Los límites de la predictabildiad electoral.

Gonzalo Barrera Borla

10 de marzo de 2014

Resumen

A partir del estudio de los resultados de las elecciones legislativas de Octubre 2013, pretendemos dar una idea empírica de los considerables márgenes de error inherentes en cualquier predicción electoral. Dado que se cuenta con datos para el total de la población, las tradicionales y complejas ecuaciones para estimar el error de *una* muestra son reemplazadas por simulaciones que lo computan para *decenas de miles* de muestra aleatorias. El estudio del error en muestras generadas de manera perfectamente aleatoria, sirve no sólo para entender el problema en abstracto, sino como cota superior para el nivel de precisión que pueden alcanzar muestreos realizados en la vida real. Discutiremos las condiciones bajo las cuales los resultados expuestos se sostienen, y las implicancias prácticas de estos.

Índice

1. Nota al lector

En este ensayo se *analiza* el margen de error inherente a la elaboración de toda hipótesi estadística, pero no *utiliza* para ello el herramental tradicional del test de hipótesis - ni tiene por qué hacerlo, de manera similar a la que para estudiar la fabricación de un automóvil no hace falta *ensamblar* uno, sino que basta con *observar* el ensamblado de varios.

Como consecuencia, y para alegría del - probablemente inexistente - lector no académico, se ha reducido a un mínimo el uso de términos técnicos. En su lugar, recurriremos principalmente al uso de gráficos, que como medio de transmisión de información suelen ser mucho más amenos e inmediatamente comprensibles que modelos y tecnicismos insípidos.

Por lo tanto, se recomienda tener los gráficos lado a lado con el cuerpo de texto, como si esto fuese una presentación narrada. ¡Inténtelo!

2. Introducción

2.1. El fenómeno

Supongamos que queremos estimar la distribución del parámetro ϕ en la población P, con precisión α. Para ello, habremos de extraer una muestra de P de tamaño n. Paradójicamente, para estimar qué n es lo suficientemente grande como para garantizar precisión α, es necesario conocer la distribucion de ϕ en P - ¡exactamente lo que prentendíamos estimar en primer lugar!.

Romper este círculo vicioso es sencillamente imposible en la enorme mayoría de los experimentos tradicionales, ya que requeriría examinar la población entera, y n suele ser varios ordenes de magnitud menor que P. En consecuencia, el n utilizado en un estudio científico o una encuesta por sí solo *no* es un buen indicador de la precisión de los resultados. La única forma segura de escapar la trampa - analizar la población entera - casi nunca es factible.

Los resultados publicados por la Dirección Nacional Electoral en el portal de Datos Públicos del Gobierno de la Nación, nos proveen de una bienvenida excepción a la regla. Entre otros, se encuentran disponibles digitalmente las cantidades de votos registrados en cada una de las mesas de la Capital Federa. Si P es ahora la población de votantes de la ciudad, y ϕ representa el partido por el cual votó cada uno, podemos estimar qué precisión en la estimación de ϕ nos garantiza un determinado n.

2.2. Estrategia de análisis

¿Qué confianza podemos tener en el estimador de ϕ derivado de una muestra de tamaño n? es la pregunta que guiará nuestro trabajo. Obviamente, conocer cómo votó la

población completa facilita este análisis, pero bajo ningún oncepto lo trivializa. Computar el error de estimación *para toda muestra posible* de n votantes se vuelve rápidamente una tarea hercúlea: con más de 1.800.000 votos emitidos, la cantidad de muestras con un mero $n = 10$ es $9{,}83 \times 10^{55}$, un 9 seguido de 55 ceros.

Imposibilitados de calcular *todas* las posibles muestras, sí podemos sin embargo simular enormes cantidades de ellas. Para hacerlo, se programaron en Ruby dos simuladores, que intentar emular las condiciones de los dos estimadores más comunes del resultado final de la eleccion: el recuento provisorio, y las encuestas pre-electorales.

El "simulador de la noche del escrutinio", primero aleatoriza el orden de recuento de los telegramas de cada mesa, y luego computa la evolución del resultado provisorio a medida que avanza el porcentaje de mesas escrutadas. Con él, podemos observar como la adición de nueva informaciø'on a un pronóstico preexistente lo va mejorando, y a qué ritmo.

El "simulador de muestras aleatorias"nos permite extraer de la población total muestras perfectamente aleatorias de tamaño n, respetando en los pesos el tamaño de cada circuito y sección electoral). Con él, podremos estudiar de manera directa la *población de muestras* del tamaño deseado, algo que el investigador con una única muestra a su disposición no puede realizar.

A partir de los datos generados por estos dos simuladores, se intentará dar una idea de la interacción entre el tamaño muestral y el error del estimador asociado, visualizando la informacion de distintas formas y extrayendo algunos - muy pocos - resultados numéricos.

Discutiremos luego las implicancias prácticas de los resultados observados, y estableceremos ciertos supuestos bajo los cuales éstos se pueden generalizar -cuidadosamente - a situaciones similares.

3. Diseño Experimental

3.1. Características del dataset

La Dirección Nacional Electoral, a través del Portal de Datos Públicos del Gobierno de la Nación, nos provee de toda la información imaginable sobre los más recientes resultados electorales. Para realizar este trabajo, descargamos el archivo 'electoral-2013-diputados-nacionales.csv' que contiene las cantidades de votos registrados en todas las mesas del país para la elección de diputados nacionales. Estas son las primeras de las más de 22 millones de líneas del archivo:

codigo_provincia	codigo_departamento	codigo_circuito	codigo_mesa	codigo_votos	votos
1	1	1	1	9001	351
1	1	1	1	9002	0
1	1	1	1	9003	0
1	1	1	1	9004	0
1	1	1	1	9005	0
1	1	1	1	9006	0
1	1	1	1	187	8
1	1	1	1	501	64
1	1	1	1	502	58
1	1	1	1	503	78
1	1	1	1	505	26
1	1	1	1	506	7

...

Para acotar el universo de análisis a una única carrera electoral, tomamos solamente los datos de las mesas de Capital Federal, es decir, aquellas cuyo codigo_provincia es '1'.

codigo_departamento, codigo_circuito y codigo_mesa, en orden de especificidad creciente, son los identificadores de toda mesa. Para Capital Federal, por ejemplo, los códigos de departamento coinciden con los números de Comuna (una mesa con codigo_departamente = 14 estará en la Comuna 14). A su vez, las comunas están divididas en un total de 167 circuitos, en los cuales hay 7342 mesas.

Por alguna razón, sólo hay datos disponibles para 7263 de las 7342 mesas, lo que pareciera decir que unos 79 telegramas nunca llegaron al centro de cómputos electoral.

Los códigos de votos del 9001 al 9006 indican varios datos como la cantidad de electores inscriptos en la mesa, de votos en blanco, nulos y recurridos. Como el resultado final de la elección *no* depende de ninguno de ellos, sólo nos concentramos en la cantidad de votos afirmativos registrados para los restantes seis 'codigo_votos'. Estos representan a

cada uno de los seis partidos que participaron en la elección de diputados nacionales por la Capital Federal:

codigo_votos	partido
187	Autodeterminación y Libertad
501	Frente Para la Victoria
502	UNEN
503	Union PRO
505	Frente de Izquierda y de los Trabajadores
506	Camino Popular

3.2. Transformación de los datos

Para manipular más fácilmente la información, con la ayuda de un poco de código en SQL (ver Anexo II) se transformó la tabla en una equivalente, donde cada mesa está representada por una única línea:

mesa	AYL	FPV	UNEN	PRO	FIT	CP
1	8	64	58	78	26	7
2	11	53	71	70	18	6
3	14	79	61	72	12	7
4	6	79	63	75	16	5
5	7	74	51	65	12	5
7	12	65	63	70	21	9

En este punto fueron eliminadas del conjunto unas 7 mesas que indicaban un total de cero votos afirmativos, dejando 7256 mesas.

Sumando los resultados por mesa a nivel sección y circuito, se confeccionaron tablas con formato idéntico a esta última, pero donde cada línea representa la cantidad de votos por partido a nivel de secciones y circuitos, respectivamente.

3.3. Diseño de los simuladores

En esta secciói̇n explicaremos brevemente y sin tecnicismos los pasos que sigue nuestro programa para correr una simulación y generar sus resultados. Si se quiere conocer en detalle las .^entrañas"de los simuladores, recomendamos consultar directamente el código fuente, que se encuentra en su totalidad en el Anexo II. Para facilitar la lectura, se proveen comentarios detallados acerca de su funcionamiento.

3.3.1. Simulador de la noche del escrutinio

A las 18hs del día de la elección, las autoridades de mesa dan por terminado el comicio y comienzan a contar los votos. A medida que terminan dicha tarea, le entregan a personal de Correo Argentino las urnas y un telegrama por mesa con el conteo de sus votos. Una vez que toda las mesas de una escuela fueron cerradas y los recuentos terminados, los telegramas son enviados al centro de cómputos electoral, donde un pequenõ ejército de tipeadores de datos digitaliza a mano los números contenidos en cada telegrama.

Los telegramas son cargados dos veces por tipeadores distintos (que no ven lo que cargó el otro), y si lo ingresado no coincide en alguno de los campos, un tercero desempata.

Tenemos entonces dos fuentes de aleatoriedad en el orden de escrutinio de los votos: primero, en función de qué escuelas terminan más rápido el conteo; segundo, porque en el centro de cómputos la carga de datos es realizada simultáneamente por numerosas personas, y con doble o triple chequeo.

Nuestro supuesto simplificador entonces, será que en el fondo, el orden en el que los resultados de las 7256 mesas se contabilizan la noche del escrutinio, es perfectamente aleatorio. En la realidad, dicho orden es mayormente aleatorio por las razones que acabamos de mencionar, pero no *completamente* aleatorio.

En una simulación dada, entonces, nuestro programa ejecutará los siguientes pasos:

1. Ordenar al azar las 7256 mesas (simulando la carga de los telegramas al sistema).

2. Reemplazar los votos de la mesa en la enésima posición, por la suma de los votos de las mesas hasta dicha posición inclusive (simulando el conteo provisorio hasta el momento).

3. Transformar a cada paso las sumas parciales de votos que recibió cada partido, en los porcentajes correspondientes.

Cada simulación producirá, finalmente, una matriz de 7256×6, donde el elemento en la posición (i, j) es el porcentaje de votos recibidos por el partido 'j' cuando se llevan contabilizadas 'i' mesas. La última fila de la matriz será idéntica en todas las simulaciones y coincidirá necesariamente con el verdadero porcentaje de los votos que recibió cada partido, uan vez escrutada la totalidad de las mesas.

3.3.2. Simulador de muestras aleatorias

Este simulador se puede pensar como una función que toma dos parámetros: un tamaño muestral n y un nivel de agregación geográfica g, donde

- n puede ser cualquier numero positivo entre 1 y 1.800.000 (la cantidad de votos afirmativos); y

- g puede ser 'mesas', 'circuitos' o 'secciones'

A pesar de tomar muestras aleatoias dentro de cada mesa, curcuito o sección, el simulador mantiene siempre la representatividad greográfica de los agregados. EN nuestro análisis, las muestras serán generadas siempre al nivel de los 167 circuitos.

Con n y g determinados, el simulador determinará cuántos votos tomar de cada g, para que el tamaño esperado de la muestra sea n. Siendo p_i la población del agregado i y P la población total, la cantidad de votos m_i a tomar de dicho agregado será:

$$m_i = \frac{p_i}{P} \times n$$

Si m_i no resultase entero, se tomarán aleatoriamente el piso o techo de m_i votos de manera que a la larga, en promedio, se estén tomando exactamente m_i votos de dicho agregado.

Por ejemplo, si hubiese sólo dos mesas con 34 y 66 votos cada una, y se especificara $n = 10$, habrá que tomar 3,4 votos de la primer mesa, y 6,6 de la segunda. Para ello, de la primer mesa se tomarán 3 votos en el %60 de los casos, y 4 votos el %40 restante. De la segunda mesa, se tomaran 6 votos el 40 % de los casos, y 7 el %60 restante.

Luego, se calculan los pesos asociados a cada elemento de la muestra, dependiendo de qué agregado vengan, como la razón entre la cantidad total de votos en el agregado y la cantidad de votos del agregado en la muestra. En el ejemplo anterior, si tomamos 3 votos de la mesa de 34, cada voto contará por $34/3 = 11,333...$ votos, pero si tomamos 4, cada uno contará por $34/4 = 8,5$.

Una vez hechos todos los cálculos, se extraen al azar de cada agregado la cantidad de votos previamente determinada, se los pondera por su peso asociado, y con todos ellos juntos se construye la muestra final.

Por el elemento aleatorio que utilizamos para volver entera la cantidad de votos tomados de cada agregado, el tamaño final de las muestras puede no ser exactamente n sino un poco mayor o menor. Sin embargo, esto es irrelevante, pues al calcular los pesos de cada observación en la muestra, el peso final de cada agregado es exactamente proporcional al tamaño de su población.

El producto de una simulación dada entonces, será un vector de 6 componentes, que representa las cantidades de votos que cada partido hubiese obtenido en la elección si la población entera votase exactamente igual que la muestra elegida.

3.3.3. Generación de datos experimentales.

Simulador del escrutinio Con el simulador del escrutinio, se generaron mil (1.000) posibles ordenes de carga de los telegramas, y para cada uno de ellos se computó la evolución de los porcentajes de votos recibidos por cada candidato a medida que se escrutaban cada

una de las 7256 mesas. Estas son las primeras y últimas 10 líneas de la matriz generada por una simulación cualquiera:

n	AYL	FPV	UNEN	PRO	FIT	CP
1	1.61	9.64	38.96	46.18	2.41	1.20
2	1.21	19.56	32.26	41.53	4.03	1.41
3	2.02	18.46	31.67	39.76	6.20	1.89
4	2.29	23.36	28.63	36.88	6.76	2.09
5	2.69	24.41	28.98	35.35	6.45	2.12
6	2.68	24.82	28.09	35.79	6.49	2.14
7	2.82	25.50	26.66	36.85	6.33	1.84
8	2.97	25.13	29.87	33.37	6.73	1.93
9	3.11	24.94	30.62	32.86	6.45	2.01
10	3.55	24.49	31.02	32.45	6.45	2.04
...	...	...	...	...	...	...
7247	3.78	21.59	32.23	34.46	5.65	2.29
7248	3.78	21.59	32.23	34.46	5.65	2.29
7249	3.78	21.59	32.23	34.46	5.65	2.29
7250	3.78	21.59	32.23	34.46	5.65	2.29
7251	3.78	21.59	32.23	34.46	5.65	2.29
7252	3.78	21.59	32.23	34.46	5.65	2.29
7253	3.79	21.59	32.23	34.46	5.65	2.29
7254	3.79	21.59	32.23	34.46	5.65	2.29
7255	3.79	21.59	32.23	34.46	5.65	2.29
7256	3.79	21.59	32.23	34.46	5.65	2.29

A simple vista se puede observar que los prorcentajes varían considerablemente al comienzo, para estabilizarse alrededor de sus valores reales hacia el final del escrutinio, aunque todavía se pueden observar pequeñas variaciones centesimales como la de AYL al contabilizar la mesa número 7253.

Simulador de muestras Con el simulador de muestras, se consideraron 21 tamaños muestrales distintos, creciendo a razón geométrica entre $n = 10$ hasta $n = 100{,}000$. El íesimo n está dado por la ecuación:

$$n_i = 10^{1+4\times\frac{i}{20}}$$

... de modo que $n_0 = 10^1 = 10$ y $n_{20} = 10^5 = 100{,}000$. Se generaron luego 10.000 muestras aleatorias para cada n, totalizando unas doscientas diez mil muestras (210.000). A continuación se observan los porcentajes de votos en muestras tomadas al azar de 10, 100, 1.000 y 10.000 elementos.

n	AYL	FPV	UNEN	PRO	FIT	CP
	6.20	51.25	15.61	26.94	0.00	0.00
	11.87	33.25	6.06	41.97	6.85	0.00
10	0.00	40.18	20.81	33.77	5.24	0.00
	10.17	15.33	45.15	24.92	0.00	4.43
	7.82	20.31	35.83	28.61	7.42	0.00
	4.12	16.45	31.01	44.04	2.75	1.63
	4.86	17.44	47.44	24.74	4.37	1.15
100	1.33	22.48	34.50	31.36	6.67	3.65
	4.26	19.61	32.84	35.99	5.11	2.19
	3.09	16.75	32.52	35.92	9.45	2.27
	2.99	21.38	30.31	37.73	5.07	2.52
	4.37	20.79	31.78	35.27	6.07	1.73
1.000	3.77	21.08	31.81	35.65	5.74	1.95
	3.36	23.29	33.98	32.26	4.77	2.35
	3.91	21.02	32.51	34.51	6.10	1.95
	3.75	21.11	33.06	34.66	5.25	2.18
	3.82	21.48	32.85	34.16	5.56	2.13
10.000	3.79	21.42	31.93	35.24	5.42	2.20
	4.14	21.66	31.39	35.02	5.76	2.04
	3.85	21.39	32.52	34.09	5.95	2.20

Aquí se observa claramente que a medida que el tamaño muestral considerado aumenta, los porcentajes de votos por candidato se estabilizan alrededor de los valores reales, cor márgenes de error continuamente decrecientes.

4. Desarrollo

4.1. Sobre la medición del error de predicción

A partir de una muestra cualquiera, es un hecho bien conocido que la frecuencia relativa de ocurrencia de un fenómeno es un estimador insesgado de su verdadera frecuencia relativa poblacional. A la hora de estimar porcentajes de votos, esto significa que de 100 personas 15 dicen votar a Fulano, lo más probable es que el 15 % de la población vote a Fulano. Mucho menos trivial sin embargo, a la hora de evaluar el desempeõ de una predicción, es la medición de su error.

La primer fórmula que viene a la mente, aunque sólo sea por su ubicuidad, es la suma del cuadrado de los desvíos. Siendo e_i el porcentaje de votos estimado para el candidato i, y r_i el porcentaje real de votos obtenido, podemos definir el error del estimador como:

$$E(e|r) = \sum_{i=0}^{n} (e_i - r_i)^2$$

Un problema no trivial con ella, es que al elevar al cuadrado, las unidades en las que está expresado el error no guardan ninguna relación directa con las unidades en la que está expresada la predicción: puntos porcentuales.

A su vez, aunque esta fórmula la hemos usado cientos de veces para medir la varianza de una distribución en clases de Estadística, siempre la utilizamos para medir la varianza de *un* parámetro. Sin embargo, al estimar los porcentajes de votos para c candidatos, estamos en efecto produciendo $c - 1$ estimadores: uno para cada uno de los candidatos, salvo el último que forzosamente dará cien menos la suma de todos los demás.

Al entrar en territorio tan distinto al del estudio tradicional, no tenemos a priori razones para suponer que otras fórmulas de medición del error de un estimador no son al menos tan buenas como ésta. Una forma interesante de medir el error, por ejemplo, sería calculando por D'Hont cuántas bancas le corresponden a cada partido, y observando luego por cuántas bancas dicha predicción difiere de la realidad.

Para mantener el análisis lo más sencillo posible, en este trabajo elegimos una fórmula similar a la original: la suma de los desvíos absolutos.

$$E(e|r) = \sum_{i=0}^{n} |e_i - r_i|$$

Esta sencilla fórmula, tiene una gran ventaja a la hora de interpretar sus valores: está en las mismas unidades que el estimador analizado. Así, representa crudamente "por cuánto le erró"la predicción global al resultado real. Si dos candidatos A y B obtuviesen el 30 y 70 % de los votos cada uno, y una estimación los pusiera en 35 y 65 %, su *error absoluto* sería de 10 puntos. De hecho, 10 % es exactamente el porcentaje de los votos totales que el estimador predijo incorrectamente: un 5 % que suponíamos iba a votar a A y no lo hizo, y otro 5 % que creíamos no iba a votar a B, pero lo votó.

Mas allá de la discusión teórica, vale remarcar que la utilidad de estimar el error para nuestros propósitos, no radica tanto en estimar el error *per se*, sino en la observación de su evolución: a medida que se reduce la variación del error, aumenta la estabilidad del pronóstico, y con ella la confianza que nosotros deberíamos tenerle.

4.2. ¿Cuándo es hora de felicitar a los ganadores?

Obviamente es necesario esperar hasta el final de la elección para conocer el resultado exacto, y dada la particularidades del sistema D'Hont para asignar bancas, a veces diferencias porcentuales muy pequeñas pueden mantener en vilo a los candidatos. Aunque no podamos estimar exactamente nunca este tipo de incertidumbres, sí podemos al menos

intentar identificar tendencias. Veamos por ejemplo cómo les fue a los principales contendientes, UNEN y el PRO en nuestras mil simulaciones. Dirija Ud. lector su atención ahora al **Gráfico I: Áreas de incertidumbre para los porcentajes de votos obtenidos por UNEN y PRO**.

(Si aún no está leyendo este trabajo con los gráficos al lado, es un buen momento para comenzar. Si no, todo el palabrerío le va a resultar muy confuso.)

En el eje x se encuentra representada la cantidad de mesas escrutadas en un momento dado en escala logarítmica: dadu que cada nueva mesa agrega progresivamente menos información nueva al pronóstico, es de esperar que éste mejore mucho y rápido al comienzo, y poco y lento sobre el final. Si la escala fuese lineal, toda la información útil se comprimiría sobre el eje de las y.

Las áreas rojas y azules representan respectivamente, el espacio donde se encuentran los votos de UNEN y PRO en el 90 % (con dos colas de 5 %, por debajo del percentil 5 y encima del percentil 95) de las simulaciones corridas, mientras que la línea sólida intermedia es la mediana. Finalmente, las líneas punteadas indican los resultados finales.

Esto quiere decir, por ejemplo, que con 100 mesas escrutadas, en el 90 % de los escrutinios simulados UNEN recibía entre el 31 % y 33,25 % de los votos. A esa misma altura, 90 % de las simulaciones (aunque no necesariamente *las mismas* 90 %) ponían al PRO con más del 33,5 % y menos del 35,5 % de los votos.

Con esos números, sería más que razonable concederle una futura victoria casi segura al PRO. De hecho, unas pocas mesas antes podemos identificar un hecho que nos permite contabilizar la probabilidad de esa tendencia: alrededor de las 85 mesas, se cruzan las curvas del percentil 5 del PRO con la del percentil 95 de UNEN. En otras palabras, a esa altura el 95 % de las simulaciones le daban *al menos* 33,3 % al PRO, y un 95 % le daba *a lo sumo* un 33.3 % a UNEN. El cálculo no es trivial, pero podemos aproximar las chances de ganar de UNEN pensando que los únicos casos en los que todavía tiene chances con esos números, es si a su vez UNEN está por encima de su percentil 95, y PRO por debajo de su percentil 5. Un rápido cálculo estima esa probabilidad en $0{,}5^2 = 0{,}0025$, o 400 a 1 en contra, asumiendo que ambos hechos son independientes entre sí (téngase en cuenta que todavía no ha habido ni cerca de esa cantidad de elecciones nacionales en la historia del país).

Obviamente, los votos por partido no son independientes entre sí, ya que a más votos a un partido, menos para los demás: cada voto para el PRO es un voto que no recibe UNEN, disminuyendo sus posibilidades de aparecer como momentáneo ganador en el escrutinio cuando ya se han contabilizado al menos 85 mesas. Todo suena perfectamente plausible hasta aquí, salvo por un pequeño detalle: 85 mesas escrutadas esquivale a un mero 1,17 % de las 7256 totales. No es ridículamente temprano para afirmar tendencias en una elección tan reñida? La respuesta, es sí pero no tanto.

1,17 % de las mesas escrutadas es un número muy pequeño, pero no debemos olvidar

los supuestos de los que partimos: al suponer *perfectamente* aleatorio el orden del escrutinio, eliminamos la muy real posibilidad de que los votos vengan "de a tandas". Mapeos de estos mismos resultados electorales han mostrado que en general UNEN fue ganador claro en el centro de la ciudad, el PRO en el corredor norte, y algunos reductos del sur fueron para el FPV. Si por alguna razón, los votos de una misma comuna o secciones adyacentes se cargasen más o menos seguidos, deberíamos esperar mucho más ruido y variación en las tendencias iniciales.

Una forma de medir empíricamente cúanto más lenta es la convergencia hacia los resultados finales en la realidad que en nuestro modelo, es recurriendo al archivo. La noche del escrutinio, cuando el ministro Randazzo anunció por primera vez los resultados provisorios, después de la tradicional admonició sobre cómo éstos pueden variar durante la noche, dio los siguientes números: 9,87 % de las mesas escrutadas, PRO 34,15 %, UNEN 29,95 %, FPV 24,19 %

En nuestras simulaciones, a las 700 mesas escrutadas (9,64 %), en el 95 % de los casos UNEN tenía más del 31,75 % de los votos, y el 95 % de las ocasiones el PRO recibía *al menos* el resultado anunciado por el ministro. Claramente nuestro modelo y la realidad tienen grandes diferencias. Podemos concluir con seguridad que el orden de carga de los telegramas está lejos de ser perfectamente aleatorio.

Una forma de considerar esta correlación en el orden de carga de las mesas, en el futuro, podría ser aleatorizar .ª medias": si se acaba de carga la mesa x, la próxima mesa en cargarse sera elegida al azar la mitad de las veces, y la otra mitad será la $n + 1$, que casi siempre esta geográficamente muy cerca.

Todavía nos queda por resolver sin embargo cuál es la verdadera causa de la divergencia entre la simulaciones y la realidad. Los resultados generados pueden resultar antiintuitivos porque 85 mesas parecen ser muy pocas mesas, pero si recordamos que cada mesa tiene unos 250 votos, estamos hablando de una muestra de más de 21.000 votos, número que hace babear a cualquier sociólogo. El modelo no inventa los resultados, pero hay algo de la realidad que no çaza".

Lamentablemente, hay prácticamente infinitas explicaciones posibles, desde que el centro de cómputos decida cargar los telegramas por circuito o comuna para mantener un orden, pasando por que los fiscales de los barrios céntricos son más lentos que sus contrapartes de barrios sureños, hasta la más perniciosas, donde explícitamente se cargan primero ciertos circuitos para disminuir la diferencia parcial entre el partido oficialista y los dos primeros (aunque en el anuncio de Randazzo UNEN le lleva menos de 5 puntos al FPV, finalmente la diferencia fue de más de 11). Lo único que podemos decir con seguridad, es que el orden del escrutinio evidentemente está lejos de ser perfectamente aleatorio, y por lo tanto las tendencias se formarán más -bastante más - lentamente que en nuestro modelo.

4.3. A falta de confianza, estabilidad

Supongamos que uno fuese candidato a diputado en estas elecciones, y con el 50 % de las mesas escrutadas tenemos el 11 % de los votos, porcentaje que nos alcanza para obtener una banca en el Congreso. Sabiendo que si ese número baja del 10 %, nos quedamos afuera, ¿qué confianza deberíamos tener en terminar por arriba el 10 %? La respuesta honesta a dicha pregunta debería ser un rotundo "ni idea". Todo depende de que se hayan cargado o no las mesas más favorables a nuestra causa, cosa que sólo podemos distinguir si a la vez revisamos constantemente la página de resultados provisorios y sabemos con exactitud cuáles son esos los circuitos en que mejor nos va.

Una tarea un poco más sencilla que la de estimar la confianza que debemos tener en el resultado parcial, es la de estimar su estabilidad: mucho más probable es que terminemos por encima del 10 % si venimos por encima "hace muchas mesas"que si recién lo cruzamos con un salto fortuito. Así considerado, podemos aproximar una respuesta a este problema observando la *variación temporal del error absoluto*. El **Gráfico II: Variación del Error Absoluto en función de las mesas escrutadas, 200 simulaciones**, intenta representar esta idea.

Los primerísimos momentos del escrutinio son decididamente caóticos, como se puede observar en el gráfico. Aunque el eje y está cortado en (-10,10) para optimizar el espacio disponible se observa que buena parte de los resultados parciales con 1, 2 o 3 mesas le erran al final por mucho más de 10 puntos. Ya con 10 mesas escrutadas (siempre manteniendo el supuesto de perfecta aleatoridad en el orden del escrutinio), son muy pocas las ocasiones en las que una mesa puede mover el resultado parcial más de 5 puntos. Y pasadas las 70, en ninguna de las 200 corridas del simulador graficadas una mesa pudo mover el resultado un punto entero.

La razón por la que esto pasa es doble. Obviamente, cada nueva mesa que se suma 'diluye' el peso de todas las anteriores y el suyo propio un poco más, por lo cual necesariamente su impacto será menor. Para compensar por ello es que se ha dibujado uan vez más el eje x en escala logarítmica. En adición, también actúa aquí la tan mentada perfecta aleatoriedad. Aún si existen diferencias geográficas en los patrones de votos, 70 mesas tomadas aleatoriamente alcanzan con creces para tener datos de todos los grandes barrios. En consecuencia, las diferencias geográficas ya se habrán licuado, y por más espectacular que sea el resultado de una mesa, al promediarla con un pastiche general, su particularidad desaparece del todo.

Si recordamos que describimos al error absoluto de un pronóstico como el total de puntos porcentuales por los que la estimación en consideración está errada, la utilidad de analizar la estabilidad de su evolución resulta transparente: el hecho que nos preocupaba era perder 1 punto porcentual de votos, lo cual será casi perfectamente seguro si el error absoluto aún varía de a 10 puntos, posible si varía de a 1 punto, poco probable si no salta

de a más de 1 décima, y casi imposible si la variación está en las centésimas. Este tipo de predicciones cualitativas está lejos de ser perfecta, pero gracias a que considera situaciones distintas entre sí por órdenes enteros de magnitud, sus resultados son útiles aún en su imprecisión.

En el **Gráfico 3: Cantidad de mesas en las que se estabiliza la variación del error absoluto**, se puede observar esquemáticamente esta idea. Su lectura es bastante sencilla a pesar de que pueda parecer intimidante. La línea violeta por ejemplo, marca a partir de qué mesa la variación del error en una simulación es siempre menor a 10. Por todo lo dicho anteriormente, no es sorprendente que casi 950 de las 1000 simulaciones ya no peguen saltos de tal magnitud con una única mesa cargada, y ninguna pasadas las 4.

Si en vez de seguir las curvas nos concentramos en las frecuencias acumuladas del eje y, podemos estimar ciertas probabilidades, independientemente de en qué simulación particular transcurra nuestro escenario de momentánea victoria. Observando por ejemplo la interseccio de $y = 980$ con las curvas de niveles, podemos deducir que en el 98 % de las simulaciones, la varicion del error se volvió permanentemente menor a 1 en las primeras cincuenta mesas. De manera similar, se puede inferir del gráfico que en un 88 % de los casos, pasadas las 500 mesas el error ya no varía más que de a décimas. Para resolver nuestro dilema original, nos basta sencillamente con ubicarnos en el gráfico, y deducir las posibilidades.

Dado que planteamos que habían sido escrutadas la mitad de las mesas (unas 3600, digamos), nos encontramos bien hacia la derecha en el gráfico. A esa altura, todas las simulaciones varían a lo sumo de a décimas, y un pequeõ porcentaje sólo de a centésimas. Sabiendo esto, podemos suponer con confianza considerable que mantendremos nuestra ventaja (o algún margen de ella) hasta el final del escrutinio. No es una predicción .ª prueba de balas", pero funciona, básicamente gracias a que el evento opuesto (perder un punto entero de ventaje de a décimas y centésimas) es infinitamente más complicado.

4.4. Los límites de lo predecible

Hasta el momento, tratamos únicamente las simulaciones del escrutinio. Nuestro principal problema, al tratar de comparar los resultados con la experiencia práctica, fue provocado por un supuesto particularmente fuerte, el de perfecta aleatoriedad en el orden de cómputo. Este supuesto provocó que los resultados obtenidos converjan hacia los valores reales mucho más rápido de lo que lo hacen en la vida real: nuestro modelo estimaba el resultado final tanto mejor que lo empíricamente posible que compararlos fue un tanto infructuoso.

Hay un ámbito, sim embargo, donde este tipo de simulaciones puede tener una utilidad práctica bastante directa: el análisis del error muestral en las encuestas previas a la elección.

Un generador de muestras geográficamente representativas, que además es perfecta-

mente riguroso en la elección al azar de encuestados al interior de cada agregado, tendrá utilidad *negativa*. No nos sirve para saber cuál es el error asociado a la muestra tomada en una cierta encuesta, pero sí nos permitirá dilucidar si dada una cierta muestra, el error que se le imputa es factible o la predicción es demasiado çonfianzuda".

Una lista con todas las razones por las cuales aún la mejor muestra aleatoria en el mundo real es fundamentalmente menos aleatoria de lo que creemos ser haría larga, pero podemos mencionar algunas claves.

En primer lugar, la muestra se toma muchos días antes de la elección, tiempo en el cual los votantes aún pueden cambiar de opinión. Además, la mayoría de las encuestas no se realizan en un único día, sino que pueden cubrir tanto como una semana laboral entera, y los resultados que de ella se deriven no se podrán asociar con ningún momento particular del intervalo de tiempo.

Los medios a través de los cuales se realizan las encuestas también tienen su número de falencias. Sea que la encuesta se realice en persona en la calle o por teléfono, la participación es casi siempre voluntaria, y por enden habrá un sesgo de selección inherente en los participantes que sí la contestan.

A veces, las encuestas telefónicas ni siquieran son realizadas por una persona, y las opiniones que uno pronuncia en voz alta ante otro ser humano son mucho mas confiables que los numeritos que uno presiona para que la máquina haga su trabajo.

La calidad de la representatividad geográfica es una de las pocas cosas en las que una encuesta telefónica puede llegar a ser mejor que una en persona, ya que los teléfonos fijos por definició no se mueven mucho - aunque casi nadie los atiende, tampoco.

Cuando la encuesta se realiza en la calle, se suele recortar trabajo utilizando "puntos muestra": se realizan varias encuestas en el radio de unas pocas cuadras, y esos resultados se extrapolan luego a todo el barrio. Habiendo militado en un partido político durante las elecciones presidenciales del 2011, tuve la oportunidad de realizar varias de estas encuestas, y sé por experiencia propia que unir varios puntos muestra en uno, .ªcomodar"ligeramente los puntajes de nivel socioeconómico para cumplir con la cuota de ciudadanos de clase media-alta exigdida, y otras tantas pequeñas alteraciones son moneda corriente. Cuando además el trabajo es pago en lugar de voluntario, los incentivos para alterar respuestas son todavía mayores.

La idea entonces, será pensar nuestro "simulador de muestrasçomo el ideal platónico de la muestra polietápica (primero estratificada, y luego aleatoria): muestras reales"nunca podrán ser tan buenas como las que construyamos, pero estas últimas pueden servir como vara para medir el desempeño relativo de las primeras.

Veamos en el **Gráfico IV**, cómo se distribuye el **error absoluto para muestras de tamaño 100, 500, 2000 y 10.000**. 100 es un n irrisorio para ciudades del tamaño de Capital Federal, y 10.000 es un número lo suficientemente grande como para que casi nunca se hagan encuestas de ese tamaño. Como referencia, se consideran también 500, un

n superior a los tradicionalmetne usados en encuestas electorales presenciales en Capital Federal, y 2.000, un n típico para sondeos telefónicos.

Aunque parezcan números pequeños (¿cómo podemos inferir los patrones de votación de 1.800.000 personas de sólo 500 de ellas?), en la prćtica son muy usados, y hasta recuerdo que los menores n que se consideraban para realizar una encuesta a nivel nacional (en buena parte por su costo), comenzaban en 1.200 personas.

Como todos los gráficos hasta ahora, el Gráfico IV no es la excepción, y en el eje x se utilizó una escala logarítmica. De lo contrario, no podríamos observar juntos los tamanos del error para muestras con n tan dispares. En este análisis, a diferencia del anterior, trataremos directamente con los valores del error absoluto y no su variación, concepto que no tiene sentido en una muestra estática.

Si nos enfocamos en la curva roja, que representa los errores observados en 10.000 muestras de $n = 100$, podemos inferir que una muestra de este tamaño tiene un poder predictivo realmente bajo. En 9.000 casos (90 %), el error observado estuvo por encima de 9 puntos (cosa que determinamos observando donde $y = 1{,}000$ corta a la curva de $n = 100$), y en el peor 20 % de los casos, superó los 20 puntos. En otras palabras, una muestra tan pequeña es una lotería. El observador casual, sin embargo, puede llegar a derivar conclusiones significativa aún con $n = 100$: por cuestiones culturales probablemente, "100.$^{\text{es}}$ algo así como el "primer número grande"que conocemos, y no somos buenos para distinguir "números grandes.$^{\text{en}}$tre sí.

En el otro extremo, tenemos las muestras de $n = 10{,}000$, que resultaron muy precisas. En el 20 % de los casos su error absoluto no supera la unidad, el 80 % no supera los dos puntos, y en ninguna de las diez mil simulaciones el error absoluto fue mayor a 4. En general, podemos decir que si alguien produjiese una encuesta electoral con $n = 10{,}000$ y tuviese infinito cuidado de evitar caer en los problemas antes mencionados, esa sí que sería una estimación confiable.

La vida real, para variar, no ocurre en ninguno de los dos extremos, sino en un gris intermedio. Es por ello que vale la pena analizar los casos $n = 2{,}000$ y $n = 500$. Para $n = 500$, el 95 % de las muestras tuverion un error absoluto de 11,64 o menos, mientras que para $n = 2{,}000$ el percentil 95 está en 5,81 puntos de error absoluto. Puesto en otras palabras, alrededor de 1 de cada 20 muestras con $n = 500$ yerran el resultado final por más de 11 puntos, y 1 de cada 20 muestras con $n = 2{,}000$, yerran por casi 6 unidades. Ninguno de los resultados es demasiado alentador.

Otra forma de representar estos mismos datos, que nos permite analizar más directamente el famoso ïntervalo de confianza"de un estimador, está en el **Gráfico V: Percentiles seleccionados para el error absoluto de una muestra en función de su tamaño**.

La utilidad de este gráfico reside en el hecho de que para cada n, entre dos bandas de un mismo color encontraremos distintos intervalos de confianza a dos colas: al 98 % entre las rectas rojas, 90 % entre las azules, y 80 % entre las violetas. La recta verde marca la

mediana: una simulación al azar tiene las mismas chances de tener un error mayor a la mediana (50 %) o menor a ella (el otro 50 %).

En esta última representación, además de en el eje x, también la escala en el eje y es logarítmica. Llamativamente, los percentiles que delimitan los intervalos de confianza para distintos tamaõs de muestra forman líneas rectas al ser graficadas de esta forma, lo cual está indicando que un mismo aumento porcentual del tamaño muestral implicará siempre una misma reducción porcentual del error de la muestra, independientemente del n del que se arranque y el nivel de confianza que se desee (pues todas las rectas representando percentiles son esencialmente paralelas).

Si aproximamos las variaciones porcentuales sobre la curva de la mediana, observamos que

$$\frac{ErrAbs(muestra|n = 1{,}000)}{ErrAbs(muestra|n = 100)} = \frac{4,639}{15,513} = 0,299$$
$$\frac{ErrAbs(muestra|n = 10{,}000)}{ErrAbs(muestra|n = 1{,}000)} = \frac{1,462}{4,639} = 0,315$$

Podemos concluir entonces que a grandes rasgos, multiplicar por 10 (o aumentar en un 900 %) el tamaño muestral, disminuirá el error absoluto de la muestra en un 70 %. Este intercambio no es el más favorable en términos de costo-beneficio, y puede explicar en buena parte por qué muchas veces los n considerados en distintos experimentos y encuestas son muy inferiores a los que el margen de error declarado precisaría.

¿Se puede observar este efecto en la práctica? Para comprobarlo, intentaremos poner en contexto algunas predicciones hechas en el mundo real. A pesar de que su principal trabajo es predecir el futuro, rara vez encontramos una evaluación de desempeño de los vaticinios de las consultoras profesionales. Con este gráfico, podemos hacer algunas pruebas caseras.

4.5. Caso de estudio: Poliarquía, La Nación 6-Oct-2013

Revisando los archivos de los principales diarios, es fácil encontrarse con numerosas historias que reportan sobre encuestas realizadas en plena campaña electoral. Tomaremos como ejemplo una de las encuestas más prominentes en esas semanas, publicada en La Nación y realizada Poliarquía 21 días antes de la elección. La encuesta en cuestión, contiene un 8,7 % de "No Sabe/Indeciso", y por ende los porcentajes para los seis candidatos están lejos de sumar 100 %. No intentaremos entonces calcular su error absoluto, pero sí estudiar cómo se describe a sí misma y su margen de error. En el rincón inferior derecho del comunicado de prensa, se lee:

"Universo: población mayor de 18 años, residente en la ciudad de Buenos Aires, en condiciones de votar en las próximas elecciones. Tipo de encuesta: telefónica por sistema

CATI e IVR. Características de la muestra: aleatoria, polietápica, estratificada por zonas para la selección de las características y números telefónicos, y por cuotas de edad y sexo para la selección del entrevistado. Tamaño total de la mestra: 2380 casos. *Error estadístico: +/- 2,05 para un nivel de confianza del 95 %.* Fecha del trabajo de campo: del 2 al 4 de octubre de 2013."

La única cosa que esta encuesta realiza que nuestro modelo no toma en cuenta(simplemente porque el voto es anónimo) es la segmentación por sexo y edad. Al ser telefónica, ya podemos imaginar que la calidad de la información obtenida sera muy inferior a la de nuestro simulador, que posee los resultados oficiales por mesa. Lo que realmente nos interesa, son dos datos. Primero, que $n = 2{,}380$, y segundo, el sumamente vago "error estadístico" de +/- 2,05 al 95 % de confianza.

Se puede sobreentender que por .$^{\text{er}}$ror estadístico"se hace referencial al error muestral, causado por observar una muestra en lugar de la población entera. Este error es básicamente el único que realmente puede afectar a nuestra simulaciones. EL problema para comparar la encuesta con lo expuesto hasta ahora, es que el error de +/- 2,05 es reportado *sin ninguna mención* de las unidades en que se expresa, ni si representa el margen de error de cada predicción indiviual o el margen de error colectivo.

Del Gráfico 5, podemos establecer que el 95 % de las muestras aleatorias de aproximadamente el mismo tamaño tuvieron a lo sumo 5 puntos de error. Si el +/-2,05 de Poliarquía se refiere a un único candidato, está siendo ingenuamente optimista: ¡ la *mitad* de las muestras con $n = 2{,}500$ en nuestra simulaciones tienen *al menos* 3 puntos de error!

Si el +/-2,05 se refiere a cada candidato, la situación podría llegar a ser plausible. En nuestras simulaciones, el 99 % de las iteraciones devolvieron un error muestral menor a 6,14 para $n = 2{,}500$, número que está más en línea con lo reportado por Poliarquía. Sin embargo, no queda claro qué representa dicho intervalo en casos como los de CP, cuyos votos la encuesta pone en 1,6 % (¿¡o sea que puede sacar entre *-0,45* y 3,65 %!?)

Por último, está el tema de la comparación de efectividad con el simulador anterior. Aún cuando la Dirección Nacional Electoral está usando el mismo método que nosotros para estimar el resultado provisorio - contar votos *reales* -, por lo que su tarea es poco incierta, sus resultados convergían con al resultado final *mucho* más lento que nuestro simulador.

Es una razonable suposición lógica que predecir el resultado de una elección a partir de una encuesta es mucho más difícil que contando votos la noche del escrutinio. Si la transitividad se mantiene, debería ser imposible que a Poliarquía le vaya relativamente mejor que a la Dirección Nacional Electoral cuando ambas çompitençon los simuladores programados.

Sin embargo, por lo que vimos, Poliarquía tiene tanta fe en sus encuestas telefónicas como la que nosotros podemos derivar empíricamente de muestras perfectamente aleatorias. A esta altura, estamos en condiciones de afirmar con cierta vehemencia que el margen de

error provisto por Poliarquía es absolutamente inverosímil en un caso (si lo consideramos sobre el total de la predicción), o posible pero demasiado confiado en el otro (cuando se lo aplica a cada candidato).

4.6. Generalizando nuestro análisis

Por último, aunque sea obvio es necesario aclarar que las condiciones en las cuales estos resultados son replicable para otras elecciones, son particularmente restrictivas. Cada caso requiere su análisis particular, pero con algo de sentido común podemos establecer que :

- las elecciones con más candidatos serán menos predecibles (es decir, tendrán mayores intervalos de confianza del error absoluto para idénticos tamaños muestrales) que las de menos candidatos. En democracias bipartidarias como la Argentina pre-2001 o los Estados Unidos durante toda su historia, los resultados son generalmente más fáciles de anticipar que en casos como el nuestro actual, con cinco o seis partidos sacando cantidades no-negligibles de votos;

- las elecciones menos polarizadas serán menos predecibles que las más polarizadas. Una elección entre 10 candidatos, en los cuales sólo 2 o 3 tienen chances de ganar, es más predecible que otra con sólo cinco pero con probabilidades más parejas para todos. En el fondo, una mayor polarización se puede pensar como equivalente a una reducción en el número de çandidatos efectivos".

Nuestro caso fue el de un número considerable de candidatos (6), y una elección relativamente poco polarizada, con tres partidos sacando más del 20 % de los votos. EN términos generales entonces, la mayoría de las elecciones deberían ser al menos ligeramente más predecibles que esta, aunque dicha afirmación bajo ningúnconcepto permitE extrapolar los resultados a otra circunstancias sin uan revisión crítica de los supuestos utilizados.

5. Conclusiones

Analizando los resultados producidos por el simulador de la noche del escrutinio y comparándolos con lo que realmente sucedió el día de la votacíon al cierre de urnas, encontramos que el funcionamiento del simulador y la evolución de los hechos reales seguían caminos demasiado diferentes como para ser comparados directamente. Muy probablemente a causa del supuesto de perfecta aleatoriedad en el orden de carga de los telegramas de votos, sobreestimamos la velocidad a la que los resultados provisorios convergen hacia el resultado final. Mientras que en nuestras simulaciones poco más del 1 % de las mesas

escrutadas alcanzaban para apreciar tendencias definitivas, en los primeros resultados provisorios anunciados oficialmente, aún con casi el 10 % del recuento completo los márgenes de error eran considerables, por encima de los 5 puntos.

Una futura línea de investigación entonces, será incluir en la simulación una cierta medida de correlación entre los telegramas cargados sucesivamente, y verificar si el modelo corregido se ajusta mejor a la realidad. Mientras tanto, aunque sus resultados no describan empíricamente la evolución del recuento provisorio en la realidad, sí nos sirvieron para observar gŕaficamente como a medida que aumenta la cantidad de datos procesados, la cantidad de nueva información incorporada disminuye exponencialmente.

Este mismo fenómeno se observó en el simulador de muestras aleatorias: cada nueva persona incorporada a la muestra mejora la calidad global del estimador, pero un poco menos que la anterior. Para toda consultora profesional (y para los partidos que las contratan), cada individuo encuestado tiene un costo, y si la información que se obtiene tiene rendimientos marginales decrecientes, es de esperar que los tamaños muestrales elegidos tengan una cota superior dura: es por esto que casi nunca observamos muestras $n = 10{,}000$, por ejemplo.

Al disminuir *exponencialmente* en vez de linealmente, los beneficios marginales percibidos al incorporar un nuevo individuo a la muestra pueden verse rápidamente anulados en la práctica, máxime si se considera que existen varias potenciales fuentes de ruido en la información, y que el investigador casi nunca tiene una segunda muestra de control con la cual realizar estudios comparados.

Si nuestras observaciones fuesen son correctas, deberíamos observar una marcada sobre-confianza en los pronósticos electorales aún - o especialemnte - entre quienes dependen financieramente de ello. Dicha subestimación de la incertidumbre inherente a la predicción se suele manifestar en forma de un intervalo demasiado pequeño para el nivel de confianza pretendido. A modo de ejemplo y sin intenciones de generalizar demasiado, se probó esta hipótesis con una encuesta publicada tres semanas antes de la elección por Poliarquía, que confirmó nuestras sospechas.

Estimamos también que al menos en nuestras simulaciones, un 900 % de aumento en el tamaño muestral reduce en un 70 % el error. Un tradeoff así de negativo entre costo de la información linealmente creciente en n, contra rendimiento marginal exponencialemnte decreciente de la información en n, es evidencia suficiente para reconocer que la solución al problema de la sobreestimación de la confianza en los resultados no puede pasar por realizar muestras más grandes. La alternativa que resta, será ser más cuidadoso en el cálculo de los intervalos de confianza.

En dicho contexto, simuladores como este representana una herramienta más para refinar nuestros análisis. Asumiendo que las muestras que genera tienen un nivel de confianza órdenes de magnitud superior al de encuestas reales, podemos utilizarlo como un *check de realidad*: si los márgenes de error que le damos a una muestra tomada en el campo son

similares a los que se observan en el ideal platónico de muestral aleatoria del simulador, sin duda estamos siendo demasiado confianzudos.

6. Referencias

- El archivo de datos utilizado fue descargado del siguiente link: http://www.datospublicos.gob.ar/data/storage/f/2013-10-29T14%3A45%3A50.732Z/electoral-2013-diputados-nacionales.csv
- El archivo con la descripción general del dataset es http://www.datospublicos.gob.ar/data/storage/f/2013-10-29T15%3A42%3A53.979Z/electoral-2013-descripcion-general.txt
- Las descripciones del funcionamiento del escrutinio provienen de experiencia personal, y conversaciones con funcionarios de la DINE.
- Las afirmaciones sobre las regiones donde cada partido dominó las urnas resultan de consultar el hack realizado por Manuel Aristarán, con los resultados por establecimiento geolocalizados. interactivos.lanacion.com.ar/mapa-elecciones-2013
- Las declaraciones de Florencio Randazzo la noche del escrutinio fueron tomadas del archivo de la TV Publica. https://www.youtube.com/watch?v=yW1sXTgKJIg

7. Anexo I: Gráficos

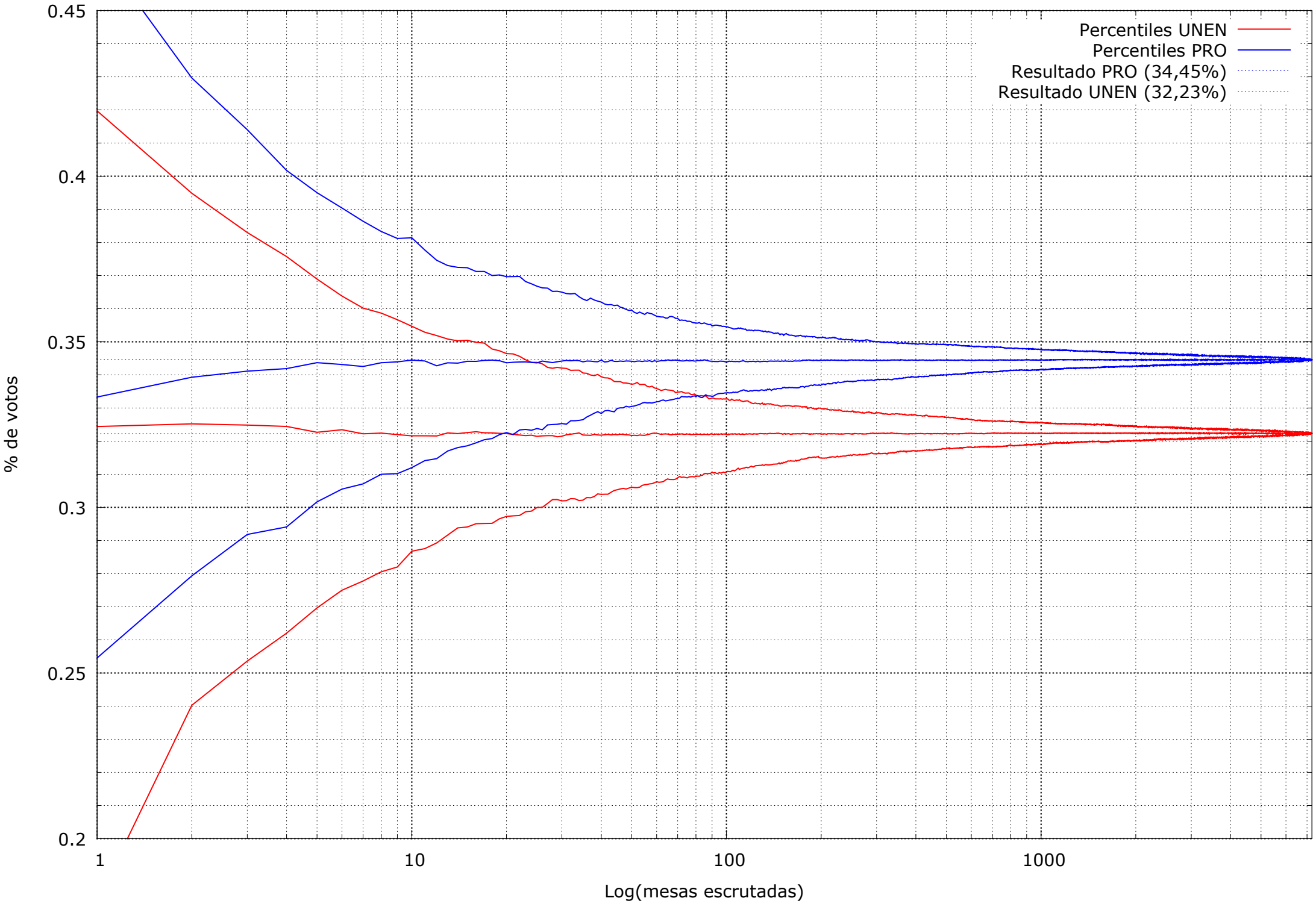


Grafico I: Percentiles 5, 50 y 95 de los votos de UNEN y PRO en 1000 simulaciones.
Percentiles UNEN
Percentiles PRO
Resultado PRO (34,45%)
Resultado UNEN (32,23%)
0.45
0.4
0.35
0.3
0.25
0.2
% de votos
1
10
100
1000
Log(mesas escrutadas)

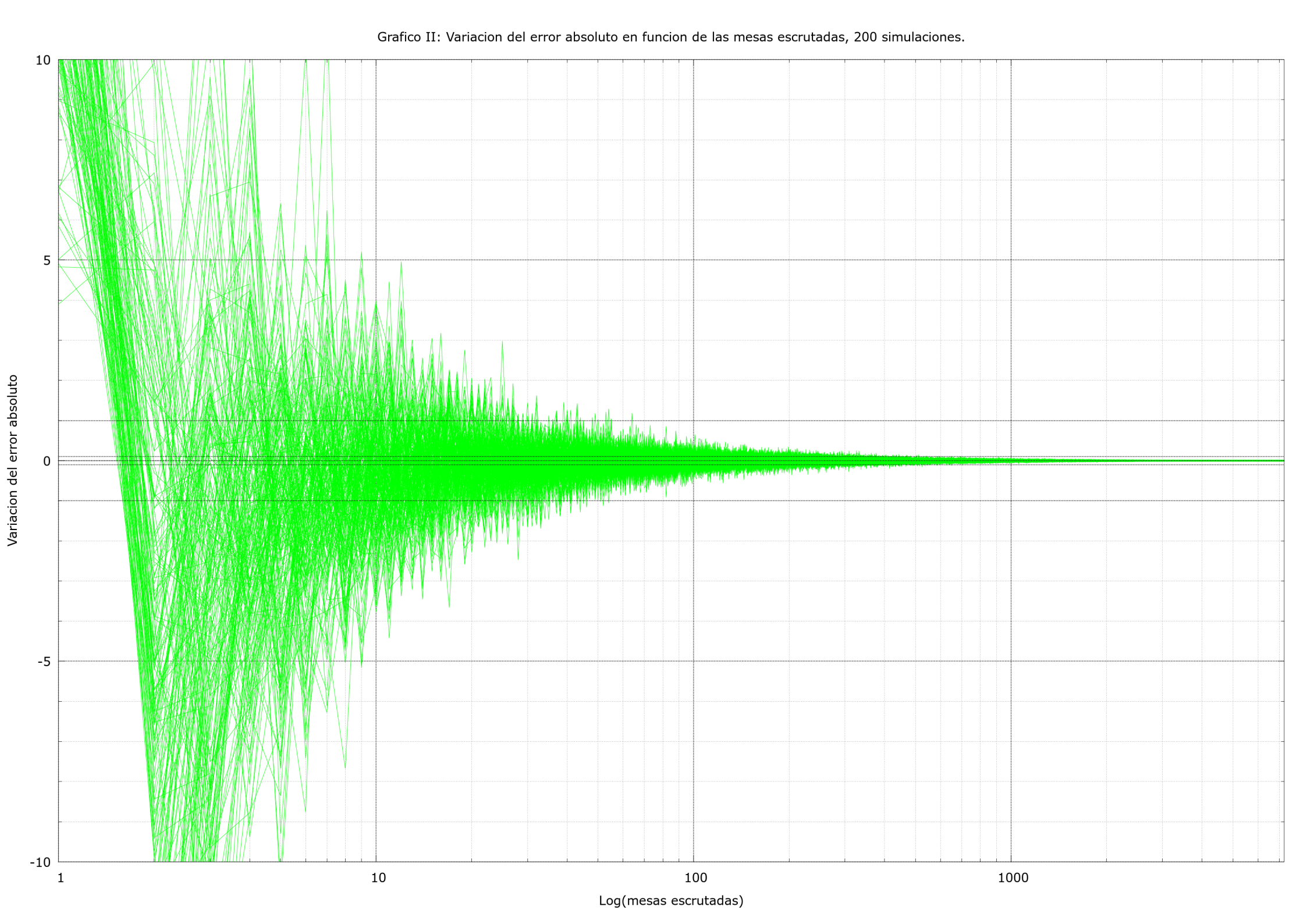


Grafico II: Variacion del error absoluto en funcion de las mesas escrutadas, 200 simulaciones.
10
5
0
-5
-10
Variacion del error absoluto
1
10
100
1000
Log(mesas escrutadas)

Grafico III: Cantidad de mesas en las que se estabiliza la variacion del error absoluto.

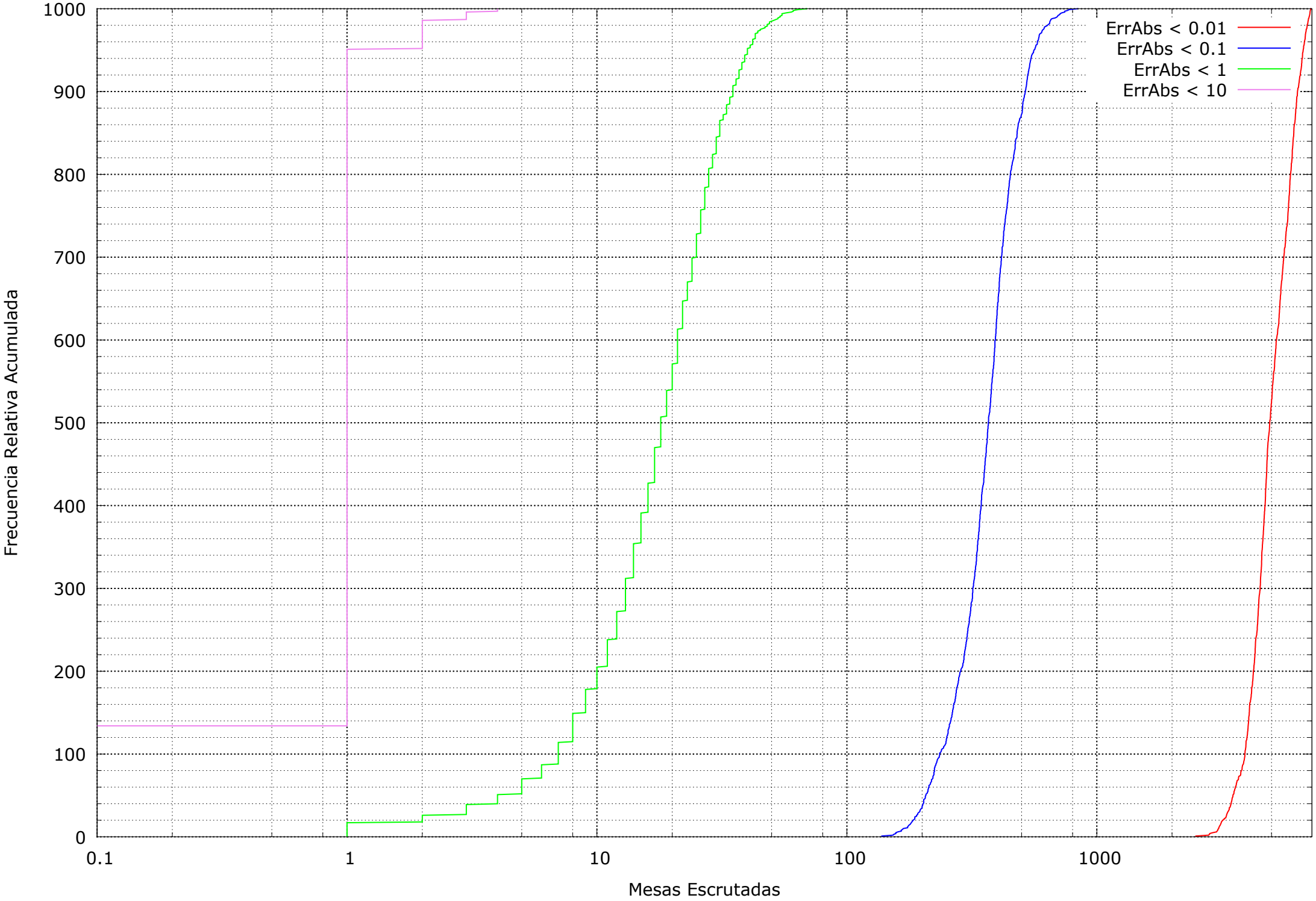
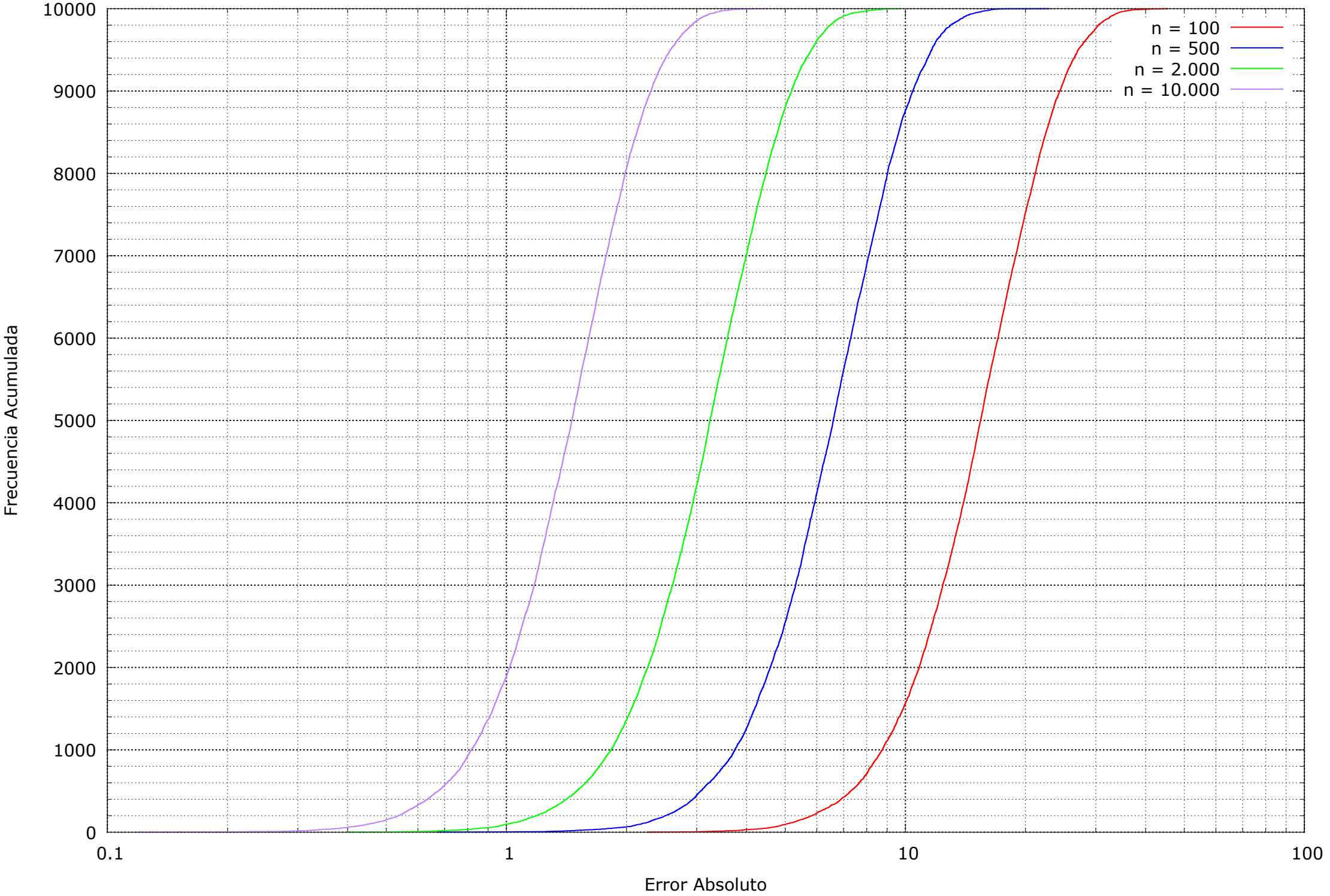


Grafico IV: Frecuencia acumulada del error absoluto para muestras elegidas (10.000 simulaciones).



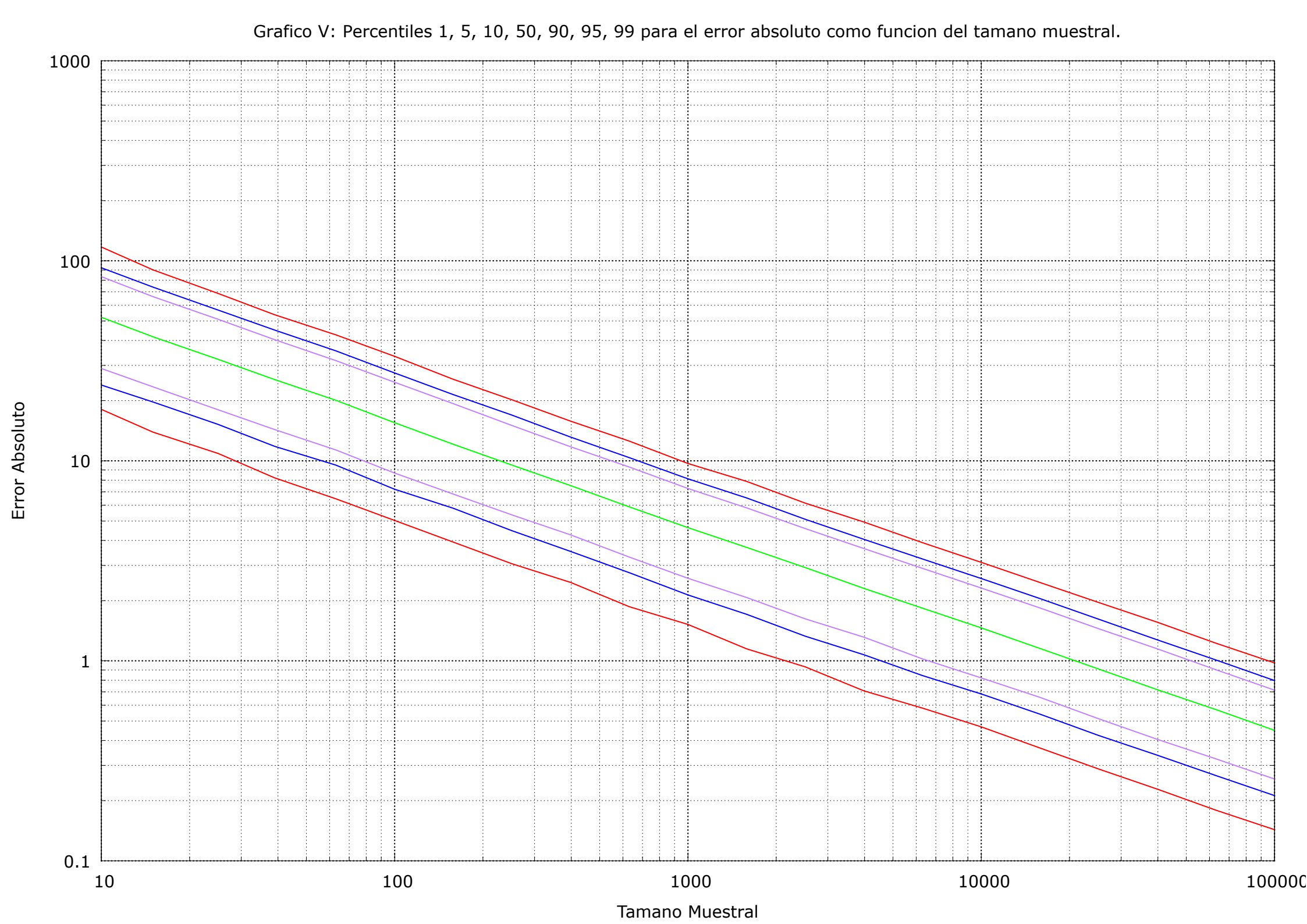


Grafico V: Percentiles 1, 5, 10, 50, 90, 95, 99 para el error absoluto como funcion del tamano muestral.
1000
100
10
1
0.1
Error Absoluto
10
100
1000
10000
100000
Tamano Muestral

8. Anexo II: Código Fuente

No se incluye aquí el código para transformar la información del formato original a la versión utilizada por los simuladores, con cada mesa, circuito o sección representada por un unico vector de 6 componentes. Si se desea, dicho código (escrito en SQL) se puede revisar en el repositorio de GitHub de este trabajo:

https://github.com/gonzalobb/tesis_mesis

Para cada función, se provee una breve descripción de lo que realiza, y qué elementos devuelve, en el siguiente formato.

mi_funcion(arg1, arg2) =>resultado

Esto debe interpretarse como "la funcion 'mi_funcion' toma los argumentos 'arg1' y arg2', y devuelve 'resultado'".

Tanto los argumentos como los resultados de las funciones son de tres grandes tipos:

- Numeros enteros, representados con letras minusculas. El tamano muestral a considerar, la cantidad de muestras uqe se desea extraer, el numero de mesas a escrutar.
- Vectores (o *arrays*), representados con []. Son casi siempre un dexteto de numeros representando alguna caracteristica relacionada a los partidos participantes: pueden estar expresados en votos absolutos, proporciones entre cero y 1, y porcentajes (entre 0 y 100)
- "Matrices"(o *arrays bidimensionales*, representadas con [[]]. Tecnicamente son arrays que tienen un array en cada una de sus posiciones. Como un vector es un conjunto de numeros, una matriz es un conjunto de vectores. Una simulacion de la noche dle escutinio, por ejemplo, consistira en uan matriz donde la fila i contiene los resultados parciales, habiendose escrutado i mesas. Un conjunto de 1000 muestras aleatorias de tamano X, estara representado por 1000 vectores de 6 elementos, o una matriz de 1000×6.

```
########################################
#                                      #
#       PROCESAMIENTO DE DATOS         #
#                                      #
########################################

# Tomo los datos a cada nivel de agregacion, y los cargo en sendos [[]]
mesas_raw     = IO.readlines(Dir.pwd + "/mesas_raw.tsv")
circuitos_raw = IO.readlines(Dir.pwd + "/circuitos_raw.tsv")
secciones_raw   = IO.readlines(Dir.pwd + "/secciones_raw.tsv")

# emprolijar_agregado([[]]) => ([[]])
# --Toma el input 'crudo' generado con SQL y devuelve los votos agregados a
   nivel mesa, circuito o seccion en el formato 'vector de 6 elementos' que
   utilizan los simuladores.
def emprolijar_agregado(agregado)

  # Elimino la primer linea del archivo, que contiene los titulos de las
     columnas:
  agregado.shift()

  # Formateo cada division del agregado (mesas, circuitos o secciones) para
     transformarla en un array de numeros enteros y las guardo en
     mesas_completas.
  agregado_prolijo = []
  for division in agregado
    # quito el \n del final de la linea, la corto sobre los tabs y transformo
       a enteros el array resultante
    agregado_prolijo << division.chomp.split("\t").map { |x| x.to_i}
  end

  # Hasta aqui, cada division del agregado tiene ocho campos: su numero
     identificador en la primera, el total de votos que cada uno de los 6
     partidos saco de la 2 a la 7, y blancos en la 8.
  # Con la proxima linea eliminamos el primer elemento de cada array, para
     dejar solo las cantidades de votos.
  agregado_prolijo.each {|division| division.shift(1)}

  # Si se quiere realizar el analisis CON VOTOS EN BLANCO, descomentar la
     proxima linea:
  agregado_prolijo.each {|division| division.pop(1)}

  return agregado_prolijo
end

# Guarda el resultados de 'emprolijar' mesas, circuito y secciones en sendas
   [[]].
votos_mesas     = emprolijar_agregado(mesas_raw)
votos_circuitos = emprolijar_agregado(circuitos_raw)
votos_secciones = emprolijar_agregado(secciones_raw)
```

```
########################
#                      #
#  Funciones Generales #
#                      #
########################

# errcuad([], []) => n
# --Suma los cuadrados de las diferencias entre los elementos de dos vectores
   . Ej: ecm([1,2], [4,3]) => 10
def errcuad(a, b)
  dif = [a, b].transpose.map {|x| x.reduce(:-)}
  dif.map! {|x| x**2}
  return dif.reduce(:+)
end

# errabs([], []) => n
# --Suma las diferencias absolutas entre los elementos de dos vectores. Ej:
   ecm([1,2], [4,3]) => 4
def errabs(a, b)
  dif = [a,b].transpose.map {|x| x.reduce(:-).abs}
  return dif.reduce(:+)*100
end

# normalizar_una([]) => []
# --Toma los votos por partido obtenidos en una mesa/circuito/seccion ("m/c/s
   "), y devuelve que proporcion del total representan. Ej: normalizar_una
   ([2,3,5]) => [0.2,0.3,0.5]
def normalizar_una(mesa)
  peso = mesa.reduce(:+)
  peso = 1 unless peso != 0 # En caso de que se tomen 0 votos de una mesa,
     esta linea impide que se dividan los resultads por cero.
  norma = mesa.map {|votos| (votos.to_f / peso).round(10)}
  return norma
end

############################
#                          #
# Simulador de Escrutinios #
#                          #
############################

# mezclar_mesas([[]]) => [[]]
# -- Toma la matriz de resutlados por mesa, y reordena las filas al azar. Ej:
    mezclar_mesas([[1,2],[3,4],[5,6],[7,8]]) => [[3,4],[7,8],[5,6],[1,2]]
def mezclar_mesas(mesas)
  mesas_mezcladas = mesas.shuffle
  return mesas_mezcladas
end
```

```ruby
# sumar_arrays([[]]) => []
# --Toma un conjunto de vectores, y realiza su suma escalar. Ej: sumar_arrays
   ([1,2],[7,4],[2,8]) = [10,14]
def sumar_arrays(arrays)
  arrays.transpose.map {|x| x.reduce(:+)}
end

# suma_parcial([[]]) => [[]]
# --Toma un conjunto de vectores, y devuelve para cada vector la suma escalar
    de todos los vectores desde el primero hasta el inclusive. Ej:
   suma_parcial([1,2],[7,4],[2,8]) = [[1,2],[8,6],[10,14]]
def suma_parcial(mesas)
  sumas_parciales = []
  sumas_parciales[0] = mesas[0]
  for i in (1..mesas.length-1)
    sumas_parciales[i]= sumar_arrays([sumas_parciales[i-1], mesas[i]])
  end
  return sumas_parciales
end

# normalizar_muchas([[]]) => [[]]
# --Aplica la funcion normalizar_una([]) => [] a cada uno de los elementos de
    un conjunto de vectores.
def normalizar_muchas(mesas)
  normas = []
  mesas.each do |mesa|
    normas << normalizar_una(mesa)
  end
  return normas
end

# simular_un_escrutinio([[]]) => [[]]
# --Combinando las funciones anterior, corre una simulacion entera de la
   evolucion de los resutlados la noche del escrutinio. Primero mezcla las
   mesas, luego hace las sumas parciales hasta cada mesa, y finalmente
   normaliza los resultados. El ultimo vector de la matriz coincide
   necesariamente ocn el resultado oficial de la eleccion, siempre.
def simular_un_escrutinio(mesas)
  normalizar_muchas(suma_parcial(mezclar_mesas(mesas)))
end

#########################
#                       #
# Simulador de Muestras #
#                       #
#########################

# pesos([[]]) => []
```

```ruby
# --Toma una matriz, y devuelve un vector donde el i-esimo elemento
   corresponde a la suma de los componentes del i-esimo vector de la matriz
   original. Ej: pesos([1,2],[7,4],[2,8]) = [3,11,10]
def pesos(mesas)
  pesos_mesas = []
  mesas.each do |mesa|
    pesos_mesas << mesa.reduce(:+)
  end
  return pesos_mesas
end

# tamanos_muestra_discreta([[]], n) => []
# --Dado un nivel de agregados m/c/s y un tamano de muestra n a extrar,
   calcula cuantos elementos tomar de cada m/c/s para mantener la
   representatividad de la muestra general. En caso de encontrar cantidades
   no enteras, las "fracciones de voto" se distribuyen aleatoriamente.
def tamanos_muestra_discreta(votos_agregado, n)

  peso_total = pesos(votos_agregado).reduce(:+)
  votos_por_mesa = pesos(votos_agregado).map {|x| (x.to_f / peso_total * n).
     round(2)}
  votos_por_mesa_discretos = []
  votos_por_mesa.each do |votos|
    if rand < votos % 1
      votos_por_mesa_discretos << (votos.to_i + 1)
    else
      votos_por_mesa_discretos << (votos.to_i)
    end
  end
  return votos_por_mesa_discretos
end

# agregar_votos([]) => []
# --Toma un conjunto de votos, y cuenta los totales para cada candidato,
   estando los candidatos representados por los numeros del 0 al 5.
# --Ej: agregar_votos([1,2,3,1,1,0,5,3]) = [1,3,1,3,0,1]
def agregar_votos(votos)
  votos_agregados = Array.new(6, 0)
  votos.each do |voto|
    votos_agregados[voto] += 1
  end
  return votos_agregados
end

# pesar_muestra([], n) => []
# --Toma una muestra de votos correspondiente a cierta m/c/s, y la pondera
   por el numero de votantes en ella. Ej: pesar_muestra([1,3,2], 300) =
   [50,150,100]
def pesar_muestra(muestra, peso)
  return normalizar_una(muestra).map {|votos| votos * peso}
```

```
end

# muestra_por_mesa([], n) => []
# --Dada una m/c/s y un tamano muestral n, extrae de la m/c/s una muestra de
   n votos al azar.
def muestra_por_mesa(mesa, cantidad)
  copia_mesa = mesa.dup
  votos_elegidos = []
  peso_mesa = mesa.reduce(:+)
  while votos_elegidos.length < cantidad
    n = copia_mesa.reduce(:+)
    sorteo = rand(1..n)
    voto_actual = 0

    while sorteo > copia_mesa.first(voto_actual+1).reduce(:+)
      voto_actual +=1
    end

    votos_elegidos << voto_actual
    copia_mesa[voto_actual] -= 1
  end

  return pesar_muestra(agregar_votos(votos_elegidos), peso_mesa)
end

# muestral_general([[]], n) => []
# Dado un conjunto de m/c/s y un tamano muestral, crear una muestra con
   dichas caracteristicas. Involucra primero una llamada a
   tamano_muestra_discreta() que devuelve los votos a tomar por m/c/s, y
   luego a muestra_por_mesa(), que toma los votos anteriormente indicados de
   cada m/c/s.
def muestra_general(mesas, cantidad)
  muestras_por_mesa = []

  votos_por_mesa = tamanos_muestra_discreta(mesas, cantidad)

  for i in 0..mesas.length-1
    muestras_por_mesa << muestra_por_mesa(mesas[i], votos_por_mesa[i])
  end

  muestra_general = sumar_arrays(muestras_por_mesa)

  return normalizar_una(muestra_general)
end

############################
#                          #
#    Extraccion de Datos   #
#                          #
############################
```

```
# Variables auxiliares.
resultado_4dec = [0.379, 0.2162, 0.3221, 0.3446, 0.0564, 0.0228]
votos_por_partido = [68_246, 389_128, 581_096, 621_167, 101_862, 41_194]
resultado_exacto = normalizar_una(votos_por_partido)
=begin
# Creacion de 1000 simulaciones de la noche del escrutinio.
escrutinios = []
i = 1
1_000.times do
  escrutinios << simular_un_escrutinio(votos_mesas)
  p i
  i +=1
end

# 1. Conos de incertidumbre para Carrio y Bergman
# Del array de s simulaciones, tomo unicamente los porcentajes
   conrrespondientes a 'candidato' y desecho el resto. Obtengo un array de s
   * n (n = numero de agregados)
# Traspongo el array y obtengo uno donde en cda posicion, estan los s
   porcentajes que el candidato obtuvo hasta la mesa i.
# Tomo el percentil indicado del array en cada posicion.
# Devuebo un array de n posiciones, cada una con el valor que acumula el
   percentil indicado por el candidato hasta entonces.
puts "1"

def tomar_valores_candidato(mesas, candidato)
  valores_candidato = []
  mesas.each do |mesa|
    valores_candidato << mesa[candidato]
  end
  return valores_candidato
end

def limites(simulaciones, candidato, posicion)
  valores_candidato = []
  simulaciones.each do |sim|
    valores_candidato << tomar_valores_candidato(sim, candidato)
  end
  porcentajes_parciales = valores_candidato.transpose
  porcentajes_parciales.map {|n| n.sort!}
  resultado = []
  porcentajes_parciales.each do |percs|
    resultado << percs[posicion]
  end
  return resultado
end

for candidato in (2..3)
  for percentil in [1, 10, 50, 100, 500, 900, 950, 990, 1000]
    curfile = File.open("#{candidato}_#{percentil}.dat", "w")
```

```
    limites(escrutinios, candidato, percentil-1).each_with_index do |x, i|
      curfile.puts "#{i + 1} #{x}"
    end
    curfile.close
  end
end

# 2. Variacion del error absoluto de 200 escrutinios en funcion del numero de
    mesas computadas.
puts "2"

def errabs_escrutinio(escrutinio)
  errabs = []
  for i in 0..escrutinio.length-1
    errabs[i] = errabs(escrutinio[i], escrutinio[-1])
  end
  return errabs
end

def delta_serie(serie)
  delta_serie = []
  for i in 0..(serie.length-1)
    delta_serie << serie[i]-serie[i-1]
  end
  return delta_serie
end

escrutinios.first(200).each_with_index do |escrutinio, index|
  curfile = File.open("escrutinio#{index}.dat", "w")
  deltas = delta_serie(errabs_escrutinio(escrutinio))
  deltas.each_with_index do |delta, i|
    curfile.puts "#{i + 1} #{delta}"
  end
  curfile.close
end

# 3. Frecuencias acumuldas de los valores para los cuales la estabilidad del
   error absoluto aumenta en un orden de magnitud. En otras palabras, se
   busca cual es la mesa 'n' a partirde la cual el errror absoluto se
   estabiliza por debajo de 10/1/0,1/0,01 puntos.
puts "3"

def estabilizacion_errabs(escrutinios)
  resultados = []
  escrutinios.each do |escrutinio|
    deltas_invertidos = delta_serie(errabs_escrutinio(escrutinio)).reverse!
    puntos_criticos = []
    deltas_invertidos.each_with_index do |paso, i|
      if paso > 0.01 && puntos_criticos.length == 0
        puntos_criticos << escrutinio.length - i
```

```
      elsif paso > 0.1 && puntos_criticos.length == 1
        puntos_criticos << escrutinio.length - i
      elsif paso > 1 && puntos_criticos.length == 2
        puntos_criticos << escrutinio.length - i
      elsif paso > 10 && puntos_criticos.length == 3
        puntos_criticos << escrutinio.length - i
        break
      else
      end
    end
    puntos_criticos.push(0) if puntos_criticos.length == 3
    resultados << puntos_criticos
  end
  return resultados
end

puntos_criticos_por_nivel = estabilizacion_errabs(escrutinios).transpose.map
   {|x| x.sort!}
puntos_criticos_por_nivel.each_with_index do |puntos, i|
  curfile = File.open("quiebres#{i}.dat", "w")
  puntos.each_with_index do |p, i|
    curfile.puts "#{i␣+␣1}␣#{p}"
  end
  curfile.close
end
=end
# 4. Error absoluto de 10.000 muestras, para tamanos muestrales de 100, 500,
   2000 y 10000.
puts "4"

for tamano in [100, 500, 2_000, 10_000]
  curfile = File.open("graf4_#{tamano}.dat", "w")
  errores_muestras = []
  10_000.times do
    errores_muestras  << errabs(muestra_general(votos_circuitos, tamano),
       resultado_exacto)
  end
  errores_muestras.sort!
  errores_muestras.each_with_index do |errabs, i|
    curfile.puts "#{i␣+␣1}␣#{errabs}"
  end
  curfile.close
end

# 5. Percentiles comparados para varios tamanos meustrales
puts "5"

enes_grafico_cinco = []
for i in 0..20
  enes_grafico_cinco << (10**(1 + 4.0 / 20 * i)).to_i
```

```
end

muestras_grafico_cinco = []

enes_grafico_cinco.each do |ene|
  muestras_ene = []
  10_000.times do
    muestras_ene << errabs(muestra_general(votos_circuitos, ene),
       resultado_exacto)
  end
  muestras_ene.sort!
  muestras_grafico_cinco << muestras_ene
end

percentiles =  [1, 5, 10, 50, 90, 95, 99]
for perc in percentiles do
  curfile = File.open("perc#{perc}.dat", "w")
  muestras_grafico_cinco.each_with_index do |muestras, i|
    curfile.puts "#{enes_grafico_cinco[i]} #{muestras[perc*100]}"
  end
  curfile.close
end
```