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:

```r
# Solution

primeFactor = function(n) {
  factors = c() # empty vector to store prime factors
  i = 2 # starts checking divisibility from the smallest prime number, 2

  while(n > 1){
    while (n %% i == 0) { # checks if i is a factor of n
      factors = c(factors, i) # i gets added to vector of prime factors
      n = n/i # to find other factors of n other than the current i
    }
    i = i + 1 # increment to check next potential factor
    if (i * i > n){ # checks if n is prime or has no smaller divisors left
      if (n > 1){ # if n is greater than one after the last conditional it must be prime
        factors = c(factors, n) # add the prime number n to the vector
      }
      break # once all factors are found break loop
    }
  }
  return (factors) # returns all factors of n
}

product.of.unique.prime = function(n){
  factors = primeFactor(n) # gets the prime factors of n
  return (length(factors) == length(unique(factors))) # returns TRUE if number of factors equals number of unique factors
}

replacement = function(n){
  # defines the number below and above the "wrong" number
  l = n - 1
  u = n + 1

  while(TRUE){ # runs infinitely until a replacement is found
    if (product.of.unique.prime(l)){ # checks if the lower number is a product of unique primes
      return (l)
    }
    if (product.of.unique.prime(u)){
      return (u) # checks if the higher number is a product of unique primes
    }
    # expands the search decreasing l and increasing u
    l = l - 1
    u = u + 1
  }
}

# Problem
nums = 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)

for (num in nums) {
  if (!product.of.unique.prime(num)){
    print(num)
    repl = replacement(num)
    print(paste("Replace", num, "with", repl))
  }
}
```

```
## [1] 75
## [1] "Replace 75 with 74"
```

This R script analyzes a list of numbers (nums) to determine whether each number is a product of unique prime factors. The `primeFactor()` function extracts all prime factors of a given number, while the `product.of.unique.prime()` function checks whether all factors are distinct. If a number contains repeated prime factors, it is considered invalid. For these invalid numbers, the `replacement()` function searches for the nearest valid number checking decreasing (n-1) and increasing (n+1) values until it finds a valid number.