500} \frac{1}{500} \cdot \mathbb{I}_{\eta_{1,i} > (1+m) \cdot N_0}$$

where i is the i^{th} simulation. This is the percentage of estimators η_1 that is higher than the decision margin. In other words: this are all the bars that are right of the decision margin.

Sample size	Percentage of accepted requests
1000	38
1147	26.4
2000	13.2
2294	9.8
3000	2.8
4000	0.6
4588	0

Table 1: Different sample sizes and their corresponding percentage of accepted requests

From Table 1 we see that halving the sample size means that 16.6 percent more requests will be falsely accepted. And by doubling the sample size we do have a 0 percentage of falsely accepting a request, but here we only gain 9.8 percent while 2294 more requests need to be checked.

Choosing a sample size is, of course, not only a statistical problem, but also a matter of policy, how much effort do you want to put in checking all the requests in the sample and how certain do you want to be? But from the results of Table 1 we see that the new method does a very good job in finding a balance between accuracy and sample size.

4 Improvement of the method

Now that the new method is defined, and the simulations in Chapter 3 have shown that we have a significant increase in accuracy, it is now time to look at possibilities to improve the method. In the process of developing this method, the Dutch government came up with a slightly more specific question they want answered, which is:

A 1 percent acceptable chance on rejecting H_0 incorrectly with an error margin of 2 percent.

Formally, this means that we want to find a sample size n that satisfies:

$$\mathbb{P}(\hat{\eta_1} \geq N_0 \,|\, \eta_1 = (1-m) \cdot N_0) = \alpha \tag{4.1}$$

where: $\alpha = 0.02$ and $m = 0.01$. Recall that α is the allowed probability on an error and m is the decision margin, a boundary to make sure that we do not falsely accept an invalid request. If we look back at the examples from Chapter 3 we see that when $N \gg N_0$ this accuracy is easily reached with the new method, the sample size will then be chosen big enough to ensure this level of accuracy. Unfortunately, we run into trouble when $N \sim N_0$. Because of the fact that $\hat{\eta_1}$ is normally distributed, $\hat{\eta_1}$ lies 50 percent of the times above the true value of η_1 and 50 percent of the times below the true value of η_1. This means that we cannot ensure the accuracy that is desired in this request.

The most problematic case we can think of is when $N = N_0$, we can reason that in this case the biggest sample size is needed to ensure the foregoing accuracy. What will be done now is that a sample size is calculated so that it ensures the desired level of accuracy when $N = N_0$, since this sample size then ensures the accuracy for our *worst case* $N = N_0$, it definitely works for all other cases.

4.1 Determining the sample size

First of all, we note that if $N = N_0$, $X_1 = n$, which in turn implies that:

$$\begin{aligned}
\hat{\eta_1} &= \frac{2\pi_{2,2} + \pi_{1,2}}{2\pi_{1,1}\pi_{2,2}} \cdot X_1 - \frac{\pi_{1,2} n}{2\pi_{1,1}\pi_{2,2}} \\
&= \frac{n}{\pi_{1,1}} \\
&= \frac{n}{\frac{\binom{1}{1}\binom{N-1}{n-1}}{\binom{N}{n}} \cdot \frac{1}{1-\frac{\binom{1}{0}\binom{N-1}{n}}{\binom{N}{n}}} \cdot \frac{n}{N}} \\
&= \frac{n}{\frac{\frac{n}{N}}{1-\frac{N-n}{N}} \cdot \frac{n}{N}} \\
&= \frac{N}{n} \cdot n \\
&= N.
\end{aligned}$$

With this information in mind, we can read Equation (4.1) as: what is the probability that $X_1 = n$, *given that* $\eta_1 = (1-m) \cdot N_0$. Formally:

$$\mathbb{P}(X_1 = n \,|\, \eta_1 = (1-m) \cdot N_0) = \alpha. \tag{4.2}$$

Before we proceed, we make a visualization of the situation in the form of a number line. All the single requests (η_1 $(= (1-m)\cdot N_0)$) will be arranged on the left side of a number line and all the double requests (η_2 $(= \frac{mN_0}{2})$) on the right side. This gives the following picture:

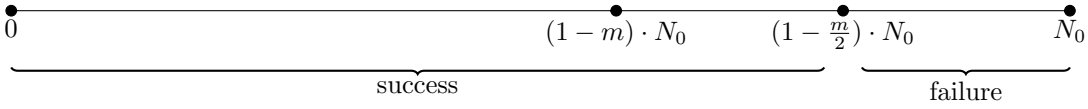


Figure 2: A visual representation of the distribution of the requests, where the single requests are positioned on the left and the double requests on the right

From Equation (4.2) we know that we are interested in the probability that we pick n single requests, in other words: n out of n successes. From Figure 2 we conclude that a there is a success when a request is taken out of the first $(1-\frac{m}{2})\cdot N_0$ requests, and a failure if a request is picked out of the last $\frac{m}{2}\cdot N_0$ requests. With this estimation procedure we arrive at the following inequality:

$$\begin{aligned}\alpha &= \mathbb{P}(X_1 = n \,|\, \eta_1 = (1-m)\cdot N_0) \\ &\leq \mathbb{P}\left(\text{n out of n successes} \mid \left(1-\frac{m}{2}\right)\cdot N_0 \text{ successes in } N = N_0\right)\end{aligned}$$

This can be calculated using the hypergeometric distribution, which means we want to solve the following equality:

$$\frac{\binom{(1-\frac{m}{2})\cdot N_0}{n}}{\binom{N_0}{n}} \leq \alpha$$

With the use of Matlab (Appendix E) , we get the following results:

$$\begin{aligned}\text{Introductory request: } &\frac{\binom{9900}{n}}{\binom{10000}{n}} \leq 0.01 \\ &\Rightarrow n \geq 448 \\ \text{Definitive request: } &\frac{\binom{29,7000}{n}}{\binom{30,0000}{n}} \leq 0.01 \\ &\Rightarrow n \geq 458\end{aligned}$$

Where these sample sizes are approximations, since Matlab is not able to calculate an analytic solution for these equations. The problem that now arises is that the conclusion arises that we have a lower bound for the sample size, which is not very informative. Also, the sample sizes are quite small compared to the sample sizes from Chapter 3. It may be a good idea to take a different approach.

4.2 The birthday problem

The birthday problem [Johnson, 1997] is a well-known problem in probability theory, where the following question needs to be answered:

Given a group of n people, what is the probability that at least two will have the same birthday?

Going back to the question we would like to answer, it can be seen that this can also be transformed into a birthday problem:

$$\begin{aligned}\alpha &= \mathbb{P}(X_1 = n \,|\, \eta_1 = (1-m)\cdot N_0) \\ &= \mathbb{P}(\text{No two people have the same birthday} \,|\, \eta_1 = (1-m)\cdot N_0).\end{aligned}$$

In this situation the total sample size is n, so there are n people in the birthday problem. Saying that $X_1 = n$ is equivalent to saying that we don't want to have a double request in our sample. This is in turn equivalent to saying that we don't want two people having the same birthday. Define:

$$\begin{aligned}p = &\text{ The probability that two randomly chosen people have the same birthday} \\ (= &\text{ two randomly chosen requests form a double request)} \\ = &\frac{\frac{mN_0}{2}}{N_0} \cdot \frac{\frac{mN_0}{2}}{N_0 - 1}\end{aligned}$$

Which implies:

$$\begin{aligned}\alpha &= \mathbb{P}(\text{No two people have the same birthday} \,|\, \eta_1 = (1-m)\cdot N_0) \\ &= (1-p)^{\binom{n}{2}} \\ &= \left(1 - \frac{mN_0}{2} \cdot \frac{1}{N_0 \cdot (N_0 - 1)}\right)^{\binom{n}{2}}\end{aligned}$$

Solving for n:

$$\begin{aligned}\binom{n}{2} &= \frac{log(\alpha)}{log(1 - \frac{mN_0}{2N_0\cdot(N_0-1)})} \\ \Rightarrow n^2 - n - &\frac{2log(\alpha)}{log(1 - \frac{mN_0}{2N_0\cdot(N_0-1)})} = 0 \\ \Rightarrow n &= \frac{1}{2} \pm \frac{\sqrt{1 + \frac{8log(\alpha)}{log\left(1 - \frac{mN_0}{2N_0(N_0-1)}\right)}}}{2} \\ \Rightarrow n &= \frac{1}{2} + \frac{\sqrt{1 + \frac{8log(\alpha)}{log\left(1 - \frac{mN_0}{2N_0(N_0-1)}\right)}}}{2} \qquad (n > 0)\end{aligned}$$

Filling in with the desired values gives:

$$\begin{aligned}\text{Introductory request: } n &= 3035 \\ \text{Definitive request: } n &= 16,623\end{aligned} \tag{4.3}$$

These two sample sizes now work respectively for all $N \geq 10,000$ and $N \geq 300,000$. The sample sizes are bigger than the sample sizes determined before (from Chapter 3 we know that for $N = 13,000$ we had $n = 2294$ and for $N = 400,000$ we had $n = 13,811$). This makes sense, since the sample sizes from Equation (4.3) are determined using the *worst case* $N = N_0$. For all cases where $N > N_0$ we can now pick the corresponding sample size and it will now always give the desired accuracy.

Concluding remarks

In this thesis, we have seen that the method used to estimate the number of valid requests for the past few years had some serious problems and could have made requests pass which should have been rejected. The new method, with a new way to define the estimated number of valid requests and the sample size shows with the use of simulations a big improvement over the old method and the Washington method. The new request from the government that required a higher accuracy has been answered with a different determined sample size. Depending on whether we want to check an introductory request or definitive request, for both situations there is now a fixed sample size that gives the desired accuracy.

Of course, assumptions were needed and deciding that there are only single and double requests present in the population makes the estimator conservative, which is also the biggest limitation of the method. The fact that the estimator is normally distributed makes situations where there are barely enough valid requests hard to judge. For the future it may therefore be interesting to see if there is a way to get rid of the conservative nature of the estimator, although there are probably other assumptions that can cause other limitations.

For now, we can be very sure that the next request for a referendum in the Netherlands can be rejected or accepted with a lot more confidence than before.

Acknowledgments

First of all, I would like to thank my first supervisor prof. Ernst C. Wit for providing me with this very interesting subject and taking the time to answer my questions with his big expertise in statistics. Because of this thesis I am now sure that I want to proceed my academic career in statistics, thank you!

I would also like to thank my good friend Luuk Wester for helping me out with programming when needed, and Mozhgan Arabpour for being available for questions when prof. Wit wasn't.

References

Besluit raadgevend referendum. http://wetten.overheid.nl/BWBR0036521/2017-04-01. Accessed: 19-11-2017.

Wet raadgevend referendum. http://wetten.overheid.nl/BWBR0036443/2015-07-01#Hoofdstuk7_Paragraaf2_Artikel44. Accessed: 19-11-2017.

George Casella and Roger L Berger. *Statistical inference*, volume 2. Duxbury Pacific Grove, CA, 2002.

John M. Johnson. The birthday problem explained. *The Mathematics Teacher*, 90(1):20–22, 1997. ISSN 00255769. URL http://www.jstor.org/stable/27970048.

Washington State Legislature. Random sampling procedure. WAC 434-379-010: District Court. WD Washington.

E.C. Wit. Advies ter aanpassing controle steekproef in de wet raadgevend referendum, 2016.

Appendix

For all the code listed below, R studio version 1.0.153 was used.

A The different situations

```
N <-     c(13000, 13000, 13000, 13000, 400000, 400000, 400000, 400000, 400000
           , 10100)
N0 <-    c(10000, 10000, 10000, 10000, 300000, 300000, 300000, 300000, 300000
           , 10000)
eta1 <- c(10000, 12000,  9999, 10100, 300000, 260000, 340000, 300100, 0
           ,  9999)
situation <- c( "Exactly enough single requests, 10.000 needed"
                ,"More than enough single requests, 10.000 needed"
                ,"Almost enough single requests, 10.000 needed"
                ,"Just enough single requests, 10.000 needed"
                ,"Exactly enough single requests, 300.000 needed"
                ,"Way too little single requests, 300.000 needed"
                ,"More than enough single requests, 300.000 needed"
                ,"Just enough single requests, 300.000 needed"
                ,"Not a single valid requests, 300,000 needed"
                ,"Very few double requests, 10,000 needed")

m       <- 0.05
n.iter <- 500
alpha1 <- 0.10
alpha  <- alpha1 / 2
n.sim  <- length(N)
n       <- NULL
```

B New method

```
vr <- function(sample, totaal.requests){
  sample <- rank(sort(sample))
  prob <- function(i, j, n, N){
    n * dhyper(i, j, N-j, n) * j / N / (1 - dhyper(0, j, N-j, n))
  }
  N <- totaal.requests
  n <- length(sample)
  X <- tabulate(tabulate(sample))
  m <- length(X)
  if (m == 1){
    X.pseudo <- c(X, 0)
  } else {
    X.pseudo <- c(X[1], sum(X[2:m] * (2:m) / 2))
  }
  X = X.pseudo
  m = 2
  eta <- rep(NA, 2)
  eta[2] <- X[2] / prob(m, m, n, N)
  eta[1] <- (X[1]- prob(1, 2, n, N) * eta[2]) / prob(1, 1, n, N)
  return(list(observed=X, estimates=round(eta, 1), valid=max(eta[1], 0)))
}

size <- function(N, N0, m, alpha){
  za <- qnorm(1 - alpha)
  a <- m^2 * N0^2
  b <- za^2 * (N - N0)^2
  c <- -N^2 * (N - N0)* za^2
  n <- c(n, round((-b + sqrt(b^2 - 4 * a * c)) / (2 * a)))
  return(n)
}

new_method <- function() {
  res <- NULL
  for (i in 1:n.sim) {
    # make population
    Pop <- c((1:eta1[i]), eta1[i] + rep(1:((N[i] - eta1[i]) / 2), each=2))
    # determine sample size
    n <- c(n, size(N[i], N0[i], m, alpha))  # add new size to n
    x <- NULL
    est <- NULL
    est2 <- NULL
    for (j in 1:n.iter) {
      smpl <- sample(Pop, n[i], replace=FALSE)
      sim.vr <- vr(smpl, N[i])  # find the number of valid reqeusts
      x <- cbind(x, sim.vr$observed[1])
      est <- cbind(est, sim.vr$estimates[1])  # add sim.vv$estimates[1] to est
      est2 <- cbind(est2, sim.vr$valid)  # add sim.vv$valid toe aan est2
    }
    # save results in res
    res[[i]] <- list(N.N1.N0.n=c(N[i], eta1[i], N0[i], n[i]),
                     x=round(c(mean(x), sd(x)), 1),
                     est.all=est2,
                     est=round(c(mean(est), sd(est)), 1),
                     est2=round(c(mean(est2), sd(est2)), 1),
                     prob=sum(est > (1 + m) * N0[i]) / n.iter)
  }
  return(res);
}
```

```
new_res  = new_method()

#code for histograms

dec_new <- NULL
for (i in 1: 10) { #change situation here
        dev.new()
        hist(new_res[[i]]$est.all, breaks=15, xlab="Estimated number of valid single
         requests", main=paste("New method. ", situation[i]), ylim=c(0, 500)
         , xlim=c(6000,12000))
        dec_new[i] <- (1 + m) * N0[i]

        lines(c(dec_new[i], dec_new[i]), c(0,500), col="red")
        text(bdec_new[i], 500, "Decision margin", pos=4)
}
```

C Old method

```
r_old_method <- function() {
  res <- NULL
  for (i in 1: n.sim) {
    # make population
    Pop <- c((1: eta1[i]), eta1[i] + rep(1: ((N[i] - eta1[i]) / 2), each=2))
    # determine sample size
    n = (N * 2.576^2 * 0.25) / (2.576^2 * 0.5 * 0.5 + (N - 1) * 0.02^2)
    x <- NULL
    est <- NULL
    est2 <- NULL
    for (j in 1: n.iter) {
      smpl <- sample(Pop, n[i], replace=FALSE)
      x <- cbind(x, length(unique(smpl)))
      est[j] <- x[j] * N[i] / n[i]
    }
    # save results in res
    res[[i]] = list(N.N1.N0.n=c(N[i], eta1[i], N0[i], n[i]), est = est)
  }
  return(res);
}

r_old_res  = r_old_method()

#code for histograms

dec_old <- NULL
for (i in 1: 10) { #change situation here
        dev.new()
        hist(r_old_res[[i]], breaks=15, xlab="Estimated number of valid single
        requests", main=paste("Old method. ", situation[i]), ylim=c(0, 300))
        dec_old[i] <- (1 + m) * N0[i] #decision margin

        lines(c(dec_old[i], dec_old[i]), c(0,260), col="red")
        text(dec_old[i], 270, "Decision margin", pos=4)
}
```

D The Washington method

```
for (i in 1: 10) {           #Change the situation here
  number_accepted = 0
  n <- N[i] * 0.06
  eta2 <- (N[i] - eta1[i]) / 2
  Pop <- c((1: eta1[i]), eta1[i] + rep(1: eta2, each=2))
  for (j in 1: n.iter) {
    washsample <- sample(Pop, n, replace=FALSE)
    X <- tabulate(tabulate(washsample))

    max.allowed.pairs <- N[i] - N0[i]
    expected.pairs <- (n / N[i])^2 * max.allowed.pairs
    acceptable.pairs <- expected.pairs - 1.65 * sqrt(expected.pairs)

    if (length(X) == 1) {
      X = c(X, 0)
    }
    if (X[2] < acceptable.pairs) {
      number_accepted <- number_accepted + 1
    } else {
      number_accepted <- number_accepted
    }
  }

  percentage_accepted <-number_accepted / n.iter * 100
  print(percentage_accepted)
}
```

E Calculating the sample size

For the code below, Matlab R2017a - EDU was used.

```
syms n
b = nchoosek(9900,  n)
c = nchoosek(10000, n)

eqn = b/c == 0.01;
soln = solve(eqn,n)
```

```
syms n
b = nchoosek(297000, n)
c = nchoosek(300000, n)

eqn = b/c == 0.01;
soln = solve(eqn,n)
```