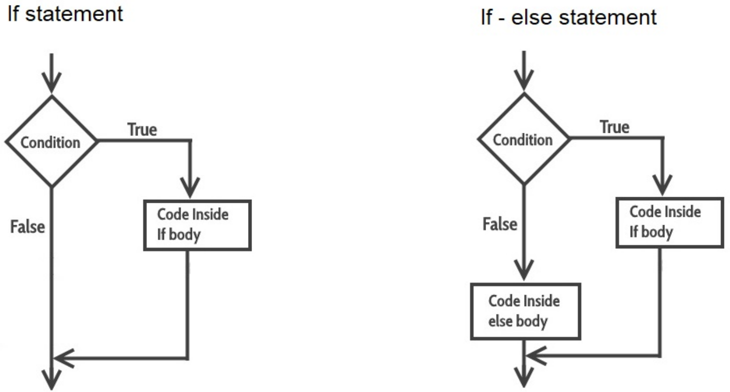
Basic R - Control flow,loops and writing function

Control statements in R

- Allow us to control the flow of our programming and cause different things to happen depending on the values of tests.
- Tests result in a logical, "TRUE", or "FALSE".
- The main control statements are "if", "else" and "ifelse".



Control statements in R

Control statements in R

List of checks in R (which result in TRUE or FALSE)

Check	meaning
==	equal to
!=	not equal to
<	less than
<=	less than or equal to
>	greater than
>=	greater than or equal to

Control statements "If"

Let try the following codes

```
x<-5
if (x < 10){
  print("less than 10")
}
```

```
## [1] "less than 10"
```

```
x<-5
if (x == 10){
  print("equal to 10")
}
```

```
if (sum(is.na(Nile))==0){
  print("There isn't any NA value in data Nile")
}
```

```
## [1] "There isn't any NA value in data Nile"
```

Control statements “If”

"President" dataset provides quarterly approval rating for the President of the United States from the first quarter of 1945 to the last quarter of 1974.

- Is there any **NA** value in the dataset?
- If yes, replace all **NA** value by 50.
- Calculate the mean of quarterly rating for the President.

Control statements "If"

"President" dataset provides quarterly approval rating for the President of the United States from the first quarter of 1945 to the last quarter of 1974.

- Is there any **NA** value in the dataset?
- If yes, replace all **NA** value by 50.
- Calculate the mean of quarterly rating for the President.

```
sum(is.na(presidents))
```

```
## [1] 6
```

```
if(sum(is.na(presidents))>0){
  ind<-is.na(presidents)
  presidents[ind]<-50
}
mean(presidents)
```

Control statements “If” and “else”

```
if (sum(is.na(presidents))==0){
  print("There isn't any NA value in data presidents")
} else {
  print("There are NA values in data presidents")
}
```

```
## [1] "There isn't any NA value in data presidents"
```

Example 1 Write a R code, result in "Yes" or "No", to check if there is at least a value in AirPassengers less than 125

Control statements “If” and “else”

```
if (sum(is.na(presidents))==0){
  print("There isn't any NA value in data presidents")
} else {
  print("There are NA values in data presidents")
}
```

```
## [1] "There isn't any NA value in data presidents"
```

Example 1 Write a R code, result in "Yes" or "No", to check if there is at least a value in AirPassengers less than 125

```
if (sum(AirPassengers<125)>0){
  print("Yes")
} else {
  print("No")
}
```

```
## [1] "Yes"
```

Control statements "Ifelse"

Similar to "If" statement in excel.

```
ifelse (sum(AirPassengers<125)>0,"Yes","No")
```

```
## [1] "Yes"
```

Example 2 Vector **x** consists of values 0 and 1. Create vector **y** such that *x[i] = 0* then *y[i] = "Tail"* and *x[i] = 1* then \textit{y[i] = "Head".

Control statements "Ifelse"

Similar to "If" statement in excel.

```
ifelse (sum(AirPassengers<125)>0,"Yes","No")
```

```
## [1] "Yes"
```

Example 2 Vector **x** consists of values 0 and 1. Create vector **y** such that *x[i] = 0* then *y[i] = "Tail"* and *x[i] = 1* then \textit{y[i] = "Head".

```
x<-c(0,0,1,1,0,1,1)
y<-ifelse(x==0,"Tail","Head")
x
```

```
## [1] 0 0 1 1 0 1 1
```

```
y
```

```
## [1] "Tail" "Tail" "Head" "Head" "Tail" "Head" "Head"
```

Control statements: Compound test

"&" stands for "and" while "|" stands for "or" operations in R. Remember the following combination of "&" and "|" operations

Compound test	result
TRUE & TRUE	TRUE
TRUE & FALSE	FALSE
FALSE & FALSE	FALSE
TRUE | TRUE	TRUE
TRUE | FALSE	TRUE
FALSE | FALSE	FALSE

```
(1==1)&(2>3)
```

```
## [1] FALSE
```

```
(1==1)|(2>3)
```

```
## [1] TRUE
```

For Loops

When we want to iterate over elements of a vector, list, or data.frame. To print numbers from 1 to 8, we could use for loops as follows:

```
for (i in 1:8)
  {
  print(i)
  }
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
```

For Loops

Example 1: Using *for loops* to calculate the following sums

$$\sum_{k=1}^{1000} k = ?$$

$$\sum_{k=1}^{1000} \frac{1}{k^2} = ?$$

$$\sum_{k=1}^{1000} \frac{1}{k!} = ?$$

Example 2: Using *for loops* to verify if $10^{12} + 10^{6} + 101$ is a prime number ?

For Loops

The vector in *for loops* does not have to be sequential; it can be **any vector.**

```
myvector1<-c("pie","banana","apple","mango")
for (s in myvector1)
  {
  print(s)
}
```

```
## [1] "pie"
## [1] "banana"
## [1] "apple"
## [1] "mango"
```

While Loops

The code inside the braces runs repeatedly as long as the tested condition is true.

```
x<-1
while(x<=5){
  print(x)
  x<-x+1
}
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
```

Controlling Loops

A *next* statement is used inside a loop to skip the iterations and flow the control to next step.

```
myvector1<-c("pie","banana","apple","mango")
myvector2<-vector(mode="numeric",length(myvector1))
for (i in 1:length(myvector1))
  {
  if (i==2){next}
  myvector2[i]<-nchar(myvector1[i])
  }
myvector2
```

```
## [1] 3 0 5 5
```

Controlling Loops

A *break* statement is used inside a loop to stop the iterations and flow the control outside of the loop.

```
myvector1<-c("pie","banana","apple","mango")
myvector2<-vector(mode="numeric",length(myvector1))
for (i in 1:length(myvector1))
  {
  if (i==2){break}
  myvector2[i]<-nchar(myvector1[i])
  }
myvector2
```

```
## [1] 3 0 0 0
```

Controlling Loops

A number is prime number or not.

```
n<-25
bl<-TRUE
for (i in 2:n){
  if(i^2>n){
    break
  }
  if(n %% i ==0){
    bl<-FALSE
  }
}
bl
```

```
## [1] FALSE
```

Writing functions

If we have to run the same code repeatedly, it is a good idea to turn it into a function.

```
h<-function(x)#h is a function of variable x, or parameter x
  {
  h<-x^0.5
  }
print(h(5))
```

```
## [1] 2.236068
```

```
g<-function(x,y,z)#g is a function of variables x,y and z
  {g<-x*y-z}
print(g(3,4,5))
```

```
## [1] 7
```

Writing function

Write the following functions of n, calculate values of these function at $n = 5, 10, 15$ and 20

$$
\begin{aligned}
f_1(n) &= \sum_{k=1}^{n} k \\
f_2(n) &= \sum_{k=1}^{n} \frac{1}{k^2} \\
f_3(n) &= \sum_{k=1}^{n} \frac{1}{k!}
\end{aligned}
$$

Writing function

```
f1<-function(n){ # khai bao ham so
  s0<-0 # Code giong nhu tinh tong 1:n
  for (i in 1:n){
    s0<-s0+i
  }
  f1<-s0 # luon luon ket thuc bang cach gan f1
}
print(f1(5))
```

```
## [1] 15
```

```
print(f1(10))
```

```
## [1] 55
```

```
print(f1(15))
```

```
## [1] 120
```

Writing function

Function f_2

```
f2<-function(n){ # khai bao ham so
  s0<-0 # Code giong nhu tinh tong 1:n
  for (i in 1:n){
    s0<-s0+1/i^2
  }
  f2<-s0 # luon luon ket thuc bang cach gan f1
}
print(f2(5))
```

```
## [1] 1.463611
```

```
print(f2(10))
```

```
## [1] 1.549768
```

```
print(f2(15))
```

Writing function - multivariable

Writing a function of n and k

$$1^k + 2^k + 3^k + \cdots + n^k$$

```
f<-function(n,k){ # khai bao ham so
  s0<-0 # Code giong nhu tinh tong 1:n
  for (i in 1:n){
    s0<-s0+i^k
  }
  f<-s0 # luon luon ket thuc bang cach gan f
}
print(f(10,1.5))
```

```
## [1] 142.6723
```

Writing functions

A function and arrays

```
h<-function(x){h<-x^2}
x<-c(1,2,3,4)
print(h(x))
```

```
## [1]  1  4  9 16
```

```
g<-function(x,y,z){g<-x*y-z}
x<-c(1,2,3)
y<-c(4,5,6)
z<-c(7,8,9)
print(g(x,y,z))
```

```
## [1] -3  2  9
```

Writing functions

Example 1: Let $f(n) = \sum_{k=1}^{n} k^{-2}$. Prove that $\lim_{n\to\infty} f(n) = \frac{\pi^2}{6}$ by calculating $f(5)$,$f(10)$, $f(15)$,$\cdots$, $f(100)$.

Writing functions

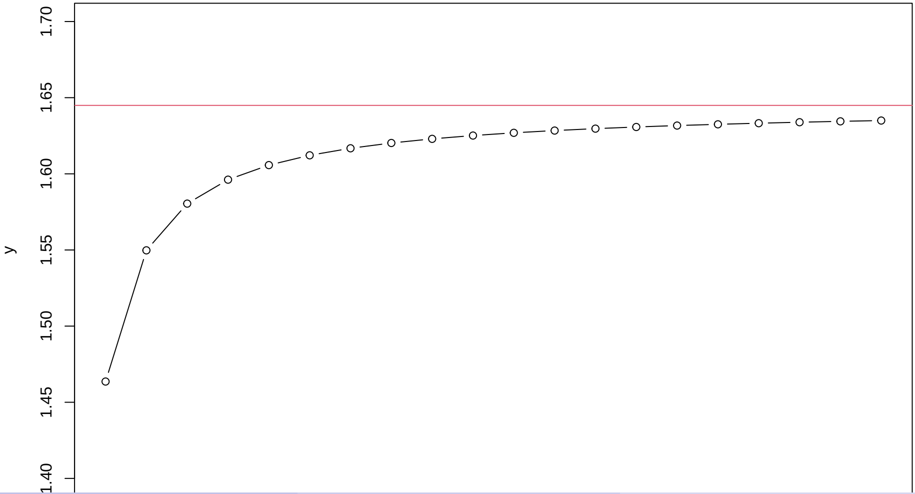
<u>Example 1:</u> Let $f(n) = \sum_{k=1}^{n} k^{-2}$. Prove that $\lim_{n\to\infty} f(n) = \frac{\pi^2}{6}$ by calculating $f(5)$,$f(10)$, $f(15)$,$\cdots$, $f(100)$.

```
f<-function(n){
  tg<-0
  for (k in 1:n){tg<-tg+k^(-2)}
  f<-tg
}
x<-5*(1:20)
y<-vector(mode="numeric",length(x))
for (i in 1:length(x)){y[i]=f(x[i])}
print(y[15:20])
```

```
## [1] 1.631689 1.632512 1.633238 1.633884 1.634463 1.634984
```

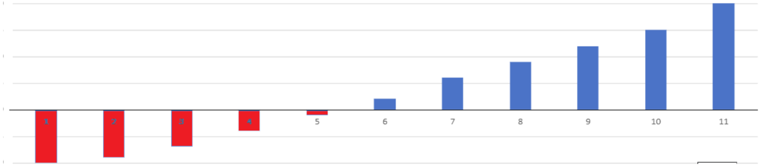
Writing functions

```
plot(x,y,ylim = c(1.4,1.7),type="b")
abline(h=pi^2/6,col=2)
```



Writing function

Write a function f to calculate the present value of a cash flow with 2 parameters: The first parameter, CF, is the vector consisting of annual cash flows; and the second parameter i is the annual effective rate.



$$PV = \frac{CF_1}{(1+i)} + \frac{CF_2}{(1+i)^2} + \cdots + \frac{CF_n}{(1+i)^n}$$

Writing function

With i $= 10\%$, calculate the PV of the following cash flow

```
##       [,1]  [,2]  [,3] [,4] [,5] [,6] [,7] [,8]
## CF -10000 -5000 -1000 2000 5000 8000 9000 4000
```

Writing function

With i = 10%, calculate the *PV* of the following cash flow

```
##       [,1]  [,2]  [,3] [,4] [,5] [,6] [,7] [,8]
## CF -10000 -5000 -1000 2000 5000 8000 9000 4000
```

```
PV<-function(CF,i){#bat dau tu nam 1
  n<-length(CF)
  s0<-0
  for (k in 1:n){
    s0<-s0+CF[k]/(1+i)^(k)
  }
  PV<-s0
}
x<-c(-10000,-5000,-1000,2000,5000,8000,9000,4000)
i<-0.1
print(PV(x,i))
```

```
## [1] 1496.426
```

Writing function

In practice, cash flows are often started at year 0. Write a function of 3 parameters CF, i and p calculating the present value of cash flow CF with annual interest rate i and $p = 0$ corresponds to cash flows started at year 0

Writing function

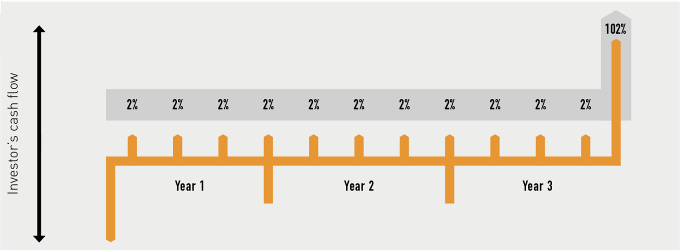
In practice, cash flows are often started at year 0. Write a function of 3 parameters *CF*, *i* and *p* calculating the present value of cash flow *CF* with annual interest rate *i* and $p = 0$ corresponds to cash flows started at year 0

```
PV<-function(CF,i,p){#bat dau tu nam 1
  n<-length(CF)
  s0<-0
  for (k in 1:n){
    s0<-s0+CF[k]/(1+i)^(k)
  }
  PV<-ifelse(p==1,s0,s0*(1+i))
}
```

Writing function

Write a function to calculate price of a bond with following parameters:

- F: face value (or par value).
- c: coupon.
- T: Maturity.
- k: number of coupons per year (each coupon payment is $c/k \times F$).
- i: nomial annual interest rate (one period interest rate is i/k).



Writing function

```r
PVBond<-function(F,c,T,k,i){
  n<-k*T
  CF<-rep(c/k*F,n)+c(rep(0,(n-1)),F)
  PVBond<-PV(CF,i/k,1)
}
F<-10^8
c<-8/100
T<-3
k<-4
i<-6/100
print(PVBond(F,c,T,k,i))
```

```
## [1] 105453753
```

Writing functions

Example 2: Let $f(n) = \sum_{k=0}^{n} \frac{1}{k!}$. Prove that $\lim_{n\to\infty} f(n) = e$ by calculating $f(1)$,$f(2)$, $f(3)$,$\cdots$, $f(10)$.

Writing functions

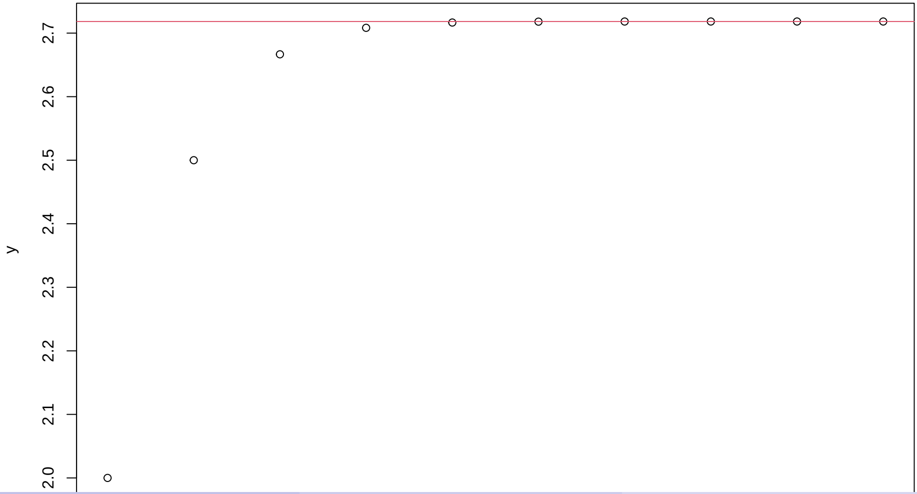
Example 2: Let $f(n)=\sum_{k=0}^{n}\frac{1}{k!}$. Prove that $\lim_{n\to\infty} f(n)=e$ by calculating $f(1)$,$f(2)$, $f(3)$,$\cdots$, $f(10)$.

```
f<-function(n){
  tg<-0
  for (k in 0:n){tg<-tg+1/factorial(k)}
  f<-tg
}
x<-(1:10)
y<-vector(mode="numeric",length(x))
for (i in 1:length(x)){y[i]=f(x[i])}
print(y[5:10])
```

```
## [1] 2.716667 2.718056 2.718254 2.718279 2.718282 2.718282
```

Writing functions

```
plot(x,y)
abline(h=exp(1),col=2)
```



Writing functions

Example 3: Write a function to solve the equation $a_1x^2 + a_2x + a_3 = 0$.

Writing functions

Example 3: Write a function to solve the equation $a_1x^2 + a_2x + a_3 = 0$.

```
f<-function(a1,a2,a3){ D<-a2^2-4*a1*a3
    if (D < 0){tg<-"No solution"}
    if (D == 0){tg<- -a2/(2*a1)}
    if (D > 0){tg<-c((-D^0.5-a2)/(2*a1),(D^0.5-a2)/(2*a1))}
    f<-tg}
print(f(1,1,1))#x^2+x+1 = 0
```

```
## [1] "No solution"
```

```
print(f(1,-2,1))#x^2-2x+1 = 0
```

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

```
print(f(1,-4,3))#x^2-4x+3 = 0
```

```
## [1] 1 3
```

Writing functions

Exercises 1: Write the following fomulars as function of i and n:

Note that v is discount factor: $v = \dfrac{1}{1+i}$.

$$\textit{Annuity Immediate :} \quad a_{\overline{n}|i} = \sum_{t=1}^{n} v^t$$

$$\textit{Annuity Due :} \quad \ddot{a}_{\overline{n}|i} = \sum_{t=0}^{n-1} v^t$$

$$\textit{Annuity Continuous :} \quad \bar{a}_{\overline{n}|i} = \int_{0}^{n} v^t dt$$

Writing functions

Exercises 1: (continue)

$$\textit{Increasing Annuity Immediate}: \quad (Ia)_{\overline{n}|i} = \sum_{t=1}^{n} t \times v^t$$

$$\textit{Increasing Annuity Due}: \quad (I\ddot{a})_{\overline{n}|i} = \sum_{t=0}^{n-1} (t+1) \times v^t$$

$$\textit{Decreasing Annuity Immediate}: \quad (Da)_{\overline{n}|i} = \sum_{t=1}^{n} (n-t+1) \times v^t$$

$$\textit{Decreasing Annuity Due}: \quad (D\ddot{a})_{\overline{n}|i} = \sum_{t=0}^{n-1} (n-t) \times v^t$$

Writing functions

- Exercise 2: Write a function that returns TRUE when a number if a prime and returns FALSE otherwise.

- Exercise 3: Write a function to approximate $\int_a^b g(x)\, dx$ where $g(x)$ is a continuous function. Using the following approximation

$$\int_a^b g(x)\, dx = \frac{b-a}{n} \sum_{i=1}^{n} g\left(a + i \times \frac{b-a}{n}\right)$$

- Exercise 4: Write a function to approximate $\int_a^b \int_c^d g(x, y)\, dx\ dy$ where $g(x, y)$ is a continuous function.