Entrega #3: Bootstrap

Autor: Tomás Notenson

3 de diciembre de 2021

1. Consignas

A) 1) Agregar calculo de bias al script `bootstrap_vs_toy` que usamos ayer

2) Estudiar cómo evoluciona la estimación de la incerteza de un estimador a elección conforme aumenta el número de réplicas para tres tamaños de muestra diferente. Comparar en un mismo grafico.

B) Calcular la correlación angular en un experimento con fotones entrelazados usando los datos disponibles en la solapa Material Adicional, y agregar una barra de error por bootstrap para cada $\Delta\theta$.

2. Respuestas

2.1. Parte A

1) **Bias**

El cálculo del bias ya estaba implementado en el código `bootstrap_vs_toy` como $\hat{bias} = \hat{\theta^*} - t(\hat{F})$ donde $\hat{\theta^*} = \sum_{r=1}^{N_{rep}} \frac{\theta^*(r)}{N_{rep}}$ es el promedio de los estadísticos calculados usando las réplicas y $t(\hat{F})$ es el estadístico calculado a partir de la muestra que se utiliza como una distribución de probabilidad empírica para hacer las réplicas.

2) **Incerteza en función del número de réplicas**

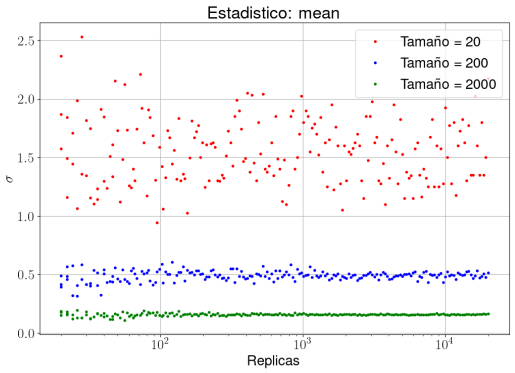
Se implementó en python la función `bootstrap_poisson` que recibe los siguientes parámetros:

- `estadistico`: Una función de python que se aplique a una tira de datos para calcular el estadístico correspondiente. Por ejemplo: `np.mean`, `np.median`, `np.max`, etc.
- `sample_size`: Tamaño de la muestra madre sobre la cual se van a tomar las réplicas.
- `replicas`: Cantidad de réplicas que se van a utilizar.
- `CL=0.6827` (optativo): Confidence level para el caso en el que el error se calcule tomando un intervalo para el estadístico utilizando los cuantiles $\alpha = \frac{1-CL}{2}$ y $1-\alpha$ (en este programa esa opción se denomina "frecuentista" y se pasa en el parámetro `error`).
- `mu=50` (es optativo cambiarlo): Parámetro real de la distribución de Poisson de la cual se genera la muestra madre.
- `error='frecuentista'` (es optativo cambiarlo): Recibe dos posibles strings: 'frecuentista' y 'std', para indicar con qué método se calcula la incerteza del estimador.

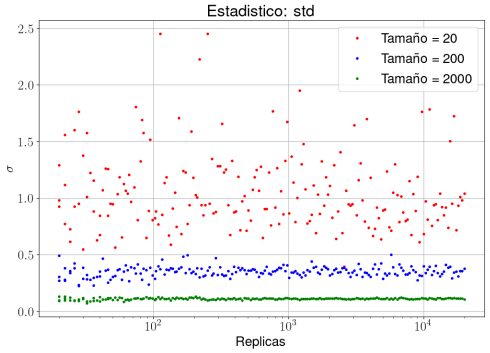
Esta función toma una muestra madre de una distribución $Poisson(\mu)$ que se guarda en la variable local `sample` de la cual luego se van a tomar un número `replicas` de réplicas a partir de la función `np.random.choice` que samplea el array `sample` una cantidad `sample_size` veces, con repetición. En para cada una de las réplicas se computa el estadístico indicado con el parámetro `estadistico` y se guarda en la variable local `bootstrap_sample` para luego calcular la incerteza y el bias a partir de estos valores. Estos últimos cálculos son los que devuelve la función a través de las variables `bootstrap_sigma`, `bootstrap_bias`. El error se calcula llamando a la función `errores(sample, CL, tipo='std')` que recibe como parámetros la muestra `sample`, el nivel de confianza `CL` y un string que especifica por cual de los dos métodos se calcula el error `tipo='std'`. Para calcular

los errores usando `tipo='std'` utiliza la función `np.std` sobre `sample`. Para calcular los errores usando `tipo='frecuentista'` se utiliza la variable `CL`. Para calcular el intervalo la función toma los cuantiles usando `np.quantile` y luego devuelve la mitad del intervalo (aproximándolo por un intervalo simétrico).

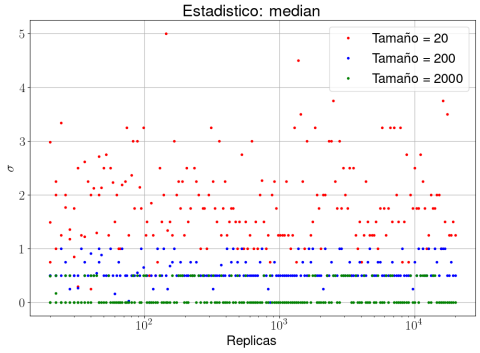
En la figura 1 se muestra la incerteza en función del número de réplicas para algunos estadísticos (la media, la desviación estándar, la mediana y el máximo de los datos) y para tres tamaños distintos de muestra madre. Para la media y la desviación estándar esta incerteza se hace más pequeña al aumentar el tamaño de las muestras y al aumentar el número de réplicas. En cambio, para la mediana y el máximo se observa que esta estimación se hace más pequeña en promedio al aumentar el tamaño de las muestras pero no al aumentar el número de réplicas. Esto se debe a que la muestra madre está fija y entonces la mediana y el máximo varían solamente cuando el dato que corresponde a la mediana y al máximo de la muestra madre no están incluidos en la réplica correspondiente. En general en la figura se puede observar que esta variación es de 1 o de 2 teniendo en cuenta que los datos toman valores discretos ya que vienen de una distribución de Poisson.



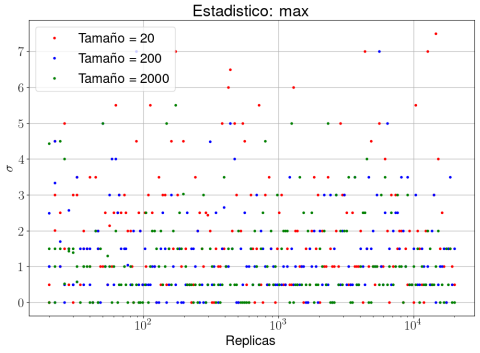
(a) media



(b) desviación estándar



(c) mediana



(d) máximo

Figura 1: Incerteza en función del número de réplicas para distintos estadísticos

2.2. Parte B

Se guardó el número de fotones contados f_i para cada inclinación con ángulos $\theta = 10°.i$ con $i = 1, ..., 36$ en un array de numpy. Como el objetivo general es calcular la correlación y su incerteza entre los f_i y los f_{i+n} con $n = 1, ..., 36$ e i fijo, se definió la función `traslacion(fotones,n)` que recibe el array con los f_i y traslada sus índices en `n` para retornar f_{i+n}. Para calcular la correlación entre ambos arrays se definió la función `corr(a,b)` que recibe dos arrays y devuelve la correlación entre ellos. Para computar la correlación utilizando la muestra madre para cada valor de n se definió la función `corr_mean(fotones)` que recibe el array con los datos del conteo de fotones para cada inclinación y devuelve los 36 valores de correlación entre los $(f_i; f_{i+n})$ para $n = 1, ..., 36$.

La función `bootstrap_fotones(fotones, n, n_replicas=100, CL = 0.6827, error='frecuentista')` recibe los siguientes parámetros:

- `fotones`: Array con los valores de f_i.
- `n`: Valor correspondiente a n para ejecutar la traslación $f_i \to f_{i+n}$.
- `n_replicas=100` (es optativo cambiarlo): Número de réplicas a tomar de la muestra madre.
- `CL = 0.6827` (optativo): Nivel de confianza para calcular el error a partir de los cuantiles.
- `error = 'frecuentista'` (es optativo cambiarlo): Recibe dos posibles strings: 'frecuentista' y 'std', para indicar con qué método se calcula la incerteza del estimador.

El principal objetivo de esta función es estimar el error de la correlación (además calcula el bias y el valor de correlación estimado como el promedio de las réplicas) para `n` fijo. Para cumplirlo, samplea con repetición de la muestra madre conformada por los 36 pares ordenados $(f_i; f_{i+n})$ una réplica de 36 pares ordenados y calcula la correlación (es importante notar que el método de bootstrap permite que se repitan los pares ordenados en la misma réplica). La implementación de esto último se realizó mediante un ciclo `for 1:n_replicas` para el cual se toman índices al azar, con repetición, mediante la función `np.random.choice` sobre un array de valores $(0, 1, ..., 35)$. Estos índices se usaron luego para samplear pares ordenados sobre la muestra madre tomando los f_i del array `fotones` y los f_{i+n} del array retornado por `traslacion(fotones,n)`. Por último computa la correlación entre estos dos arrays y guarda su valor en el array `correlacion`. Al final el ciclo, calcula el bias como se detalló anteriormente y la incerteza con el método indicado en el parámetro `error` (para esto llama a la función `errores` ya presentada anteriormente).

Finalmente, la última función definida es `bootsrap_correlacion(fotones, n_replicas, CL = 0.68, error='frecuentista')` que recibe prácticamente los mismos parámetros que la función anterior a excepción de `n`. El objetivo de esta función es calcular la incerteza, el bias y la correlación por bootstrap para todos los valores de $n = 1, ..., 36$ utilizando la función detallada en el párrafo anterior.

El resultado obtenido para un número de réplicas `n_replicas = 10000` se puede observar en la figura 2

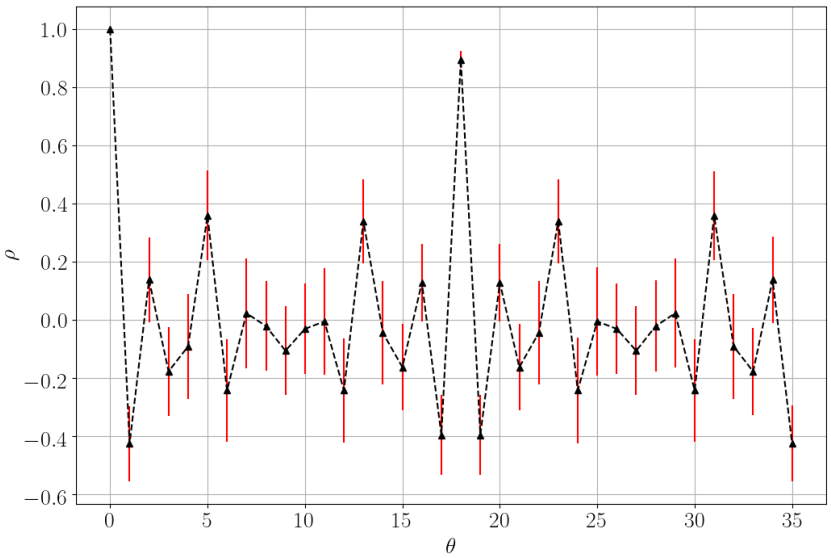


Figura 2: Correlación con su incerteza estimada utilizando bootstrap en función del ángulo de rotación para 10000 réplicas.