

# InteresSimpleEquipo[6]

Miguel Sandoval, Juan Carrillo, Aldo Hernandez y Rosendo Hernandez



Integrantes

- Miguel Sandoval Millán 2408819j@umich.mx
- Juan Carrillo Salazar 2408812d@umich.mx
- Aldo Hernandez Perez 2405155c@umich.mx
- Rosendo Hernandez Arias 2350465c@umich.mx

## Condicionales

Los condicionales se usan para que el programa se ejecute en base a una condicion. Si la condicion se cumple el programa realizara la accion puesta dentro del condicional, pero si la condicion no se cumple el programa no realizara. Hay diferentes tipos de condicionales de los cuales tocaremos dos:

- if
- else if

### Ejemplos usando if

```
x = 5
if(x>1){
    resultado = "x es mayor a 1" # Nos regresa el valor
}
print(resultado)
```

```
[1] "x es mayor a 1"
```

## Ejemplos usando else if

```
x = 5
if(x>1){
  resultado = "x es mayor a 1" # Nos indicara si x es mayor a 1
}else if(x<1){
  resultado= "x es menor a 1" # Nos indicara si x es menor a 1
}
print(resultado)
```

```
[1] "x es mayor a 1"
```

## Funcion con Condicional (Valor Futuro)

Una vez visto los condicionales. Necesitaremos meter un condicional en la funcion, donde este detectara los argumentos que se encuentran dentro de la funcion.

Utilizaremos el if para la funcion, ahora necesitamos que la funcion sea capaz de calcular el valor futuro si y solo si se dectan los los argumentos en la funcion.

Investigando un poco vimos que necesitamos un “verificador de argumentos” para comprobar si uno de los argumentos es nulo (no tiene valor o no existe).

Entonces ocuparemos este verificador de argumento.

```
"!is.null ( ) ”
```

Probemos meter la condicional dentro de la funcion.

```
interesSimpleBase = function(va, r, n){
  if(!is.null(va) && !is.null(r) && !is.null(n)){
    VF = va*(1 + r*n)
    return(VF)}
}
```

```
miVF = interesSimpleBase(1,1,1)
miVF
```

```
[1] 2
```

## Funcion de Interes Simple con cualquier argumento

Pero ahora necesitamos que la computadora entienda que necesitamos calcular con los argumentos que le damos

```
interesSimple = function(va = NULL, r = NULL, n = NULL, VF = NULL) {  
  
  if(is.null(VF)){  
    VF = va*(1 + r*n)  
    return(VF)  
  }  
  
  else if(is.null(va)){  
    va = VF/(1 + r*n)  
    return(va)  
  }  
  
  else if(is.null(r)){  
    r = ((VF/va) - 1)/n  
    return(r)  
  }  
  
  else if(is.null(n)){  
    n = ((VF/va) - 1)/r  
    return(n)  
  }  
}
```

## Prueba con R

Para garantizar el “éxito” en nuestro trabajo, agarraremos los siguientes valores para los parametros y verificaremos que el codigo y los calculos cumplen con lo que buscamos.

$$VA = 1000 \quad VF = 1100 \quad n = 2 \quad r = 0.05$$

Llamamos la funcion:

```
MiVF=interesSimple(va = 1000, r = 0.05, n = 2)
MiVF
```

```
[1] 1100
```

```
MiVA=interesSimple(VF = 1100, r = 0.05, n = 2)
MiVA
```

```
[1] 1000
```

```
MiR=interesSimple(va = 1000, VF = 1100, n = 2)
MiR
```

```
[1] 0.05
```

```
MiN=interesSimple(va = 1000, VF = 1100, r = 0.05)
MiN
```

```
[1] 2
```

Efectivamente cumple con lo que necesitabamos

## Python

Podríamos intentar hacer el código de otra manera, pero la mejor opción será traducir del lenguaje “R” al lenguaje de “python”.

```
def interesSimple(va = None, r = None, n = None, VF = None):  
  
    if VF is None:  
        VF = va*(1 + r*n)  
        return(VF)  
  
    elif va is None:  
        va = VF/(1 + r*n)  
        return(va)  
  
    elif r is None:  
        r = ((VF/va) - 1)/n  
        return(r)  
  
    elif n is None:  
        n = ((VF/va) - 1)/r  
        return(n)
```

## Prueba con Python

```
MiVF = interesSimple(va=1000, r=0.05, n=2)
print(MiVF)
```

1100.0

```
MiVA = interesSimple(VF=1100, r=0.05, n=2)
print(MiVA)
```

999.9999999999999

```
MiR = interesSimple(VF=1100, va=1000, n=2)
print(MiR)
```

0.0500000000000000044

```
MiN = interesSimple(VF=1100, va=1000, r=0.05)
print(MiN)
```

2.00000000000000018