Ran Simulation for 2000 Rounds (each round consisted for simulated 4775 pages)

**Page Level Algorithm**

N = number of pages to simulate

For x=0;x<N;x++:

expAges = [] //Empty List

respSizes = [] //Empty List

//Determine whether or not page will contain expAges >= 86400

Y = randomvalue(1,4775)

If Y < 3046:

Large\_expAge = True //ExpAges CAN be >= 86,400 (can also still be < 86400 too)

Else:

Large\_expAge = False //ExpAges can’t be >= 86,400

//Determine initial proportion of expAges that will be set to zero

PropInitialZeros = **generateProportionZeros()**

**//**Determine number of responses for page

numResponses = **generateNumResponses()** //numResponses must be > 1

//Multiply proportion of zero-value expAges requests by number of requests to get number of //requests to initially set expAge to zero for.

numZeroExpAges = Math.ceil(numResponses \* PropInitialZeros)

For r=0;r<numZeroExpAges;r++ //Put in initial proportion of zero values expAges into list

expAges.append(0)

For s = 0;s< numResponses – numZeroExpAges;s++ //Generate rest of expAges into list

expAges.append(**generateExpAge(**Large\_expAge**)**)

For t = 0;t< numResponses – numZeroExpAges;t++ //Generate respSizes into list

respSizes.append(**generateRespSize()**)

//Output average of expAges list for average expAges of page

//Output average of respSizes list for average respSizes of page

**generateNumResponses()**

return Math.Ceiling(gengamma(alpha=3.1119,k=0.6167,scale=8.2912,loc=0.17238))

**generateExpAge (boolean Large\_expAge)**

expAge = **determine\_bin\_expAge(**random\_value(0,289355))

if(Large\_expAge == True):

while expAge > 604800: //Not concerned with expAges > 1 week, redraw value

expAge = **determine\_bin\_expAge**(random\_value(0,289355))

else:

while expAge >= 86400: //Page expAge must be < 86400 (large\_expAge is false)

expAge = **determine\_bin\_expAge**(random\_value(0,289355))

return expAge

**determine\_bin\_expAge(int x)** //Maps the integer to the appropriate range to generate expAge within

If x < 122004:

expAge = 0

ElIf x < 156484:

expAge = Generate expAge value from fisk distribution (c=0.88945,loc=1,scale=2600.7)

ElIf x < 160381

expAge = Generate random expAge between [36001, 42000]

ElIf x < 162203

expAge = Generate random expAge [43201,79200]

ElIf x < 176796

expAge = Generate random expAge [79201,86400]

ElIf x < 177367

expAge = Generate random expAge [86401,165600]

ElIf x < 179332

expAge = Generate random expAge [165601,172800]

ElIf x < 179788

expAge = Generate random expAge [172801,252000]

ElIf x < 181320

expAge = Generate random expAge [252001,259200]

ElIf x < 182440

expAge = Generate random expAge [259201,424800]

ElIf x < 183139

expAge = Generate random expAge [424801,432000]

ElIf x < 183985

expAge = Generate random expAge [432001,597600]

ElIf x < 199734

expAge = Generate random expAge [597601,604800]

Else

Value over 604,800. Only concerned with values less than or equal to week. Generate a different expAge (generate another random number and try to map it to the

**generateRespSize()**

return weibull\_min(a=0.27,scale=700,loc=2.5) //Weibull\_min dist, aka Frechet right distribution

**generateProportionZeros()** //Generates the the proportion of responses that will be have value of zero

return normal(0.629075947,0.300892868) //Normal Distribution