Lesson-1—16.10.2023.r

bramu

2023-10-17

```
2 + 2
```

```
## [1] 4
```

```
sqrt(25)
```

```
## [1] 5
```

```
exp(0)
```

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

```
log(1)
```

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

```
x <- 15
x
```

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

```
rm(x)

x <- c(1, 2, 3)
x
```

```
## [1] 1 2 3
```

```
is.vector(x)
```

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

```
ls()
```

```
## [1] "x"
```

```
rm(list = ls())

a <- 1:10
a
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

```
b <- seq(from = 1, to = 10, by = 1)

set.seed(1234)
rnorm(n = 10, mean = 0, sd = 1)
```

```
##  [1] -1.2070657  0.2774292  1.0844412 -2.3456977  0.4291247  0.5060559
##  [7] -0.5747400 -0.5466319 -0.5644520 -0.8900378
```

```r
length(seq(from = 1, to = 10, by = 0.1))
```

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

```r
min(rnorm(n = 100, mean = 0, sd = 10))
```

```
## [1] -21.8004
```

```r
max(rnorm(n = 100, mean = 0, sd = 10))
```

```
## [1] 30.43766
```

```r
5^2
```

```
## [1] 25
```

```r
100 >= 100
```

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

```r
99 > 100
```

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

```r
2 + 2 == 4
```

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

```r
99 != 100
```

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

```r
!(1==1)
```

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

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

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

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

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

```r
a <- c(1, 2)
b <- c(1, 2, 3, 4, 5, 6)
a + b
```

```
## [1] 2 4 4 6 6 8
```

```r
a <- c(1, 2)
b <- c(1, 2, 3, 4, 5)
a + b
```

```
## Warning in a + b: Länge des längeren Objektes
##       ist kein Vielfaches der Länge des kürzeren Objektes
```

```
## [1] 2 4 4 6 6
```

```r
x <- vector(mode = "numeric", length = 100)
is.vector(x)
```

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

```r
y <- numeric(100)
is.vector(y)
```

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

```
y[1] <- 50

y[1]
```

```
## [1] 50
```

```
x <- 10
y <- 2
if (x <= y) {

  print("x is smaller or equal y")

} else {

  print("x is larger than y")

}
```

```
## [1] "x is larger than y"
```

```
x <- rnorm(10, 0, 1)
ifelse(x <= 0, "x is smaller or equal zero", "x is larger than zero")
```

```
##  [1] "x is larger than zero"      "x is smaller or equal zero"
##  [3] "x is larger than zero"      "x is smaller or equal zero"
##  [5] "x is smaller or equal zero" "x is larger than zero"
##  [7] "x is smaller or equal zero" "x is smaller or equal zero"
##  [9] "x is smaller or equal zero" "x is larger than zero"
```

```
x <- 1:5
for (i in x) {

  print("Hello")

}
```

```
## [1] "Hello"
## [1] "Hello"
## [1] "Hello"
## [1] "Hello"
## [1] "Hello"
```

```
x <- c("Monday", "Tueasday", "Wednesday", "Thursday", "Friday")

for (i in x) {

  print("Hello")

}
```

```
## [1] "Hello"
## [1] "Hello"
## [1] "Hello"
## [1] "Hello"
## [1] "Hello"
```

```r
x <- 1:5

for (i in x) {

  print(i^2)

}
```

```
## [1] 1
## [1] 4
## [1] 9
## [1] 16
## [1] 25
```

```r
x <- c("Monday", "Tueasday", "Wednesday", "Thursday", "Friday")

for (i in x) {

  print(paste("Today is", i))

}
```

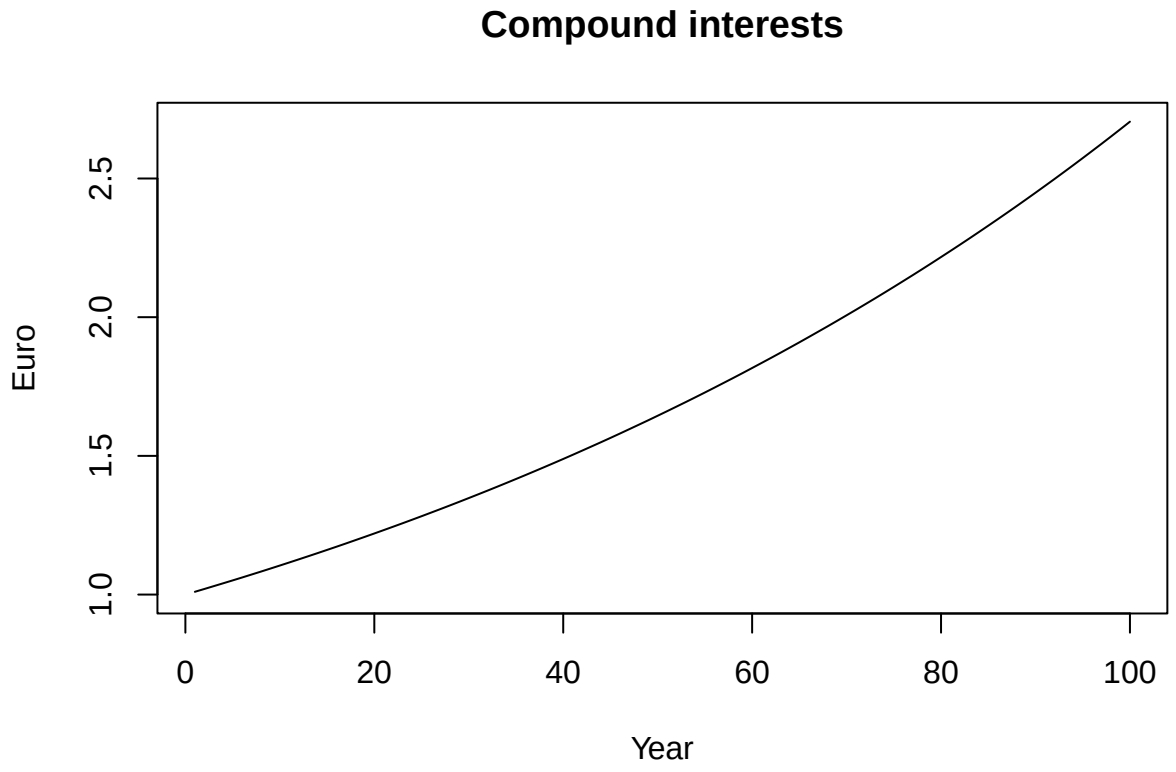
```
## [1] "Today is Monday"
## [1] "Today is Tueasday"
## [1] "Today is Wednesday"
## [1] "Today is Thursday"
## [1] "Today is Friday"
```

```r
# Formula
# A = P (1 + r)^t

years <- 100
A <- numeric(length = years)
r <- 0.01
P <- 1

for (t in 1:years) {

  A[t] <- P * (1 + r)^t

}

P * (1 + r)^(1:years)
```

```
##   [1] 1.010000 1.020100 1.030301 1.040604 1.051010 1.061520 1.072135 1.082857
##   [9] 1.093685 1.104622 1.115668 1.126825 1.138093 1.149474 1.160969 1.172579
##  [17] 1.184304 1.196147 1.208109 1.220190 1.232392 1.244716 1.257163 1.269735
##  [25] 1.282432 1.295256 1.308209 1.321291 1.334504 1.347849 1.361327 1.374941
##  [33] 1.388690 1.402577 1.416603 1.430769 1.445076 1.459527 1.474123 1.488864
##  [41] 1.503752 1.518790 1.533978 1.549318 1.564811 1.580459 1.596263 1.612226
##  [49] 1.628348 1.644632 1.661078 1.677689 1.694466 1.711410 1.728525 1.745810
##  [57] 1.763268 1.780901 1.798710 1.816697 1.834864 1.853212 1.871744 1.890462
##  [65] 1.909366 1.928460 1.947745 1.967222 1.986894 2.006763 2.026831 2.047099
##  [73] 2.067570 2.088246 2.109128 2.130220 2.151522 2.173037 2.194768 2.216715
##  [81] 2.238882 2.261271 2.283884 2.306723 2.329790 2.353088 2.376619 2.400385
```

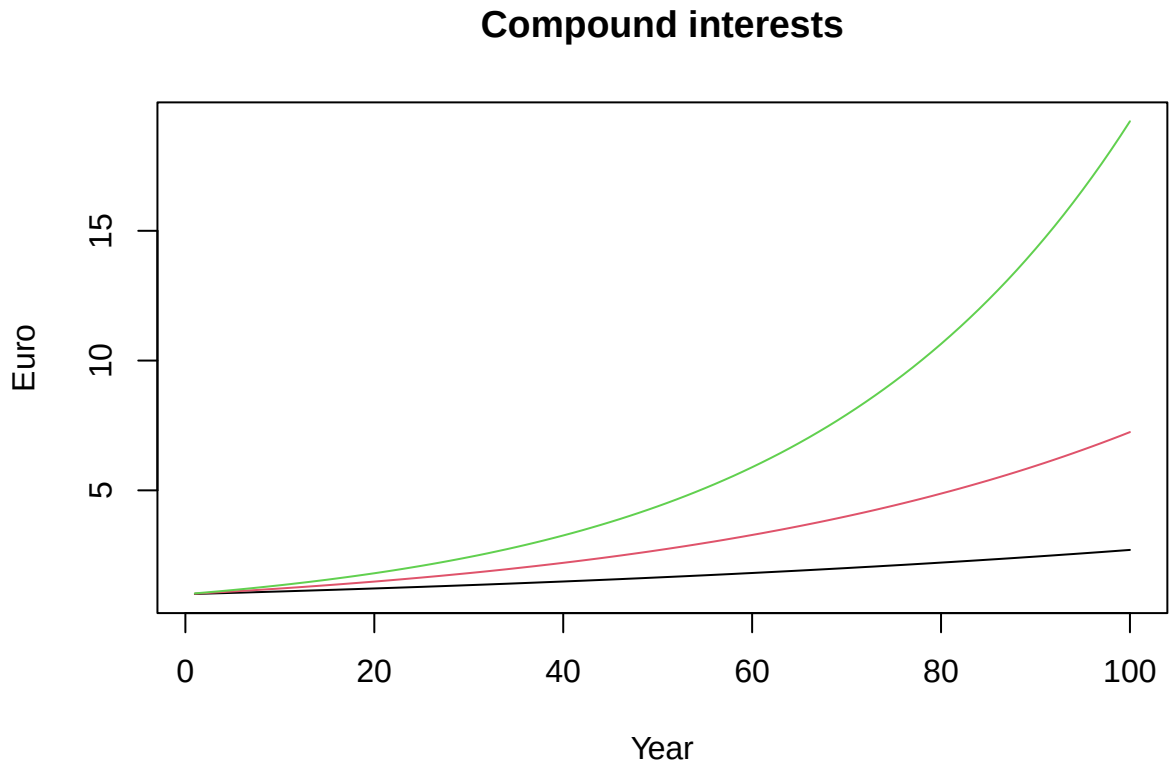
```
##  [89] 2.424389 2.448633 2.473119 2.497850 2.522829 2.548057 2.573538 2.599273
##  [97] 2.625266 2.651518 2.678033 2.704814
```

```r
plot(x = 1:years,
     y = A,
     xlim = c(1, years),
     ylim = c(1, max(A)),
     type = "l",
     col = "black",
     main = "Compound interests",
     xlab = "Year",
     ylab = "Euro")
```




```r
years <- 100
r <- c(0.01, 0.02, 0.03)
A <- matrix(nrow = years, ncol = length(r), byrow = TRUE)
P <- 1

for (t in 1:years) {

  for (s in 1:length(r)) {

    A[t,s] <- P * (1 + r[s])^t

  }

}
```

```r
plot(NULL,
     xlim = c(1, nrow(A)),
     ylim = c(1, max(A)),
     main = "Compound interests",
     xlab = "Year",
     ylab = "Euro")

for (l in 1:ncol(A)) {

  lines(A[,l], type = "l", col = l)

}
```




```r
my_fun <- function(x) {

  square <- x^2

  return(square)

}

my_fun(x = 5)
```

```
## [1] 25
```

```r
my_fun_alternative <- function(x) x^2

my_fun_alternative(x = 5)
```

```
## [1] 25
```

```r
my_fun2 <- function(x, y) {

  to_power <- x^y

  return(to_power)

}

my_fun2(x = 5, y = 2)
```

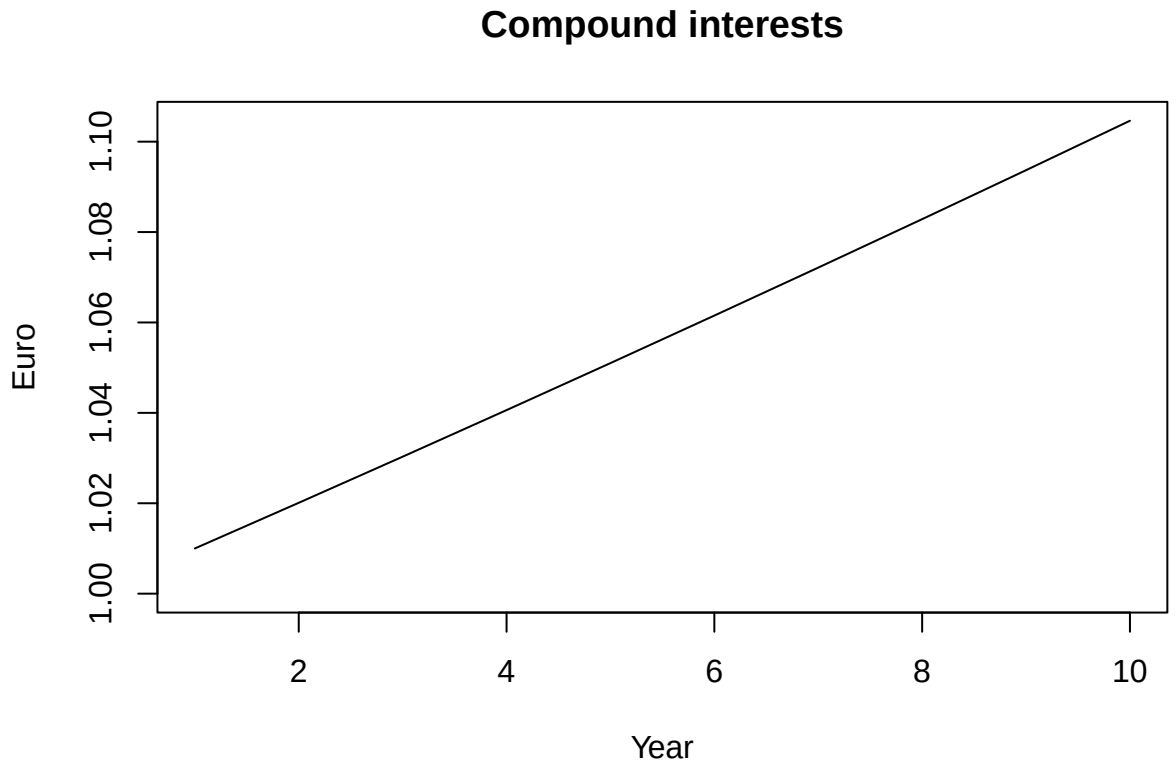
```
## [1] 25
```

```r
# With default value for y
my_fun2_alternative <- function(x, y = 2) {

  to_power <- x^y

  return(to_power)

}

my_fun2_alternative(x = 5)
```

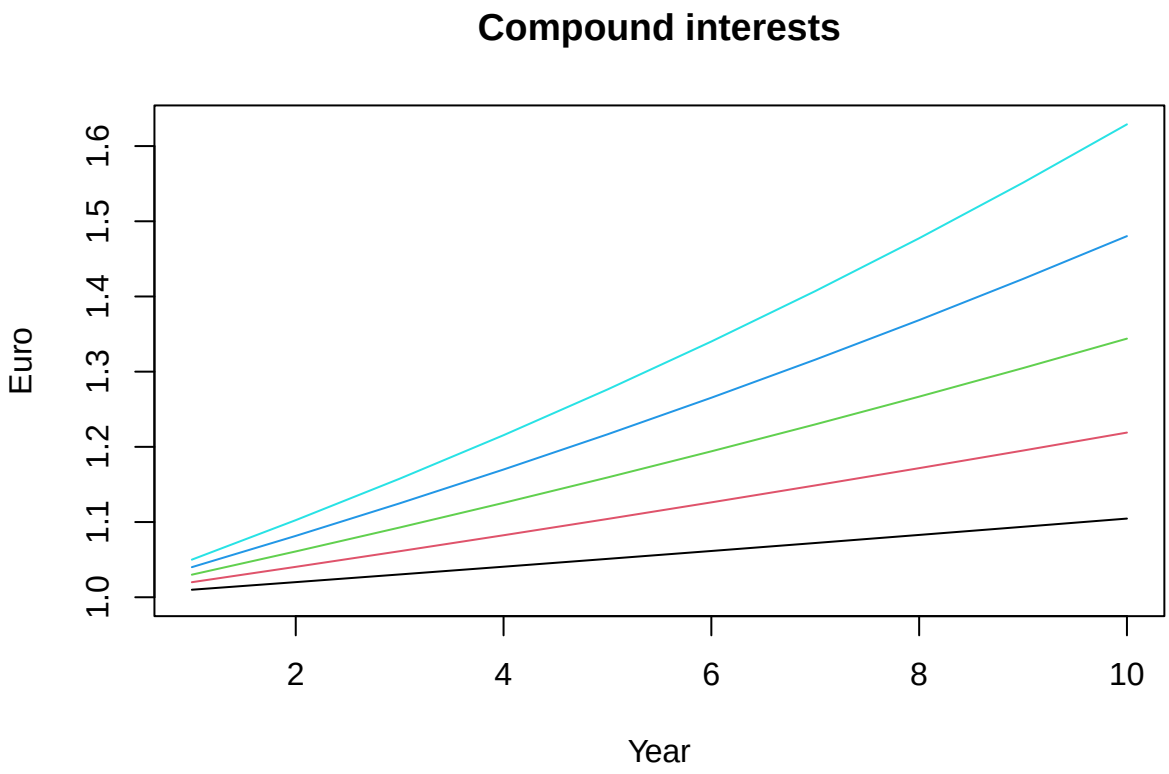
```
## [1] 25
```

```r
my_fun <- function(time, scenarios) {

r <- scenarios

A <- matrix(nrow = time, ncol = length(r), byrow = TRUE)

P <- 1

for (t in 1:time) {

  for (s in 1:length(r)) {

    A[t,s] <- P * (1 + r[s])^t

  }

}

plot(NULL,
     xlim = c(1, nrow(A)),
     ylim = c(1, max(A)),
     main = "Compound interests",
     xlab = "Year",
     ylab = "Euro")
```

```
for (l in 1:ncol(A)) {

  lines(A[,l], type = "l", col = l)

  }

}

my_fun(time = 10, scenarios = 0.01)
```




```
my_fun(time = 10, scenarios = seq(0.01, 0.05, 0.01))
```


Compound interests
Euro
1.0
1.1
1.2
1.3
1.4
1.5
1.6
2
4
6
8
10
Year