Problema 1

Los siguientes 15 datos forman una muestra aleatoria de una distribución Gama con parámetro de forma $\alpha = 3$ y parámetro de escala $\beta = 2$ (la media es $\alpha\beta$ y la varianza $\alpha\beta^2$).

$$14.18, 10.99, 3.38, 6.76, 5.56, 1.26, 4.05, 4.61, 1.78, 3.84, 4.69, 2.12, 2.39, 16.75, 4.19$$

Encuentre un intervalo de confianza para la media de la distribución.

Estudiemos la muestra proporcionada en el enunciado. Siendo $X_1, X_2, ..., X_{15} \sim Gamma(\alpha = 3, \beta = 2)$ la muestra, calculamos su media muestral como estimador de la media real,

$$\bar{X}_{15} = \frac{1}{15}\sum_{i=1}^{15} X_i = 5.77$$

Queremos encontrar un intervalo de confianza para la media $\theta := \alpha\beta$ de nuestra distribución basandonos en la muestra proporcionada. Para ello generamos B muestras Bootstrap x^* de tamaño n, remuestreando los datos originales con remplazamiento. Para cada una de estas muestras calculamos el estimador de la media obteniendo $(\hat{\theta}_1^*, \hat{\theta}_2^*, ..., \hat{\theta}_B^*)$. Así obtenemos una muestra como aproximación a la distribución del estimador $\hat{\theta}$ de la media.

Algorithm 1: Intervalos de confianza Bootstrap

```
def bootstrap_ci(sample, resample, estimator, confidence=.95):
    # Estimate with sample
    est = estimator(sample)

    # Estimated values list
    bestimated_lst = []

    # Resample multiple times and estimate
    for idx in range(resample):
        bsmpl = bootstrap_sample(sample)
        bestimated_lst.append(estimator(bsmpl))

    # Estimate .95, .5 quantiles from sample
    alpha = 1.0 - confidence
    quantiles = np.quantile(bestimated_lst, [alpha / 2.0, 1.0 - alpha / 2.0])

    # Basic confidence interval
    basic_ci = (2 * est - quantiles[1], 2 * est - quantiles[0])

    # Percentile confidence interval
    percen_ci = (quantiles[0], quantiles[1])
```

```
    return (basic_ci, percen_ci)
```

Utilizando nuestra implementación en python para aproximar la verdadera distribución del estimador $\hat{\theta}$ daremos las estimaciones de los intervalos de confianza de la media usando el **método básico** y el **método de percentiles**, que se calculan de la siguiente manera,

$$Basico = [2\hat{\theta} - \hat{\theta}^*_{1-\alpha/2}, 2\hat{\theta} - \hat{\theta}^*_{\alpha/2}] \quad (1)$$

$$Percentil = [\hat{\theta}^*_{1-\alpha/2}, \hat{\theta}^*_{\alpha/2}] \quad (2)$$

Un comentario de la notas de Bootstrap compartidas en clase [1], es que el intervalo **Percentil** no se recomienda utilizar por no tener una justificación sólida. A comparación sabemos que de el intervalo **Básico**, la distribución del estimador Bootstrap $\hat{\theta}^* - \hat{\theta}$ converge al distribución empírica cuando B se hace grande, y a su vez a la distribución real cuando n se hace grande (Sección 6 [1]).

Así obtuvimos las estimaciones de los intervalos, usando $B = 10,000$ que es el número de remuestreos Bootstrap a hacer,

$$Basico = [3.3744, 7.7973] \quad (3)$$

$$Percentil = [3.7426, 8.1655] \quad (4)$$

Problema 2

En la librería boot de R accesar los datos cd4, los cuales son conteos de células CD4 en pacientes HIV-positivos al inicio y después de 1 año de tratamiento con un antiviral.

a) **Construya un intervalo bootstrap para el coeficiente de correlación entre los conteos base y los conteos después del tratamiento.**

En este caso queremos encontrar un intervalo de confianza para el coeficiente de correlación los datos CD4, siendo X el conteo de celulas CD4 al inicio y Y el conteo despues de un año de tratamiento.

Con una muestra $(X_1, Y_1), (X_2, Y_2), ..., (X_{20}, Y_{20})$ podemos calcular un estimador de su coefficiente de correlación como, siendo en nuestro caso partícular igual a,

$$\hat{\theta}_{xy} = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n\sum x_i^2 - (\sum x_i)^2}\ \sqrt{n\sum y_i^2 - (\sum y_i)^2}} = 0.7231$$

0	baseline	oneyear
1	2.12	2.47
2	4.35	4.61
3	3.39	5.26
4	2.51	3.02
5	4.04	6.36
6	5.10	5.93
7	3.77	3.93
8	3.35	4.09
9	4.10	4.88
10	3.35	3.81
11	4.15	4.74
12	3.56	3.29
13	3.39	5.55
14	1.88	2.82
15	2.56	4.23
16	2.96	3.23
17	2.49	2.56
18	3.03	4.31
19	2.66	4.37
20	3.00	2.40

Figura 1: Dataset CD4

.

Utilizamos los mismos métodos que en el ejercicio anterior para aproximar la distribución de nuestro estimador $\hat{\theta}$ utilizando el método de **Bootstrap No Parámetrico** y contruimos los dos intervalos de confianza con los métodos **Básico** y **Percentil**. Obteniendo los siguientes resultados, con $B = 10,000$,

$$Basico = [0.5825, 0.9455] \tag{5}$$

$$Percentil = [0.5007, 0.8638] \tag{6}$$

b) **Calcule el coefiente estimado de correlación corregido por sesgo usando el jackknife.**

A continuación presentamos el estimador Jacknife para nuestro estimador muestral del coeficiente de correlación $\hat{\theta}$:

$$\hat{\theta}_{(.)} = \frac{1}{n} \sum_{i=1}^{n} \hat{\theta}_{(i)}$$

Siendo $\hat{\theta}_{(i)}$ el estimador resultante de quitar el **i-esimo** elemento (x_i, y_i) de la muestra y volver a calcular el coefficiente de correlación. Finalmente el estimador corregido por sesgo usando jacknife se calcula de la siguiente manera,

$$\hat{\theta}_{\text{jack}} = \hat{\theta} - \widehat{\text{bias}}_{(\theta)} = n\hat{\theta} - (n-1)\hat{\theta}_{(.)}$$

Esto se implemento en python de la siguiente manera,

Algorithm 2: Posterior Normal

```
def jacknife_bias_corrected(sample, estimator):
    # Estimate with sample
    est = estimator(sample)

    # Sample size
    n = len(sample)

    # Estimated values list
    jkestimated_lst = []

    # Jacknife estimator
    for idx in range(n):
        # Drop idx element
        jckknf_smpl = sample[:idx] + sample[idx + 1:]
        jkestimated_lst.append(estimator(jckknf_smpl))

    jkestimator = np.mean(jkestimated_lst)

    # Estimate bias
    bias = (n - 1.0) * (jkestimator - est)

    # Bias corrected
    bc_jkestimator = est - bias

    return bc_jkestimator
```

Obteniendo el siguiente resultado para **coefiente estimado de correlación corregido por sesgo**,

$$\hat{\theta}_{\text{jack}} = 0.7299$$

Referencias

[1] Notas sobre Bootstrap - Rogelio Ramos Quiroga

[2] Jackknife Resampling - https://en.wikipedia.org/wiki/Jackknife_resampling