

---

# MODULO 2

---

A PREPRINT

**María Emilia Gullé**  
Facultad de Ingeniería  
Universidad Nacional de Cuyo  
Mendoza, Chacras de Coria 5507  
emigulle@gmail.com

**Guadalupe Gual Grossi**  
Facultad de Ingeniería  
Universidad Nacional de Cuyo  
Mendoza, San Martín 5570  
guadalupegual03@gmail.com

**Lucía Boscafiore Ferrando**  
Facultad de Ingeniería  
Universidad Nacional de Cuyo  
Mendoza, Chacras de Coria 5507  
luciaboscafiore@gmail.com

**Martín Federico Herrera**  
Facultad de Ingeniería  
Universidad Nacional de Cuyo  
Mendoza, San Martín 5570  
martinfederico.herrera@gmail.com

**Delfina Di Paola**  
Facultad de Ingeniería  
Universidad Nacional de Cuyo  
Mendoza, San Martín 5570  
delfinadipaola2003@gmail.com

**María Paz Antich**  
Facultad de Ingeniería  
Universidad Nacional de Cuyo  
Mendoza, San Martín 5570  
mariapazantich2@gmail.com

April 17, 2024

## Abstract

En este documento se muestran las distintas herramientas aprendidas en la materia Técnica y Herramientas Modernas I.

## 1 Módulo 1

### 1.1 HTML

El Lenguaje de Marcado de Hipertexto (HTML) es el código que se utiliza para estructurar y desplegar una página web y sus contenidos. Por ejemplo, sus contenidos podrían ser párrafos, una lista con viñetas, o imágenes y tablas de datos.

#### 1.1.1 Experiencia programando con HTML

Aprendimos sobre la programación de una página web mediante la utilización del lenguaje HTML. Lo que hicimos fue realizar nuestra propia página web, la cual muestra el nombre del grupo, los participantes del mismo y detalla información extra de cada participante como su email y su perfil de github (el cual se detalla más adelante). Para la realización de la misma escribimos el código en las notas de nuestras computadoras y lo guardamos como “.html”, esta acción es muy importante ya que está indicando que ese archivo es un documento HTML, es decir, un documento que contiene código HTML y que está destinado a ser interpretado y mostrado por un navegador web. En la página pudimos incluir imágenes, así como también vínculos los cuales llevan al lector a la redacción de un mail con un destinatario ya establecido, así como también vínculos que permiten acceder a los perfiles de github de cada participante. A continuación se muestran algunos códigos HTML que nos permitieron colocar imágenes, vínculos y el cambio de tipografía respectivamente:

```

<a href="https://github.com/emigulle">Perfil Github</a>
<h1> Equipo Linces</h1>
```

La realización de la página fue una experiencia muy enriquecedora ya que ninguno de nosotros había trabajado con algo así antes y pudimos aprender mucho.

### 1.1.2 Primer código HTML

A continuación se muestra la programación del primer página web realizada durante el cursado:

```
<html>

  <head>

    <h1> Equipo Linces</h1> <p>
  </head>

<body>

<h3> Integrantes del grupo: </h3>
<ul>
<li> Antich, Maria Paz
<li> Boscafiori, Lucia
<li> Di Paola, Delfina
<li> Gual, Guadalupe
<li> Gulle Linares, María Emilia
<li> Herrera, Martín
</ul> <p>

<a href="https://github.com/MartinFH2002/MODULO1
">MODULO 1</a>

<hr>
 <p>

<h3>Emilia Gullé Linares </h3> <p>
mail:
  <a href="mailto:emigulle@gmail.com
">emigulle@gmail.com</a>
<p>
<a href="https://github.com/emigulle">Perfil Github</a>

<hr>
 <p>

<h3>Guadalupe Gual Grossi </h3><p>
mail:
  <a href="mailto:guadalupegual03@gmail.com
">guadalupegual03@gmail.com</a>
<p>
<a href="https://github.com/Guada7890">Perfil Github</a>

<hr>
 <p>
```

```

<h3> Lucia Boscafiori </h3> <p>
mail:
  <a href="mailto:luciaboscafiori@gmail.com"
">luciaboscafiori@gmail.com</a>
<p>
<a href="https://github.com/luciaboscafiori"
">Perfil Github</a>

<hr>
 <p>

<h3> Martin Herrera </h3> <p>
mail:
  <a href="mailto:martinfederico.herrera@gmail.com"
">martinfederico.herrera@gmail.com</a>
<p>
<a href="https://github.com/MartinFH2002"
">Perfil Github</a>

<hr>
 <p>

<h3> María Paz Antich </h3> <p>
mail:
  <a href="mailto:mariapazantich2@gmail.com">mariapazantich2@gmail.com</a>
<p>
<a href="https://github.com/mariapazantich">Perfil Github</a>

<hr>
 <p>

<h3> Delfina Di Paola </h3> <p>
mail:
  <a href="mailto:delfinadipaola2003@gmail.com">delfinadipaola2003@gmail.com</a>
<p>
<a href="https://github.com/delfinadipa">Perfil Github</a>

</body>
</html>

```

## 1.2 Github

### 1.2.1 Creación y uso

En la segunda clase de Técnicas y Herramientas Modernas aprendimos como crear usuarios en GitHub, una plataforma líder en desarrollo colaborativo que utiliza el sistema de control de versiones Git, ya que como equipo de desarrollo entendemos la importancia de utilizar herramientas colaborativas eficientes para gestionar nuestros proyectos de software.

Para comenzar, visitamos el sitio web de GitHub y seleccionamos la opción “Sign up” (Registrarse) en la esquina superior derecha de la página. Luego, completamos el formulario de registro con nuestra información personal, incluyendo un nombre de usuario único, una dirección de correo electrónico válida y una contraseña segura. Después de verificar nuestra dirección de correo electrónico, nuestro usuario de GitHub fue creado exitosamente.

GitHub nos proporciona un espacio centralizado para almacenar y gestionar nuestro código fuente. Los repositorios en GitHub actúan como contenedores para nuestros proyectos, permitiéndonos colaborar con otros miembros del equipo y realizar un seguimiento de los cambios realizados en el código. Además, GitHub ofrece herramientas como solicitudes de extracción, problemas y seguimiento de problemas, que facilitan la colaboración y la gestión de proyectos de software de manera eficiente.

Además, tuvimos que elegir un nombre para el grupo de trabajo, no solo para identificarnos en GitHub sino también para identificarnos en las clases de trabajo y en el resto del cursado. Es por eso que nuestro grupo se llama “Equipo Linces”. Creamos una pestaña en GitHub en la cual subimos datos del grupo como por ejemplo fotos de cada uno. Al enlace de esta pestaña lo pegamos en la página web del grupo que armamos esa misma clase, y además pegamos los links de GitHub de cada participante del grupo, de modo tal que si se aprieta alguno de estos, se dirige directamente al perfil de GitHub de esta persona.

GitHub sirve para una variedad de propósitos útiles para desarrolladores de software y equipos de trabajo en proyectos colaborativos:

- **Gestión de versiones:** GitHub utiliza Git, un sistema de control de versiones, para permitir a los desarrolladores mantener un historial completo de cambios en su código. Esto facilita la colaboración, el seguimiento de errores y la gestión de versiones de software.
- **Colaboración:** GitHub ofrece una plataforma centralizada donde los desarrolladores pueden colaborar en proyectos de software. Permite a los equipos trabajar juntos en el mismo código, realizar revisiones de código, enviar y revisar solicitudes de extracción y fusionar cambios de manera eficiente.
- **Almacenamiento en la nube:** GitHub proporciona almacenamiento en la nube para repositorios de código, lo que permite a los desarrolladores acceder y compartir su código desde cualquier lugar con conexión a internet.
- **Seguimiento de problemas y tareas:** GitHub incluye herramientas para seguir problemas, administrar tareas y realizar un seguimiento del progreso del trabajo en equipo. Los equipos pueden organizar y priorizar tareas, asignar responsabilidades y mantener un registro de los problemas encontrados en el código.
- **Despliegue continuo:** GitHub proporciona integración con servicios de despliegue continuo (CI/CD), lo que permite a los equipos automatizar pruebas y despliegues de software. Esto ayuda a mejorar la calidad del código y a acelerar el proceso de entrega de software.
- **Comunidad y colaboración abierta:** GitHub es una plataforma social donde los desarrolladores pueden descubrir, contribuir y colaborar en proyectos de código abierto. Esto fomenta el intercambio de conocimientos y la construcción de comunidades en torno a proyectos de software.

### 1.2.2 Página web de Github

Se realizó una página web del equipo de trabajo en Github, en lenguaje HTML, donde aparecen los integrantes del grupo, sus perfiles de Github y el link de las entregas de los distintos módulos

```
<html>

  <head>

    <h1> Equipo Linces</h1> <p>
  </head>

<body>

<h3> Integrantes del grupo: </h3>
<ul>
<li> Antich, Maria Paz
<li> Boscafiori, Lucia
<li> Di Paola, Delfina
<li> Gual, Guadalupe
<li> Gulle Linares,María Emilia
<li> Herrera, Martín
```

```

</ul> <p>

<hr>
  <h2> MODULO 1 <a href="https://github.com/MartinFH2002/MODULO1">Link al Módulo 1-Martín Herrera</a>
  <h2> MODULO 2 <a href="https://github.com/delfinadipa/MODULO-2">Link al Módulo 2-Delfina Di Paola</a>

<hr>
 <p>

<h3>Emilia Gullé Linares </h3> <p>
mail:
  <a href="mailto:emigulle@gmail.com">emigulle@gmail.com</a>
<p>
<a href="https://github.com/emigulle">Perfil Github</a>

<hr>
 <p>

<h3>Guadalupe Gual Grossi </h3><p>
mail:
  <a href="mailto:guadalupegual03@gmail.com">guadalupegual03@gmail.com</a>
<p>
<a href="https://github.com/Guada7890">Perfil Github</a>

<hr>
 <p>

<h3> Lucia Boscafiori </h3> <p>
mail:
  <a href="mailto:luciaboscafiori@gmail.com">luciaboscafiori@gmail.com</a>
<p>
<a href="https://github.com/luciaboscafiori">Perfil Github</a>

<hr>
 <p>

<h3> Martin Herrera </h3> <p>
mail:
  <a href="mailto:martinfederico.herrera@gmail.com">martinfederico.herrera@gmail.com</a>
<p>
<a href="https://github.com/MartinFH2002">Perfil Github</a>

<hr>
 <p>

<h3> María Paz Antich </h3> <p>
mail:
  <a href="mailto:mariapazantich2@gmail.com">mariapazantich2@gmail.com</a>
<p>

```

```

<a href="https://github.com/mariapazantich">Perfil Github</a>

<hr>
 <p>

<h3> Delfina Di Paola </h3> <p>
mail:
<a href="mailto:delfinadipaola2003@gmail.com">delfinadipaola2003@gmail.com</a>
<p>
<a href="https://github.com/delfinadipa">Perfil Github</a>

</body>
</html>

```

### 1.3 LaTeX

LaTeX es un sistema de composición de textos de alta calidad que se utiliza ampliamente en el ámbito académico, científico y técnico para la creación de documentos con un diseño profesional y una presentación estética superior. A diferencia de los procesadores de texto convencionales, como Microsoft Word, que se centran en la apariencia visual directa, LaTeX se enfoca en la estructura y el contenido del documento, permitiendo a los usuarios concentrarse en el texto en lugar de preocuparse por el diseño. Algunas características del sistema LaTeX:

- Estructura modular: Los documentos LaTeX se dividen en secciones, subsecciones, capítulos y otros elementos estructurales. Esto permite una organización clara y facilita la gestión de documentos largos y complejos.
- Bibliografía y referencias cruzadas: LaTeX facilita la gestión de bibliografías y referencias cruzadas, lo que es especialmente útil para la escritura académica y científica. Los usuarios pueden generar automáticamente listas de referencias y citas en diferentes estilos de formato.
- Soporte para matemáticas: LaTeX es especialmente valorado por su capacidad para escribir y formatear ecuaciones matemáticas complejas de manera clara y elegante. Ofrece un conjunto completo de herramientas para notación matemática, fórmulas, símbolos y más.
- Tipografía de alta calidad: LaTeX está diseñado para producir documentos con una tipografía excepcionalmente buena. Utiliza algoritmos avanzados para el ajuste de líneas, el espaciado entre palabras y la disposición de elementos en la página.

Para utilizar LaTeX, los usuarios necesitan un editor LaTeX (como TeXShop, TeXstudio, Overleaf, entre otros) y un compilador LaTeX (como pdfLaTeX, XeLaTeX, LuaLaTeX) que convierta el código LaTeX en un documento final en formato PDF, DVI, PS u otro formato deseado. A continuación se describe el editor Overleaf que fue el utilizado en clase y con el cual se realizó este documento.

#### 1.3.1 Overleaf

Overleaf es un editor online de textos en formato LaTeX que permite la colaboración en tiempo real de los editores, así como escribir artículos científicos, técnicos, tesis, informes y cualquier documento que se quiera presentar en el formato indicado.

- Colaboración sencilla: Overleaf permite la colaboración en tiempo real entre autores. Todos los autores pueden trabajar en el mismo documento al mismo tiempo y comunicarse a través del chat.
- Visualización en tiempo real: Los cambios realizados por los autores son visibles al instante, lo que evita conflictos y facilita el trabajo conjunto a distancia.
- Facilidad de uso: Aunque LaTeX requiere conocimientos de código, Overleaf ofrece un editor de texto enriquecido que se asemeja a un editor tradicional, lo que facilita la escritura sin una curva de aprendizaje pronunciada.

-Resultados profesionales: A pesar de la simplicidad del editor, los documentos generados en LaTeX tienen un aspecto profesional y de alta calidad.

En resumen, Overleaf es una herramienta valiosa para la escritura colaborativa en LaTeX, en nuestro caso la utilizamos justamente para crear este documento.

##Rmarkdown La teoría general del lenguaje de marcas de texto se refiere al estudio de los principios fundamentales y la estructura subyacente de los lenguajes de marcado utilizados para formatear y estructurar texto en documentos electrónicos. Algunos aspectos clave que suelen abordarse en esta teoría son:

-Concepto de marcado: En el corazón de la teoría del lenguaje de marcas de texto está el concepto de marcado, que implica la inserción de etiquetas o caracteres especiales en un texto para indicar su estructura y formato. Estos marcadores son interpretados por programas informáticos o aplicaciones para renderizar el texto de acuerdo con las instrucciones proporcionadas.

-Sintaxis y gramática: Cada lenguaje de marcado tiene su propia sintaxis y gramática, que define cómo se deben escribir las etiquetas y cómo se relacionan entre sí para formar un documento coherente. Esta parte de la teoría aborda cómo se construyen y combinan las etiquetas para lograr el formato deseado.

-Elementos y atributos: Los lenguajes de marcado suelen tener elementos o etiquetas específicas que representan diferentes partes de un documento, como párrafos, encabezados, listas, enlaces, imágenes, etc. Además, estos elementos pueden tener atributos que proporcionan información adicional sobre ellos, como el tamaño de una imagen o el destino de un enlace.

-Interpretación y renderizado: Otro aspecto importante de la teoría del lenguaje de marcas de texto es cómo los programas informáticos interpretan y renderizan el texto marcado. Esto implica entender cómo las etiquetas se convierten en formatos visuales o estructuras de datos que pueden ser mostradas en pantallas o impresoras.

-Aplicaciones y usos: Se considera también el contexto de aplicación de los lenguajes de marcado, que abarca una amplia gama de campos, desde la creación de documentos web hasta la redacción de correos electrónicos y la preparación de documentos científicos. Esta parte de la teoría explora cómo los diferentes lenguajes de marcado se adaptan a diferentes necesidades y entornos de uso.

Es una forma de codificar un documento que, junto con el texto, incorpora etiquetas o marcas que contienen información adicional acerca de la estructura del texto o su presentación. El lenguaje de marcas más extendido es el HTML (HyperText Markup Language, lenguaje de marcado de hipertexto), el cual utilizamos en la primera y segunda clase de la materia.

## 2 Módulo 2

### 2.1 INFORME CLASE 4

#### 2.1.1 Comandos aprendidos de posit cloud

*Plot*

Dentro de la función para graficar, vimos que debemos colocar las variables deseadas o elegir una base de datos y la variable a graficar del mismo. Se pueden agregar detalles en el gráfico utilizando los siguientes comandos: “p” para utilizar puntos “l” para líneas “b” para ambos (puntos y líneas) “h” para hacer un tipo histograma con líneas verticales. Esto se agrega en el código de la siguiente manera: `plot(x, y, ...)` donde los comandos descriptores anteriormente se colocan en la posición que aparecen los 3 puntos (...) en el código de `plot`. -Summary: función genérica que se utiliza para producir resúmenes de los resultados de varias funciones de ajuste de modelos. Como argumentos del comando se debe colocar el objeto del cual se desea obtener el resumen, el número de dígitos de los datos que figurarán en el mismo, entre otros. -data.entry: Un editor similar a una hoja de cálculo para ingresar o editar datos. Como argumentos de la función se debe colocar: Una lista de variables: actualmente deben ser vectores numéricos o de caracteres o una lista que contenga dichos vectores. Los modos que se utilizarán para las variables. Los nombres que se utilizarán para las variables. Una lista de vectores numéricos y/o de caracteres.

### 2.2 Conclusion

Luego de todo lo aprendido en la clase 4 de técnicas y herramientas modernas llegamos a la conclusión de que al comienzo de las clases con el lenguaje HTML la complejidad de sintaxis fue alta, pero a su vez nos resultó

sencillo aprender el funcionamiento del mismo. Luego, con markdown la sintaxis nos resultó fácil y pudimos realizar la tarea en clase sin dificultad. Con latex, la sintaxis nos volvió a resultar compleja pero luego de un tiempo pudimos dominar la plataforma overleaf sin problema. Por ultimo, con posit cloud pudimos integrar todos los conocimientos previamente adquiridos. Sin embargo, el funcionamiento del mismo no requiere de previo conocimiento ya que te aporta toda la ayuda necesaria para su utilización.

## 2.3 Vectores

Un vector es una estructura de datos que almacena numeros de doble precisión.

```
mi_vector_a <- c(12,34,12,54,23,12,65,34,12,56,66)
mi_vector_b <- 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
```

## 2.4 Matrices

las matrices se parecen a vectores, pero tienen filas y columnas. se alimentan de vectores

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

```
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
```

Para acceder a un elemento de la matriz se pone el número de fila y de columna entre corchetes.

```
mi_matriz_c[2,3]
```

```
## [1] 7
```

¿Cómo traer la fila 4 completa?

```
mi_matriz_c[4,]
```

```
## [1] 13 14 15 16
```

¿Cómo traer una columna?

```
mi_matriz_c[,2]
```

```
## [1] 2 6 10 14
```

¿Que haría este comando?

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

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

Este comando trae toda la matriz sin la fila 2.



### 3 Introduction

Here goes an introduction text

### 4 Headings: first level

You can use directly LaTeX command or Markdown text.

LaTeX command can be used to reference other section. See Section 4. However, you can also use **bookdown** extensions mechanism for this.

#### 4.1 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  $z = x + y$

##### 4.1.1 Headings: third level

Another paragraph.

### 5 Examples of citations, figures, tables, references

You can insert references. Here is some text (Kour and Saabne 2014b, 2014a) and see Hadash et al. (2018).

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>

#### 5.1 Figures

You can insert figure using LaTeX directly.

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

But you can also do that using R.

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

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

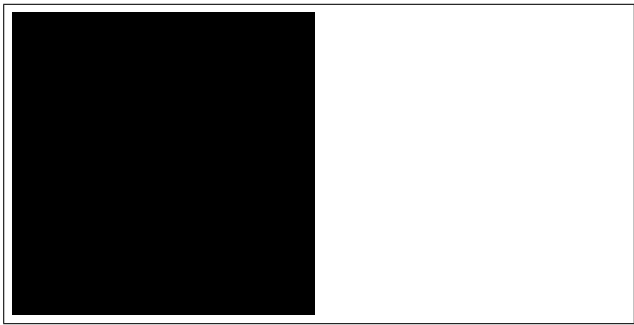


Figure 1: Sample figure caption.

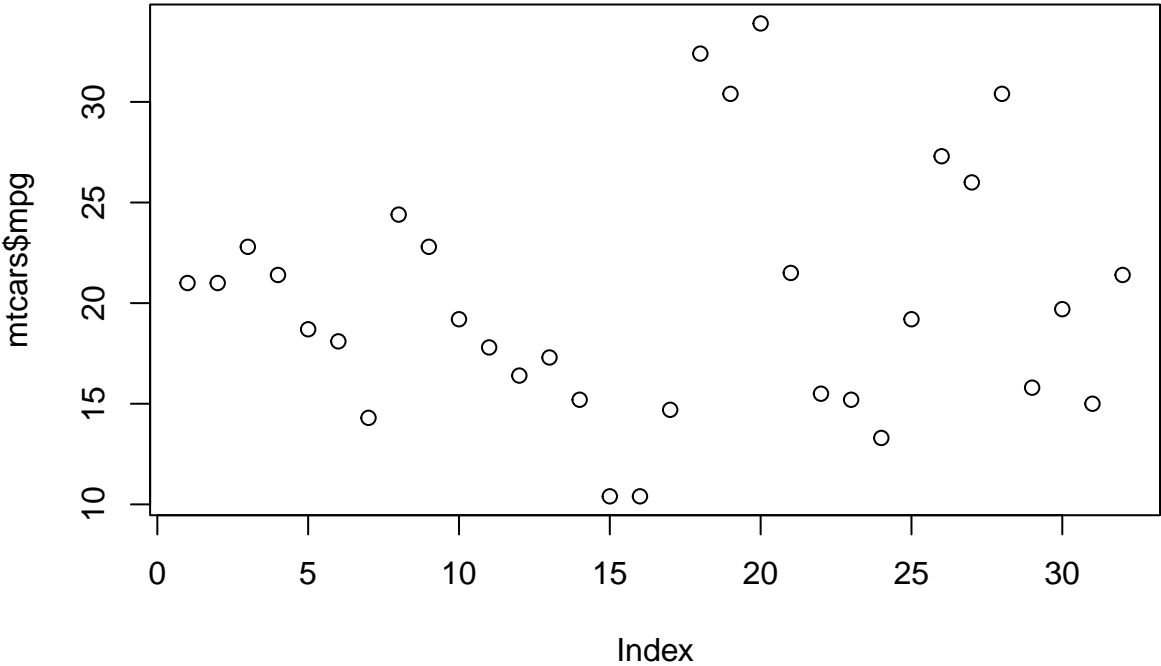


Figure 2: Another sample figure

Table 1: Sample table title

Part		
Name	Description	Size ( $\mu\text{m}$ )
Dendrite	Input terminal	$\sim 100$
Axon	Output terminal	$\sim 10$
Soma	Cell body	up to $10^6$

## 5.2 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.

```
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

## 5.3 Lists

- Item 1
- Item 2
- Item 3

# 6 Penitencia de Gauss

## 6.1 Primera versión

```
vec_gauss <- seq (1:6000)
start_time <- Sys.time()
pares <- (vec_gauss[6000])/2
suma <- (vec_gauss[1]+vec_gauss[6000])*pares
end_time <- Sys.time()
end_time - start_time
```

```
## Time difference of 0.003279924 secs
```

```
suma
```

```
## [1] 18003000
```

## 6.2 Segunda Versión

```
suma1<- 0
start_time<- Sys.time()
for (i in 1:6000)
suma1<- suma1+vec_gauss[i]
end_time<-Sys.time()
end_time-start_time
```

```
## Time difference of 0.02152109 secs
```

```
print(suma1)
```

```
## [1] 18003000
```

## 7 Fibonacci

```
start_time<- Sys.time()
a<-0
b<-1
c<- a+b
it<-0
while(c<=1000000){
a<-b
b<-c
c<-a+b
it<-it+1
}
end_time<-Sys.time()
end_time-start_time
```

```
## Time difference of 0.0398469 secs
```

```
c
```

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

```
it
```

```
## [1] 29
```

## 8 Método Burbuja

```
start_time1<-Sys.time()
# Tomo una muestra de 10 números ente 1 y 100
x<-sample(1:100,10)
# Creo una funcion para ordenar
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)
#Muestra obtenida
x
```

```
## [1] 97 26 7 28 19 75 92 12 59 20
```

```
## [1] 7 71 10 72 37 28 64 82 19 88
```

```
#Muestra Ordenada
```

```
res
```

```
## [1] 7 12 19 20 26 28 59 75 92 97
```

```
## [1] 7 10 19 28 37 64 71 72 82 88
end_time1<-Sys.time()
end_time1-start_time1
```

```
## Time difference of 0.01282001 secs
```

```
#Ordanaci?n con el coamando SORT de R-Cran
start_time<- Sys.time()
sort(x)
```

```
## [1] 7 12 19 20 26 28 59 75 92 97
```

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

```
## Time difference of 0.002014875 secs
```

Hadash, Guy, Einat Kermany, Boaz Carmeli, Ofer Lavi, George Kour, and Alon Jacovi. 2018. “Estimate and Replace: A Novel Approach to Integrating Deep Neural Networks with Existing Applications.” *arXiv Preprint arXiv:1804.09028*.

Kour, George, and Raid Saabne. 2014a. “Fast Classification of Handwritten on-Line Arabic Characters.” In *Soft Computing and Pattern Recognition (SoCPaR), 2014 6th International Conference of*, 312–18. IEEE.

———. 2014b. “Real-Time Segmentation of on-Line Handwritten Arabic Script.” In *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*, 417–22. IEEE.