---

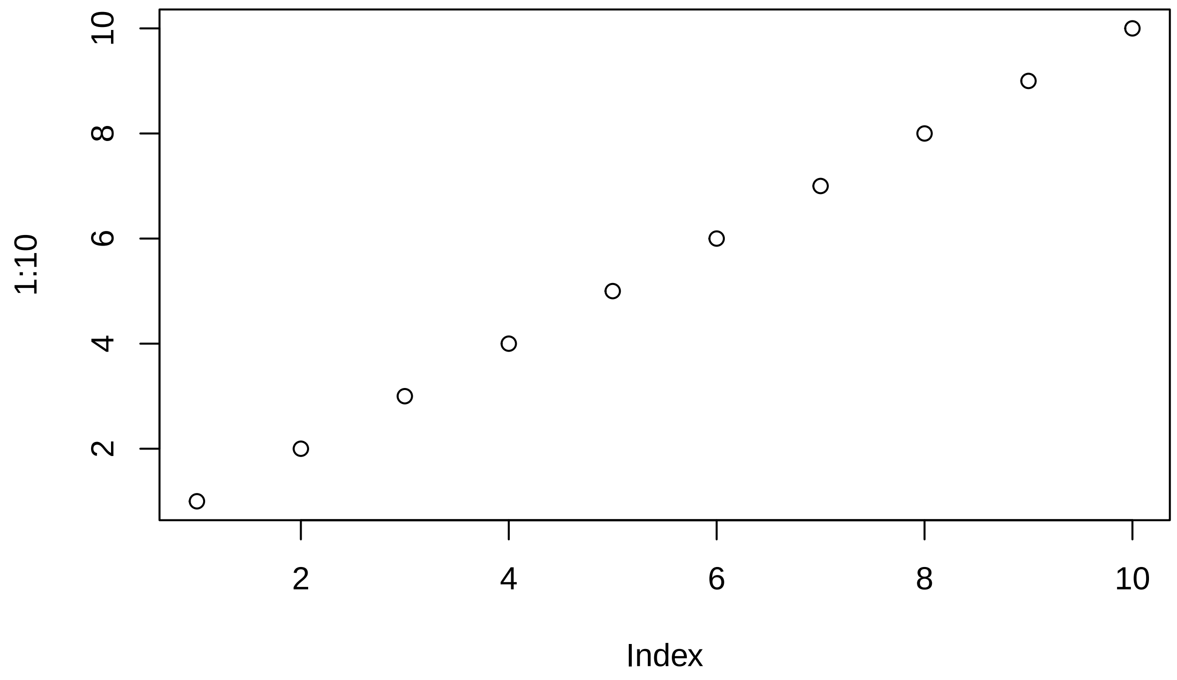
title: 'Class6: R Functions' author: "R(PID:A59010419" date: "10/15/2021" output: html_document

Quick Rmarkdown Tutorial

We can Write text. **Style** or *style*

student1 <- c(100, 100, 100, 100, 100, 100, 100, 90) student2 <- c(100, NA, 90, 90, 90, 90, 97, 80)

```
plot(1:10)
```



```
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)
```

First I think we should Find if there is an NA and set it to 0 then find and keep the place in the array that arent the minimum (logical)

```
student<-student1
student[is.na(student)] = 0
a=student[-which.min(student)]
mean(a)
```

```
## [1] 100
```

The is.na() function returns logical True if NA my intution is what we ended up doing as a class.

```
studentp<-student2
studentp[is.na(studentp)] = 0
mean(studentp[-which.min(studentp)])
```

```
## [1] 91
```

```
studentp<-student3
studentp[is.na(studentp)] = 0
mean(studentp[-which.min(studentp)])
```

```
## [1] 12.85714
```

This works for all the students lets make the functuin actually lets make numeric first

```
x <- student3
x <- as.numeric(x)
x[is.na(x)]=0
mean(x[-which.min(x)])
```

```
## [1] 12.85714
```

okay now function

```
grade <- function(x) {
x <- as.numeric(x)
x[is.na(x)]=0
mean(x[-which.min(x)])
}
```

test

```
grade(student1)
```

```
## [1] 100
```

```
grade(student2)
```

```
## [1] 91
```

```
grade(student3)
```

```
## [1] 12.85714
```

now lets play with this finle

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

```
##            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
```

Cool we have a data set now

We havent used this function yet but APPLY() applies function to our data set… this is cool because it replaces a loop

```
ans <- apply(scores,MARGIN=1,grade)
```

Q2 top scorer

```
which.max(ans)
```

```
## student-18
##         18
```

Q3 Which HW was the hardest (lowest score)

```
apply(scores,2,grade)
```

```
##      hw1      hw2      hw3      hw4      hw5
## 89.36842 76.63158 81.21053 89.63158 83.42105
```

Well this drops one lowest score oops replace NA in grade book

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

Now function, I think sum is better than average personally

```
sumscore<-apply(mask,2,sum)
which.min(sumscore)
```

```
## hw2
##   2
```

Homework 2 was the hardest probably.

okay we can use the mean

```
sumscore<-apply(mask,2,mean)
which.min(sumscore)
```

```
## hw2
##   2
```

Q4

```
cor(mask$hw5,ans)
```

```
## [1] 0.6325982
```

Or use apply apply(mask,2,cor)

```
corvec<-apply(mask,2,cor,ans)
which.max(corvec)
```

```
## hw5
##   5
```