Grupo Hudson

Abstract

Enter the text of your abstract here.

Introduction

Here goes an introduction text

Headings: first level

You can use directly LaTeX command or Markdown text.

LaTeX command can be used to reference other section. See Section . However, you can also use **bookdown** extensions mechanism for this.

Headings: second level

You can use equation in blocks

$$\xi_{ij}(t) = P(x_t = i, x_{t+1} = j | y, v, w; \theta) = \frac{\alpha_i(t) a_{ij}^{w_t} \beta_j(t+1) b_j^{v_{t+1}}(y_{t+1})}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i(t) a_{ij}^{w_t} \beta_j(t+1) b_j^{v_{t+1}}(y_{t+1})}$$

But also inline i.e $z = x + y$

Headings: third level

Another paragraph.

Examples of citations, figures, tables, references

You can insert references. Here is some text [@kour2014real; @kour2014fast] and see @hadash2018estimate.

The documentation for `natbib` may be found at

You can use custom blocks with LaTeX support from **rmarkdown** to create environment.

http://mirrors.ctan.org/macros/latex/contrib/natbib/natnotes.pdf%7D

Of note is the command `\citet`, which produces citations appropriate for use in inline text.

You can insert LaTeX environment directly too.

```
\citet{hasselmo} investigated\dots
```

produces

> Hasselmo, et al. (1995) investigated...

https://www.ctan.org/pkg/booktabs

Figures

You can insert figure using LaTeX directly.

See Figure 1. Here is how you add footnotes. [^Sample of the first footnote.]

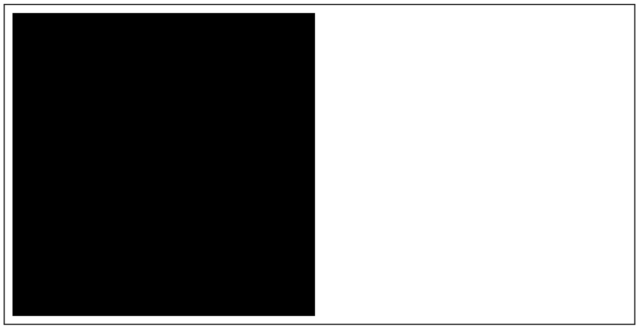

Figure 1: Sample figure caption.

But you can also do that using R.

```r
plot(mtcars$mpg)
```

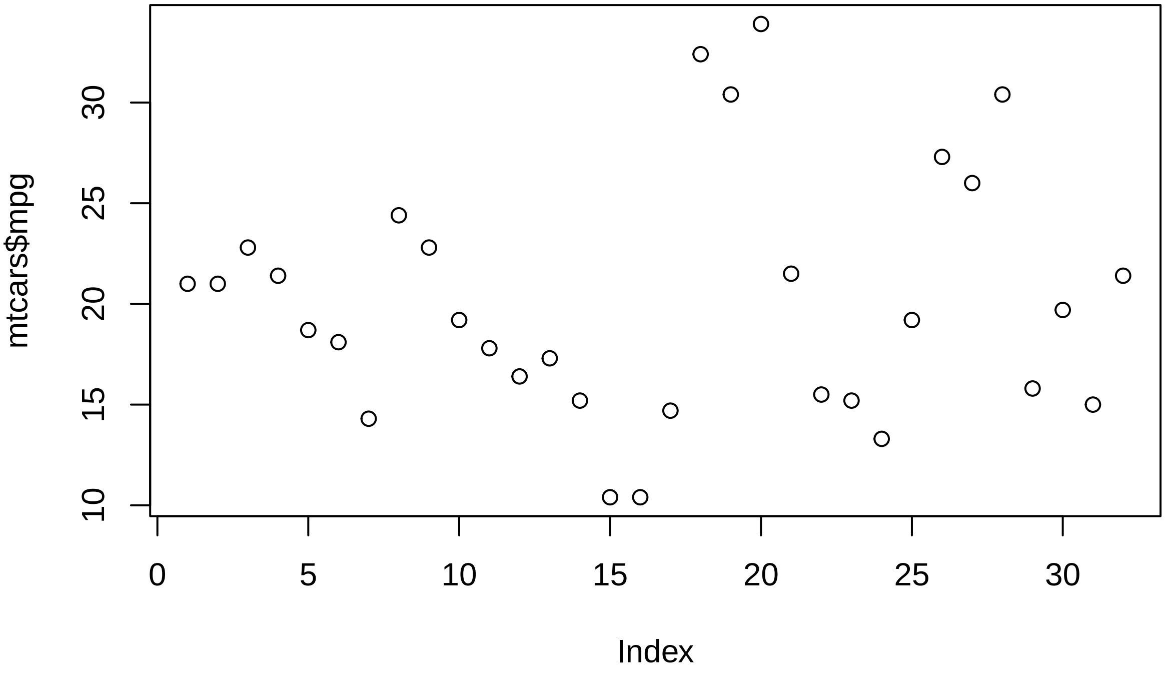

Figure 2: Another sample figure

You can use **bookdown** to allow references for Tables and Figures.

Tables

Below we can see how to use tables.

See awesome Table~1 which is written directly in LaTeX in source Rmd file.

You can also use R code for that.

Table 1: Sample table title

Part		
Name	Description	Size (μm)
Dendrite	Input terminal	∼100
Axon	Output terminal	∼10
Soma	Cell body	up to 10^6

```r
knitr::kable(head(mtcars), caption = "Head of mtcars table")
```

Table 2: Head of mtcars table

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225	105	2.76	3.460	20.22	1	0	3	1

Lists

- Item 1
- Item 2
- Item 3

Estructura de datos

Vectores

Un vector es una estructura de datos que almacena números de doble presición.

```
mi_vector_a <- c(12,34,12,54,23,12,65,34,12,56,66)
mi_vector_b <- c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16)#seq(1:16)

mi_vector_a
```

```
##  [1] 12 34 12 54 23 12 65 34 12 56 66
```

```
mi_vector_b
```

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

Suma de vectores

```
sum(mi_vector_a,mi_vector_b)
```

```
## [1] 516
```

Matrices

Las matrices se parecen a los vectores, pero tienen filas y columnas. Se alimentan de vectores.

```
mi_matriz_c <- matrix(mi_vector_b, nrow=4, byrow=4)
mi_matriz_c
```

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

Llenar por fila o columna la matriz

byrow=TRUE, me llena por fila byrow=FALSE, me llena por columna

¿Cómo accedo a un elemento de la matriz?

```
mi_matriz_c[2,4]
```

```
## [1] 8
```

¿Cómo traer una fila completa?

```
mi_matriz_c[2, ]
```

```
## [1] 5 6 7 8
```

¿Cómo traer una columna completa?

```
mi_matriz_c[ ,3]
```

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

¿Cómo accedo a toda la matriz menos la fila/columna 2?

```
mi_matriz_c[-2, ]
```

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

Tiempo que se demora en ejecutar un algoritmo

Usando Sys.time

Cálculo del tiempo que se demora en armar la matriz el algoritmo:

```
mi_vector_d <- seq(1:100)
start_time <- Sys.time()
mi_matriz_e <- matrix(mi_vector_d, nrow=10, byrow=TRUE)
end_time <- Sys.time()
end_time - start_time
```

```
## Time difference of 0.001140118 secs
```

Método tictoc

```
library(tictoc)
mi_vector_f <- seq(1:100)
tic("Tiempo que se demora en hacer la matriz g")
mi_matriz_g <- matrix(mi_vector_d, nrow=10, byrow=TRUE)
mi_vector_f
```

```
##   [1]   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
##  [19]  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
##  [37]  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
##  [55]  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
##  [73]  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
##  [91]  91  92  93  94  95  96  97  98  99 100
```

```
mi_matriz_g
```

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
##  [1,]    1    2    3    4    5    6    7    8    9    10
##  [2,]   11   12   13   14   15   16   17   18   19    20
##  [3,]   21   22   23   24   25   26   27   28   29    30
##  [4,]   31   32   33   34   35   36   37   38   39    40
##  [5,]   41   42   43   44   45   46   47   48   49    50
##  [6,]   51   52   53   54   55   56   57   58   59    60
##  [7,]   61   62   63   64   65   66   67   68   69    70
##  [8,]   71   72   73   74   75   76   77   78   79    80
##  [9,]   81   82   83   84   85   86   87   88   89    90
## [10,]   91   92   93   94   95   96   97   98   99   100
```

```
toc()
```

```
## Tiempo que se demora en hacer la matriz g: 0.003 sec elapsed
```

Penitencia de Gauss

A este método lo realizamos de 2 formas:

Sumas y multiplicación

```
start_time <- Sys.time()
suma <- 0
n<-10000
for (i in 1:n) {
    suma <- suma + i
}
suma
```

```
## [1] 50005000
```

```
end_time <- Sys.time()
end_time - start_time
```

```
## Time difference of 0.01700115 secs
```

```
n<-500
mi_vector_b<- seq(1:n)
S1<-0
R<-0
S1<-mi_vector_b[1]+mi_vector_b[n]
R<-(n-1)/2*S1
R
```

```
## [1] 124999.5
```

For con bucle

```
m<-500
mi_vector_a<- seq(1:m)
R <- 0

for (i in 1:m) {
  R <- R + mi_vector_a[i]
}
R
```

```
## [1] 125250
```

Serie Fibonacci

```
start_time <- Sys.time()
a<-0
b<-1
c<-a+b

  while (c<=1000000) {
    a<-b
    b<-c
    c<-a+b
  }
c
```

```
## [1] 1346269
```

```
end_time <- Sys.time()
end_time - start_time
```

```
## Time difference of 0.005337715 secs
```

Método Burbuja

```
x<-sample(1:100,10)
start_time <- Sys.time()
burbuja <- function(x){
n<-length(x)
for(j in 1:(n-1)){
for(i in 1:(n-j)){
if(x[i]>x[i+1]){
temp<-x[i]
x[i]<-x[i+1]
x[i+1]<-temp
}
}
}
return(x)
}
res<-burbuja(x)
end_time <- Sys.time()
end_time - start_time
```

```
## Time difference of 0.01128387 secs
```

```
Código html w3
```

```
<html>
<head>
Titulo
</head>
<h1> Titulo </h1>
</head>
</head>
```

Este código es compatible con w3 Consortium Ver: [@w3extensible] .

Está conformado siguien las regles de paridad da tags. esto quiere decir que todo tag que se abre, luego se cierra.

Referencias Bibliográficas

Listado de biboiográfía páginas de web y material consultado para este trabajo.

.