

## R-Assignment 1

### Math 362

The purpose of this assignment is to become accustomed to the R. Go through the following bullets:

- Learn how to access help. Enter  
> help.start()
- To assign a value to a variable, use the command = or <-  
To assign the value 3 to the variable x, enter  
> x=3  
> x  
To add 4 to the value of x, enter  
>x+4
- To set up a vector, named x, with the numbers 3 5 4 1, use the command,  
> x=c(3, 5, 4,1)  
and enter  
> x  
to view the vector x.
- The vector c(3, 4, 5, 6) is the same as the vector 3:6,  
> 3:6
- To set up a vector with numbers 3.0 3.5 4.0 4.5 5.0 5.5 6.0, enter  
> seq(3,6,0.5)
- To find the number of components of the vector x, enter  
> length(x)  
and enter  
> x[2]  
to select the second component.
- To find the sum of the components of the vector x, enter  
> sum(x)  
and enter  
> mean(x)  
to get the mean of x which is the same as  
> sum(x)/length(x)
- For the variance and standard deviation, enter  
> var(x)  
> sd(x)
- Enter  
> hist(x)  
to get the histogram

- The command  
`> sort(x)`  
gives the same vector as `x`, but sorted in increasing order.
- To repeat the vector `x` three times, i.e. 3541 3541 3541, enter  
`> rep(x, 3)`  
and to repeat each component three times, i.e. 333 555 444 111, enter  
`> rep(x, each=3)`
- Enter  
`> f= function(x) x^2+1`  
for the function `f` with input `x` and output `x^2+1`. Enter  
`> f(2)`  
to get the second component of `f`.
- The command  
`> objects()`  
display the names of the objects which are currently stored within R.  
To remove the objects `x` and `f`, enter  
`> rm(x,f)`
- Enter  
`> x= seq(-2,2, len=5)`  
for a vector `x` with 5 equally spaced values in the interval `[-2,2]`. Enter  
`> y= seq(-3,3, len=4)`  
for a vector `y` with 4 equally spaced values in the interval `[-3,3]`.  
Then enter  
`> f= outer(x,y, function(x,y) x^2+y)`  
where `f` is a 5 by 4 matrix with rows and columns indexed by `x` and `y` respectively, of values of the function `x^2+y`.
- Simulate 50 tosses of a fair coin. Enter  
`> x=rbinom(50, 1, 0.5)`  
and view the result  
`> x`  
Now add the total number of heads by typing  
`> t=cumsum(x)`  
Plot the result with steps (`type="s"`)  
`> plot(t, type="s")`  
Enter  
`> data.frame(x,t)`  
to display `x` and `t` in a table.
- Now use the sample command to simulate the number of heads in 50 coin tosses. Let `tail=0` and `head=1`. `T=TRUE`. Enter  
`> x=c(0,1)`  
`> data=sample(x,50,replace = T)`

To get the number of heads in 50 coin tosses, enter  
> sum(data)

- Plot a parametrized curve and start with the sequences x and y. Enter  
> theta=seq(-pi, pi, 0.01)  
> x=sin(5\*theta)  
> y= cos(7\*theta)  
> plot(x,y, type="l",xlab="x",ylab="y", main="Parametrized curve",col="green")  
The (type="l") is the letter l which connects the values in the sequence with lines.  
The (xlab="x") and (ylab="y") label the x-axis and y-axis.  
The (main="Parametrized curve") gives the title.

- We will now look at a surface plot. Define a function f that depends on x and y.  
> f=function(x,y) (sin(x)+sin(3\*y))/(1+x^2+y^2)  
where x and y take the values,  
> x=seq(-2,2,.1);  
> y=seq(-2,2,.1);

Enter

```
> z=outer(x,y,f)
```

where z is a square matrix with rows and columns indexed by x and y respectively, of values of the function f. Now draw a surface plot.

```
> persp(x=x,y=y,z=z, col="blue")
```

Change the viewing angle

```
> persp(x=x,y=y,z=z, theta=10,phi=40, col="blue")
```

Plot the density

```
> ncolors =30
```

```
> col= terrain.colors (ncolors)
```

```
> persp(x=x,y=y,z=z, theta=10,phi=40, col=col)
```

- We will now look at data frames and its indexing. To create a data frame, enter

```
> cost=c(50,30,24,32,44)
```

```
> amount=c(4,6,2,4,1)
```

```
> d=data.frame(cost,amount)
```

```
> d
```

To access the individual variable cost we use \$. To access the vector cost from the data frame, enter

```
> d$cost
```

To access the element in row 1 and column 2, enter

```
> d[1,2]
```

To access row 1 in the data frame, enter

```
> d[1,]
```

To access column 1, enter

```
> d[,1]
```

To access the first two row elements, enter

```
> d[1:2,]
```

To extract the data in the data frame for where the cost is greater than 30, enter

```
> d[d$cost>30,]
```

Entering,

```
> d$cost>30
```

 gives a logical vector with the value TRUE for where the cost is greater than 30.

- Here is an example of a for loop. We will calculate the product

$\prod_{k=1}^i (1 - k/9)$  for  $i=1,2,\dots,10$ .

Enter

```
> x=numeric(10)
```

```
> for (i in 1:10){
```

```
  x[i]=prod(1-(1:i)/9)
```

```
}
```

```
>data.frame(c(1:10),x)
```

Here numeric(10) creates a vector of zeros and is the initialization of the vector.

x[i] gives the values of the product for  $i=1,2,\dots,10$ .

- To create 10 normal random variables with mean 3 and standard deviation 1, enter  

```
>rnorm(10,3,1)
```