Project Dynamica: oefenopgaven met R
Uitwerkingen

Opgave 1:

```
> x <- c(23, 0.149, -5.15, 36)
> x <- c(x, -28)
> y <- x[order(x)]
> 4 * x
> round(x, 1)
> x[x>0]
> x[3] <- 7
```

Opgave 2:

```
> x2 <- seq(2, 4, 0.1)
> m <- matrix(x2, nrow=7, byrow=FALSE)
> m[2,1]
> m[,3]
> mean(m)
> apply(m, 2, mean)
```

Opgave 3:

```
> drawballs <- function(n=100, mr=40, mw=60){
+ p <- c(mr/(mr+mw), mw/(mr+mw))
+ x <- c(0,1)
+ y <- sample(x, n, replace=TRUE, prob=p)
+ white <- sum(y)
+ red <- n - white
+ return(list(nRed=red, nWhite=white))
+ }
```

Opgave 4:

```
> round(pi * sqrt(2), d=5)
```

Opgave 5:

```
> # definieer de functie
> fx <- function(x){ x ^ 2 - 3 + 1 }
> # bepaal nu de wortel
> uniroot(fx, c(0, 1))
```

Opgave 6:

```
> # definieer de functie
> fx <- function(x){ 5*x - 1 - cos(x) }
```

```
> # bepaal de wortel
> root <- uniroot(fx, c(0,100))$root
> # definieer domein van de x-as
> x <- c(0:100)/50
> # plot functies en S bij het snijpunt
> plot(5*x - 1 ~ x, type="l")
> lines(cos(x) ~ x)
> text(root, 5*root-1+0.3, "S")
```

Opgave 7:

```
> # definieer de functie
> fx <- function(x){ x * atan(x) }
> # integreer de functie
> integrate(fx, 0, 100)
```

Opgave 8:

```
> # laad het package
> library(orthopolynom)
> # activeer de help functie
> ?hermite.h.polynomials
> # vraag 4-de graads polynoom op
> hermite.h.polynomials(4, normalized=FALSE)
```

Opgave 9:

```
> # definieer de matrix A
> A <- matrix(NA, 3, 4)
> A[1,] <- c(1, -1, 0, 2)
> A[2,] <- c(2, 3, 5, 4)
> A[3,] <- c(-1, 6, 5, -2)
> # definieer de vector b
> b <- matrix(NA, 3, 1)
> b[,1] <- c(1, 2, -1)
> # bepaal de lengte van b
> sqrt(t(b) %*% b)
> # laad benodigd package
> library(MASS)
> # basis voor de nulruimte van A
> Null(A)
> # bepaal x zdd Ax = b
> solve(A, b)
```

Opgave 10:

```
> eigen(C)
```

Opgave 11:

```
> P <- matrix(NA, 3, 3)
> P[1,] <- c(0.70, 0.10, 0.20)
> P[2,] <- c(0.50, 0.25, 0.25)
> P[3,] <- c(0.40, 0.30, 0.30)
> P[3,1]
```

```
> (P %*% P %*% P)[3,1])
> (P %*% P %*% P %*% P %*% P %*% P %*% P %*% P %*% P %*% P %*% P)[1,]
```

Opgave 12:

```
> library(gmp)
> teller <- 0
> for (i in 100:200){
+ if (isprime(i) == 2){ print(i) }
+ if (isprime(i) == 2){ teller <- teller + 1 }
+ }
> print(teller)
```

Opgave 13:

```
> succDiffCutoff <- 10 ^ (-9)
> succDiff <- 10 ^ 9
> fx <- function(x){ x ^ 2 - 2 }
> dfx <- function(x){ 2 * x }
> xold <- 1
> while (succDiff > succDiffCutoff){
+ xnew <- xold - fx(xold) / dfx(xold)
+ succDiff <- abs(xnew - xold)
+ xold <- xnew
+ }
> xnew - sqrt(2)
```