ELEC IT-E1

IT ELECTIVE 1

Name: Eustaquio, Maria Luisa O.	**Year & Section:** BSIT 3-5	**Date:** November 25, 2024

ACTIVITY #6 - USING APPLY FAMILY FUNCTIONS

Instructions: You are given a dataset with students' grades in three subjects: Math, Science, and English.

```
students <- data.frame(
  Name = c("John", "Alice", "Bob", "David", "Emma"),
  Math = c(85, 90, 78, 92, 88),
  Science = c(82, 88, 76, 95, 85),
  English = c(89, 91, 80, 88, 94)
)
```

students

Filter

	Name	Math	Science	English
1	John	85	82	89
2	Alice	90	88	91
3	Bob	78	76	80
4	David	92	95	88
5	Emma	88	85	94

- Using the apply() function, calculate the average score for each student across all three subjects (Math, Science, and English). Add a new column to the students dataset that contains the average score.

```
students$Average <- apply(students[, c("Math", "Science", "English")], 1, mean)
```

Name	Math	Science	English	Average
John	85	82	80	82.33333
Alice	90	88	85	87.66667
Bob	78	76	70	74.66667
David	92	95	90	92.33333
Emma	88	85	75	82.66667

- Using the lapply() function, calculate the mean score for each subject (Math, Science, English). Display the result as a list where each element represents the mean score of a subject.

```
students_mean <- lapply(students[, c("Math", "Science", "English")], mean)
```

```
> print(students_mean)
$Math
[1] 86.6

$Science
[1] 85.2

$English
[1] 80
```

- Using the sapply() function, calculate the total score for each subject (Math, Science, and English). Display the result as a vector where each element represents the total score of a subject.

```
students_total <- sapply(students[, c("Math", "Science", "English")], sum)
```

```
> print(students_total)
  Math Science English
   433     426     400
```

- Using the apply() function, calculate the highest score in each subject (Math, Science, and English). Use apply() to perform the operation on the columns (i.e., find the maximum value in each column).

```
students_max <- apply(students[, c("Math", "Science", "English")], 2, max)
```

```
> print(students_max)
  Math Science English
    92      95      90
```

- Using lapply(), create a new list where each element contains a vector of scores greater than or equal to 85 for each subject. For example, in the "Math" subject, the result should be a list of scores in that subject that are greater than or equal to 85.

```
students_high <- lapply(
  students[, c("Math", "Science", "English")],
  function(x) x[x >= 85]
)
```

```
> print(students_high)
$Math
[1] 85 90 92 88

$Science
[1] 88 95 85

$English
[1] 85 90
```

- Using sapply(), calculate the standard deviation of scores for each subject. Display the result as a vector where each element is the standard deviation of a subject.

```
students_sd <- sapply(students[, c("Math", "Science", "English")], sd)
```

```
> print(students_sd)
    Math  Science  English
5.458938 7.049823 7.905694
```

- Using the apply() function, create a new column in the students dataset that contains "Pass" if the average score of the student is greater than or equal to 85, and "Fail" if it is less than 85. Use apply() to check each student's average score and assign the respective result.

```
students$Result <- apply(
  students[, c("Math", "Science", "English")], 1,
  function(row) if (mean(row) >= 85) "Pass" else "Fail"
)
```

Name	Math	Science	English	Average	Result
John	85	82	80	82.33333	Fail
Alice	90	88	85	87.66667	Pass
Bob	78	76	70	74.66667	Fail
David	92	95	90	92.33333	Pass
Emma	88	85	75	82.66667	Fail

- Create a new list using lapply() that contains the count of scores greater than or equal to 90 for each subject. Then, use the apply() function to find the total number of students with scores greater than or equal to 90 across all subjects.

```
students_count_90 <- lapply(
  students[, c("Math", "Science", "English")],
  function(x) sum(x >= 90)
)
```

```
> print(students_count_90)
$Math
[1] 2

$Science
[1] 1

$English
[1] 1
```

```
students_total_90 <- apply(
  students[, c("Math", "Science", "English")], 1,
  function(row) sum(row >= 90)
)
```

```
> print(students_total_90)
[1] 0 1 0 3 0
```