

# Solution Review: Find the Highest Achiever

In this review, we give a detailed analysis of the solution to the problem of finding the highest achiever.

## We'll cover the following

- Solution: Merging Data Frames
  - Explanation
    - Steps Performed:

## Solution: Merging Data Frames #





main.r

math.csv

english.csv

science.csv

All code files are copied to end of the page...



## Explanation #

The code starts executing from **line number 47** when the function `findTopper()` is executed.

This function starts executing from **line number 22** (main file handling code)

Steps Performed: #

- **Line number 23-25:** Read all the subject files in the variables: `mathData`, `englishData` and `scienceData` respectively. These will also act as the data frames.

Remember, data fetched from a `.csv` is already in the form of a data frame.

- **Line number 27-28:** Merge the three data frames into one data frame. In the code snippet above, we have broken merging of the three data frames into two steps. First, merge `mathData` and `englishData` and save in `tempData`. Then merge `tempData` and `scienceData` in `finalData`.
- Now that we have all the data compiled in one data frame `finalData`, we can begin performing our analysis on it.
- **Line number 32-43:** We use nested `for` loop to iterate over the whole data frame. The outer loop:

```
for(student in 1:length(finalData))
```

keeps track of the rows/students. Since the value of `length(finalData)` is 4 we are basically executing loop from student 1 to student 4.

The inner loop:

```
for(i in 2:ncol(finalData))
```

iterates over all the columns (math, english, science). Notice, we iterate from column 2 to `ncol(finalData)` because the 1<sup>st</sup> column is just names of students. Then we add the marks of all subjects of each student. The loop can be illustrated as follows:

finalData

name	maths	english	science
Andrew	2.5	8.2	5.2
Dany	1.9	7.5	7.1
Mathew	5.9	2.5	9.5
Phillip	9.1	9.3	1.9

finalData

name	maths	english	science
Andrew	2.5	8.2	5.2
Dany	1.9	7.5	7.1
Mathew	5.9	2.5	9.5
Phillip	9.1	9.3	1.9

temp



result



finalData

name	maths	english	science
Andrew	2.5	8.2	5.2
Dany	1.9	7.5	7.1
Mathew	5.9	2.5	9.5
Phillip	9.1	9.3	1.9

temp

15.9

result



finalData

name	maths	english	science
Andrew	2.5	8.2	5.2
Dany	1.9	7.5	7.1
Mathew	5.9	2.5	9.5
Phillip	9.1	9.3	1.9

temp

16.5

result

15.9

finalData

name	maths	english	science
Andrew	2.5	8.2	5.2
Dany	1.9	7.5	7.1
Mathew	5.9	2.5	9.5
Phillip	9.1	9.3	1.9

temp

17.9
------

result

15.9	16.5
------	------

finalData

name	maths	english	science
Andrew	2.5	8.2	5.2
Dany	1.9	7.5	7.1
Mathew	5.9	2.5	9.5
Phillip	9.1	9.3	1.9

temp

20.3

result

15.9 16.5 17.9



finalData

name	maths	english	science
Andrew	2.5	8.2	5.2
Dany	1.9	7.5	7.1
Mathew	5.9	2.5	9.5
Phillip	9.1	9.3	1.9

temp



result

15.9	16.5	17.9	20.3
------	------	------	------

finalData

name	maths	english	science
Andrew	2.5	8.2	5.2
Dany	1.9	7.5	7.1
Mathew	5.9	2.5	9.5
Phillip	9.1	9.3	1.9

temp



result

15.9	16.5	17.9	20.3
------	------	------	------

Maximum marks of student 4 (here index 4)

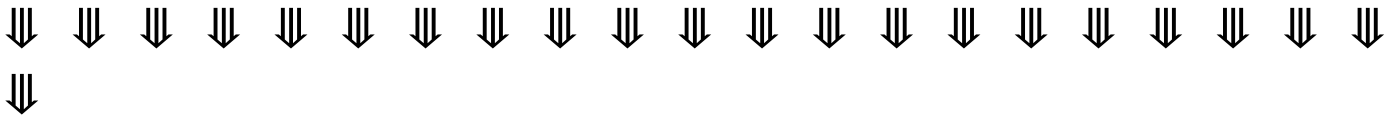
8 of 8



The `findIndexWithMaxNum()` returns the index of the largest/maximum number in an array. We have created this as a helper function to find the index of the maximum element in the `result` vector.

In the next chapter, we discuss installing and loading packages in **R** language.

## Code Files Content !!!



```
-----  
|  main.r [1]  
-----
```

```
findTopper <- function()  
{  
  findIndexWithMaxNum <- function(myVector) # helper function that returns the index  
                                              # of the element that has the highest value  
  {  
    maxNumber = -Inf # We want this to be the lowest possible value for comparison  
    maxIndex = 0  
  
    index = 1  
    for(i in myVector)  
    {  
      if(maxNumber < i)  
      {  
        maxNumber = i # set the max element  
        maxIndex = index # set the max element's index  
      }  
      index = index + 1  
    }  
    return(maxIndex) # return the index of the max element  
  }  
  
  # MAIN FILE HANDLING CODE  
  mathData = read.csv("math.csv") # fetch data from math.csv  
  englishData = read.csv("english.csv") # fetch data from english.csv  
  scienceData = read.csv("science.csv") # fetch data from science.csv  
  
  tempData <- merge(mathData, englishData) # we use the merge function on data frames  
  finalData <- merge(tempData, scienceData) # another merge function to merge the remaining data  
  
  print(finalData)  
  result <- vector("numeric", 0) # vector to store the total marks of each student  
  
  for(student in 1:length(finalData)) # loop over all the rows/students  
  {  
    temp <- 0.0 # temporarily stores the total marks of the current student  
  
    for(i in 2:ncol(finalData)){ # loop over all the columns (math, english, science)  
      # We iterate from 2 to ncol(finalData) because the 1st column is just names of students  
  
      temp <- temp + as.double(finalData[student, i]) # fetch respective student's marks  
    }  
    result <- c(result, temp) # store the total marks of the current student  
  }  
  return(findIndexWithMaxNum(result)) # return the index of the highest scoring student  
}  
  
# Driver Code  
findTopper()
```

-----  
math.csv [1]

Name,	Math
Andrew,	2.5
Mathew,	5.9
Dany,	1.9
Philip,	9.1

-----  
english.csv [1]

Name,	English
Andrew,	8.2
Mathew,	2.5
Dany,	7.5
Philip,	9.3

-----  
science.csv [1]

Name,	Science
Andrew,	5.2
Mathew,	9.5
Dany,	7.1
Philip,	1.9

\*\*\*\*\*