Uitleg R scripts

**Uitleg van de volgende segmenten:**

[Problem 1 2](#_Toc61703588)

[Dataframe 2](#_Toc61703589)

[Objecten 2](#_Toc61703590)

[For loop om de overwinningen van AOpen te tellen 3](#_Toc61703591)

[Loop om de wins van Novak Djokovic te vergelijken met Roger Federer 4](#_Toc61703592)

[Volledige code 5](#_Toc61703593)

[Problem 2 6](#_Toc61703594)

[Functie toewijzen 6](#_Toc61703595)

[Waarde instellen van y 6](#_Toc61703596)

[Omzetten decimalen en maandelijkse waarde berekenen 6](#_Toc61703597)

[While loop configureren 6](#_Toc61703598)

[Return value 6](#_Toc61703599)

[Volledige code 7](#_Toc61703600)

[Problem 3 8](#_Toc61703601)

[Functie toewijzen 8](#_Toc61703602)

[If/else loop 8](#_Toc61703603)

[Volledige code 9](#_Toc61703604)

# Problem 1

## Dataframe

Allereerst heb je in problem1.txt het dataframe meegekregen (dubbelklik op het icoontje om te openen):

In het eerste commando wordt er aangegeven dat de eerste kolom optellend is van de values 2004 tot en met 2016. Dus er komt sws in het datafame een lijst te staan vanaf 2004 tot en met 2016.

Code is als volgt:

AOpen <- data.frame(2004:2016,c("TENNISERS")

colnames(AOpen) <- c("Year","Champion")

In het commando erna wordt “colnames” uitgevoerd, dit is een afkorting voor columnnames. Nu je dat weet kan je zien dat de eerste kolom “Year” gaat heten en de tweede kolom “Champion”.

Voor visualisatie komt het er zo uit te zien in een Excel bestand:

|  |  |
| --- | --- |
| Year | Champion |
| 2004 | Roger Federer |
| 2005 | Marat Safin |
| 2006 | Roger Federer |
| 2007 | Roger Federer |
| 2008 | Novak Djokovic |
| 2009 | Rafael Nadal |
| 2010 | Roger Federer |
| 2011 | Novak Djokovic |
| 2012 | Novak Djokovic |
| 2013 | Novak Djokovic |
| 2014 | Stan Wawrinka |
| 2015 | Novak Djokovic |
| 2016 | Novak Djokovic |

Zoals je kan zien zijn er meerder overwinningen voor Roger Federer & Novak Djokovic.

Roger heeft er 4 en Djokovic heeft er 6. We gaan dit tellen met behulp van een scriptje.

## Objecten

Om de overwinningen te tellen van deze 2 tennissers worden er 2 objecten aangemaakt. Voor Roger Federer en voor Novak Djokovic.

Dit doen we zo:

Wins\_ND <- 0

Wins\_RF <- 0

De benamingen spreken voor zich, we zetten beiden objecten op 0 omdat we door het bovenste dataframe heen willen zoeken naar voorkomingen van winsten.

## For loop om de overwinningen van AOpen te tellen

Om de hoeveelheids overwinningen van de AOpen te tellen gaan we een for loop gebruiken. En for loop kan je gebruiken om door elk item in een lijst/dataframe heen te gaan.

De code ziet er als volgt uit:

for(y in AOpen[,2]) {

  if(y == "Novak Djokovic") {

    Wins\_ND <- Wins\_ND + 1

  }

  if (y == "Roger Federer") {

    Wins\_RF <- Wins\_RF + 1

  }

}

In de eerste zin geven we aan dat we een FOR loop gebruiken. Y kan hier elke letter zijn. Het dient om dus aan te geven dat 1 object uit het dataframe voortaan y heet. Daarna geven we aan dat we willen werken met het DataFrame AOpen, daarna kan je de door de [,2] zien dat we uit de 2e kolom gaan werken. De dubbele == staat voor **IS GELIJK AAN.** We gaan de for loop doorlopen en beginnen met de eerste waarde van column 2 (zie tabel vorige pagina):

1. Hier heeft y als eerst de waarde gekregen “Roger Federer”
2. Dan gaat hij door de loop, if (als) y hier gelijk aan “Novak Djokovic” tel 1 bij Wins\_ND op, onze waarde is niet gelijk aan “Novak Djokovic” dus gaat het door naar de volgende if statement.
3. Dan gaat het door naar de volgende if (als) y hier gelijk is aan “Roger Federer” tel dan 1 op bij Wins\_RF. Onze waarde van y is hier gelijk aan “Roger Federer”
4. En zo gaat het de hele lijst door van boven naar beneden.

Als dit voltooid is zullen Wins\_ND en Wins\_RF corresponderen met de data in de tabel die ik gemaakt heb. Dat betekent dat Wins\_ND nu een waarde heeft van 6 en Wins\_RF een waarde van 4.

Deze waardes zijn nu bijgewerkt en kunnen we verder mee gaan werken.

## Loop om de wins van Novak Djokovic te vergelijken met Roger Federer

In het laatste code segment gaan we de 2 waardes met elkaar vergelijken door middel van een if (als) loop. De code van dit segment ziet er zo uit:

if(Wins\_ND > Wins\_RF) {

  print("Djokovic has more wins")

} else{

  if (Wins\_ND == Wins\_RF) {

    print("Both have the same number of wins")

  } else{

    print("Federer has more wins")

  }

}

De code werkt als volgt:

1. Eerst gaan we kijken of Wins\_ND > (groter is dan) Wins\_RF als dit het geval is printen we “Djokovic has more wins”. Print betekent output deze tekst (ook string genoemd) in dit geval.
2. Als dit niet het geval is gaan we door naar de volgende if, dit wordt aangegeven door else. Als dit niet klopt doe dan het volgende
3. Bij het 2e segment gaan we kijken of de Wins\_ND == (gelijk is aan) Wins\_RF als dit het geval is printen we “Both have the same number of wins”
4. Als dit niet het geval is gaan we door naar de volgende opdracht
5. Bij het 3e segment is er geen vergelijking meer nodig dus printen we in het geval dat allebei de bovenstaande condities niet zijn voldaan. Hier printen we “Federer has more wins”.

## Volledige code

# Hier wordt het dataframe gedefineerd

AOpen <- data.frame(2004:2016,c("TENNISERS"))

colnames(AOpen) <- c("Year","Champion")

#Stel de objecten wins van Federer & Djokovic in

Wins\_ND <- 0

Wins\_RF <- 0

#Loop en tel de wins in column 2 (Column 2 kan je zien heet champion)

for(y in AOpen[,2]) {

  if(y == "Novak Djokovic") {

    Wins\_ND <- Wins\_ND + 1

  }

  if(y == "Roger Federer") {

    Wins\_RF <- Wins\_RF + 1

  }

}

#Ga de wins van de 2 objecten met elkaar vergelijken door een if loop

if(Wins\_ND > Wins\_RF) {

  print("Djokovic has more wins")

} else{

  if (Wins\_ND == Wins\_RF) {

    print("Both have the same number of wins")

    } else{

    print("Federer has more wins")

  }

}

# Problem 2

## Functie toewijzen

Het eerste wat we gaan doen is Bob een functie maken en daar de variabelen aan koppelen. Dit doen we door de volgende code:

Bob <- function(x, m, r) {

Hier geven we aan dat Bob de variabelen x, m & r meekrijgt. Deze variabelen kunnen we zelf een waarde geven in de R console.

## Waarde instellen van y

Daarna zetten we de waarde van y naar 0 (maanden) met behulp van de volgende code:

    y <- 0

## Omzetten decimalen en maandelijkse waarde berekenen

R moeten we nog omzetten naar decimalen en we willen de maandelijkse waarde bereken. Dit doen we met de volgende code:

    monthly\_rate\_interest <- r / (12 \* 100)

Hier geven we monthly\_rate\_interest de waarde van r gedeeld door (12 \* 100). Hierdoor krijgen we maandelijks de decimale waarde.

## While loop configureren

Eerst controleren we of de waarde van x > (groter is dan) 0, als deze statement klopt gaan we door met de code.

   while (x > 0) {

        y <- y + 1

        x <- x - m

        x <- x \* (1 + monthly\_rate\_interest)

    }

De rest van de loop werkt als volgt:

1. We Passen de waarde van y aan naar + 1 (maand), hierdoor krijgt y dus een waarde van 1.
2. Hierna halen we de maandelijkse betaling van de waarde van x af.
3. Met de laatste opdracht berekenen we het totaal met de monthly\_rare\_interest erbij

## Return value

Met return(y) krijgen we de het aantal maanden terug dat er nodig is om de lening terug te betalen.

    return(y)

## Volledige code

Bob <- function(x, m, r) {

    y <- 0

    monthly\_rate\_interest <- r / (12 \* 100)

    while (x > 0) {

        y <- y + 1

        x <- x - m

        x <- x \* (1 + monthly\_rate\_interest)

    }

    return(y)

}

# Problem 3

## Functie toewijzen

Het eerste wat we gaan doen is Bob een functie maken en daar de variabelen aan koppelen. Dit doen we door de volgende code:

Bob <- function(x, m, r) {

Hier geven we aan dat Bob de variabelen x, m & r meekrijgt. Deze variabelen kunnen we zelf een waarde geven in de R console.

## If/else loop

   if (x <- m) {

        return(1)

    }

    else {

        monthly\_rate\_interest <- r / (12 \* 100)

        return(1 + Bob((x - m) \* (1 + monthly\_rate\_interest), m, r))

    }

}

De code werkt als volgt:

1. Eerst gaan we kijken of het openstaande bedrag =< (hetzelfde of minder dan) M is

Als dit het geval is dan heeft Bob 1 maand nodig dus returnen we 1

1. Als dit niet het geval is gaan we door met de code door middel van else, we willen de waarde van 1 (maand) plus het aantal maanden dat ie nog nodig heeft na de betaling van deze maand

## Volledige code

Bob <- function(x, m, r) {

    if (x <- m) {

        return(1)

    }

    else {

        monthly\_rate\_interest <- r / (12 \* 100)

        return(1 + Bob((x - m) \* (1 + monthly\_rate\_interest), m, r))

    }

}