-- Estimate overall k by trial and error

DROP TABLE #PossibleK, #K

CREATE TABLE #PossibleK (K decimal (3,2))

INSERT INTO #PossibleK VALUES (-1.), (-.95),(-.9),(-.85), (-.8),(-.75), (-.7),(-.65), (-.6),(-.55), (-.5),(-.45), (-.4),(-.35), (-.3) , (-.25), (-.2),(-.15), (-.1), (-.05),(.05),(+.1), (.15),(+.2),(.25),(+.3),(.35),(+.4),(.45),(+.5),(.55),(+.6), (.65),(+.7),(.75),(+.8),(.85)

 ,(+.9),(.95),(+1.) -- inserts possible k values into table #PossibleK

SELECT top 1 K -- several k values may fit the equation, select 1

 , (-1+ Exp(sum(Log(1+ (CASE WHEN Intercept is null THEN 0. Else Intercept End)\*k))))/k as [Near One]

INTO #K -- Save the optimal k value

FROM #Intercept cross join #PossibleK -- try different k values

Group by K

HAVING (-1+ Exp(sum(Log(1+ (CASE WHEN Intercept is null THEN 0. Else Intercept End) \*k))))/k between 0.99 and 1.01 -- right side of equation divided by left side should be near 1

ORDER BY RAND() -- select among correct k values randomly