# Class 6: R functions

Georgina Canto-Encalada (A59021295)

## Our first silly function

All functions in R have 3 parts. They have:

- a name
- input arguments (none,one or more)
- a body

A function to add two numbers

```
sillyadd <- function(x,y=1) {
  x+y
}
```

Let me try out this function

```
sillyadd(100)
```

```
[1] 101
```

## Let's do something more useful

## Since Covid is in the air, I will not be so harsh with them. Then, I will consider a missing assigment as the lowest one. If there is more than one missing assigment, one of them will be dropped and the others will be considarated as zero

```
student1 <- c(100, 100, 100, 100, 100, 100, 100, 90)
student2 <- c(100, NA, 90, 90, 90, 90, 97, 80)
student3 <- c(90, NA, NA, NA, NA, NA, NA, NA)
grades <- data.frame(student1,student2,student3)
grades[is.na(grades)] <- 0
grades
```

```
  student1 student2 student3
1      100      100       90
2      100        0        0
3      100       90        0
4      100       90        0
5      100       90        0
6      100       90        0
7      100       97        0
8       90       80        0
```

```
grade<- function(x,drop.lowest=TRUE){

  # Transform all NA values to 0
  x[is.na(x)] <- 0
  if(drop.lowest){
    # Find the index of the minimum value
    min <- x[-which.min(x)]
    # Measure the mean of their grades
    ans<-mean(min)
  }
  else {
    ans<-mean(x)

  }
```

```
}
```

```
grade(student1)
grade(student2)
grade(student3)
```

Read a class gradebook CSV file from here: "https://tinyurl.com/gradeinput"

```
url<- "https://tinyurl.com/gradeinput"
gradebook <- read.csv(url,row.names=1)
gradebook
```

```
           hw1 hw2 hw3 hw4 hw5
student-1  100  73 100  88  79
student-2   85  64  78  89  78
student-3   83  69  77 100  77
student-4   88  NA  73 100  76
student-5   88 100  75  86  79
student-6   89  78 100  89  77
student-7   89 100  74  87 100
student-8   89 100  76  86 100
student-9   86 100  77  88  77
student-10  89  72  79  NA  76
student-11  82  66  78  84 100
student-12 100  70  75  92 100
student-13  89 100  76 100  80
student-14  85 100  77  89  76
student-15  85  65  76  89  NA
student-16  92 100  74  89  77
student-17  88  63 100  86  78
student-18  91  NA 100  87 100
student-19  91  68  75  86  79
student-20  91  68  76  88  76
```

We can "apply" our new grade() function over wither the rows or the columns of the gradebook, with MARGIN=1 pr MARGIN=2

```
results <- apply(gradebook,1,grade)
results
```

```
student-1  student-2  student-3  student-4  student-5  student-6  student-7
```

```
     91.75      82.50      84.25      84.25      88.25      89.00      94.00
 student-8  student-9 student-10 student-11 student-12 student-13 student-14
     93.75      87.75      79.00      86.00      91.75      92.25      87.75
student-15 student-16 student-17 student-18 student-19 student-20
     78.75      89.50      88.00      94.50      82.75      82.75
```

**Q2.** Using your grade() function and the supplied gradebook, Who is the top scoring student overall in the gradebook? [3pts]

```
which.max(results)
```

```
student-18
        18
```

**Q3.** From your analysis of the gradebook, which homework was toughest on students (i.e. obtained the lowest scores overall? [2pts]

```
which.min(apply(gradebook,2,mean,na.rm=T))
```

```
hw3
  3
```

**Q4.** Optional Extension: From your analysis of the gradebook, which homework was most predictive of overall score (i.e. highest correlation with average grade score)? [1pt]

```
mask<-gradebook
mask[is.na(mask)]<-0
mask
```

```
           hw1 hw2 hw3 hw4 hw5
student-1  100  73 100  88  79
student-2   85  64  78  89  78
student-3   83  69  77 100  77
student-4   88   0  73 100  76
student-5   88 100  75  86  79
student-6   89  78 100  89  77
student-7   89 100  74  87 100
student-8   89 100  76  86 100
student-9   86 100  77  88  77
student-10  89  72  79   0  76
```

```
student-11  82  66  78  84 100
student-12 100  70  75  92 100
student-13  89 100  76 100  80
student-14  85 100  77  89  76
student-15  85  65  76  89   0
student-16  92 100  74  89  77
student-17  88  63 100  86  78
student-18  91   0 100  87 100
student-19  91  68  75  86  79
student-20  91  68  76  88  76
```

```
which.max(apply(mask,2,cor,y=results))
```

```
hw5
  5
```