R

Bill Last Updated:

23 March, 2020

Contents

Preface: Motivation **1**

1 apply, lapply, sapply **3**

1.1 apply 3

1.2 lapply 3

1.3 sapply 4

2 C **7**

3 GLM and GLiM **9**

Preface: Motivation

All the notes I have done here are about R. While I have tried my best, probably there are still some typos and errors. Please feel free to let me know in case you find one. Thank you!

Chapter 1

apply, lapply, sapply

1.1 apply

```
m_trying <- matrix(C<-(1:10),nrow=2, ncol=5)
m_trying
```

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

```
## Operating on the columns
apply(m_trying, 2, sum)
```

```
## [1]  3  7 11 15 19
```

```
## Operating on the rows
apply(m_trying, 1, sum)
```

```
## [1] 25 30
```

1.2 lapply

"lapply returns a list of the same length as X, each element of which is the result of applying FUN to the corresponding element of X."

lapply operates on lists. Thus, as we can see below, even if m_trying is not a list, each cell becomes a list.

```
results1<-lapply(m_trying,sum)
str(results1)
```

```
## List of 10
##  $ : int 1
##  $ : int 2
##  $ : int 3
##  $ : int 4
##  $ : int 5
##  $ : int 6
##  $ : int 7
##  $ : int 8
##  $ : int 9
##  $ : int 10
```

```
is.list(results1)
```

```
## [1] TRUE
```

1.3 sapply

"sapply() function takes list, vector or data frame as input and gives output in vector or matrix."

```
results2<-sapply(m_trying, sum)
str(results2)
```

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

```
is.list(results2)
```

```
## [1] FALSE
```

```
is.matrix(results2)
```

```
## [1] FALSE
```

```
is.data.frame(results2)
```

```
## [1] FALSE
```

```r
is.vector(results2)
```

```
## [1] TRUE
```

Chapter 2

C

```
mydata1<-matrix(runif(4*2),4,2)
mydata1
```

```
##           [,1]       [,2]
## [1,] 0.5715006 0.18031906
## [2,] 0.3111623 0.96314857
## [3,] 0.5261412 0.09350421
## [4,] 0.5559436 0.50264512
```

```
str(mydata1)
```

```
##  num [1:4, 1:2] 0.572 0.311 0.526 0.556 0.18 ...
```

```
mydata2<-c(mydata1)
mydata2
```

```
## [1] 0.57150060 0.31116229 0.52614124 0.55594361 0.18031906 0.96314857 0.09350421
## [8] 0.50264512
```

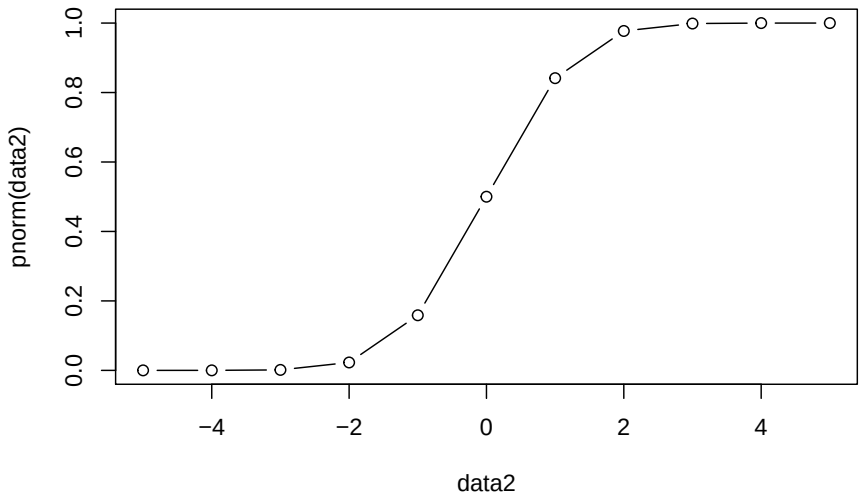
```
str(mydata2)
```

```
##  num [1:8] 0.572 0.311 0.526 0.556 0.18 ...
```

Chapter 3

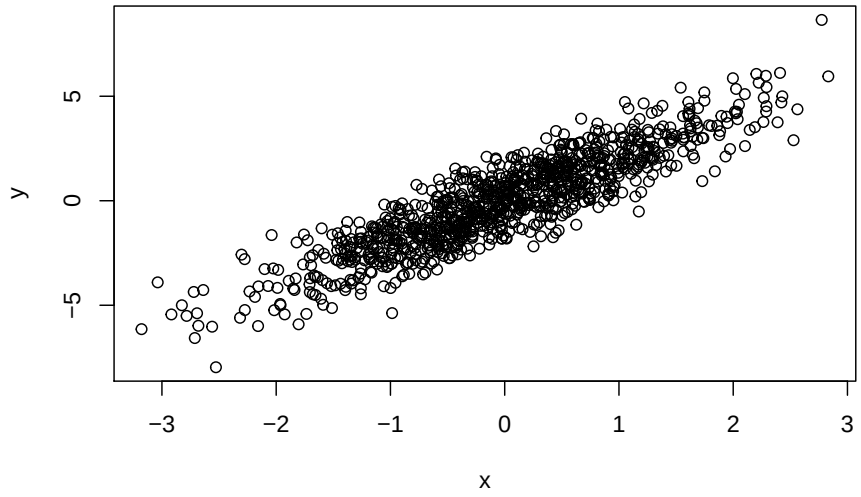
GLM and GLiM

```
data2<-seq(-5,5,1)
plot(data2,pnorm(data2),type = "b")
```

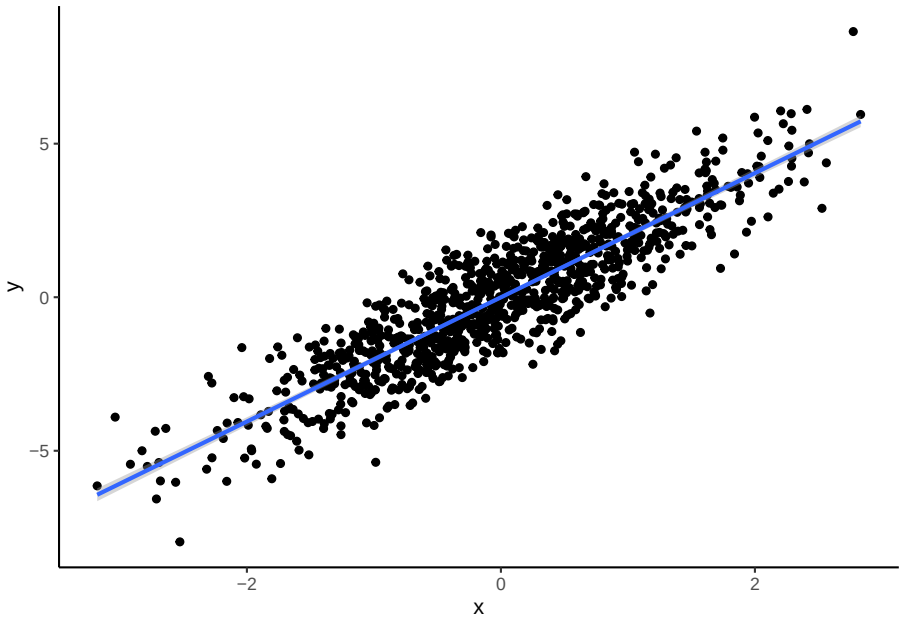



```
b_1<-2

x<-rnorm(1000)
```

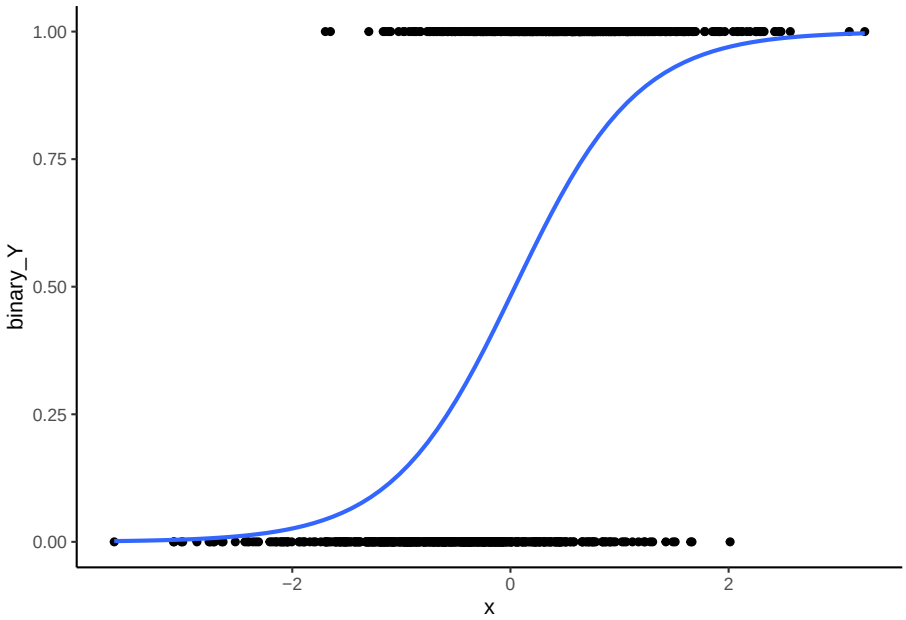
```
y=b_1*x+rnorm(1000)
plot(x,y)
```




```
data_1<-as.data.frame(cbind(y,x))

library(ggplot2)
ggplot(data_1,aes(x, y)) +
  geom_point() +
  geom_smooth(method = "lm")+theme_classic()
```




```r
x<-rnorm(1000)
y=b_1*x+rnorm(1000)
mu=exp(y)/(1+exp(y))


u<-runif(1000)
binary_Y<-ifelse(mu>u,1,0)
data_2<-as.data.frame(cbind(binary_Y,x))

library(ggplot2)
ggplot(data_2,aes(x, binary_Y)) +
  geom_point() +
  geom_smooth(method = "glm", method.args = list(family = "binomial"),
              se = FALSE)+theme_classic()
```


1.00
0.75
0.50
0.25
0.00
binary_Y
−2
0
2
x