Exercise 1

Let $H$ be a set with $n$ elements and $f : 2^H \rightarrow \mathbb{R}$ a function that maps subsets of $H$ to real numbers. We say that $h \in H$ violates $G \subseteq H$ if $f(G \cup \{h\}) \neq f(G)$ (it follows that $h \not\in G$). We also say that $h \in H$ is extreme in $G$ if $f(G \setminus \{h\}) \neq f(G)$ (it follows that $h \in G$).

Now we define two random variables $V_r, X_r : \binom{H}{r} \rightarrow \mathbb{R}$ where $V_r$ maps an $r$-element set $R$ to the number of elements that violate $R$, and $X_r$ maps an $r$-element set $R$ to the number of extreme elements in $R$.

Prove the following equality for $0 \leq r < n$:

$$\frac{E(V_r)}{n - r} = \frac{E(X_{r+1})}{r + 1}.$$ 

Exercise 2

Imagine instead of doubling the ballots of the unhappy house owners in the Swiss Algorithm, we would multiply their number by some integer $t \in \mathbb{N}$. Does the analysis of the algorithm improve (i.e., does one get a better bound on the expected number of rounds, following the same approach)?

Exercise 3

We have shown that for $d = 2$ and sample size $r = 13$, the Swiss algorithm takes an expected number of $O(\log n)$ rounds. Compute the constants, i.e., find numbers $c_1, c_2$ such that the expected number of rounds is always bounded by $c_1 \log_2 n + c_2$. Try to make $c_1$ as small as possible.