**A**nd here is the third LeMonde puzzle  open for competition:

*Consider for this puzzle only integers with no zero digits. Among these an integer x=a¹a²a³… is* refined *if it is a multiple of its scion, the integer defined as x without the first digit, y=a²a³…. Find the largest refined integer x such the sequence of the successive scions of x with more than one digit is entirely refined. An integer x=a¹a²a… is distilled if it is a multiple of its grand-scion, the integer defined as x without the first two digits, z=a³… Find the largest distilled integer x such the sequence of the successive scions of x with more than two digits is entirely distilled.*

Another puzzle amenable to a R resolution by low-tech exploration of possible integers, first by finding refined integers, with  no solution between 10⁶ and 10⁸ [*meaning there is no refined integer larger than 10⁶*] and then checking which refined integers have refined descendants, e.g.,

raf=NULL

for (x in (1e1+1):(1e6-1)){

y=x%%(10^trunc(log(x,10)))

if (y>0){

if (x%%y==0)

raf=c(raf,x)}}

that returns 95 refined integers. And then

for (i in length(raf):1){

gason=raf[i]

keep=(gason%in%raf)

while (keep&(gason>100)){

gason=gason%%(10^trunc(log(gason,10)))

keep=keep&(gason%in%raf)}

if (keep) break()}}

that returns 95,625 as the largest refined integer with the right descendance. Rather than finding all refined integers at once, going one digit at a time and checking that some solutions have proper descendants is actually faster.

Similarly, running an exploration code up to 10⁹ produces 1748 distilled integers, with maximum 9,977,34,375, so it is unlikely this is the right upper bound but among these the maximum with the right distilled descendants is 81,421,875. As derived by

rad=(1:99)[(1:99)%%10>0]

for (dig in 2:12){

for (x in ((10^dig+1):(10^{dig+1}-1))){

y=x%%(10^{dig-1})

if (y>0){

if (x%%y==0){

if (min(as.integer(substring(x,seq(nchar(x)),seq(nchar(x)))))>0){

rad=c(rad,x)

y=x%%(10^dig)

keep=(y%in%rad)

while (keep&(y>1e3)){

y=y%%(10^trunc(log(y,10)))

keep=keep&(y%in%rad)}

if (keep) solz=x}}}}

if (solz<10^dig) break()

topsol=max(solz)}