**PC1a: Intro R**

Name: Vu The Doan

Student nr: 12918687

Below, you can find exercises that need to be handed in at the end of this computer lab. Write your answers in the shaded rectangles.

**1.** Calculate the values of the next expressions (use the help function: help(tan) or ?tan):

a.

1. 6.59
2. 0.5
3. 0.7853982
4. 0.5

b.

c.

d.

2. Determine the value of the function for the *x*-values and

: 0.9316606

: 0.1417561

(hint: use abs to calculate ).

3. Complete the following table:

|  |  |  |  |
| --- | --- | --- | --- |
|  | R code | value |  |
|  | sin(pi/2) | 1 |  |
|  | cos(pi/3) | 1/2 |  |
|  | tan(pi/3) | 1.732 = sqrt(3) |  |
|  |  | Determine the function for two *x*-values | |
|  |  |  |  |
|  | log(x + sqrt(x^2+1)) | 0.4812118 | 0.8813736 |
|  |  |  |  |
|  | x/((x^2+1)\*sin(x)) | 0.6869657 | 0.4530184 |

4. Complete the table below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | round | ceil | floor | trunc |
|  | 0 | 1 | 0 | 0 |
|  | 0 | 1 | 0 | 0 |
|  | 0 | 1 | 0 | 0 |
|  | 0 | 1 | 0 | 0 |
|  | 2 | 2 | 1 | 1 |
|  | -1 | -1 | -2 | -1 |

What are the definitions of the functions round, ceil, floor and trunc?

round rounds the values in its first argument to the specified number of decimal places (default 0).

ceiling returns the smallest integers not less than the corresponding elements of x.

floor returns the largest integers not greater than the corresponding elements of x.

trunc  returns the integers formed by truncating the values in x toward 0.

trunc takes a single numeric argument x and returns a numeric vector containing the integers formed by truncating the values in x toward 0.

round rounds the values in its first argument to the specified number of decimal places (default 0). See ‘Details’ about “round to even” when rounding off a 5.

5. Install the library pracma using install.packages("pracma"). Fill in the next table:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *n* | 3 | 3 | 3 | 3 | 4 | 4 | 4 | 4 | 5 | 5 | 5 | 5 |
| *m* | 3 | 4 | –4 | 6 | 3 | 4 | –4 | 6 | 3 | 4 | –4 | 6 |
| rem(n,m) | 0 | 3 | 3 | 3 | 1 | 0 | 0 | 4 | 2 | 1 | 1 | 5 |
| mod(n,m) | 0 | 3 | -1 | 3 | 1 | 0 | 0 | 4 | 2 | 1 | -3 | 5 |

What are the definitions of the functions rem(n,m), mod(n,m)? When do they give different answers? Check rem(3,–4) and mod(3,–4) manually.

* Both rem and mod return n\mod\m.
* Outputs are the same if n and m has the same sign.

6. Evaluate the functions below for *x* from 1 to 2 in steps of 0.1. Hint: x<-seq(1,2,0.1):

|  |  |
| --- | --- |
|  | R code |
|  | y=x^3+3\*x^2+1; y |
|  | y <- sin(x^2); y |
|  | y <- sin(x)^2; y |
|  | y <- sin(2\*x) + x\*cos(4\*x); y |
|  | y <- x/(x^2+1); y |
|  | y <- cos(x)/(1+sin(x)); y |
|  | y <- 1/x + x^3/(x^4 + 5\*x\*sin(x)); y |

How could you generate the same vector *x* using the command seq(1,2,l=…)?

x <- seq(1, 2, l = 11)

7. Evaluate the function for to in steps of 0.01.

R code

x <- seq(3, 5, 0.01)

y <- x/(x + 1/(x^2)); y

8. Evaluate the function: for to in steps of 0.1.

R code

x <- seq(-2, -1, by = 0.1)

y <- 1/(x^3) + 1/(x^2) + 3/x; y

9. The following code is supposed to evaluate the function

for (in 200 points). Correct the code and check that

rm(list=ls())

x <-seq(0,1,l=200)

g <- x^3+1

h <- x+2

z <- x^2

y <- cos (pi\*x)

f <- (z\*y)/(g\*h)

f[200] == -1/6

x <-seq(0,1)

rm(list=ls())

g <- x^3+1

H <- x+2

z <- x.^2

y <- cos pix

f <- z\*y/g\*h

10. Debug the code which is supposed to plot the polynomial between and using 21 points in total.

library(Pracma)

x <- seq(-2,2,0.1)

coef <- c(1,0,0,-1)

y <- polyval(coef, x)

plot(y,x)

library("pracma")

x <- seq(-2,2,l=21)

coef <- c(1,0,0,-1)

y <- polyval(coef, x)

plot(y,x)

11. Debug the code which is supposed to set up the function on the grid to 3 in steps of 0.1 and show the value of the function at and .

x <- seq(0,3,l=30)

f <- x^3\*COS x+1

% value for x equals 2

f[21]

% value for x equals 3

Tail(f,2)

x <- seq(0,3,by=0.1)

f <- x^3\*cos(x+1)

f[21]

f[length(f)]