* En primer lugar, cree el vector de datos con una distribución normal (con media y desviación que prefiera, pero al menos 10 datos)
* Defina su primera función target como la función identidad
* Llame al mecanismo de Laplace sobre su función target

mech = DPMechLaplace(target = miFuncionTarget);

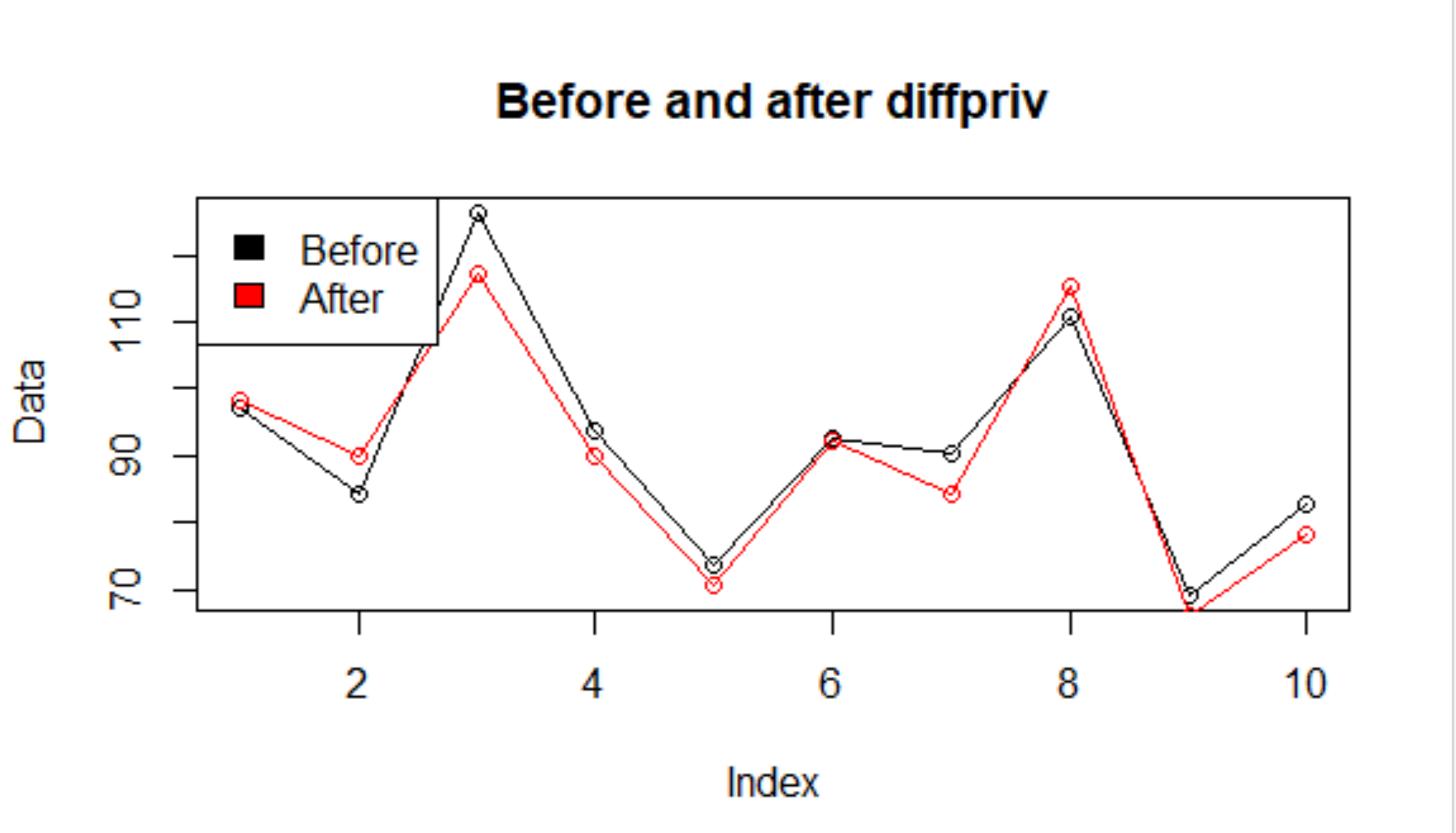
* Cree una función que llame a la distribución normal y recalcule mech asegurándose que n sea igual al tamaño de su vector de datos. Finalmente calcule r

distr <- function(n) rnorm(n);

mech <- sensitivitySampler(mech, oracle = distr, n = length(data), gamma = 0.1);

r = releaseResponse(mech, privacyParams = DPParamsEps(epsilon = 1), X = misDatos)

* Por último, grafique sus datos iniciales y sus datos nuevos (puede obtener los datos nuevos usando r$response). Debería obtener un resultado como el siguiente:



Para la segunda parte deseamos comparar epsilon (que debe ser cambiado en los parámetros del releaseResponse) contra la precisión, entiéndase por precisión como que tan cercano es un dato modificado del original. Para ello mediremos precisión como 1-error porcentual entre la media de los datos originales y la media de los datos aplicándoles privacidad diferencial. Usaremos entonces mean() como nuestra función target, que hace las veces del query que nos interesa evaluar sobre la base de datos, y calcularemos 1 – error porcentual para epsilons entre 0.01 y 100 (use al menos 1000 puntos en medio)

Sugerencia: para la gráfica utilice escala logarítmica

Nota: Entiéndase que un error de 0% indica una alta precisión y que entre más negativo el error más baja es la precisión

Su grafica debería lucir así:

