

1. This week's Problem of the Week in Math is described as follows:

There are thirty positive integers less than 100 that share a certain property. Your friend, Blake, wrote them down in the table to the left. But Blake made a mistake! One of the numbers listed is wrong and should be replaced with another. Which number is incorrect, what should it be replaced with, and why?

The numbers are listed below.

6	10	14	15	21
22	26	33	34	35
38	39	46	51	55
57	58	62	65	69
75	77	82	85	86
87	91	93	94	95

Use the fact that the “certain” property is that these numbers are all supposed to be the product of *unique* prime numbers to find and fix the mistake that Blake made.

Reminder: Code your solution in an R script and copy it over to this `.Rnw` file.

Hint: You may find the `%in%` operator and the `setdiff()` function to be helpful.

Solution: Given a set prime numbers, we can find all possible combinations of products of unique prime numbers. For this problem, a set of the first 16 prime numbers is enough to find the solution. Given this set, we can loop through to find all products of unique primes. Given our output set of `unique.products`, we can sort it in non-decreasing order, and filter out all numbers ≥ 95 , given that the numbers we are checking against(`nums.to.check`) has highest element, 95. With our filtered `unique.products`, we can find the one number that exists in `nums.to.check` and not `unique.products`, which we find to be 75.

Also, since our `unique.products` is sorted in non-decreasing order(same as `nums.to.check`), the replacement for 75 will be the the at same index in `unique.products`, as 75 is in `nums.to.check`. We find the replacement to be 74.

```
#set of prime numbers
prime.nums <- c(2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 39, 41, 43, 47)

#create a set of the products of unique primes
i <- 1
j <- 2
index <- 1
unique.products = c()
for (i in 1:length(prime.nums)){
  for (j in (i+1):length(prime.nums)){
    product <- prime.nums[i] * prime.nums[j]
    unique.products <- c(unique.products, product)
  }
}
nums.to.check <- c( 6, 10, 14, 15, 21,
                  22, 26, 33, 34, 35,
                  38, 39, 46, 51, 55,
                  57, 58, 62, 65, 69,
                  75, 77, 82, 85, 86,
                  87, 91, 93, 94, 95)

#sorting our set of products in non-decreasing order
unique.products <- sort(unique.products)
#filtering out all numbers > 95
unique.products <- unique.products[unique.products <= 95]

#the wrong number will be the number that exists in our nums.to.check but not our unique
(nums.to.check[which(!(nums.to.check %in% unique.products))])

## [1] 75

#should be replaced by:
(replacement <- unique.products[which(nums.to.check == 75)])

## [1] 74
```