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:

We know there are 30 integers that are the product of unique prime numbers. So, to solve this problem, we first want to create a vector of those 30 correct integers. I loaded the `primes` library and used the `generate_primes()` function to do that generate a list of prime numbers, whose products I compared to Blake's list (Keyes and Egeler, 2025). Before comparing the prime products, I cleaned my vector of prime prouducts so it would only include those of two *unique* numbers with a product less than 100.

We can then compare that list with Blake's list, using `setdiff()`, which outputs the value that is different between the two vectors. This outputs the wrong number in Blake's list and the missing correct number in our list.

The wrong number in Blake's list was 75, while the corrrect number that is missing is 74.

```
#Libraries to load
library(primes)

#generating a prime number list with a max of 95 because we want
#their products that are less than 100
prime.nums = generate_primes(max = 100)

#list to be filled in later
product.of.primes = c()


#looping through the products of all the primes we generated
for (num in prime.nums){
  #list of prime products with num
  prime.products = num*prime.nums
  #gets rid of square of num
  prime.products = prime.products[which(prime.products != num*num)]
  #Keep only products <= 100
  prime.products = prime.products[which(prime.products <= 100)]
  #combines them into a list of the unique prime products less than 100
  #sorted to ascending numeric value for ease of viewing
  product.of.primes = sort(union(product.of.primes, prime.products))
}


blake.list = 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)
```

```r
wrong.num = setdiff(blake.list, product.of.primes)
correction = setdiff(product.of.primes, blake.list)

paste("The wrong number in Blake's list is ", toString(wrong.num),
      ". The correct number is ", toString(correction), ".", sep = "")


## [1] "The wrong number in Blake's list is 75. The correct number is 74."
```

References

Keyes, O. and Egeler, P. (2025). *primes: Fast Functions for Prime Numbers*. R package version 1.6.1.