R Notebook

Textbook[http://www-bcf.usc.edu/~gareth/ISL/data.html]

Basic Commands

Defining Variables

```
x <- 5
x
```

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

```
y = 9
y
```

```
## [1] 9
```

Vector

Create a Vector of Numbers

```
x = c (1,3,2,5)
x
```

```
## [1] 1 3 2 5
```

Addition of two Vectors

```
x = c (1,3,2,5)
y = c (2,4,3,6)
x + y
```

```
## [1]  3  7  5 11
```

Using Functions

```
x = c(1,3,2,5)
length(x)
```

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

Using the Help for functions bei ?functionname

```
?length()
```

Sequence

```
x = seq(1,10)
x
```

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

```
x = 2:11
x
```

```
##  [1]  2  3  4  5  6  7  8  9 10 11
```

```
x = seq ( - pi , pi , length =10)
x
```

```
##  [1] -3.1415927 -2.4434610 -1.7453293 -1.0471976 -0.3490659  0.3490659
##  [7]  1.0471976  1.7453293  2.4434610  3.1415927
```

Matrix

Create a Matrix

```
x = matrix(data=c(1,2,3,4),nrow=2,ncol=2)
x
```

```
##      [,1] [,2]
## [1,]    1    3
## [2,]    2    4
```

Short version

```
x = matrix(c(1,2,3,4),2,2)
```

Change filling order to row fill

```
matrix(c(1,2,3,4),2,2,byrow=TRUE)
```

```
##      [,1] [,2]
## [1,]    1    2
## [2,]    3    4
```

Select element from matrix

```
x[2,2]
```

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

Subsection from matrix

```
x = matrix (1:16 ,4 ,4)
x
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    5    9   13
## [2,]    2    6   10   14
## [3,]    3    7   11   15
## [4,]    4    8   12   16
```

```
x [c(1,3),c(2,4)]
```

```
##      [,1] [,2]
## [1,]    5   13
## [2,]    7   15
```

```
x [1:3,2:4]
```

```
##      [,1] [,2] [,3]
## [1,]    5    9   13
## [2,]    6   10   14
## [3,]    7   11   15
```

```
x [1:2,]
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    5    9   13
## [2,]    2    6   10   14
```

```
x [-c(1,3),]
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    2    6   10   14
## [2,]    4    8   12   16
```

Dimensions

```
dim(x)
```

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

Objects

List all of the saved objects(data and functions)

```
ls()
```

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

Remove objects

```
rm(y)
ls()
```

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

Remove all objects

```
rm(list=ls())
ls()
```

```
## character(0)
```

Random Numbers

Generate random normal distributed variables with mean of 0 and a standard deviation of 1

```
x = rnorm(10)
x
```

```
##  [1] -0.0375177 -0.9880356 -0.2538479  0.5996894 -1.6669739 -1.3206267
##  [7] -1.2978281  0.6879038 -1.0977298 -1.2107913
```

```
y = rnorm(10,mean=10,sd=0.1)
y
```

```
##  [1] 10.024141 10.041109  9.925157 10.036782 10.027682  9.962409 10.028042
##  [8] 10.047017 10.042228 10.057134
```

Set seed for the RNG

```
set.seed(232)
rnorm(10)
```

```
##  [1]  1.75298569 -1.25529692  0.40855173  0.24528662  0.01703264
##  [6]  0.51582884  0.47480988 -1.17967062  0.17695023 -1.07310557
```

Correlation

Compute the correlation of numbersets

```
x = rnorm (50)
y = x + rnorm (50 , mean =50 , sd =.1)
cor ( x , y )
```

```
## [1] 0.994691
```

Mean

```
x = rnorm(100,mean=10,sd=0.1)
mean(x)
```

```
## [1] 9.996454
```

Variance

```
x = rnorm(100,mean=10,sd=0.1)
var(x)
```

```
## [1] 0.01015385
```

Standard deviation

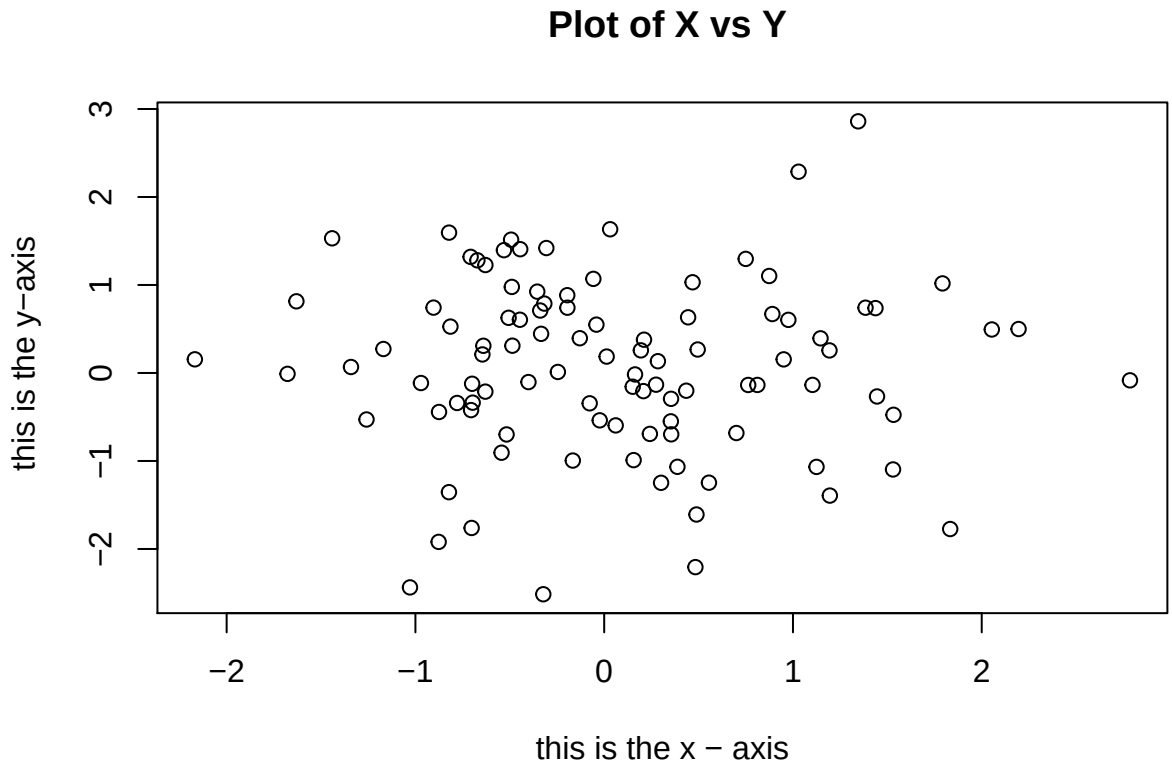
```
x = rnorm(100,mean=10,sd=0.1)
sqrt(var(x))
```

```
## [1] 0.09669588
```

Graphics

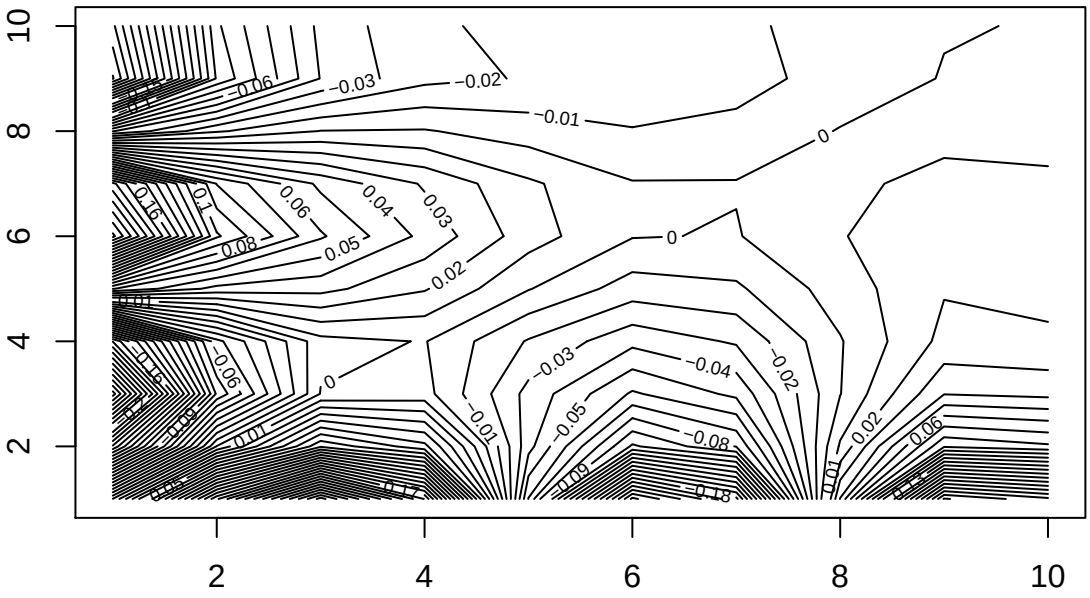
Scatter plot

```
x = rnorm (100)
y = rnorm (100)
plot (x ,y , xlab =" this is the x - axis " , ylab =" this is the y-axis", main =" Plot of X vs Y")
```



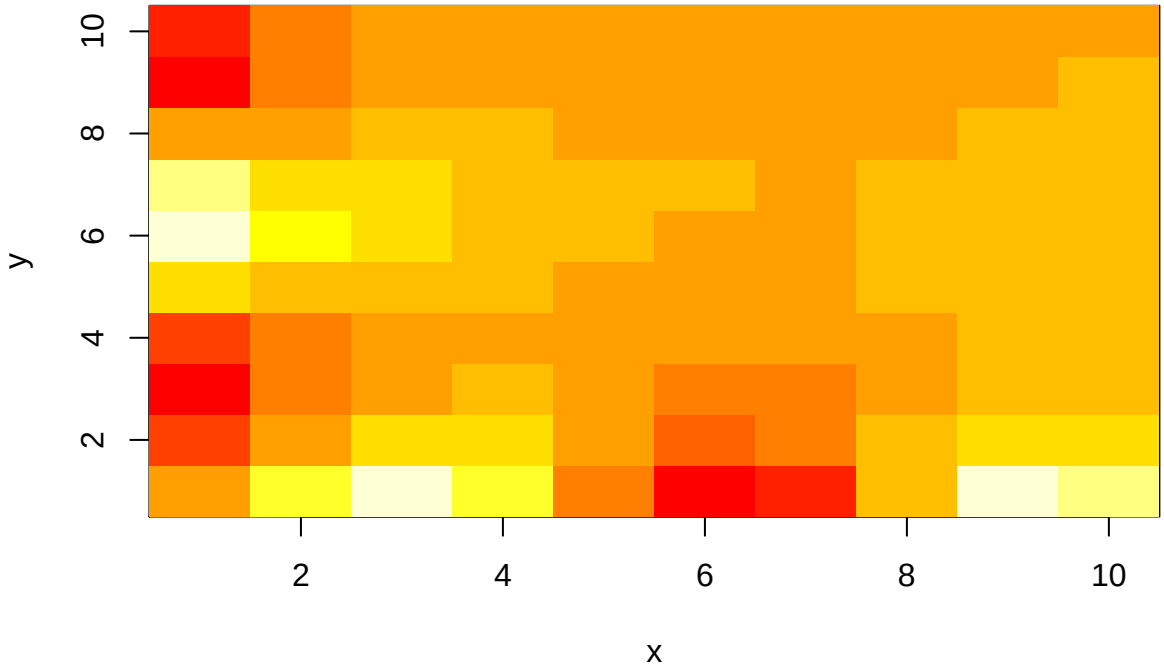
Contour plot

```
x = seq (1 ,10)
y = x
f = outer (x ,y , function (x , y ) cos ( y ) /(1+ x ^2) )
fa =( f - t ( f ) ) /2
contour (x ,y , fa , nlevels =45)
```



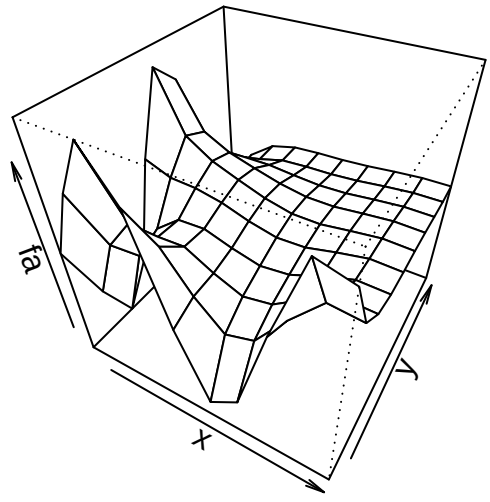
####Heat map

```
x = seq (1 ,10)
y = x
f = outer (x ,y , function (x , y ) cos ( y ) /(1+ x ^2) )
fa =( f - t ( f ) ) /2
image (x ,y , fa)
```



####Three-dimensional plot

```
x = seq (1 ,10)
y = x
f = outer (x ,y , function (x , y ) cos ( y ) /(1+ x ^2) )
fa =( f - t ( f ) ) /2
persp (x ,y , fa , theta =30 , phi =40)
```



Save Plot

Save plot to an output file. There are functions for different file types(pdf(), jpeg())

```
pdf("Figure.pdf")
plot(x,y,col="green")
dev.off()
```

```
## pdf
##   2
```

Data import/export

```
Auto=read.table("Auto.data")
fix(Auto)
```