

# Chapter4

CamilaSilva

8/19/2022

## Usuarios servidor remoto

<http://132.248.196.38:8787/> User: alumno# (1,2,...,15) Pasword: alumno#@CCM21 (1,2,...,15)

## R para Análisis Científicos Reproducibles

<https://swcarpentry.github.io/r-novice-gapminder-es/>

## Introduccion a R y RStudio

R es un lenguaje de programación y ambiente de computación estadística y gráficas; libre, de código abierto y de alto nivel.

Uno de los puntos fuertes de R es la facilidad con la que se pueden producir gráficos con calidad de publicación bien diseñados, incluidos símbolos matemáticos y fórmulas.

Proporciona una amplia variedad de métodos estadísticos y gráficos. Además es altamente extensible (“packages”).

RStudio es un ambiente de R que incluye una consola, un editor con syntax-highlighting que admite la ejecución directa de código, así como herramientas para graficar, almacenar el historial, depurar código y gestionar el espacio de trabajo.

Con R-Studio es posible desarrollar documentos HTML con la opción de Markdown. También se pueden exportar gráficas en diferentes formatos, según la necesidad.

R trae varias funciones en paquetes, pero también es posible extender su utilidad con la instalación de diferentes paquetes, según nuestra necesidad.

Los paquetes se pueden instalar con código (como se mostrará más adelante) o haciendo ‘click’ en ‘packages’, ‘install’ y buscando el paquete por su nombre.

Primero vamos a instalar el paquete ‘ALDEx2’, el cual es útil para datos estándar de secuencias de ARN.

```
#install.packages("ALDEx2") # este paquete no esta disponible para la version de R  
#para esta instalacion usamos la instalacion de bio conductor, ya que todos trabajaremos en el servidor  
#BiocManager::install()  
#por lo tanto solo es necesario llamar a la libreria  
library(ALDEx2)
```

```
## Loading required package: zCompositions
```

```
## Loading required package: MASS

## Loading required package: NADA

## Loading required package: survival

##
## Attaching package: 'NADA'

## The following object is masked from 'package:stats':
##
##      cor

## Loading required package: truncnorm
```

Si queremos ver una muestra de los paquetes instalados,

```
installed.packages()[1:5,] # que paquetes estan instalados
```

```
##      Package      LibPath      Version Priority
## ALDEx2      "ALDEx2"      "/usr/local/lib/R/site-library" "1.28.1" NA
## AlgDesign   "AlgDesign"   "/usr/local/lib/R/site-library" "1.2.1"  NA
## annotate    "annotate"    "/usr/local/lib/R/site-library" "1.74.0" NA
## AnnotationDbi "AnnotationDbi" "/usr/local/lib/R/site-library" "1.58.0" NA
## aplot       "aplot"       "/usr/local/lib/R/site-library" "0.1.6"  NA
##
## Depends
## ALDEx2      "methods, stats, zCompositions,"
## AlgDesign   NA
## annotate    "R (>= 2.10), AnnotationDbi (>= 1.27.5), XML"
## AnnotationDbi "R (>= 2.7.0), methods, utils, stats4, BiocGenerics (>=\n0.29.2), Biobase (>= 1.17.0),
## aplot       NA
##
## Imports
## ALDEx2      "Rfast, BiocParallel, GenomicRanges, IRanges, S4Vectors,\nSummarizedExperiment, multtest"
## AlgDesign   NA
## annotate    "Biobase, DBI, xtable, graphics, utils, stats, methods,\nBiocGenerics (>= 0.13.8), httr"
## AnnotationDbi "DBI, RSQLite, S4Vectors (>= 0.9.25), stats, KEGGREST"
## aplot       "ggfun (>= 0.0.6), ggplot2, ggplotify, patchwork, magrittr,\nmethods, utils"
##
## LinkingTo
## ALDEx2      NA
## AlgDesign   NA
## annotate    NA
## AnnotationDbi NA
## aplot       NA
##
## Suggests
## ALDEx2      "testthat, BiocStyle, knitr, rmarkdown"
## AlgDesign   NA
## annotate    "hgu95av2.db, genefilter, Biostrings (>= 2.25.10), IRanges,\nrae230a.db, rae230aprobe,
## AnnotationDbi "hgu95av2.db, G0.db, org.Sc.sgd.db, org.At.tair.db, RUnit,\nTxDb.Hsapiens.UCSC.hg19.knownGene"
## aplot       "ggtree"
##
## Enhances License      License_is_FOSS License_restricts_use
## ALDEx2      NA      "file LICENSE" NA      NA
## AlgDesign   NA      "GPL (>= 2)"  NA      NA
```

```
## annotate      NA      "Artistic-2.0" NA      NA
## AnnotationDbi NA      "Artistic-2.0" NA      NA
## applot        NA      "Artistic-2.0" NA      NA
##              OS_type MD5sum NeedsCompilation Built
## ALDEx2        NA      NA      "no"          "4.2.1"
## AlgDesign     NA      NA      "yes"         "4.2.0"
## annotate      NA      NA      "no"          "4.2.0"
## AnnotationDbi NA      NA      "no"          "4.2.0"
## applot        NA      NA      "no"          "4.2.0"
```

Para ver si fue posible instalar el paquete que necesitamos, podemos buscarlo:

```
a<-installed.packages()
packages<-a[,1]
is.element("ALDEx2",packages) # está este paquete en el elemento 'packages'
```

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

## Establecer directorio de trabajo en R

‘Working directory’ es en donde se tendran los datos sin procesar, los codigos de R y las salidas de un trabajo específico.

```
getwd() #para ver en que directorio estamos (path)
```

```
## [1] "/home/betterlab/GIT/Intro_R"
```

```
setwd("/home/betterlab/GIT/Intro_R") #crea un directorio especifico para trabajar,si queremos cambiar d
getwd()
```

```
## [1] "/home/betterlab/GIT/Intro_R"
```

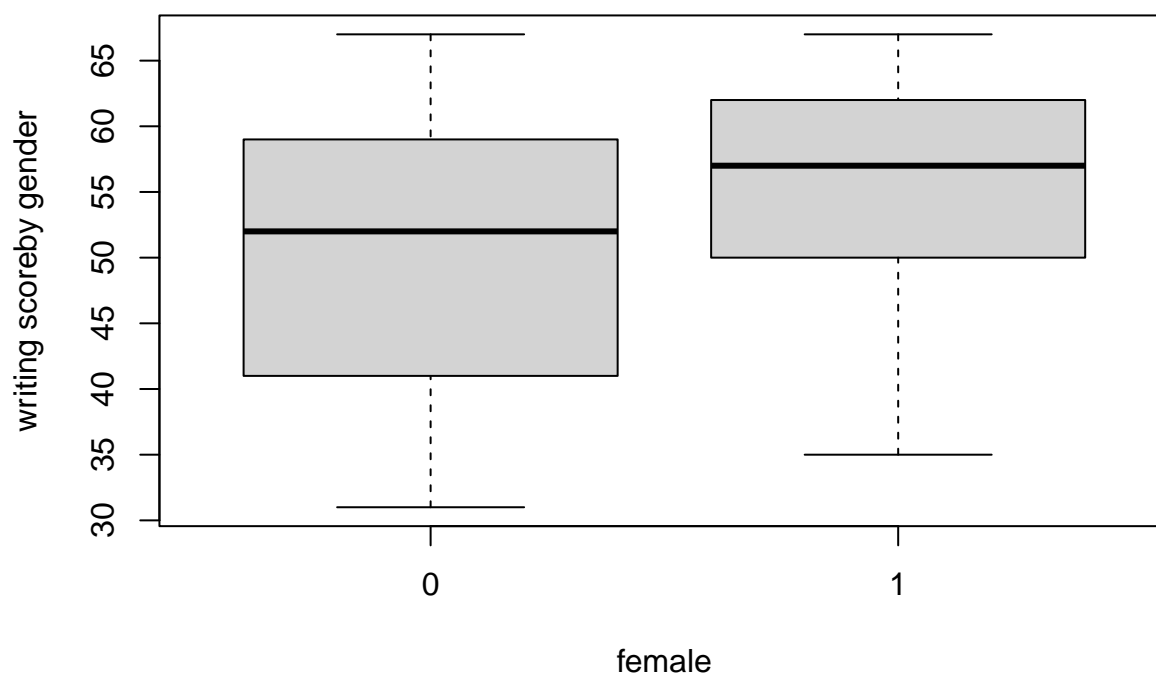
## Análisis de datos a través de RStudio

Para cargar datos externos, primero debemos tener un archivo con datos, o descargarlos; debemos tener el documento en el lugar de trabajo, o escribir el ‘path’ completo del documento.

Una manera de visualizacion de estos datos es con graficas, como la funcion ‘boxplot()’ que ya vienen predeterminadas en R. Para crear otros tipos de graficas, o de mayor calidad usaremos el paquete ‘ggplot2’

```
data <- read.csv("data/hsb2demo.csv")
boxplot(write~female,data,main="High School Students Data",slab="Gender",ylab="writing scoreby gender")
```

## High School Students Data



### Importar y exportar datos

Podemos importar y exportar datos, los formatos más usados para importar datos en estudios microbianos son: archivos de excel separados por coma (.CSV) y archivos generados por herramientas bioinformáticas y datos web variables.

Para la lectura de estos archivos en R, podemos usar:

```
#tab0 <- read.table("data/hsb2demo.csv",header=TRUE,row.names=1,sep=",")
#tab0
```

la función 'read.table' tiene varios parámetros, donde primero ponemos nuestro archivo, luego le indicamos que tiene 'cabeza' que indica que la primera entrada del archivo se interpreta como los nombres de las variables, luego tomamos la primera columna debe interpretarse como los nombres de las filas y por último indica como están separadas las columnas del archivo (las más conocidas ',', '/t')

```
tab1 <- read.table("data/hsb2demo.csv",header=TRUE,row.names=1,sep="\t")
tab1
```

## data frame with 0 columns and 200 rows

También es posible tomar datos directamente de una página web.

```
raw <- "https://raw.githubusercontent.com/swcarpentry/r-novice-gapminder/gh-pages/_episodes_rmd/data/gapm"
tab2 <- read.table(raw,sep='\t',header=TRUE,row.names=1,check.names=FALSE,stringsAsFactors=FALSE)
tab2
```

```
## data frame with 0 columns and 1704 rows
```

La función ‘read.delim()’ espera que la tabla de entrada sea texto sin formato con celdas separadas o delimitadas por tabuladores.

```
tab3 <- read.delim("data/hsb2demo.csv", header=T, row.names=1)
tab3
```

```
## data frame with 0 columns and 200 rows
```

Tambien es posible cargar directamente archivos de formato ‘csv’, ya que es uno de los mas usados.

```
tab4 <- read.csv('data/hsb2demo.csv', head=T, row.names=1, sep=',', dec='.')
tab5 <- read.csv2('data/hsb2demo.csv', head=T, row.names=1, sep=';', dec=',')
```

## Paquetes gdata y XLConnect

Los paquetes ‘gdata’ y ‘XLConnect’ nos ayudan con la lectura de archivos ‘xlsx’

```
#install.packages("gdata") # este paquete nos ayuda con la lectura de .xls directamente.
library(gdata)
```

```
## gdata: read.xls support for 'XLS' (Excel 97-2004) files ENABLED.
```

```
##
```

```
## gdata: read.xls support for 'XLSX' (Excel 2007+) files ENABLED.
```

```
##
```

```
## Attaching package: 'gdata'
```

```
## The following object is masked from 'package:stats':
```

```
##
```

```
##      nobs
```

```
## The following object is masked from 'package:utils':
```

```
##
```

```
##      object.size
```

```
## The following object is masked from 'package:base':
```

```
##
```

```
##      startsWith
```

```
tab6 <- read.xls("table.xlsx", sheet=1, header=TRUE) tab7 <- read.xls("table.xlsx", sheet=1, perl="C:/Perl64/bin/perl.exe")
```

```
#install.packages("XLConnect") # este paquete sirve para manipular archivos de excel en windows.
library(XLConnect)
```

```
## XLConnect 1.0.5 by Mirai Solutions GmbH [aut],
##   Martin Studer [cre],
##   The Apache Software Foundation [ctb, cph] (Apache POI),
##   Graph Builder [ctb, cph] (Curvesapi Java library),
##   Brett Woolridge [ctb, cph] (SparseBitSet Java library)
```

```
## https://mirai-solutions.ch
## https://github.com/miraisolutions/xlconnect
```

```
tab8 <- readWorksheetFromFile(file='table.xlsx',sheet=1,header=T,rownames=1)
```

La funcion 'write.table' puede escribir un archivo separado por comas o por tabulacion, legible por Excel.

```
write.table(hsb,file="data/hsb2demo.csv",quote=FALSE,row.names=FALSE,sep=" ") write.table(hsb,file="data/hsb2demo.t
```

## Manipulación básica de datos

En analisis de datos tenemos unos objetos llamados 'DataFrame' los cuales contienen los datos y una estructura con informacion sobre ellos.

R trae unos datos de prueba predeterminados para poder 'jugar con ellos' sin la necesidad de descargarlos o tener archivos de datos.

```
data()
attach(iris)
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1         5.1         3.5          1.4          0.2   setosa
## 2         4.9         3.0          1.4          0.2   setosa
## 3         4.7         3.2          1.3          0.2   setosa
## 4         4.6         3.1          1.5          0.2   setosa
## 5         5.0         3.6          1.4          0.2   setosa
## 6         5.4         3.9          1.7          0.4   setosa
```

Data frames (objetos de dos dimensiones): pueden tener valores numeric, character o logical. Dentro de la columna, todos los elementos tienen el mismo tipo de dato. Entre las columnas pueden tener distintos tipos de datos.

Nosotros podemos crear DataFrame con datos ya existentes.

```
#crear data frame usando indices de columna
df <- iris[,c(1,2,3)]
head(df)
```

```
##   Sepal.Length Sepal.Width Petal.Length
## 1         5.1         3.5          1.4
## 2         4.9         3.0          1.4
## 3         4.7         3.2          1.3
## 4         4.6         3.1          1.5
## 5         5.0         3.6          1.4
## 6         5.4         3.9          1.7
```

```
# crear data frame usando indices de columna con secuencias
df <- iris[,c(1:2,4:5)]
head(df)
```

```
##   Sepal.Length Sepal.Width Petal.Width Species
## 1         5.1         3.5         0.2   setosa
## 2         4.9         3.0         0.2   setosa
## 3         4.7         3.2         0.2   setosa
## 4         4.6         3.1         0.2   setosa
## 5         5.0         3.6         0.2   setosa
## 6         5.4         3.9         0.4   setosa
```

```
# crear data frame usando subset() e indices de columnas
df<- subset(iris, select=c(1,2, 4:5))
head(df)
```

```
##   Sepal.Length Sepal.Width Petal.Width Species
## 1         5.1         3.5         0.2   setosa
## 2         4.9         3.0         0.2   setosa
## 3         4.7         3.2         0.2   setosa
## 4         4.6         3.1         0.2   setosa
## 5         5.0         3.6         0.2   setosa
## 6         5.4         3.9         0.4   setosa
```

```
# crear data frame usando subset() e nombres de columnas
df <- subset(iris, select=c("Sepal.Width", "Petal.Length", "Petal.Width"))
head(df)
```

```
##   Sepal.Width Petal.Length Petal.Width
## 1         3.5         1.4         0.2
## 2         3.0         1.4         0.2
## 3         3.2         1.3         0.2
## 4         3.1         1.5         0.2
## 5         3.6         1.4         0.2
## 6         3.9         1.7         0.4
```

```
# crear data frame por seleccion de nombres de columnas
df <- iris[,c("Sepal.Width", "Petal.Length", "Petal.Width")]
head(df)
```

```
##   Sepal.Width Petal.Length Petal.Width
## 1         3.5         1.4         0.2
## 2         3.0         1.4         0.2
## 3         3.2         1.3         0.2
## 4         3.1         1.5         0.2
## 5         3.6         1.4         0.2
## 6         3.9         1.7         0.4
```

```
# crear data frame usando dataframe()
df <- data.frame(iris$Sepal.Width, iris$Petal.Length, iris$Petal.Width)
head(df)
```

```
##      iris.Sepal.Width iris.Petal.Length iris.Petal.Width
## 1           3.5           1.4           0.2
## 2           3.0           1.4           0.2
## 3           3.2           1.3           0.2
## 4           3.1           1.5           0.2
## 5           3.6           1.4           0.2
## 6           3.9           1.7           0.4
```

```
# crear data frame usando c() manualmente
Sepal.Width = c(3.5, 3.0, 3.2, 3.1, 3.6, 3.9)
Petal.Length = c(1.4, 1.4, 1.3, 1.5, 1.4, 1.7)
Petal.Width = c(0.2, 0.2, 0.2, 0.2, 0.2, 0.4)
df = data.frame(Sepal.Width, Petal.Length, Petal.Width)
df
```

```
##      Sepal.Width Petal.Length Petal.Width
## 1           3.5           1.4           0.2
## 2           3.0           1.4           0.2
## 3           3.2           1.3           0.2
## 4           3.1           1.5           0.2
## 5           3.6           1.4           0.2
## 6           3.9           1.7           0.4
```

## Operaciones Basicas

```
head(iris) #nos muestra una pequeña parte de los datos
```

```
##      Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1           5.1           3.5           1.4           0.2  setosa
## 2           4.9           3.0           1.4           0.2  setosa
## 3           4.7           3.2           1.3           0.2  setosa
## 4           4.6           3.1           1.5           0.2  setosa
## 5           5.0           3.6           1.4           0.2  setosa
## 6           5.4           3.9           1.7           0.4  setosa
```

```
attributes(iris) #imprime los nombres de las filas y columnas, y la clase de los datos
```

```
## $names
## [1] "Sepal.Length" "Sepal.Width"  "Petal.Length" "Petal.Width"  "Species"
##
## $class
## [1] "data.frame"
##
## $row.names
## [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 99 100 101 102 103 104 105 106 107 108
```



```
## [109] 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126
## [127] 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144
## [145] 145 146 147 148 149 150
```

```
class(iris) #imprime la clase de los datos
```

```
## [1] "data.frame"
```

```
dim(iris) #imprime la dimension del dataframe
```

```
## [1] 150 5
```

```
nrow(iris) #numero de filas
```

```
## [1] 150
```

```
ncol(iris) #numero de columnas
```

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

```
length(iris[, "Species"]) #imprime la longitud del vector dado, en este caso es una columna de nuestro
```

```
## [1] 150
```

```
colnames(iris) #nombre de las columnas
```

```
## [1] "Sepal.Length" "Sepal.Width" "Petal.Length" "Petal.Width" "Species"
```

```
rownames(iris) #nombre de las filas
```

```
## [1] "1" "2" "3" "4" "5" "6" "7" "8" "9" "10" "11" "12"
## [13] "13" "14" "15" "16" "17" "18" "19" "20" "21" "22" "23" "24"
## [25] "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] "49" "50" "51" "52" "53" "54" "55" "56" "57" "58" "59" "60"
## [61] "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] "85" "86" "87" "88" "89" "90" "91" "92" "93" "94" "95" "96"
## [97] "97" "98" "99" "100" "101" "102" "103" "104" "105" "106" "107" "108"
## [109] "109" "110" "111" "112" "113" "114" "115" "116" "117" "118" "119" "120"
## [121] "121" "122" "123" "124" "125" "126" "127" "128" "129" "130" "131" "132"
## [133] "133" "134" "135" "136" "137" "138" "139" "140" "141" "142" "143" "144"
## [145] "145" "146" "147" "148" "149" "150"
```

```
print(iris)
```

##	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
## 1	5.1	3.5	1.4	0.2	setosa
## 2	4.9	3.0	1.4	0.2	setosa
## 3	4.7	3.2	1.3	0.2	setosa
## 4	4.6	3.1	1.5	0.2	setosa
## 5	5.0	3.6	1.4	0.2	setosa
## 6	5.4	3.9	1.7	0.4	setosa
## 7	4.6	3.4	1.4	0.3	setosa
## 8	5.0	3.4	1.5	0.2	setosa
## 9	4.4	2.9	1.4	0.2	setosa
## 10	4.9	3.1	1.5	0.1	setosa
## 11	5.4	3.7	1.5	0.2	setosa
## 12	4.8	3.4	1.6	0.2	setosa
## 13	4.8	3.0	1.4	0.1	setosa
## 14	4.3	3.0	1.1	0.1	setosa
## 15	5.8	4.0	1.2	0.2	setosa
## 16	5.7	4.4	1.5	0.4	setosa
## 17	5.4	3.9	1.3	0.4	setosa
## 18	5.1	3.5	1.4	0.3	setosa
## 19	5.7	3.8	1.7	0.3	setosa
## 20	5.1	3.8	1.5	0.3	setosa
## 21	5.4	3.4	1.7	0.2	setosa
## 22	5.1	3.7	1.5	0.4	setosa
## 23	4.6	3.6	1.0	0.2	setosa
## 24	5.1	3.3	1.7	0.5	setosa
## 25	4.8	3.4	1.9	0.2	setosa
## 26	5.0	3.0	1.6	0.2	setosa
## 27	5.0	3.4	1.6	0.4	setosa
## 28	5.2	3.5	1.5	0.2	setosa
## 29	5.2	3.4	1.4	0.2	setosa
## 30	4.7	3.2	1.6	0.2	setosa
## 31	4.8	3.1	1.6	0.2	setosa
## 32	5.4	3.4	1.5	0.4	setosa
## 33	5.2	4.1	1.5	0.1	setosa
## 34	5.5	4.2	1.4	0.2	setosa
## 35	4.9	3.1	1.5	0.2	setosa
## 36	5.0	3.2	1.2	0.2	setosa
## 37	5.5	3.5	1.3	0.2	setosa
## 38	4.9	3.6	1.4	0.1	setosa
## 39	4.4	3.0	1.3	0.2	setosa
## 40	5.1	3.4	1.5	0.2	setosa
## 41	5.0	3.5	1.3	0.3	setosa
## 42	4.5	2.3	1.3	0.3	setosa
## 43	4.4	3.2	1.3	0.2	setosa
## 44	5.0	3.5	1.6	0.6	setosa
## 45	5.1	3.8	1.9	0.4	setosa
## 46	4.8	3.0	1.4	0.3	setosa
## 47	5.1	3.8	1.6	0.2	setosa
## 48	4.6	3.2	1.4	0.2	setosa
## 49	5.3	3.7	1.5	0.2	setosa
## 50	5.0	3.3	1.4	0.2	setosa
## 51	7.0	3.2	4.7	1.4	versicolor
## 52	6.4	3.2	4.5	1.5	versicolor
## 53	6.9	3.1	4.9	1.5	versicolor

## 54	5.5	2.3	4.0	1.3 versicolor
## 55	6.5	2.8	4.6	1.5 versicolor
## 56	5.7	2.8	4.5	1.3 versicolor
## 57	6.3	3.3	4.7	1.6 versicolor
## 58	4.9	2.4	3.3	1.0 versicolor
## 59	6.6	2.9	4.6	1.3 versicolor
## 60	5.2	2.7	3.9	1.4 versicolor
## 61	5.0	2.0	3.5	1.0 versicolor
## 62	5.9	3.0	4.2	1.5 versicolor
## 63	6.0	2.2	4.0	1.0 versicolor
## 64	6.1	2.9	4.7	1.4 versicolor
## 65	5.6	2.9	3.6	1.3 versicolor
## 66	6.7	3.1	4.4	1.4 versicolor
## 67	5.6	3.0	4.5	1.5 versicolor
## 68	5.8	2.7	4.1	1.0 versicolor
## 69	6.2	2.2	4.5	1.5 versicolor
## 70	5.6	2.5	3.9	1.1 versicolor
## 71	5.9	3.2	4.8	1.8 versicolor
## 72	6.1	2.8	4.0	1.3 versicolor
## 73	6.3	2.5	4.9	1.5 versicolor
## 74	6.1	2.8	4.7	1.2 versicolor
## 75	6.4	2.9	4.3	1.3 versicolor
## 76	6.6	3.0	4.4	1.4 versicolor
## 77	6.8	2.8	4.8	1.4 versicolor
## 78	6.7	3.0	5.0	1.7 versicolor
## 79	6.0	2.9	4.5	1.5 versicolor
## 80	5.7	2.6	3.5	1.0 versicolor
## 81	5.5	2.4	3.8	1.1 versicolor
## 82	5.5	2.4	3.7	1.0 versicolor
## 83	5.8	2.7	3.9	1.2 versicolor
## 84	6.0	2.7	5.1	1.6 versicolor
## 85	5.4	3.0	4.5	1.5 versicolor
## 86	6.0	3.4	4.5	1.6 versicolor
## 87	6.7	3.1	4.7	1.5 versicolor
## 88	6.3	2.3	4.4	1.3 versicolor
## 89	5.6	3.0	4.1	1.3 versicolor
## 90	5.5	2.5	4.0	1.3 versicolor
## 91	5.5	2.6	4.4	1.2 versicolor
## 92	6.1	3.0	4.6	1.4 versicolor
## 93	5.8	2.6	4.0	1.2 versicolor
## 94	5.0	2.3	3.3	1.0 versicolor
## 95	5.6	2.7	4.2	1.3 versicolor
## 96	5.7	3.0	4.2	1.2 versicolor
## 97	5.7	2.9	4.2	1.3 versicolor
## 98	6.2	2.9	4.3	1.3 versicolor
## 99	5.1	2.5	3.0	1.1 versicolor
## 100	5.7	2.8	4.1	1.3 versicolor
## 101	6.3	3.3	6.0	2.5 virginica
## 102	5.8	2.7	5.1	1.9 virginica
## 103	7.1	3.0	5.9	2.1 virginica
## 104	6.3	2.9	5.6	1.8 virginica
## 105	6.5	3.0	5.8	2.2 virginica
## 106	7.6	3.0	6.6	2.1 virginica
## 107	4.9	2.5	4.5	1.7 virginica

## 108	7.3	2.9	6.3	1.8	virginica
## 109	6.7	2.5	5.8	1.8	virginica
## 110	7.2	3.6	6.1	2.5	virginica
## 111	6.5	3.2	5.1	2.0	virginica
## 112	6.4	2.7	5.3	1.9	virginica
## 113	6.8	3.0	5.5	2.1	virginica
## 114	5.7	2.5	5.0	2.0	virginica
## 115	5.8	2.8	5.1	2.4	virginica
## 116	6.4	3.2	5.3	2.3	virginica
## 117	6.5	3.0	5.5	1.8	virginica
## 118	7.7	3.8	6.7	2.2	virginica
## 119	7.7	2.6	6.9	2.3	virginica
## 120	6.0	2.2	5.0	1.5	virginica
## 121	6.9	3.2	5.7	2.3	virginica
## 122	5.6	2.8	4.9	2.0	virginica
## 123	7.7	2.8	6.7	2.0	virginica
## 124	6.3	2.7	4.9	1.8	virginica
## 125	6.7	3.3	5.7	2.1	virginica
## 126	7.2	3.2	6.0	1.8	virginica
## 127	6.2	2.8	4.8	1.8	virginica
## 128	6.1	3.0	4.9	1.8	virginica
## 129	6.4	2.8	5.6	2.1	virginica
## 130	7.2	3.0	5.8	1.6	virginica
## 131	7.4	2.8	6.1	1.9	virginica
## 132	7.9	3.8	6.4	2.0	virginica
## 133	6.4	2.8	5.6	2.2	virginica
## 134	6.3	2.8	5.1	1.5	virginica
## 135	6.1	2.6	5.6	1.4	virginica
## 136	7.7	3.0	6.1	2.3	virginica
## 137	6.3	3.4	5.6	2.4	virginica
## 138	6.4	3.1	5.5	1.8	virginica
## 139	6.0	3.0	4.8	1.8	virginica
## 140	6.9	3.1	5.4	2.1	virginica
## 141	6.7	3.1	5.6	2.4	virginica
## 142	6.9	3.1	5.1	2.3	virginica
## 143	5.8	2.7	5.1	1.9	virginica
## 144	6.8	3.2	5.9	2.3	virginica
## 145	6.7	3.3	5.7	2.5	virginica
## 146	6.7	3.0	5.2	2.3	virginica
## 147	6.3	2.5	5.0	1.9	virginica
## 148	6.5	3.0	5.2	2.0	virginica
## 149	6.2	3.4	5.4	2.3	virginica
## 150	5.9	3.0	5.1	1.8	virginica

Si queremos usar o visualizar solo una columna, podemos volver esa columna un vector con nombre propio

```
Species <- iris[,"Species"]
Species
```

##	[1]	setosa	setosa	setosa	setosa	setosa	setosa
##	[7]	setosa	setosa	setosa	setosa	setosa	setosa
##	[13]	setosa	setosa	setosa	setosa	setosa	setosa
##	[19]	setosa	setosa	setosa	setosa	setosa	setosa

```
## [25] setosa      setosa      setosa      setosa      setosa      setosa
## [31] setosa      setosa      setosa      setosa      setosa      setosa
## [37] setosa      setosa      setosa      setosa      setosa      setosa
## [43] setosa      setosa      setosa      setosa      setosa      setosa
## [49] setosa      setosa      versicolor  versicolor  versicolor  versicolor
## [55] versicolor  versicolor  versicolor  versicolor  versicolor  versicolor
## [61] versicolor  versicolor  versicolor  versicolor  versicolor  versicolor
## [67] versicolor  versicolor  versicolor  versicolor  versicolor  versicolor
## [73] versicolor  versicolor  versicolor  versicolor  versicolor  versicolor
## [79] versicolor  versicolor  versicolor  versicolor  versicolor  versicolor
## [85] versicolor  versicolor  versicolor  versicolor  versicolor  versicolor
## [91] versicolor  versicolor  versicolor  versicolor  versicolor  versicolor
## [97] versicolor  versicolor  versicolor  versicolor  virginica   virginica
## [103] virginica   virginica   virginica   virginica   virginica   virginica
## [109] virginica   virginica   virginica   virginica   virginica   virginica
## [115] virginica   virginica   virginica   virginica   virginica   virginica
## [121] virginica   virginica   virginica   virginica   virginica   virginica
## [127] virginica   virginica   virginica   virginica   virginica   virginica
## [133] virginica   virginica   virginica   virginica   virginica   virginica
## [139] virginica   virginica   virginica   virginica   virginica   virginica
## [145] virginica   virginica   virginica   virginica   virginica   virginica
## Levels: setosa versicolor virginica
```

Podemos acceder solo a un valor,

```
iris[1,3] #se puede acceder por posicion
```

```
## [1] 1.4
```

```
iris["1","Petal.Length"] #o por nombre de fila y columna
```

```
## [1] 1.4
```

```
tab = read.csv("data/hsb2demo.csv",row.names=1,check.names=FALSE)
sum(tab == 0) # podemos contar cuantos elementos del archivo son cero
```

```
## [1] 91
```

```
sum(tab != 0) #y cuantos son diferentes de cero
```

```
## [1] 1909
```

Para realizar diferentes graficas, y dividir nuestro espacio de visualizacion, podemos usar 'layout()'

```
# layout(matrix, widths=w; heights=h)
# diseño(matriz, ancho=w; alto=h)
ng <- layout(matrix(c(1,3,2,3),2,2, byrow=TRUE), widths=c(5,2),height=c(3,4))
layout.show(ng)
```



## Estadísticas simples

La función 'summary()' nos genera datos estadísticos básicos de un conjunto de datos dado. Como por ejemplo: el mínimo, el máximo, la media, la mediana etc...

```
summary(iris)
```

```
##   Sepal.Length   Sepal.Width   Petal.Length   Petal.Width
##   Min.    :4.300   Min.    :2.000   Min.    :1.000   Min.    :0.100
##   1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
##   Median :5.800   Median :3.000   Median :4.350   Median :1.300
##   Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
##   3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
##   Max.    :7.900   Max.    :4.400   Max.    :6.900   Max.    :2.500
##           Species
##   setosa    :50
##   versicolor:50
##   virginica :50
##
##
##
```

El comando 'apply()' nos permite aplicar una función a través de una matriz, array o data frame.

```
iris_1 <- (iris[,-5])
head(apply(iris_1, 1, mean))
```

```
## [1] 2.550 2.375 2.350 2.350 2.550 2.850
```

Esta funcion tiene importantes aplicaciones en datos del microbioma.

```
apply(iris_1, 1, mean)
```

```
## [1] 2.550 2.375 2.350 2.350 2.550 2.850 2.425 2.525 2.225 2.400 2.700 2.500
## [13] 2.325 2.125 2.800 3.000 2.750 2.575 2.875 2.675 2.675 2.675 2.350 2.650
## [25] 2.575 2.450 2.600 2.600 2.550 2.425 2.425 2.675 2.725 2.825 2.425 2.400
## [37] 2.625 2.500 2.225 2.550 2.525 2.100 2.275 2.675 2.800 2.375 2.675 2.350
## [49] 2.675 2.475 4.075 3.900 4.100 3.275 3.850 3.575 3.975 2.900 3.850 3.300
## [61] 2.875 3.650 3.300 3.775 3.350 3.900 3.650 3.400 3.600 3.275 3.925 3.550
## [73] 3.800 3.700 3.725 3.850 3.950 4.100 3.725 3.200 3.200 3.150 3.400 3.850
## [85] 3.600 3.875 4.000 3.575 3.500 3.325 3.425 3.775 3.400 2.900 3.450 3.525
## [97] 3.525 3.675 2.925 3.475 4.525 3.875 4.525 4.150 4.375 4.825 3.400 4.575
## [109] 4.200 4.850 4.200 4.075 4.350 3.800 4.025 4.300 4.200 5.100 4.875 3.675
## [121] 4.525 3.825 4.800 3.925 4.450 4.550 3.900 3.950 4.225 4.400 4.550 5.025
## [133] 4.250 3.925 3.925 4.775 4.425 4.200 3.900 4.375 4.450 4.350 3.875 4.550
## [145] 4.550 4.300 3.925 4.175 4.325 3.950
```

```
apply(iris_1, 2, mean, na.rm = TRUE)
```

```
## Sepal.Length Sepal.Width Petal.Length Petal.Width
## 5.843333 3.057333 3.758000 1.199333
```

```
#apply(DataFrame, dimension = Son identificadas con números, 1 son renglones y 2 son columnas, funcion
tab_perc <- apply(tab, 2, function(x){x/sum(x)})
tab_perc
```

```
## female race ses schtyp prog read
## 70 0.00000000 0.005830904 0.00243309 0.004310345 0.002469136 0.005456634
## 121 0.009174312 0.005830904 0.00486618 0.004310345 0.007407407 0.006509669
## 86 0.000000000 0.005830904 0.00729927 0.004310345 0.002469136 0.004212139
## 141 0.000000000 0.005830904 0.00729927 0.004310345 0.007407407 0.006031017
## 172 0.000000000 0.005830904 0.00486618 0.004310345 0.004938272 0.004499330
## 113 0.000000000 0.005830904 0.00486618 0.004310345 0.004938272 0.004212139
## 50 0.000000000 0.004373178 0.00486618 0.004310345 0.002469136 0.004786521
## 11 0.000000000 0.001457726 0.00486618 0.004310345 0.004938272 0.003254834
## 84 0.000000000 0.005830904 0.00486618 0.004310345 0.002469136 0.006031017
## 48 0.000000000 0.004373178 0.00486618 0.004310345 0.004938272 0.005456634
## 75 0.000000000 0.005830904 0.00486618 0.004310345 0.007407407 0.005743825
## 60 0.000000000 0.005830904 0.00486618 0.004310345 0.004938272 0.005456634
## 95 0.000000000 0.005830904 0.00729927 0.004310345 0.004938272 0.006988321
## 104 0.000000000 0.005830904 0.00729927 0.004310345 0.004938272 0.005169443
## 38 0.000000000 0.004373178 0.00243309 0.004310345 0.004938272 0.004307869
## 115 0.000000000 0.005830904 0.00243309 0.004310345 0.002469136 0.004020678
## 76 0.000000000 0.005830904 0.00729927 0.004310345 0.004938272 0.004499330
```

```

## 195 0.000000000 0.005830904 0.00486618 0.008620690 0.002469136 0.005456634
## 114 0.000000000 0.005830904 0.00729927 0.004310345 0.004938272 0.006509669
## 85 0.000000000 0.005830904 0.00486618 0.004310345 0.002469136 0.005265173
## 167 0.000000000 0.005830904 0.00486618 0.004310345 0.002469136 0.006031017
## 143 0.000000000 0.005830904 0.00486618 0.004310345 0.007407407 0.006031017
## 41 0.000000000 0.004373178 0.00486618 0.004310345 0.004938272 0.004786521
## 20 0.000000000 0.001457726 0.00729927 0.004310345 0.004938272 0.005743825
## 12 0.000000000 0.001457726 0.00486618 0.004310345 0.007407407 0.003542026
## 53 0.000000000 0.004373178 0.00486618 0.004310345 0.007407407 0.003254834
## 154 0.000000000 0.005830904 0.00729927 0.004310345 0.004938272 0.006222478
## 178 0.000000000 0.005830904 0.00486618 0.008620690 0.007407407 0.004499330
## 196 0.000000000 0.005830904 0.00729927 0.008620690 0.004938272 0.004212139
## 29 0.000000000 0.002915452 0.00243309 0.004310345 0.002469136 0.004977982
## 126 0.000000000 0.005830904 0.00486618 0.004310345 0.002469136 0.004020678
## 103 0.000000000 0.005830904 0.00729927 0.004310345 0.004938272 0.007275512
## 192 0.000000000 0.005830904 0.00729927 0.008620690 0.004938272 0.006222478
## 150 0.000000000 0.005830904 0.00486618 0.004310345 0.007407407 0.004020678
## 199 0.000000000 0.005830904 0.00729927 0.008620690 0.004938272 0.004977982
## 144 0.000000000 0.005830904 0.00729927 0.004310345 0.002469136 0.005743825
## 200 0.000000000 0.005830904 0.00486618 0.008620690 0.004938272 0.006509669
## 80 0.000000000 0.005830904 0.00729927 0.004310345 0.004938272 0.006222478
## 16 0.000000000 0.001457726 0.00243309 0.004310345 0.007407407 0.004499330
## 153 0.000000000 0.005830904 0.00486618 0.004310345 0.007407407 0.003733487
## 176 0.000000000 0.005830904 0.00486618 0.008620690 0.004938272 0.004499330
## 177 0.000000000 0.005830904 0.00486618 0.008620690 0.004938272 0.005265173
## 168 0.000000000 0.005830904 0.00486618 0.004310345 0.004938272 0.004977982
## 40 0.000000000 0.004373178 0.00243309 0.004310345 0.002469136 0.004020678
## 62 0.000000000 0.005830904 0.00729927 0.004310345 0.002469136 0.006222478
## 169 0.000000000 0.005830904 0.00243309 0.004310345 0.002469136 0.005265173
## 49 0.000000000 0.004373178 0.00729927 0.004310345 0.007407407 0.004786521
## 136 0.000000000 0.005830904 0.00486618 0.004310345 0.004938272 0.006222478
## 189 0.000000000 0.005830904 0.00486618 0.008620690 0.004938272 0.004499330
## 7 0.000000000 0.001457726 0.00486618 0.004310345 0.004938272 0.005456634
## 27 0.000000000 0.002915452 0.00486618 0.004310345 0.004938272 0.005073712
## 128 0.000000000 0.005830904 0.00729927 0.004310345 0.004938272 0.003733487
## 21 0.000000000 0.001457726 0.00486618 0.004310345 0.002469136 0.004212139
## 183 0.000000000 0.005830904 0.00486618 0.008620690 0.004938272 0.006031017
## 132 0.000000000 0.005830904 0.00486618 0.004310345 0.004938272 0.006988321
## 15 0.000000000 0.001457726 0.00729927 0.004310345 0.007407407 0.003733487
## 67 0.000000000 0.005830904 0.00243309 0.004310345 0.007407407 0.003542026
## 22 0.000000000 0.001457726 0.00486618 0.004310345 0.007407407 0.004020678
## 185 0.000000000 0.005830904 0.00486618 0.008620690 0.004938272 0.006031017
## 9 0.000000000 0.001457726 0.00486618 0.004310345 0.007407407 0.004595060
## 181 0.000000000 0.005830904 0.00486618 0.008620690 0.004938272 0.004786521
## 170 0.000000000 0.005830904 0.00729927 0.004310345 0.004938272 0.004499330
## 134 0.000000000 0.005830904 0.00243309 0.004310345 0.002469136 0.004212139
## 108 0.000000000 0.005830904 0.00486618 0.004310345 0.002469136 0.003254834
## 197 0.000000000 0.005830904 0.00729927 0.008620690 0.004938272 0.004786521
## 140 0.000000000 0.005830904 0.00486618 0.004310345 0.007407407 0.004212139
## 171 0.000000000 0.005830904 0.00486618 0.004310345 0.004938272 0.005743825
## 107 0.000000000 0.005830904 0.00243309 0.004310345 0.007407407 0.004499330
## 81 0.000000000 0.005830904 0.00243309 0.004310345 0.004938272 0.006031017
## 18 0.000000000 0.001457726 0.00486618 0.004310345 0.007407407 0.004786521
## 155 0.000000000 0.005830904 0.00486618 0.004310345 0.002469136 0.004212139

```



## 97	0.000000000	0.005830904	0.00729927	0.004310345	0.004938272	0.005743825
## 68	0.000000000	0.005830904	0.00486618	0.004310345	0.004938272	0.006988321
## 157	0.000000000	0.005830904	0.00486618	0.004310345	0.002469136	0.006509669
## 56	0.000000000	0.005830904	0.00486618	0.004310345	0.007407407	0.005265173
## 5	0.000000000	0.001457726	0.00243309	0.004310345	0.004938272	0.004499330
## 159	0.000000000	0.005830904	0.00729927	0.004310345	0.004938272	0.005265173
## 123	0.000000000	0.005830904	0.00729927	0.004310345	0.002469136	0.006509669
## 164	0.000000000	0.005830904	0.00486618	0.004310345	0.007407407	0.002967643
## 14	0.000000000	0.001457726	0.00729927	0.004310345	0.004938272	0.004499330
## 127	0.000000000	0.005830904	0.00729927	0.004310345	0.004938272	0.006031017
## 165	0.000000000	0.005830904	0.00243309	0.004310345	0.007407407	0.003446295
## 174	0.000000000	0.005830904	0.00486618	0.008620690	0.004938272	0.006509669
## 3	0.000000000	0.001457726	0.00243309	0.004310345	0.004938272	0.006031017
## 58	0.000000000	0.005830904	0.00486618	0.004310345	0.007407407	0.005265173
## 146	0.000000000	0.005830904	0.00729927	0.004310345	0.004938272	0.005265173
## 102	0.000000000	0.005830904	0.00729927	0.004310345	0.004938272	0.004977982
## 117	0.000000000	0.005830904	0.00729927	0.004310345	0.007407407	0.003254834
## 133	0.000000000	0.005830904	0.00486618	0.004310345	0.007407407	0.004786521
## 94	0.000000000	0.005830904	0.00729927	0.004310345	0.004938272	0.005265173
## 24	0.000000000	0.002915452	0.00486618	0.004310345	0.004938272	0.004977982
## 149	0.000000000	0.005830904	0.00243309	0.004310345	0.002469136	0.006031017
## 82	0.009174312	0.005830904	0.00729927	0.004310345	0.004938272	0.006509669
## 8	0.009174312	0.001457726	0.00243309	0.004310345	0.004938272	0.003733487
## 129	0.009174312	0.005830904	0.00243309	0.004310345	0.002469136	0.004212139
## 173	0.009174312	0.005830904	0.00243309	0.004310345	0.002469136	0.004786521
## 57	0.009174312	0.005830904	0.00486618	0.004310345	0.004938272	0.006796860
## 100	0.009174312	0.005830904	0.00729927	0.004310345	0.004938272	0.006031017
## 1	0.009174312	0.001457726	0.00243309	0.004310345	0.007407407	0.003254834
## 194	0.009174312	0.005830904	0.00729927	0.008620690	0.004938272	0.006031017
## 88	0.009174312	0.005830904	0.00729927	0.004310345	0.004938272	0.006509669
## 99	0.009174312	0.005830904	0.00729927	0.004310345	0.002469136	0.004499330
## 47	0.009174312	0.004373178	0.00243309	0.004310345	0.004938272	0.004499330
## 120	0.009174312	0.005830904	0.00729927	0.004310345	0.004938272	0.006031017
## 166	0.009174312	0.005830904	0.00486618	0.004310345	0.004938272	0.004977982
## 65	0.009174312	0.005830904	0.00486618	0.004310345	0.004938272	0.005265173
## 101	0.009174312	0.005830904	0.00729927	0.004310345	0.004938272	0.005743825
## 89	0.009174312	0.005830904	0.00243309	0.004310345	0.007407407	0.003350565
## 54	0.009174312	0.004373178	0.00243309	0.008620690	0.002469136	0.004499330
## 180	0.009174312	0.005830904	0.00729927	0.008620690	0.004938272	0.006796860
## 162	0.009174312	0.005830904	0.00486618	0.004310345	0.007407407	0.005456634
## 4	0.009174312	0.001457726	0.00243309	0.004310345	0.004938272	0.004212139
## 131	0.009174312	0.005830904	0.00729927	0.004310345	0.004938272	0.006222478
## 125	0.009174312	0.005830904	0.00243309	0.004310345	0.004938272	0.006509669
## 34	0.009174312	0.001457726	0.00729927	0.008620690	0.004938272	0.006988321
## 106	0.009174312	0.005830904	0.00486618	0.004310345	0.007407407	0.003446295
## 130	0.009174312	0.005830904	0.00729927	0.004310345	0.002469136	0.004116408
## 93	0.009174312	0.005830904	0.00729927	0.004310345	0.004938272	0.006988321
## 163	0.009174312	0.005830904	0.00243309	0.004310345	0.004938272	0.004977982
## 37	0.009174312	0.004373178	0.00243309	0.004310345	0.007407407	0.003924947
## 35	0.009174312	0.001457726	0.00243309	0.008620690	0.002469136	0.005743825
## 87	0.009174312	0.005830904	0.00486618	0.004310345	0.002469136	0.004786521
## 73	0.009174312	0.005830904	0.00486618	0.004310345	0.004938272	0.004786521
## 151	0.009174312	0.005830904	0.00486618	0.004310345	0.007407407	0.004499330
## 44	0.009174312	0.004373178	0.00243309	0.004310345	0.007407407	0.004499330

## 152 0.009174312 0.005830904 0.00729927 0.004310345 0.004938272 0.005265173  
 ## 105 0.009174312 0.005830904 0.00486618 0.004310345 0.004938272 0.004786521  
 ## 28 0.009174312 0.002915452 0.00486618 0.004310345 0.002469136 0.003733487  
 ## 91 0.009174312 0.005830904 0.00729927 0.004310345 0.007407407 0.004786521  
 ## 45 0.009174312 0.004373178 0.00243309 0.004310345 0.007407407 0.003254834  
 ## 116 0.009174312 0.005830904 0.00486618 0.004310345 0.004938272 0.005456634  
 ## 33 0.009174312 0.002915452 0.00243309 0.004310345 0.004938272 0.005456634  
 ## 66 0.009174312 0.005830904 0.00486618 0.004310345 0.007407407 0.006509669  
 ## 72 0.009174312 0.005830904 0.00486618 0.004310345 0.007407407 0.004020678  
 ## 77 0.009174312 0.005830904 0.00243309 0.004310345 0.004938272 0.005839556  
 ## 61 0.009174312 0.005830904 0.00729927 0.004310345 0.004938272 0.007275512  
 ## 190 0.009174312 0.005830904 0.00486618 0.008620690 0.004938272 0.004499330  
 ## 42 0.009174312 0.004373178 0.00486618 0.004310345 0.007407407 0.004403599  
 ## 2 0.009174312 0.001457726 0.00486618 0.004310345 0.007407407 0.003733487  
 ## 55 0.009174312 0.004373178 0.00486618 0.008620690 0.004938272 0.004977982  
 ## 19 0.009174312 0.001457726 0.00243309 0.004310345 0.002469136 0.002680452  
 ## 90 0.009174312 0.005830904 0.00729927 0.004310345 0.004938272 0.004020678  
 ## 142 0.009174312 0.005830904 0.00486618 0.004310345 0.007407407 0.004499330  
 ## 17 0.009174312 0.001457726 0.00486618 0.004310345 0.004938272 0.004499330  
 ## 122 0.009174312 0.005830904 0.00486618 0.004310345 0.004938272 0.004977982  
 ## 191 0.009174312 0.005830904 0.00729927 0.008620690 0.004938272 0.004499330  
 ## 83 0.009174312 0.005830904 0.00486618 0.004310345 0.007407407 0.004786521  
 ## 182 0.009174312 0.005830904 0.00486618 0.008620690 0.004938272 0.004212139  
 ## 6 0.009174312 0.001457726 0.00243309 0.004310345 0.004938272 0.004499330  
 ## 46 0.009174312 0.004373178 0.00243309 0.004310345 0.004938272 0.004307869  
 ## 43 0.009174312 0.004373178 0.00243309 0.004310345 0.004938272 0.004499330  
 ## 96 0.009174312 0.005830904 0.00729927 0.004310345 0.004938272 0.006222478  
 ## 138 0.009174312 0.005830904 0.00486618 0.004310345 0.007407407 0.004116408  
 ## 10 0.009174312 0.001457726 0.00486618 0.004310345 0.002469136 0.004499330  
 ## 71 0.009174312 0.005830904 0.00486618 0.004310345 0.002469136 0.005456634  
 ## 139 0.009174312 0.005830904 0.00486618 0.004310345 0.004938272 0.006509669  
 ## 110 0.009174312 0.005830904 0.00486618 0.004310345 0.007407407 0.004977982  
 ## 148 0.009174312 0.005830904 0.00486618 0.004310345 0.007407407 0.004020678  
 ## 109 0.009174312 0.005830904 0.00486618 0.004310345 0.002469136 0.004020678  
 ## 39 0.009174312 0.004373178 0.00729927 0.004310345 0.004938272 0.006318208  
 ## 147 0.009174312 0.005830904 0.00243309 0.004310345 0.004938272 0.004499330  
 ## 74 0.009174312 0.005830904 0.00486618 0.004310345 0.004938272 0.005456634  
 ## 198 0.009174312 0.005830904 0.00729927 0.008620690 0.004938272 0.004499330  
 ## 161 0.009174312 0.005830904 0.00243309 0.004310345 0.004938272 0.005456634  
 ## 112 0.009174312 0.005830904 0.00486618 0.004310345 0.004938272 0.004977982  
 ## 69 0.009174312 0.005830904 0.00243309 0.004310345 0.007407407 0.004212139  
 ## 156 0.009174312 0.005830904 0.00486618 0.004310345 0.004938272 0.004786521  
 ## 111 0.009174312 0.005830904 0.00243309 0.004310345 0.002469136 0.003733487  
 ## 186 0.009174312 0.005830904 0.00486618 0.008620690 0.004938272 0.005456634  
 ## 98 0.009174312 0.005830904 0.00243309 0.004310345 0.007407407 0.005456634  
 ## 119 0.009174312 0.005830904 0.00243309 0.004310345 0.002469136 0.004020678  
 ## 13 0.009174312 0.001457726 0.00486618 0.004310345 0.007407407 0.004499330  
 ## 51 0.009174312 0.004373178 0.00729927 0.004310345 0.002469136 0.004020678  
 ## 26 0.009174312 0.002915452 0.00729927 0.004310345 0.004938272 0.005743825  
 ## 36 0.009174312 0.004373178 0.00243309 0.004310345 0.002469136 0.004212139  
 ## 135 0.009174312 0.005830904 0.00243309 0.004310345 0.004938272 0.006031017  
 ## 59 0.009174312 0.005830904 0.00486618 0.004310345 0.004938272 0.006222478  
 ## 78 0.009174312 0.005830904 0.00486618 0.004310345 0.004938272 0.003733487  
 ## 64 0.009174312 0.005830904 0.00729927 0.004310345 0.007407407 0.004786521

```

## 63 0.009174312 0.005830904 0.00243309 0.004310345 0.002469136 0.004977982
## 79 0.009174312 0.005830904 0.00486618 0.004310345 0.004938272 0.005743825
## 193 0.009174312 0.005830904 0.00486618 0.008620690 0.004938272 0.004212139
## 92 0.009174312 0.005830904 0.00729927 0.004310345 0.002469136 0.004977982
## 160 0.009174312 0.005830904 0.00486618 0.004310345 0.004938272 0.005265173
## 32 0.009174312 0.002915452 0.00729927 0.004310345 0.007407407 0.004786521
## 23 0.009174312 0.002915452 0.00243309 0.004310345 0.004938272 0.006222478
## 158 0.009174312 0.005830904 0.00486618 0.004310345 0.002469136 0.004977982
## 25 0.009174312 0.002915452 0.00486618 0.004310345 0.002469136 0.004499330
## 188 0.009174312 0.005830904 0.00729927 0.008620690 0.004938272 0.006031017
## 52 0.009174312 0.004373178 0.00243309 0.004310345 0.004938272 0.004786521
## 124 0.009174312 0.005830904 0.00243309 0.004310345 0.007407407 0.004020678
## 175 0.009174312 0.005830904 0.00729927 0.008620690 0.002469136 0.003446295
## 184 0.009174312 0.005830904 0.00486618 0.008620690 0.007407407 0.004786521
## 30 0.009174312 0.002915452 0.00729927 0.004310345 0.004938272 0.003924947
## 179 0.009174312 0.005830904 0.00486618 0.008620690 0.004938272 0.004499330
## 31 0.009174312 0.002915452 0.00486618 0.008620690 0.002469136 0.005265173
## 145 0.009174312 0.005830904 0.00486618 0.004310345 0.007407407 0.004020678
## 187 0.009174312 0.005830904 0.00486618 0.008620690 0.002469136 0.005456634
## 118 0.009174312 0.005830904 0.00486618 0.004310345 0.002469136 0.005265173
## 137 0.009174312 0.005830904 0.00729927 0.004310345 0.004938272 0.006031017
##      write      math      science      socst
## 70 0.004926575 0.003894007 0.004532305 0.005438412
## 121 0.005589768 0.005033716 0.006075217 0.005820055
## 86 0.003126480 0.005128692 0.005593057 0.002957733
## 141 0.004168640 0.004463862 0.005110897 0.005343002
## 172 0.004926575 0.005413620 0.005110897 0.005820055
## 113 0.004926575 0.004843765 0.006075217 0.005820055
## 50 0.005589768 0.003988983 0.005110897 0.005820055
## 11 0.004358124 0.004273910 0.003760849 0.003434787
## 84 0.005400284 0.005128692 0.005593057 0.004865948
## 48 0.005210801 0.004938741 0.004821601 0.004865948
## 75 0.004358124 0.004843765 0.005110897 0.005820055
## 60 0.006158219 0.004843765 0.006075217 0.005820055
## 95 0.005684510 0.006743280 0.005882353 0.006774163
## 104 0.005968735 0.005413620 0.005303761 0.004388894
## 38 0.005400284 0.004748789 0.002989392 0.005343002
## 115 0.004642350 0.004083959 0.004821601 0.005343002
## 76 0.004926575 0.004843765 0.004821601 0.005343002
## 195 0.005400284 0.005698547 0.005593057 0.005343002
## 114 0.006158219 0.005888498 0.005303761 0.005820055
## 85 0.003694931 0.005413620 0.005110897 0.004388894
## 167 0.004642350 0.003324152 0.006364513 0.003911840
## 143 0.005968735 0.007123184 0.006943105 0.006297109
## 41 0.003789673 0.004273910 0.005303761 0.005343002
## 20 0.004926575 0.005413620 0.005882353 0.005820055
## 12 0.004168640 0.004273910 0.003760849 0.004388894
## 53 0.003505448 0.004368886 0.003760849 0.002957733
## 154 0.006158219 0.006268402 0.005882353 0.006297109
## 178 0.005400284 0.005413620 0.005593057 0.004388894
## 196 0.003600189 0.004653813 0.003760849 0.004388894
## 29 0.004168640 0.004653813 0.005303761 0.003911840
## 126 0.002936997 0.005413620 0.004532305 0.004865948
## 103 0.004926575 0.006078450 0.006171649 0.005820055

```

## 192 0.006347703 0.005983474 0.006364513 0.006774163  
 ## 150 0.003884415 0.005413620 0.006943105 0.002957733  
 ## 199 0.005589768 0.004748789 0.005882353 0.005820055  
 ## 144 0.006158219 0.005508595 0.005882353 0.006297109  
 ## 200 0.005116059 0.007123184 0.006364513 0.006297109  
 ## 80 0.005873993 0.006458353 0.006364513 0.006297109  
 ## 16 0.002936997 0.004178934 0.003471553 0.003434787  
 ## 153 0.002936997 0.003799031 0.003760849 0.004865948  
 ## 176 0.004452866 0.003894007 0.004050145 0.004865948  
 ## 177 0.005589768 0.005888498 0.005593057 0.004865948  
 ## 168 0.005116059 0.005413620 0.005303761 0.004865948  
 ## 40 0.003884415 0.004083959 0.004821601 0.003911840  
 ## 62 0.006158219 0.004558837 0.006075217 0.006297109  
 ## 169 0.005589768 0.005983474 0.006653809 0.004388894  
 ## 49 0.003789673 0.003704055 0.004725169 0.004484305  
 ## 136 0.005589768 0.006648305 0.006075217 0.004865948  
 ## 189 0.005589768 0.005983474 0.005110897 0.004388894  
 ## 7 0.005116059 0.005603571 0.004532305 0.004865948  
 ## 27 0.005779252 0.005793523 0.005496625 0.005343002  
 ## 128 0.003126480 0.003609080 0.004532305 0.003911840  
 ## 21 0.004168640 0.005793523 0.004821601 0.004388894  
 ## 183 0.005589768 0.004653813 0.005303761 0.006774163  
 ## 132 0.005873993 0.006933232 0.006653809 0.006297109  
 ## 15 0.003694931 0.004178934 0.002507232 0.004007251  
 ## 67 0.003505448 0.003988983 0.003182257 0.003053144  
 ## 22 0.003694931 0.003704055 0.005400193 0.004388894  
 ## 185 0.005400284 0.005223668 0.005593057 0.003911840  
 ## 9 0.004642350 0.004938741 0.004243009 0.004865948  
 ## 181 0.004358124 0.004273910 0.005593057 0.005820055  
 ## 170 0.005873993 0.005793523 0.006653809 0.006297109  
 ## 134 0.004168640 0.003704055 0.003278689 0.004388894  
 ## 108 0.003126480 0.003894007 0.003471553 0.003434787  
 ## 197 0.003979157 0.004748789 0.003471553 0.005820055  
 ## 140 0.003884415 0.003799031 0.004821601 0.002480679  
 ## 171 0.005116059 0.005698547 0.005303761 0.006297109  
 ## 107 0.003694931 0.004463862 0.004050145 0.002480679  
 ## 81 0.004073899 0.005603571 0.006268081 0.004198073  
 ## 18 0.003126480 0.004653813 0.004243009 0.003434787  
 ## 155 0.004168640 0.004368886 0.003760849 0.004865948  
 ## 97 0.005116059 0.005508595 0.005593057 0.005820055  
 ## 68 0.006347703 0.006743280 0.006075217 0.006297109  
 ## 157 0.005589768 0.005508595 0.007135969 0.006297109  
 ## 56 0.004263382 0.004368886 0.005593057 0.004865948  
 ## 5 0.003789673 0.004083959 0.004339441 0.002957733  
 ## 159 0.005779252 0.005128692 0.004725169 0.005820055  
 ## 123 0.005589768 0.005318644 0.006075217 0.006297109  
 ## 164 0.003410706 0.004368886 0.003760849 0.004388894  
 ## 14 0.003884415 0.005128692 0.004050145 0.005343002  
 ## 127 0.005589768 0.005413620 0.005303761 0.005343002  
 ## 165 0.004642350 0.005128692 0.005882353 0.003434787  
 ## 174 0.005589768 0.006743280 0.006364513 0.005343002  
 ## 3 0.006158219 0.004558837 0.006075217 0.005343002  
 ## 58 0.003884415 0.003799031 0.004243009 0.003911840  
 ## 146 0.005873993 0.006078450 0.006075217 0.006297109

```

## 102 0.003884415 0.004843765 0.005110897 0.005343002
## 117 0.004642350 0.003704055 0.004050145 0.005343002
## 133 0.002936997 0.003799031 0.003278689 0.002957733
## 94 0.004642350 0.005793523 0.005882353 0.005343002
## 24 0.005873993 0.006268402 0.004532305 0.004388894
## 149 0.004642350 0.004653813 0.006364513 0.004388894
## 82 0.005873993 0.006173426 0.006653809 0.005820055
## 8 0.004168640 0.004938741 0.004243009 0.004579716
## 129 0.004168640 0.004368886 0.004532305 0.004865948
## 173 0.005873993 0.005793523 0.006075217 0.004865948
## 57 0.006158219 0.006838256 0.006364513 0.005343002
## 100 0.006158219 0.006743280 0.006653809 0.006774163
## 1 0.004168640 0.003799031 0.003760849 0.003911840
## 194 0.005968735 0.006553329 0.005882353 0.005820055
## 88 0.005684510 0.006078450 0.006653809 0.006297109
## 99 0.005589768 0.005318644 0.006364513 0.005820055
## 47 0.004358124 0.004653813 0.003182257 0.003911840
## 120 0.004926575 0.005128692 0.004821601 0.004865948
## 166 0.005589768 0.005033716 0.005882353 0.004865948
## 65 0.005116059 0.006268402 0.004050145 0.005343002
## 101 0.005873993 0.006363377 0.004821601 0.005343002
## 89 0.003315964 0.003799031 0.004918033 0.003148555
## 54 0.005116059 0.004368886 0.004821601 0.005343002
## 180 0.006158219 0.006553329 0.005593057 0.006774163
## 162 0.004926575 0.003799031 0.005882353 0.005343002
## 4 0.004737091 0.003894007 0.003760849 0.004865948
## 131 0.005589768 0.005413620 0.004435873 0.006297109
## 125 0.006158219 0.005508595 0.005689489 0.005343002
## 34 0.005779252 0.005413620 0.005303761 0.006297109
## 106 0.004168640 0.003514104 0.004050145 0.003911840
## 130 0.005116059 0.005223668 0.005303761 0.004388894
## 93 0.006347703 0.005888498 0.005593057 0.006297109
## 163 0.005400284 0.006078450 0.005593057 0.005343002
## 37 0.004452866 0.003799031 0.003760849 0.004865948
## 35 0.005116059 0.004748789 0.004821601 0.004865948
## 87 0.004926575 0.004368886 0.004821601 0.005343002
## 73 0.004926575 0.005033716 0.003760849 0.005343002
## 151 0.004358124 0.004938741 0.004628737 0.004388894
## 44 0.005873993 0.004273910 0.003278689 0.004388894
## 152 0.005400284 0.005318644 0.005593057 0.005820055
## 105 0.003884415 0.004273910 0.004243009 0.005343002
## 28 0.005021317 0.005128692 0.004821601 0.003911840
## 91 0.004642350 0.005318644 0.004532305 0.004388894
## 45 0.003315964 0.003894007 0.002796528 0.002480679
## 116 0.005589768 0.005128692 0.004821601 0.005343002
## 33 0.006158219 0.006838256 0.005207329 0.005343002
## 66 0.005873993 0.005318644 0.004821601 0.004865948
## 72 0.005116059 0.004463862 0.004532305 0.004388894
## 77 0.005589768 0.004653813 0.004243009 0.006297109
## 61 0.005968735 0.005698547 0.006460945 0.006297109
## 190 0.005589768 0.005128692 0.005593057 0.004388894
## 42 0.004926575 0.005223668 0.004243009 0.005343002
## 2 0.003884415 0.003134201 0.004050145 0.003911840
## 55 0.004642350 0.004653813 0.004243009 0.005820055

```

## 19 0.004358124 0.004083959 0.004243009 0.004865948  
 ## 90 0.005116059 0.004748789 0.004821601 0.004961359  
 ## 142 0.003979157 0.004938741 0.003760849 0.004865948  
 ## 17 0.005400284 0.004558837 0.004243009 0.003911840  
 ## 122 0.005589768 0.005508595 0.005110897 0.006297109  
 ## 191 0.004926575 0.004083959 0.004628737 0.005820055  
 ## 83 0.005873993 0.003894007 0.005303761 0.002957733  
 ## 182 0.004926575 0.004083959 0.004243009 0.004865948  
 ## 6 0.003884415 0.004368886 0.003857281 0.003911840  
 ## 46 0.005210801 0.004178934 0.003278689 0.003911840  
 ## 43 0.003505448 0.004083959 0.004050145 0.004388894  
 ## 96 0.005116059 0.005793523 0.005593057 0.005343002  
 ## 138 0.005400284 0.003799031 0.004821601 0.004865948  
 ## 10 0.005116059 0.004653813 0.005110897 0.005820055  
 ## 71 0.005873993 0.005318644 0.005593057 0.006297109  
 ## 139 0.005589768 0.005793523 0.005303761 0.006774163  
 ## 110 0.005210801 0.004748789 0.005207329 0.005820055  
 ## 148 0.005400284 0.004843765 0.004532305 0.005820055  
 ## 109 0.003694931 0.003988983 0.004050145 0.003911840  
 ## 39 0.006347703 0.006363377 0.005882353 0.006297109  
 ## 147 0.005873993 0.005033716 0.005110897 0.005820055  
 ## 74 0.004737091 0.004748789 0.004918033 0.005533823  
 ## 198 0.005779252 0.004843765 0.006075217 0.002957733  
 ## 161 0.005873993 0.006838256 0.005882353 0.005820055  
 ## 112 0.005589768 0.004558837 0.005303761 0.005820055  
 ## 69 0.004168640 0.003799031 0.003857281 0.002957733  
 ## 156 0.005589768 0.005033716 0.005882353 0.005820055  
 ## 111 0.005116059 0.003704055 0.004532305 0.003434787  
 ## 186 0.005873993 0.005983474 0.005303761 0.003911840  
 ## 98 0.005684510 0.004843765 0.005110897 0.003530198  
 ## 119 0.005400284 0.004273910 0.004821601 0.004102662  
 ## 13 0.004358124 0.003704055 0.004532305 0.005820055  
 ## 51 0.003410706 0.003988983 0.002989392 0.003721019  
 ## 26 0.005589768 0.005888498 0.005882353 0.004865948  
 ## 36 0.004642350 0.004178934 0.003375121 0.004865948  
 ## 135 0.005684510 0.006173426 0.005207329 0.006297109  
 ## 59 0.006347703 0.005983474 0.005303761 0.006774163  
 ## 78 0.005116059 0.005128692 0.005110897 0.003911840  
 ## 64 0.004926575 0.004273910 0.005593057 0.003434787  
 ## 63 0.006158219 0.005698547 0.005400193 0.004865948  
 ## 79 0.005873993 0.004653813 0.004821601 0.004865948  
 ## 193 0.004642350 0.004558837 0.003760849 0.004865948  
 ## 92 0.006347703 0.005413620 0.006075217 0.005820055  
 ## 160 0.006158219 0.005223668 0.004821601 0.005820055  
 ## 32 0.006347703 0.006268402 0.006364513 0.005343002  
 ## 23 0.006158219 0.006078450 0.005593057 0.006774163  
 ## 158 0.005116059 0.005223668 0.005110897 0.004865948  
 ## 25 0.004168640 0.003988983 0.004050145 0.003434787  
 ## 188 0.005873993 0.005318644 0.005303761 0.005820055  
 ## 52 0.004358124 0.005033716 0.005110897 0.006297109  
 ## 124 0.005116059 0.003894007 0.004050145 0.003911840  
 ## 175 0.005400284 0.003988983 0.004821601 0.003911840  
 ## 184 0.004926575 0.005033716 0.005303761 0.005343002  
 ## 30 0.005589768 0.003988983 0.003278689 0.004865948

```
## 179 0.006158219 0.005698547 0.004821601 0.005343002
## 31 0.005589768 0.004938741 0.004050145 0.005343002
## 145 0.004358124 0.003609080 0.003471553 0.004388894
## 187 0.003884415 0.005413620 0.005303761 0.004961359
## 118 0.005873993 0.005508595 0.005593057 0.005820055
## 137 0.006158219 0.006173426 0.005110897 0.005820055
```

```
tab_perc <- apply(tab[,1:ncol(tab)-1], 2, function(x){x/sum(x)})
```

```
tab_p1 <- tab[apply(tab_perc, 1, max)>0.01,]
```

```
tab_p2 <- tab[apply(tab_perc, 1, min)>0.01,]
head(tab_p2)
```

```
## [1] female race ses schtyp prog read write math science
## [10] socst
## <0 rows> (or 0-length row.names)
```

```
iris_t <- t(iris) #toma la traspuesta del dataframe
iris_t[1:5,1:6]
```

```
##           [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
## Sepal.Length "5.1"    "4.9"    "4.7"    "4.6"    "5.0"    "5.4"
## Sepal.Width  "3.5"    "3.0"    "3.2"    "3.1"    "3.6"    "3.9"
## Petal.Length "1.4"    "1.4"    "1.3"    "1.5"    "1.4"    "1.7"
## Petal.Width  "0.2"    "0.2"    "0.2"    "0.2"    "0.2"    "0.4"
## Species      "setosa" "setosa" "setosa" "setosa" "setosa" "setosa"
```

Aplicamos las dos funciones al dataframe iris y creamos un nuevo dataframe

```
# ordenar el dataframe
iris_2 <- (iris[,-c(3:5)])
sorted <- sort(iris_2$Sepal.Length)
#sort() y order(), ordenan(Sepal.Length) de forma ascendente
ordered <- order(iris_2$Sepal.Length)
new_iris <- data.frame(iris_2,sorted,ordered)
head(new_iris)
```

```
##   Sepal.Length Sepal.Width sorted ordered
## 1          5.1          3.5     4.3      14
## 2          4.9          3.0     4.4       9
## 3          4.7          3.2     4.4      39
## 4          4.6          3.1     4.4      43
## 5          5.0          3.6     4.5      42
## 6          5.4          3.9     4.6       4
```

```
rev_iris <- rev(sort(iris_2$Sepal.Length))
# ordena de forma descendente
head(rev_iris)
```

```
## [1] 7.9 7.7 7.7 7.7 7.7 7.6
```

```
head(iris[order(Sepal.Length),])
```

```
##      Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 14             4.3          3.0           1.1          0.1  setosa
## 9              4.4          2.9           1.4          0.2  setosa
## 39             4.4          3.0           1.3          0.2  setosa
## 43             4.4          3.2           1.3          0.2  setosa
## 42             4.5          2.3           1.3          0.3  setosa
## 4              4.6          3.1           1.5          0.2  setosa
```

```
head(iris[order(iris[, 'Sepal.Length']),])
```

```
##      Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 14             4.3          3.0           1.1          0.1  setosa
## 9              4.4          2.9           1.4          0.2  setosa
## 39             4.4          3.0           1.3          0.2  setosa
## 43             4.4          3.2           1.3          0.2  setosa
## 42             4.5          2.3           1.3          0.3  setosa
## 4              4.6          3.1           1.5          0.2  setosa
```

## Introduccion al paquete dplyr

<https://swcarpentry.github.io/r-novice-gapminder-es/13-dplyr/>

El paquete dplyr proporciona un conjunto de funciones extremadamente útiles para manipular data frames y así reducir el número de repeticiones, la probabilidad de cometer errores y el número de caracteres que hay que escribir.

```
#instalar y cargar el paquete
#install.packages("dplyr")
library("dplyr")
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:gdata':
##
##      combine, first, last

## The following object is masked from 'package:MASS':
##
##      select

## The following objects are masked from 'package:stats':
##
##      filter, lag

## The following objects are masked from 'package:base':
##
##      intersect, setdiff, setequal, union
```



```
tab <- read.csv('data/hsb2demo.csv')
head(tab)
```

```
##      id female race ses schtyp prog read write math science socst
## 1   70      0    4   1     1     1   57   52   41     47    57
## 2  121      1    4   2     1     3   68   59   53     63    61
## 3   86      0    4   3     1     1   44   33   54     58    31
## 4  141      0    4   3     1     3   63   44   47     53    56
## 5  172      0    4   2     1     2   47   52   57     53    61
## 6  113      0    4   2     1     2   44   52   51     63    61
```

Veremos varias de sus funciones, así como a usar los pipes (`%>%`) para combinarlas

`select()` `filter()` `group_by()` `summarize()` `mutate()`

El operador de (`%>%`) es muy útil cuando combinamos varias funciones, en lugar de leer funciones de adentro hacia afuera, las lee de izquierda a derecha.

```
tab %>% select(id,write) %>% head
```

```
##      id write
## 1   70    52
## 2  121    59
## 3   86    33
## 4  141    44
## 5  172    52
## 6  113    52
```

*#esta funcion guardará sólo las variables que seleccionemos.*

Vamos a seleccionar columnas con el comando `select()` del dataframe 'tab', estamos seleccionando las variables id, read, write, math.

```
head(select(tab, id, read, write, math))
```

```
##      id read write math
## 1   70   57    52   41
## 2  121   68    59   53
## 3   86   44    33   54
## 4  141   63    44   47
## 5  172   47    52   57
## 6  113   44    52   51
```

Ahora seleccionamos de la tabla 'tab' usando 'select' desde la columna 'read' a 'socst', seleccionando el rango completo entre estas dos columnas.

```
head(select(tab, read:socst))
```

```
##      read write math science socst
## 1    57    52   41     47    57
## 2    68    59   53     63    61
```

```
## 3  44  33  54    58  31
## 4  63  44  47    53  56
## 5  47  52  57    53  61
## 6  44  52  51    63  61
```

Tambien podemos seleccionar de la tabla ‘tab’ todas las columnas menos la columna ‘female’

```
head(select(tab, - female))
```

```
##      id race ses schtyp prog read write math science socst
## 1   70    4  1      1     1   57   52  41      47    57
## 2  121    4  2      1     3   68   59  53      63    61
## 3   86    4  3      1     1   44   33  54      58    31
## 4  141    4  3      1     3   63   44  47      53    56
## 5  172    4  2      1     2   47   52  57      53    61
## 6  113    4  2      1     2   44   52  51      63    61
```

De forma similar podemos seleccionarlas, desde que columna a que columna no queremos que aparezcan

```
head(select(tab, - (female:prog)))
```

```
##      id read write math science socst
## 1   70   57   52   41      47    57
## 2  121   68   59   53      63    61
## 3   86   44   33   54      58    31
## 4  141   63   44   47      53    56
## 5  172   47   52   57      53    61
## 6  113   44   52   51      63    61
```

También podemos seleccionar columnas basados en escritores específicos usando ‘select()’. Estos criterios pueden ser, starts\_with(), ends\_with(), matches(), contains(), and one\_of ().

Queremos seleccionar todas las columnas que comienzan con el carácter “s”

```
head(select(tab, starts_with("s")))
```

```
##      ses schtyp science socst
## 1     1      1      47    57
## 2     2      1      63    61
## 3     3      1      58    31
## 4     3      1      53    56
## 5     2      1      53    61
## 6     2      1      63    61
```

El comando ‘filter()’ nos permite seleccionar filas de un dataframe.

```
#filtra las fila de estudiantes con puntaje de lectura mayor o igual a 70.
filter(tab, read >= 70)
```

```
##      id female race ses schtyp prog read write math science socst
## 1   95      0    4  3      1     2   73   60  71      61    71
```

```
## 2 103      0    4    3      1    2    76    52    64      64    61
## 3 132      0    4    2      1    2    73    62    73      69    66
## 4   68      0    4    2      1    2    73    67    71      63    66
## 5   57      1    4    2      1    2    71    65    72      66    56
## 6 180      1    4    3      2    2    71    65    69      58    71
## 7   34      1    1    3      2    2    73    61    57      55    66
## 8   93      1    4    3      1    2    73    67    62      58    66
## 9   61      1    4    3      1    2    76    63    60      67    66
```

```
#Filtra las filas de estudiantes con un puntaje de lectura y matematica mayor o igual a 70
filter(tab, read >= 70, math >= 70)
```

```
##      id female race ses schtyp prog read write math science socst
## 1   95      0    4    3      1    2    73    60    71      61    71
## 2  132      0    4    2      1    2    73    62    73      69    66
## 3   68      0    4    2      1    2    73    67    71      63    66
## 4   57      1    4    2      1    2    71    65    72      66    56
```

La funcion 'arrange()' trabaja de manera similar a 'filter()' solo que este en lugar de seleccionar las filas las ordena.

```
#ordena por read y luego por write
head(arrange(tab, read, write))
```

```
##      id female race ses schtyp prog read write math science socst
## 1   19      1    1    1      1    1    28    46    43      44    51
## 2  164      0    4    2      1    3    31    36    46      39    46
## 3  108      0    4    2      1    1    34    33    41      36    36
## 4   45      1    3    1      1    3    34    35    41      29    26
## 5   53      0    3    2      1    3    34    37    46      39    31
## 6    1      1    1    1      1    3    34    44    40      39    41
```

```
#Usamos desc() para ordenar una columna en orden decreciente
head(arrange(tab, desc(read)))
```

```
##      id female race ses schtyp prog read write math science socst
## 1  103      0    4    3      1    2    76    52    64      64    61
## 2   61      1    4    3      1    2    76    63    60      67    66
## 3   95      0    4    3      1    2    73    60    71      61    71
## 4  132      0    4    2      1    2    73    62    73      69    66
## 5   68      0    4    2      1    2    73    67    71      63    66
## 6   34      1    1    3      2    2    73    61    57      55    66
```

```
head(arrange(tab, desc(female),read)) #ejemplo
```

```
##      id female race ses schtyp prog read write math science socst
## 1   19      1    1    1      1    1    28    46    43      44    51
## 2    1      1    1    1      1    3    34    44    40      39    41
## 3   45      1    3    1      1    3    34    35    41      29    26
## 4   89      1    4    1      1    3    35    35    40      51    33
## 5  106      1    4    2      1    3    36    44    37      42    41
## 6  175      1    4    3      2    1    36    57    42      50    41
```

The pipe operator: ‘%>%’ lo podemos usar juto a ‘arrange()’. este operados nos ayuda a ejecutar todas las ordenes concatenadas,en orden.

```
tab %>% arrange(female) %>% head
```

```
##      id female race ses schtyp prog read write math science socst
## 1    70      0   4   1     1     1   57   52   41      47    57
## 2    86      0   4   3     1     1   44   33   54      58    31
## 3   141      0   4   3     1     3   63   44   47      53    56
## 4   172      0   4   2     1     2   47   52   57      53    61
## 5   113      0   4   2     1     2   44   52   51      63    61
## 6    50      0   3   2     1     1   50   59   42      53    61
```

```
#Primero selecciona las columnas id, gender, read de tab, luego ordena las filas por gender y luego por
tab%>%select(id, female, read) %>% arrange(female, read) %>% head
```

```
##      id female read
## 1   164      0    31
## 2    11      0    34
## 3    53      0    34
## 4   108      0    34
## 5   117      0    34
## 6   165      0    36
```

```
# Filtramos las filas por 'read' con un puntaje mayor o igual a 70
tab %>% select(id, female, read) %>% arrange(female, read) %>% filter(read >= 70)
```

```
##      id female read
## 1    95      0    73
## 2   132      0    73
## 3    68      0    73
## 4   103      0    76
## 5    57      1    71
## 6   180      1    71
## 7    34      1    73
## 8    93      1    73
## 9    61      1    76
```

```
#realizamos los mismos pasos anteriores, pero con orden descendente
tab %>% select(id, female, read) %>% arrange(female, desc(read)) %>% filter(read >= 70)
```

```
##      id female read
## 1   103      0    76
## 2    95      0    73
## 3   132      0    73
## 4    68      0    73
## 5    61      1    76
## 6    34      1    73
## 7    93      1    73
## 8    57      1    71
## 9   180      1    71
```

La funcion 'mutate()' crea nuevas columnas que sean funciones de las columnas ya existentes.

```
#Calculamos los puntajes promedio de lectura y escritura
head(mutate(tab, avg_read = sum(read)/n()))
```

```
##      id female race ses schtyp prog read write math science socst avg_read
## 1   70      0    4  1      1    1  57   52  41      47    57   52.23
## 2  121      1    4  2      1    3  68   59  53      63    61   52.23
## 3   86      0    4  3      1    1  44   33  54      58    31   52.23
## 4  141      0    4  3      1    3  63   44  47      53    56   52.23
## 5  172      0    4  2      1    2  47   52  57      53    61   52.23
## 6  113      0    4  2      1    2  44   52  51      63    61   52.23
```

podemos crear nuevas columnas usando 'mutate()' y operador '%>%'

```
tab %>% mutate(avg_read = sum(read)/n()) %>% head
```

```
##      id female race ses schtyp prog read write math science socst avg_read
## 1   70      0    4  1      1    1  57   52  41      47    57   52.23
## 2  121      1    4  2      1    3  68   59  53      63    61   52.23
## 3   86      0    4  3      1    1  44   33  54      58    31   52.23
## 4  141      0    4  3      1    3  63   44  47      53    56   52.23
## 5  172      0    4  2      1    2  47   52  57      53    61   52.23
## 6  113      0    4  2      1    2  44   52  51      63    61   52.23
```

'summarise()' funciona como 'summary()',ya que con ella podemos crear estadísticas de resumen para una columna dada en el dataframe, combinando otras estadísticas de resumen, como mean(), sd(), min(), max(), median(), sum(), n().

```
#contrae un dataframe en una sola fila.
summarise(tab, avg_read = mean(read, na.rm = TRUE))
```

```
##      avg_read
## 1      52.23
```

```
tab %>% summarise(avg_read = mean(read),min_read = min(read),max_read = max(read),n = n())
```

```
##      avg_read min_read max_read    n
## 1      52.23      28      76 200
```

Esta función divide un conjunto de datos en grupos específicos de filas (agrupa). Divide el dataframe por alguna variable, aplica una función a los dataframe individuales y luego combina la salida.

```
#primero agruparemos por genero y luego muestra las estadísticas obtenidas (media, minimo y maximo)
by_gender <- group_by(tab, female)
read_by_gender <- summarise(by_gender,
                             n = n(),
                             avg_read = mean(read, na.rm = TRUE),
                             min_read = min(read,na.rm = TRUE),
                             max_read = max(read,na.rm = TRUE))
read_by_gender
```

```
## # A tibble: 2 x 5
##   female      n avg_read min_read max_read
##   <int> <int>   <dbl>   <int>   <int>
## 1     0    91    52.8     31     76
## 2     1   109    51.7     28     76
```

## Introduccion al paquete ggplot2

<https://swcarpentry.github.io/r-novice-gapminder-es/08-plot-ggplot2/index.html>

‘ggplot2’ es un paquete de R para crear gráficos elegantes para el análisis de datos, Implementada por Hardley Wickham.

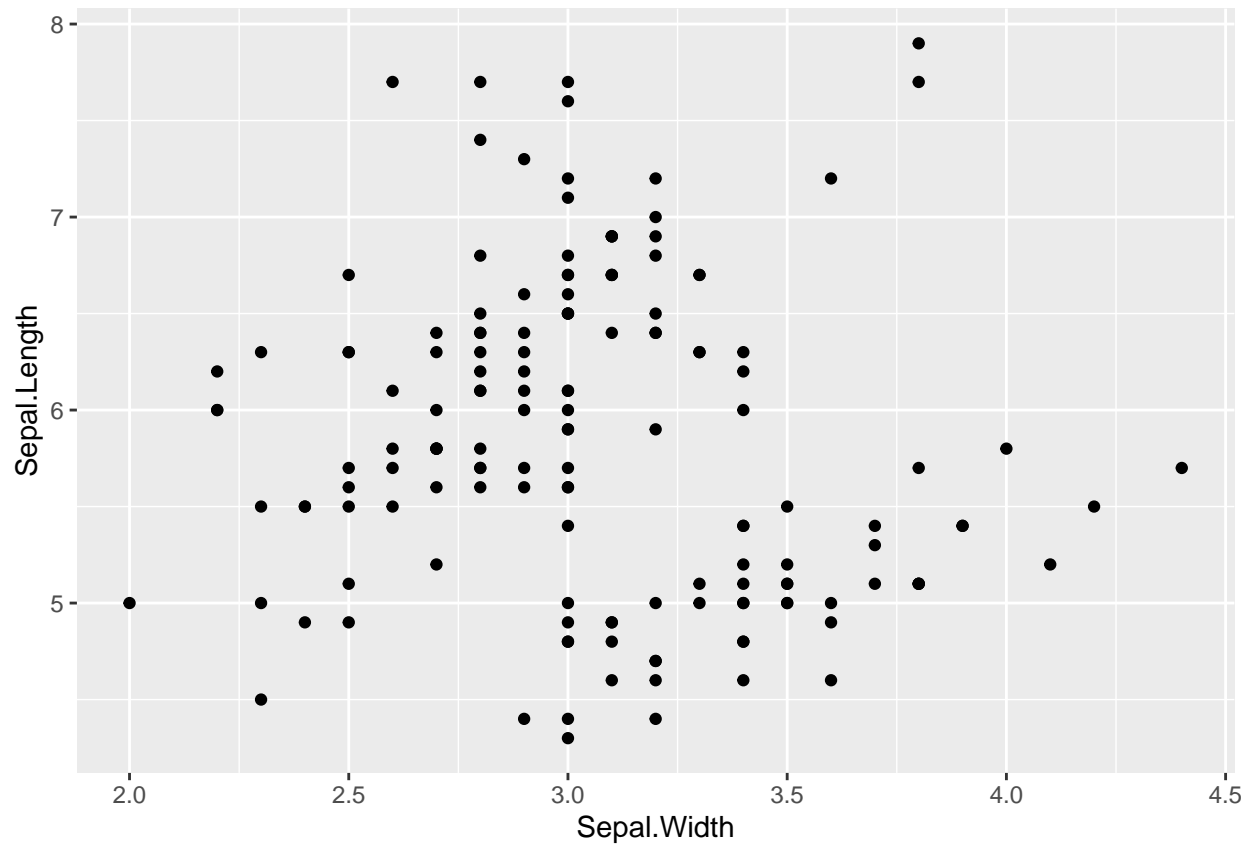
Usa el paquete (grid) creado por Paul Murrell para facilitar la generación de gráficos completos. Está basado en “The Grammar of Graphics” de Leland Wilkinson (2000). Propone una serie de ideas novedosas sobre cómo se debe generar un gráfico. Las ideas centrales son: (1) Todos los gráficos pueden generarse mediante un lenguaje regular, con una sintaxis determinada. (2) Es posible construir una serie de reglas comunes, conocidas y regulares para crear representaciones visuales de datos de interés estadístico. (3) Es un marco que sigue un enfoque de capas para describir y construir visualizaciones o gráficas de manera estructurada.

Capas o elementos de un gráfico en ggplot2 - Datos(data) -> datos a graficar, ‘ggplot2’ solo acepta un tipo de datos “data.frames” - Estéticas (aes(x,y)) -> son los datos de los ejes - Geometrías (geom) -> es el tipo grafica, ya sea en puntos, o lineas continuas (esto va por capas y es posible agregar varias capas de tipos de graficas) - Temas -> permiten hacer modificaciones a los aspectos estéticos del gráfico, incluye: ejes, etiquetas, colores de fondo, tamaño de los márgenes, paletas de colores, etc. - Facetas -> Las facetas nos permiten hacer comparaciones entre las diferentes variables. - Estadísticas - Coordenadas

```
library(ggplot2)
```

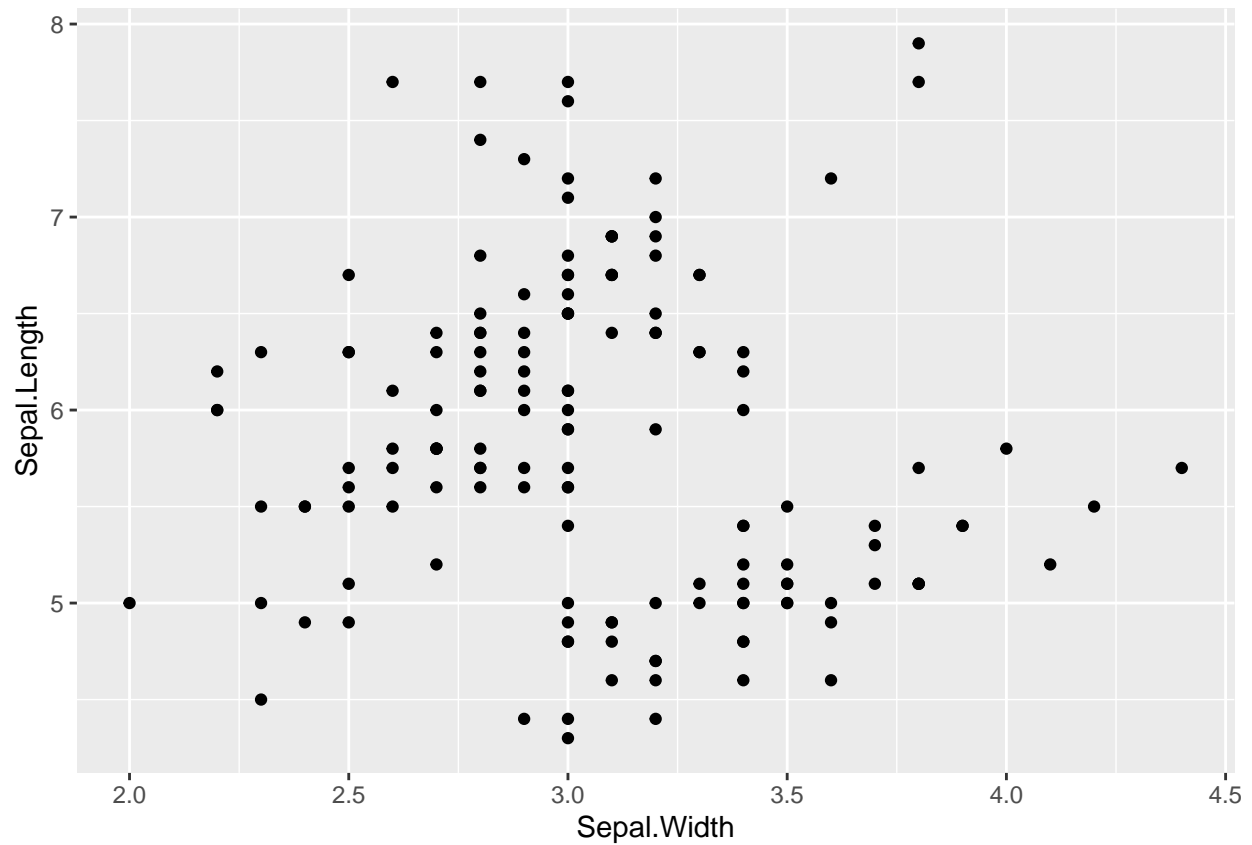
Podemos ver que una sola capa especifica los datos, el mapeo, la geom, la estadística y la posición, las dos escalas de posición continuas y un sistema de coordenadas cartesianas.

```
ggplot() +
  layer(
    data = iris,
    mapping = aes(x = Sepal.Width, y = Sepal.Length),
    geom = "point",
    stat = "identity",
    position = "identity" ) +
  scale_y_continuous() +
  scale_x_continuous() +
  coord_cartesian()
```



ggplot es una función más compleja que nos va a permitir explorar más a fondo los datos. Se pueden agregar varias capas, es importante tener en cuenta que cada capa se dibuja sobre la capa anterior.

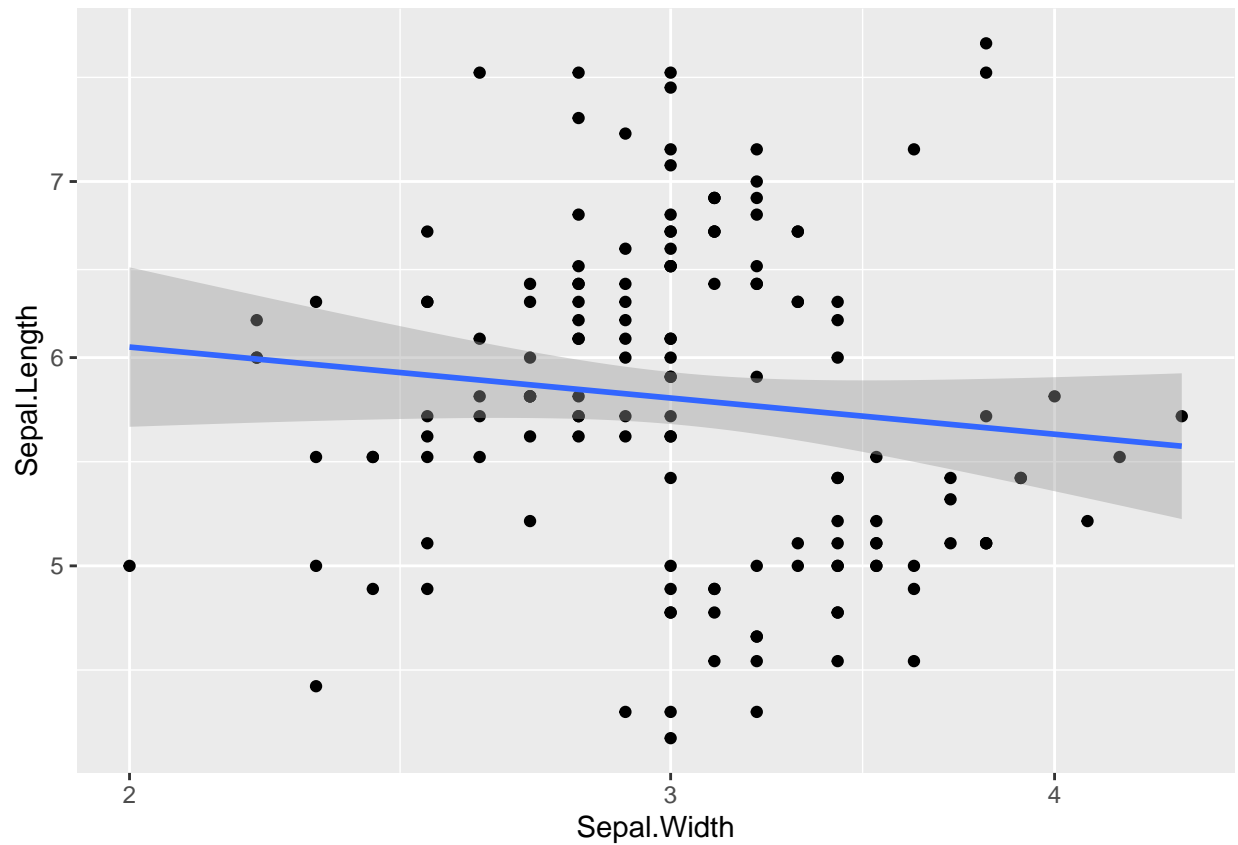
```
ggplot(iris, aes(Sepal.Width, Sepal.Length)) + geom_point()
```



```
ggplot(iris, aes(Sepal.Width, Sepal.Length)) +
  geom_point() +
  stat_smooth(method = lm) +
  scale_x_log10() +
  scale_y_log10() #podemos cambiar la escala de unidades en los ejes, usando las funciones de escala (sc

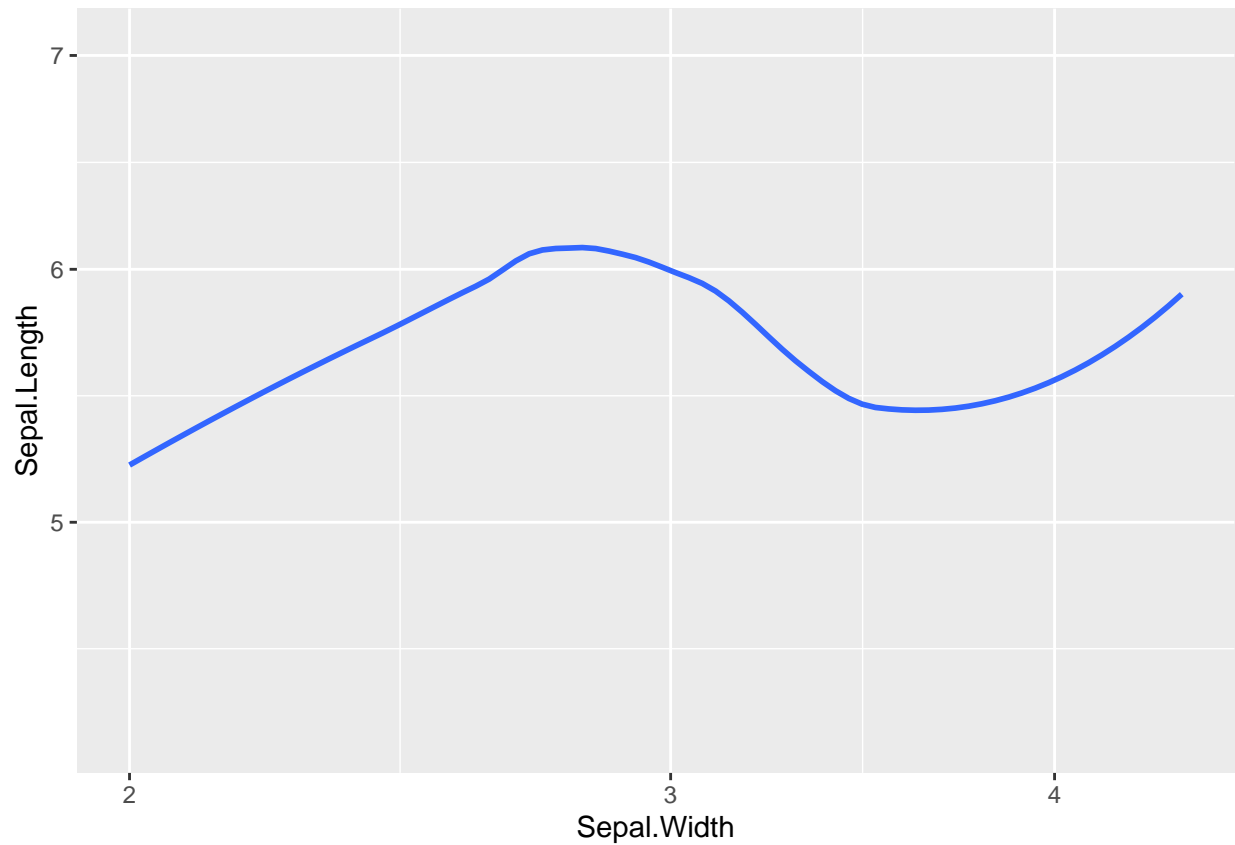
## 'geom_smooth()' using formula 'y ~ x'
```





```
ggplot() +
  layer(
    data = iris,
    mapping = aes(x = Sepal.Width, y = Sepal.Length),
    geom = "smooth",
    position = "identity",
    stat = "smooth",
  ) + scale_y_log10() + scale_x_log10() + coord_cartesian()
```

```
## 'geom_smooth()' using method = 'loess' and formula 'y ~ x'
```

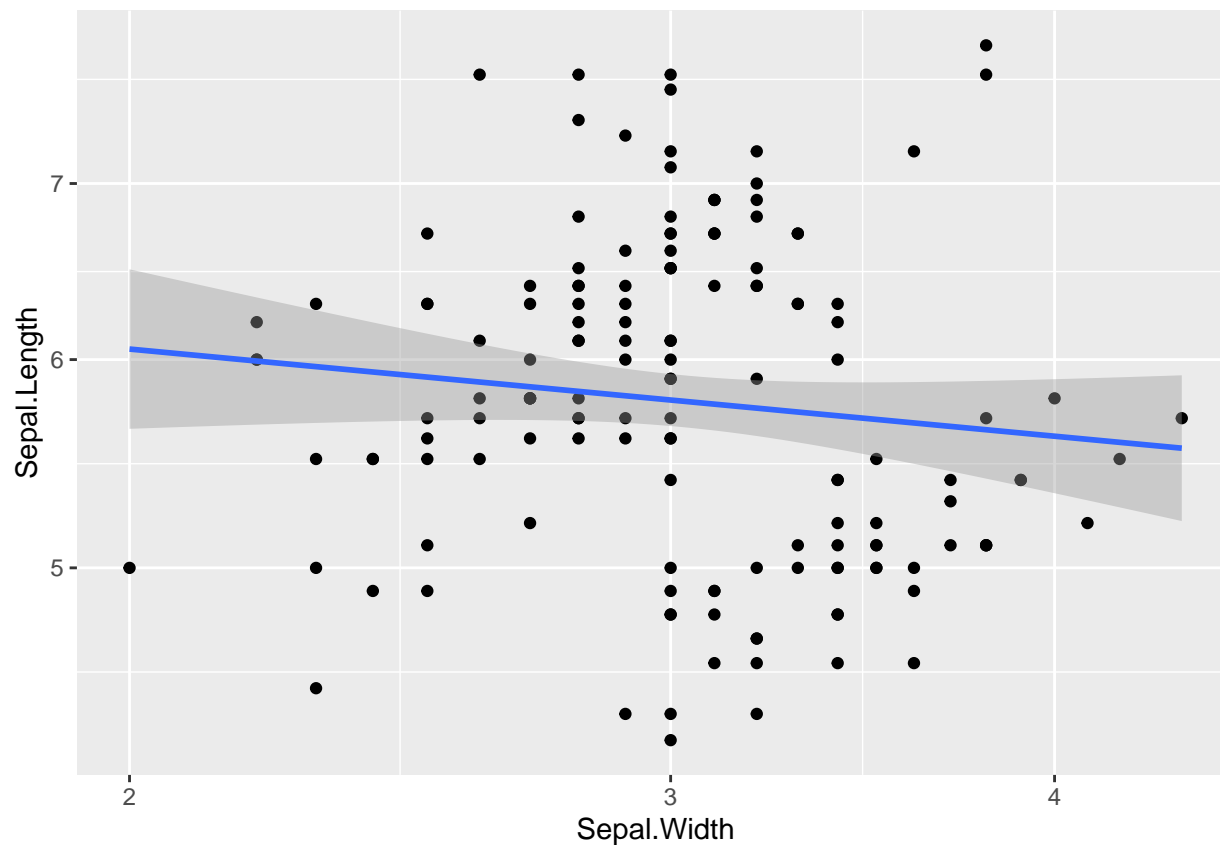


‘qplot()’ es una función que permite ver los datos de manera rápida, crea un gráfico completo con los datos, geometría y mapeos. Proporciona muchos valores por defecto.

```
qplot(Sepal.Width, Sepal.Length,  
      data = iris,  
      geom = c("point", "smooth"),  
      method = "lm",  
      log = "xy")
```

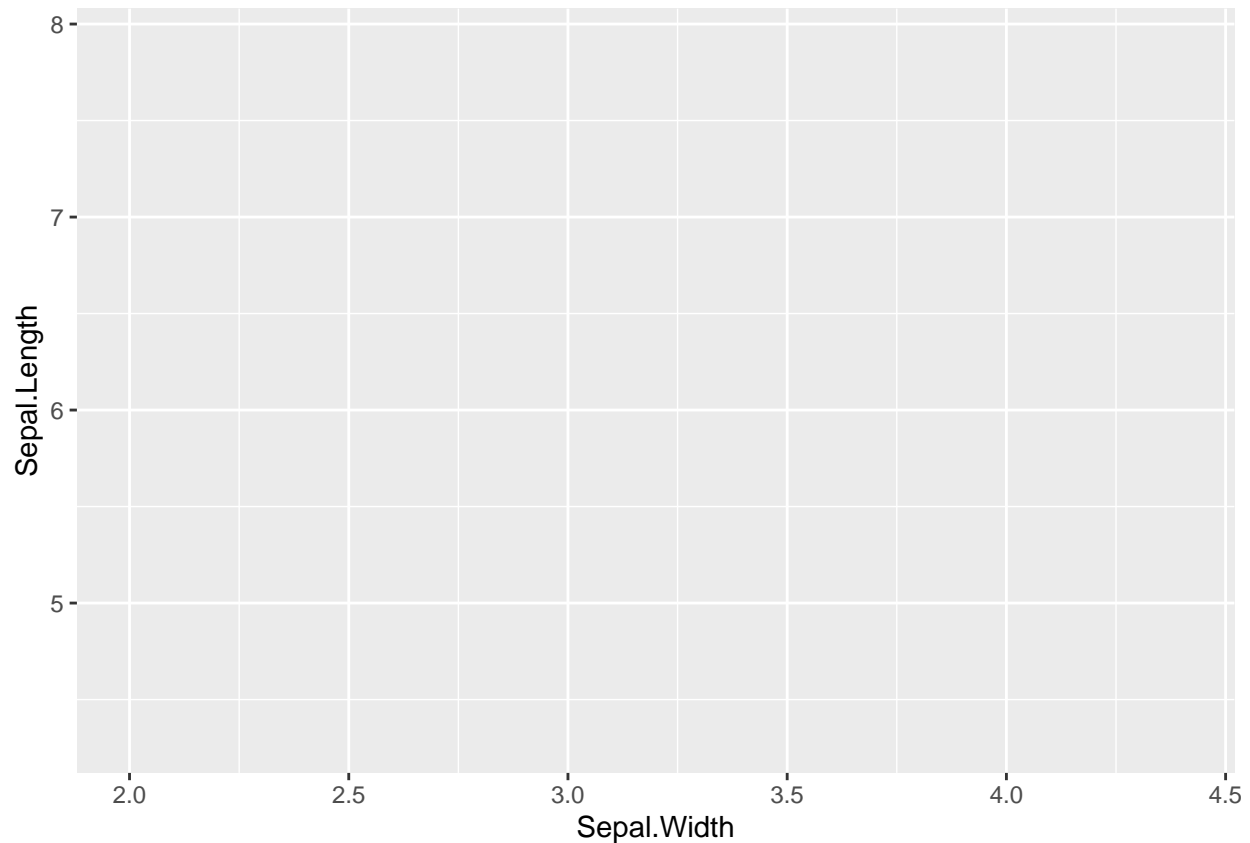
```
## Warning: Ignoring unknown parameters: method
```

```
## ‘geom_smooth()’ using formula ‘y ~ x’
```



## Paso 1 La primera capa para una grafica con 'ggplot' es elobgeto en donde seespera dibujar.

```
#library(ggplot2)
p <- ggplot(iris, aes(x=Sepal.Width, y=Sepal.Length))
# Sepal.Width y Sepal.Length son columnas en el dataframe iris
p
```



Podemos acceder a los detalles de la información usando la función de resumen () para realizar un seguimiento de qué datos se usaron exactamente y cómo se mapearon las variables.

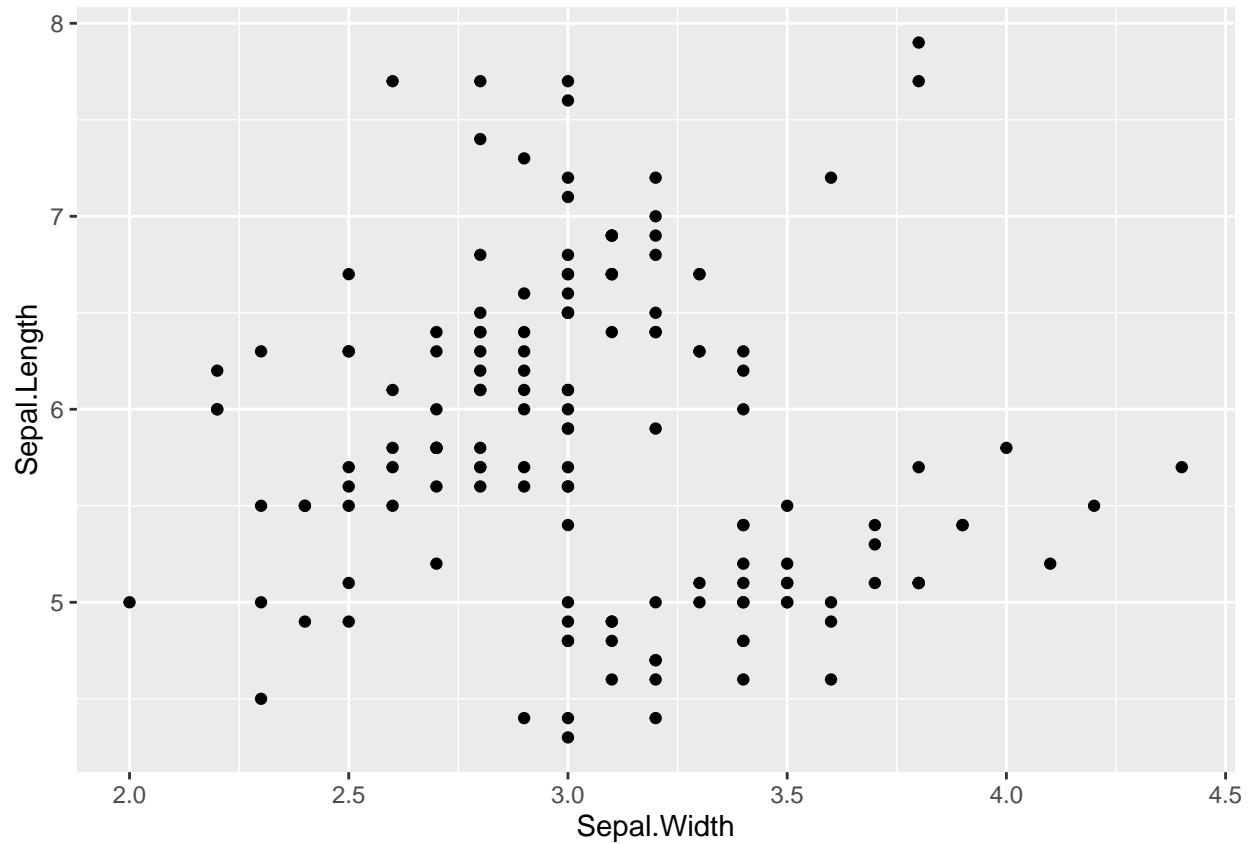
```
summary(p)
```

```
## data: Sepal.Length, Sepal.Width, Petal.Length, Petal.Width, Species
## [150x5]
## mapping: x = ~Sepal.Width, y = ~Sepal.Length
## faceting: <ggproto object: Class FacetNull, Facet, gg>
##   compute_layout: function
##   draw_back: function
##   draw_front: function
##   draw_labels: function
##   draw_panels: function
##   finish_data: function
##   init_scales: function
##   map_data: function
##   params: list
##   setup_data: function
##   setup_params: function
##   shrink: TRUE
##   train_scales: function
##   vars: function
##   super: <ggproto object: Class FacetNull, Facet, gg>
```

## Paso 2

Para dibujar un diagrama de dispersión, agregamos puntos usando una capa geom llamada `geom_point()` al objeto de la gráfica `p`.

```
p1 <- p + geom_point()
p1
```

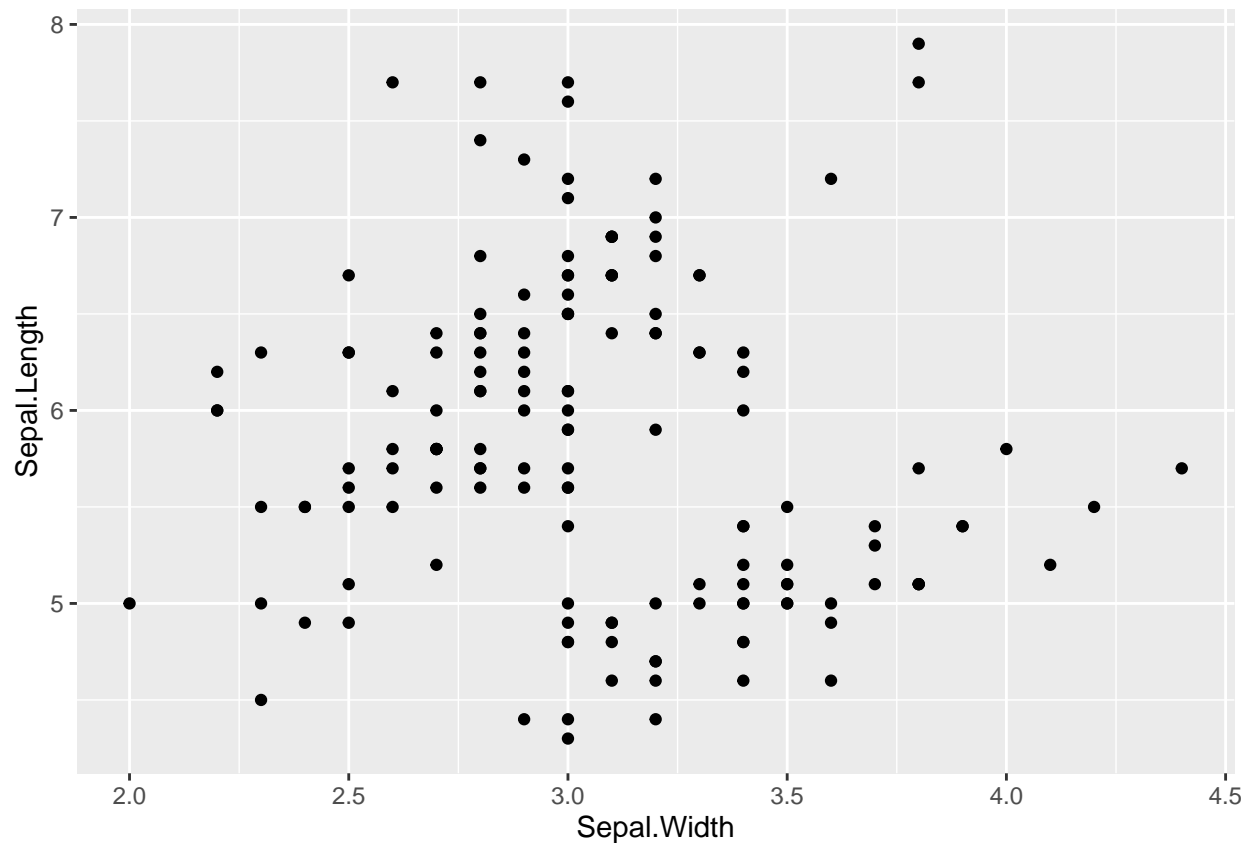


```
summary(p1)
```

```
## data: Sepal.Length, Sepal.Width, Petal.Length, Petal.Width, Species
## [150x5]
## mapping: x = ~Sepal.Width, y = ~Sepal.Length
## faceting: <ggproto object: Class FacetNull, Facet, gg>
##   compute_layout: function
##   draw_back: function
##   draw_front: function
##   draw_labels: function
##   draw_panels: function
##   finish_data: function
##   init_scales: function
##   map_data: function
##   params: list
##   setup_data: function
##   setup_params: function
```

```
## shrink: TRUE
## train_scales: function
## vars: function
## super: <ggproto object: Class FacetNull, Facet, gg>
## -----
## geom_point: na.rm = FALSE
## stat_identity: na.rm = FALSE
## position_identity
```

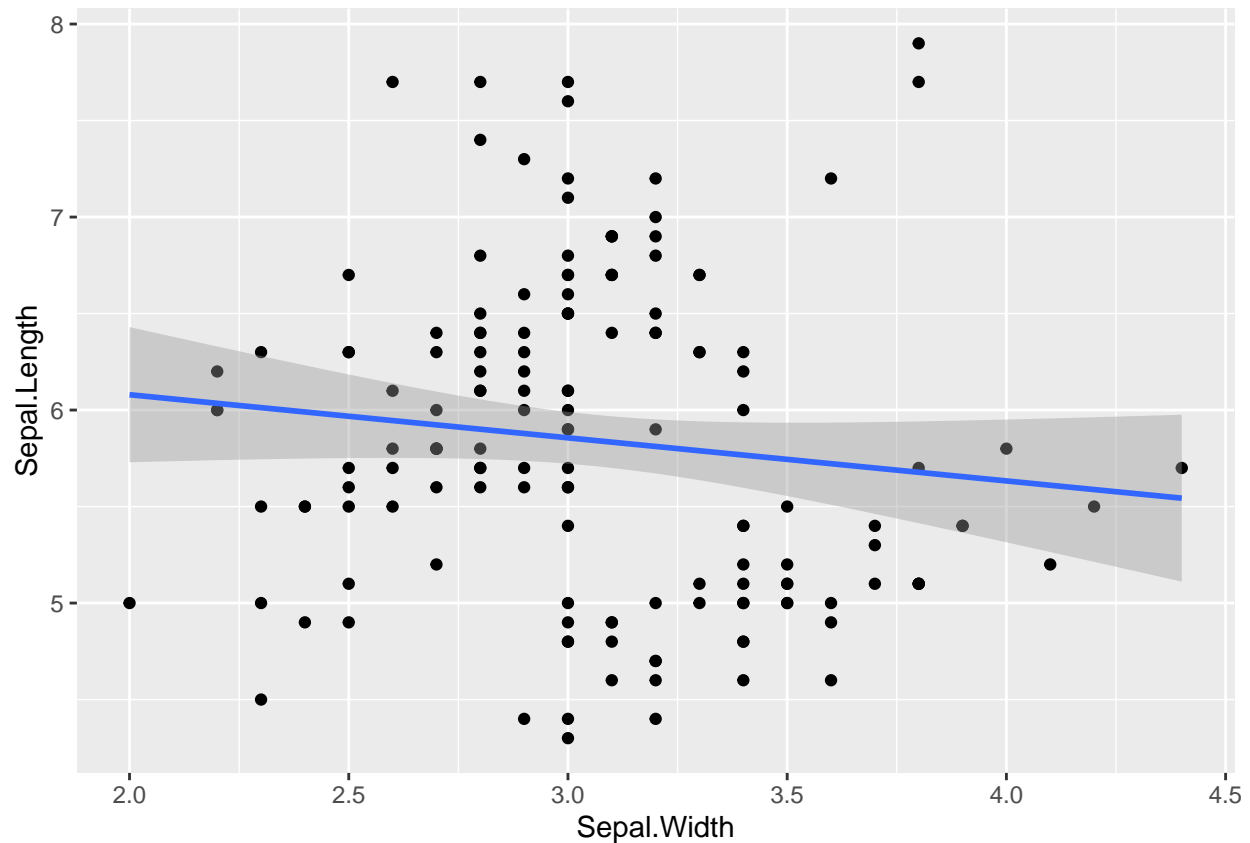
```
ggplot(iris, aes(x=Sepal.Width, y=Sepal.Length)) + geom_point()
```



## Paso 3 Para dibujar un diagrama de dispersión con una curva suave, agregamos una capa de geoma adicional llamada `geom_smooth()` al objeto de trazado anterior.

```
p2 <- p1 + geom_smooth(method="lm")
p2
```

```
## 'geom_smooth()' using formula 'y ~ x'
```



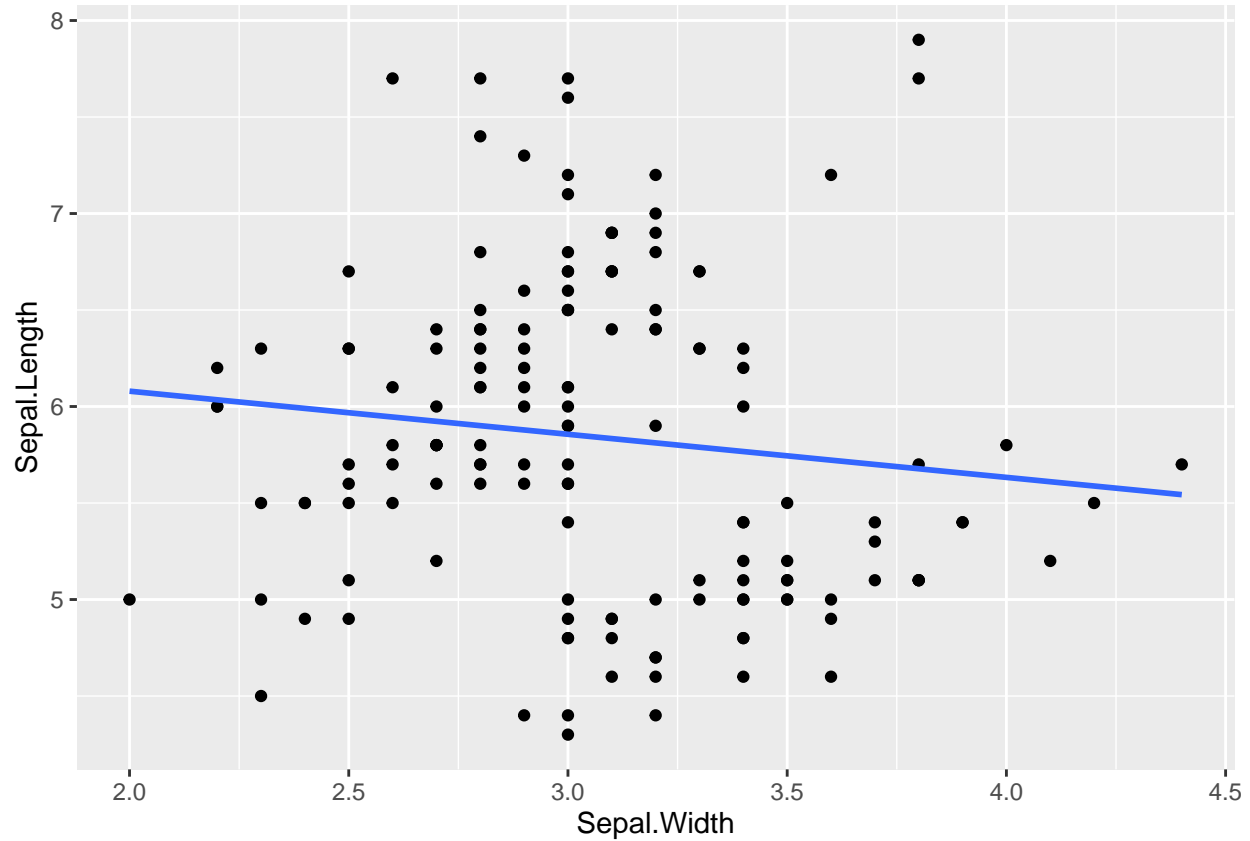
```
summary(p2)
```

```
## data: Sepal.Length, Sepal.Width, Petal.Length, Petal.Width, Species
## [150x5]
## mapping: x = ~Sepal.Width, y = ~Sepal.Length
## faceting: <ggproto object: Class FacetNull, Facet, gg>
##   compute_layout: function
##   draw_back: function
##   draw_front: function
##   draw_labels: function
##   draw_panels: function
##   finish_data: function
##   init_scales: function
##   map_data: function
##   params: list
##   setup_data: function
##   setup_params: function
##   shrink: TRUE
##   train_scales: function
##   vars: function
##   super: <ggproto object: Class FacetNull, Facet, gg>
## -----
## geom_point: na.rm = FALSE
## stat_identity: na.rm = FALSE
## position_identity
```

```
##
## geom_smooth: na.rm = FALSE, orientation = NA, se = TRUE
## stat_smooth: na.rm = FALSE, orientation = NA, se = TRUE, method = lm
## position_identity
```

```
p1 + geom_smooth(method="lm", se = FALSE)
```

```
## 'geom_smooth()' using formula 'y ~ x'
```

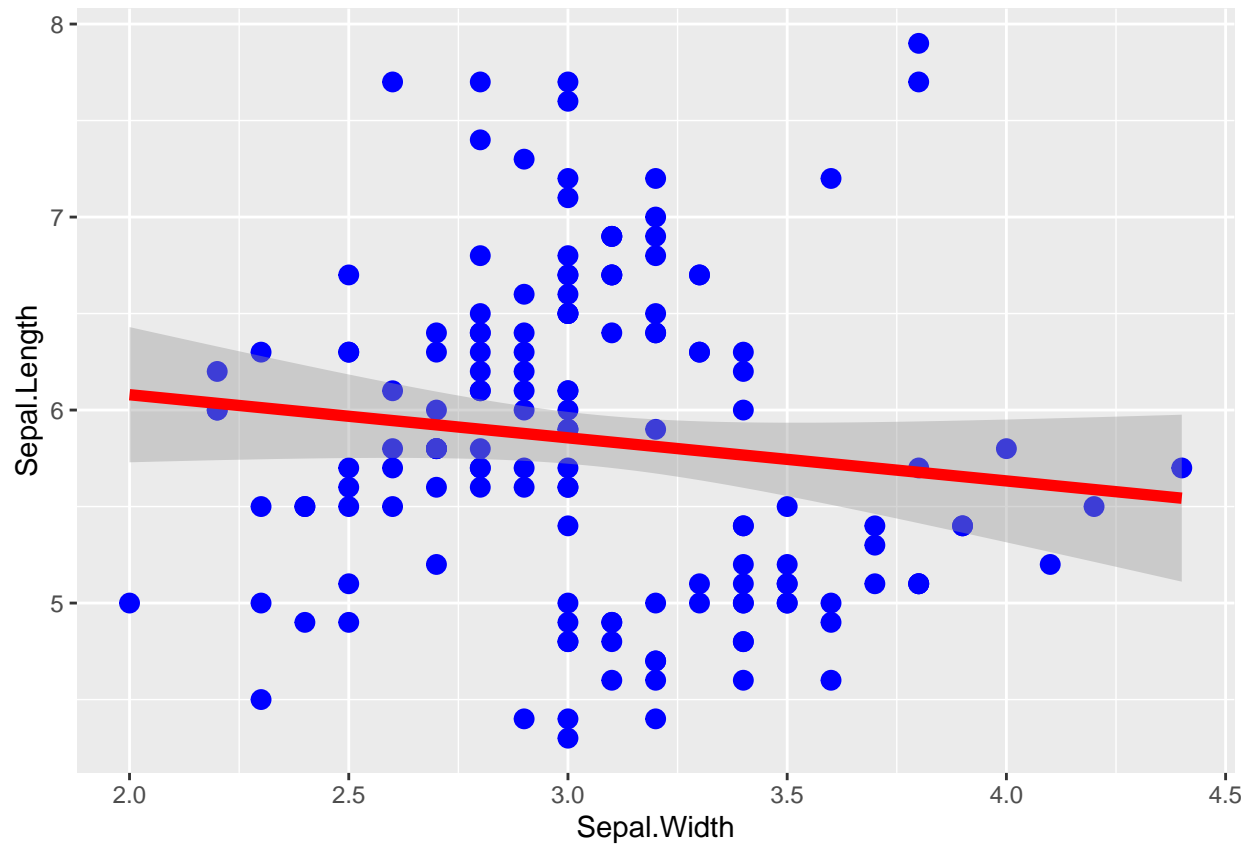


Podemos hacercambios en la estetica, escala y geometria delas graficas.

```
p3 <- ggplot(iris, aes(x=Sepal.Width, y=Sepal.Length)) +
  #Add scatterplot geom (layer1)
  geom_point(col="blue", size=3) +
  #Add smoothing geom (layer2)
  geom_smooth(method="lm", col="red", size=2)
p3
```

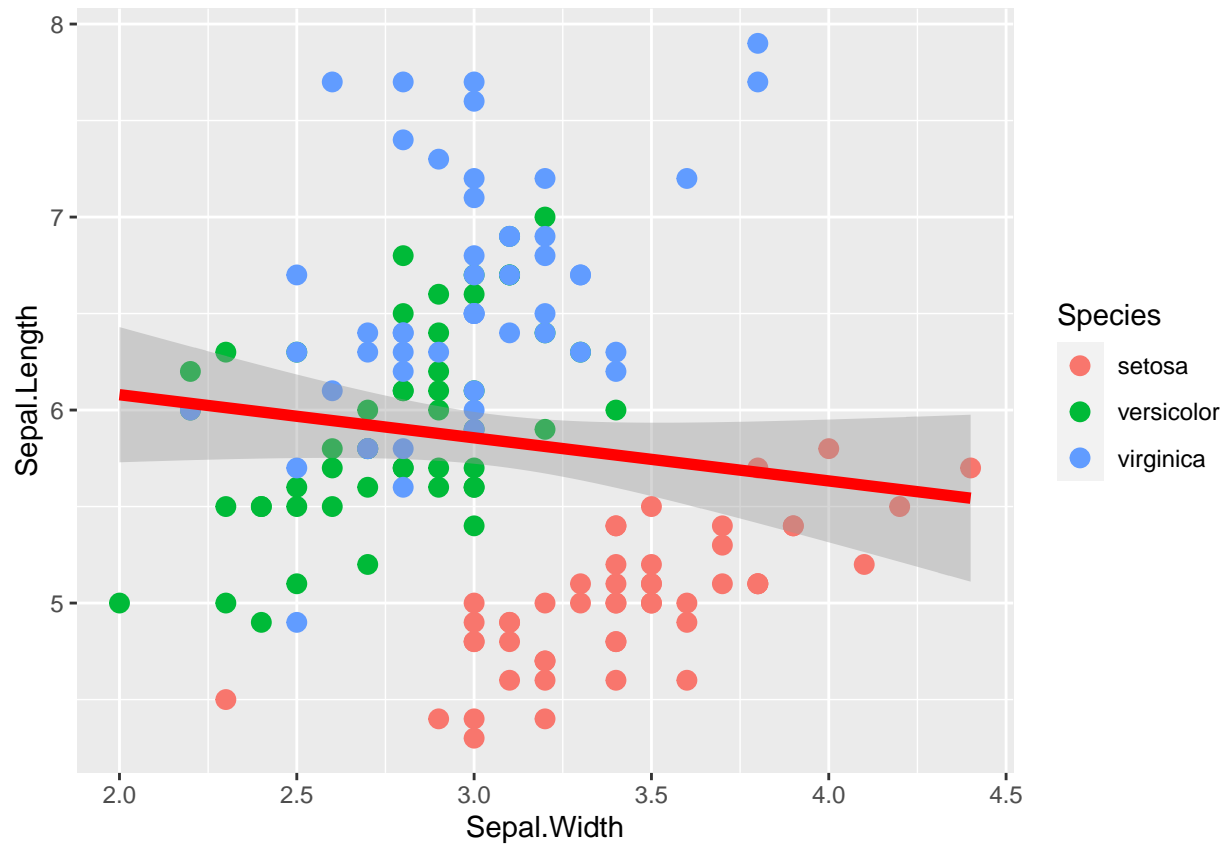
```
## 'geom_smooth()' using formula 'y ~ x'
```





```
p4 <- ggplot(iris, aes(x=Sepal.Width, y=Sepal.Length)) +  
  #Add scatterplot geom (layer1)  
  geom_point(aes(col=Species), size=3) +  
  #Add smoothing geom (layer2)  
  geom_smooth(method="lm", col="red", size=2)  
p4
```

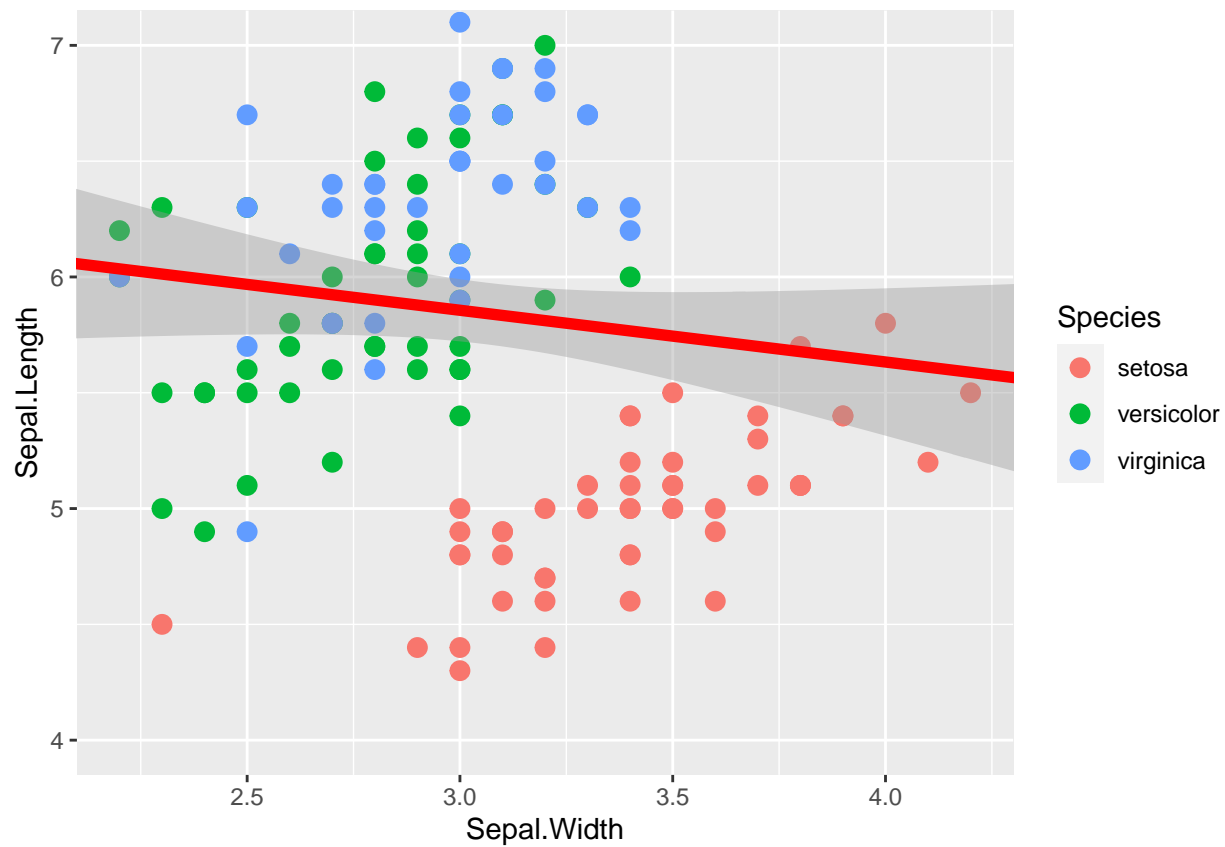
```
## 'geom_smooth()' using formula 'y ~ x'
```



Podemos tambien cambiar los limites de los ejes

```
p5 <- p4 + coord_cartesian(xlim=c(2.2,4.2), ylim=c(4, 7)) # zooms in
plot(p5)
```

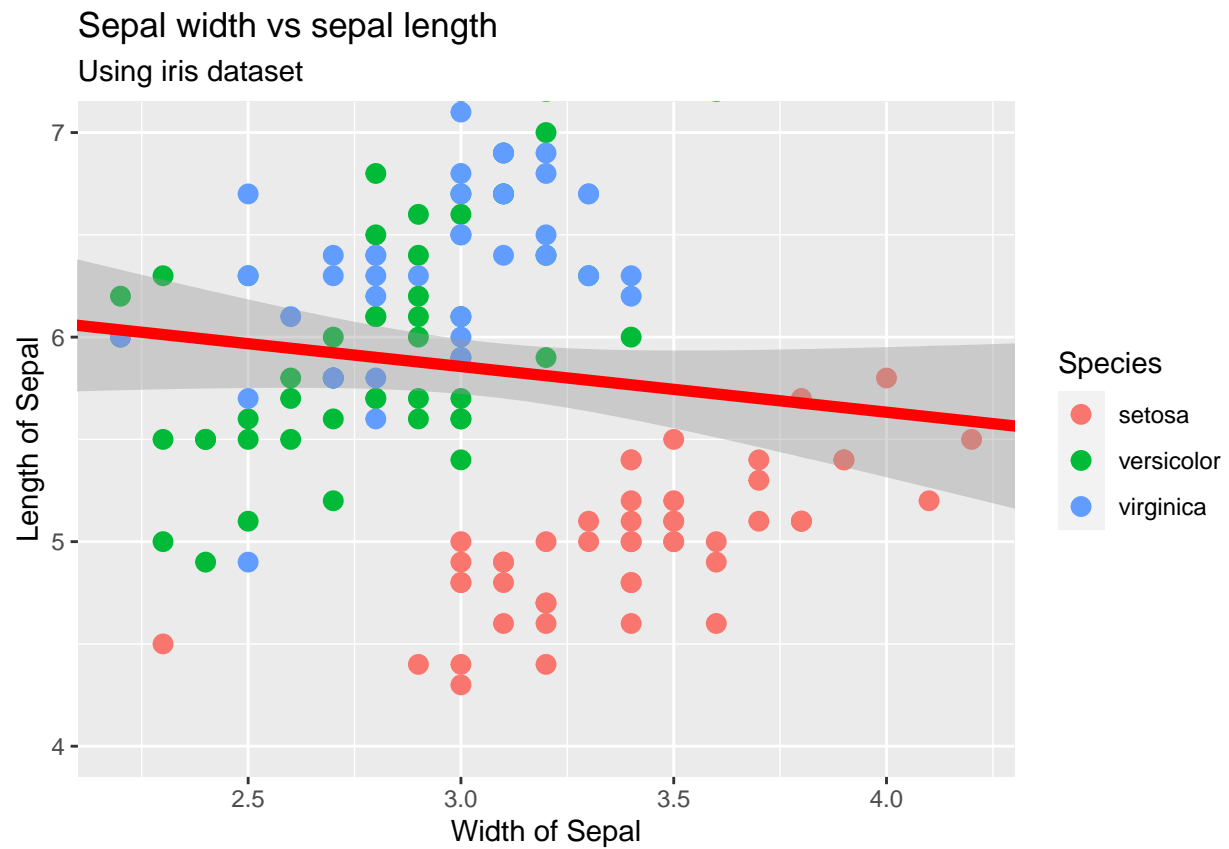
```
## 'geom_smooth()' using formula 'y ~ x'
```



Podemos agregar un titulo y labels al grafico.

```
p6 <- p5 + labs(title="Sepal width vs sepal length",
  subtitle="Using iris dataset",
  y="Length of Sepal",
  x="Width of Sepal")
print(p6) #Or plot(p6)
```

```
## 'geom_smooth()' using formula 'y ~ x'
```



```
plot(p6)
```

```
## 'geom_smooth()' using formula 'y ~ x'
```



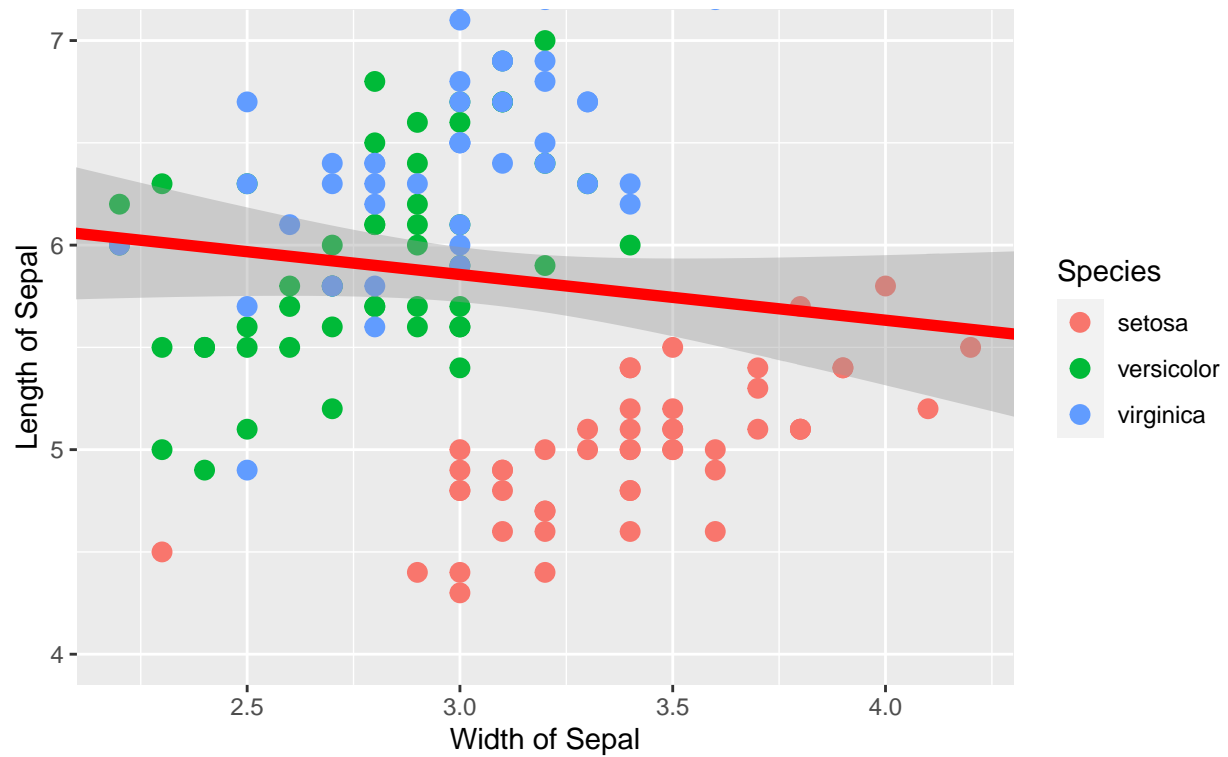
Podemos hacer la creacion del grafico en total

```
#library(ggplot2)
ggplot(iris, aes(x=Sepal.Width, y=Sepal.Length)) +
  geom_point(aes(col=Species), size=3) +
  geom_smooth(method="lm", col="red", size=2) +
  coord_cartesian(xlim=c(2.2,4.2), ylim=c(4, 7)) +
  labs(title="Sepal width vs sepal length",
        subtitle="Using iris dataset",
        y="Length of Sepal",
        x="Width of Sepal")
```

```
## 'geom_smooth()' using formula 'y ~ x'
```

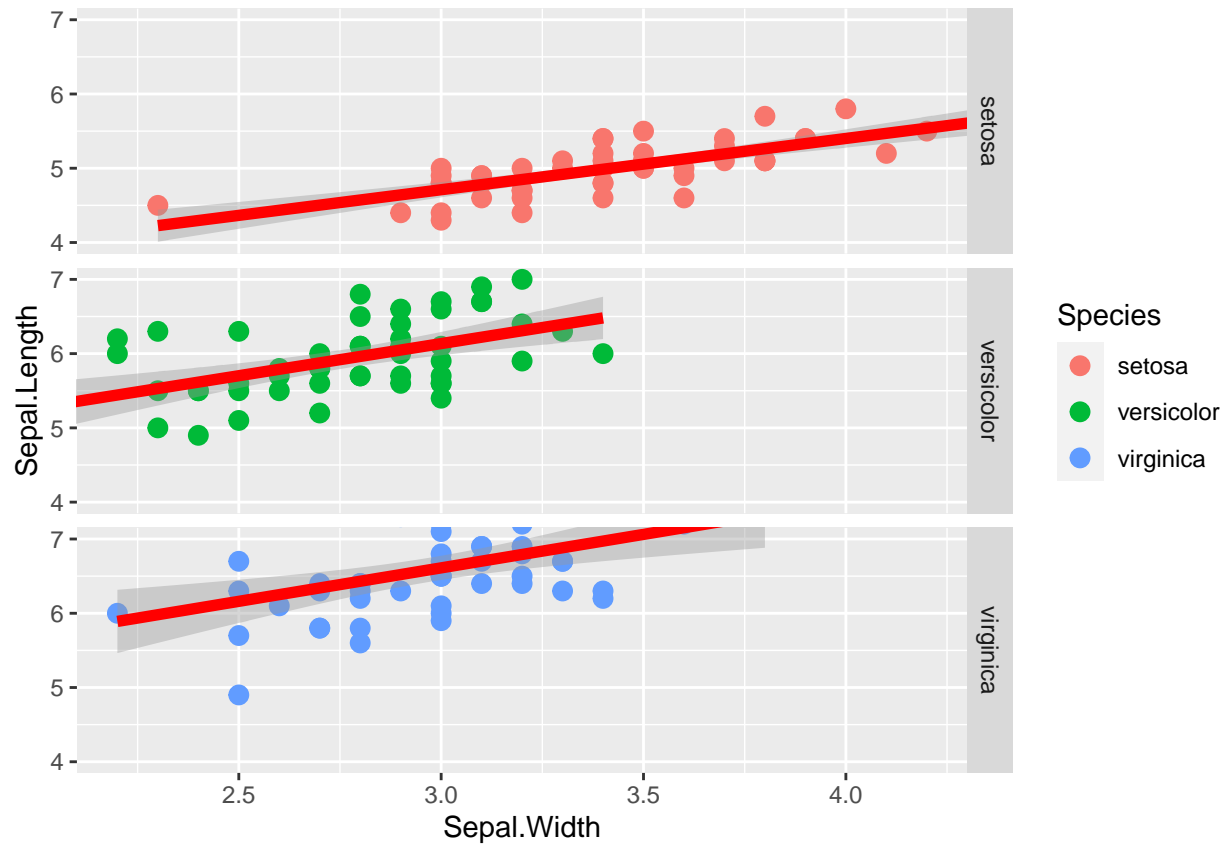
## Sepal width vs sepal length

Using iris dataset



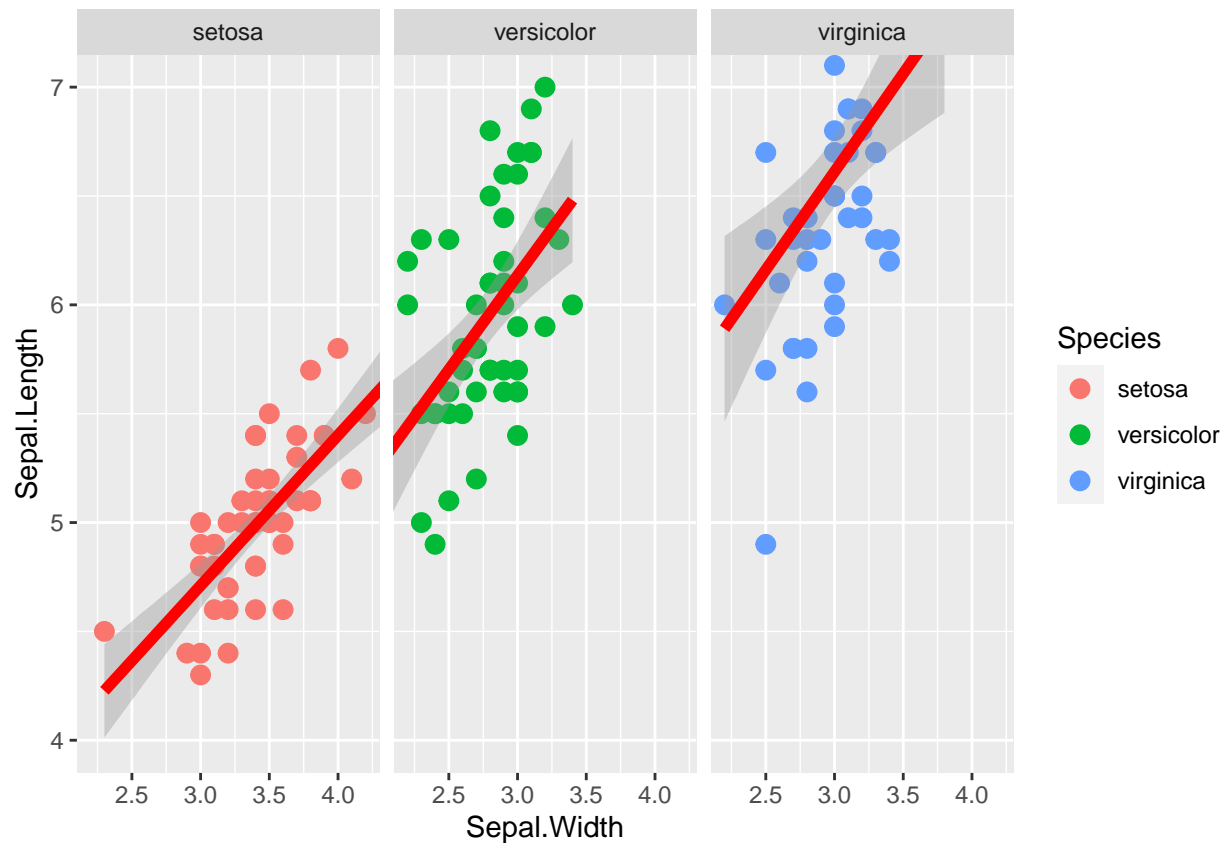
```
ggplot(iris, aes(x=Sepal.Width, y=Sepal.Length)) +  
  geom_point(aes(col=Species), size=3) +  
  geom_smooth(method="lm", col="red", size=2) +  
  coord_cartesian(xlim=c(2.2,4.2), ylim=c(4, 7)) +  
  # Add Facet Grid  
  facet_grid(Species ~.)
```

```
## 'geom_smooth()' using formula 'y ~ x'
```



```
ggplot(iris, aes(x=Sepal.Width, y=Sepal.Length)) +
  geom_point(aes(col=Species), size=3) +
  geom_smooth(method="lm", col="red", size=2) +
  coord_cartesian(xlim=c(2.2,4.2), ylim=c(4, 7)) +
  # Add Facet Grid
  facet_grid(~ Species)
```

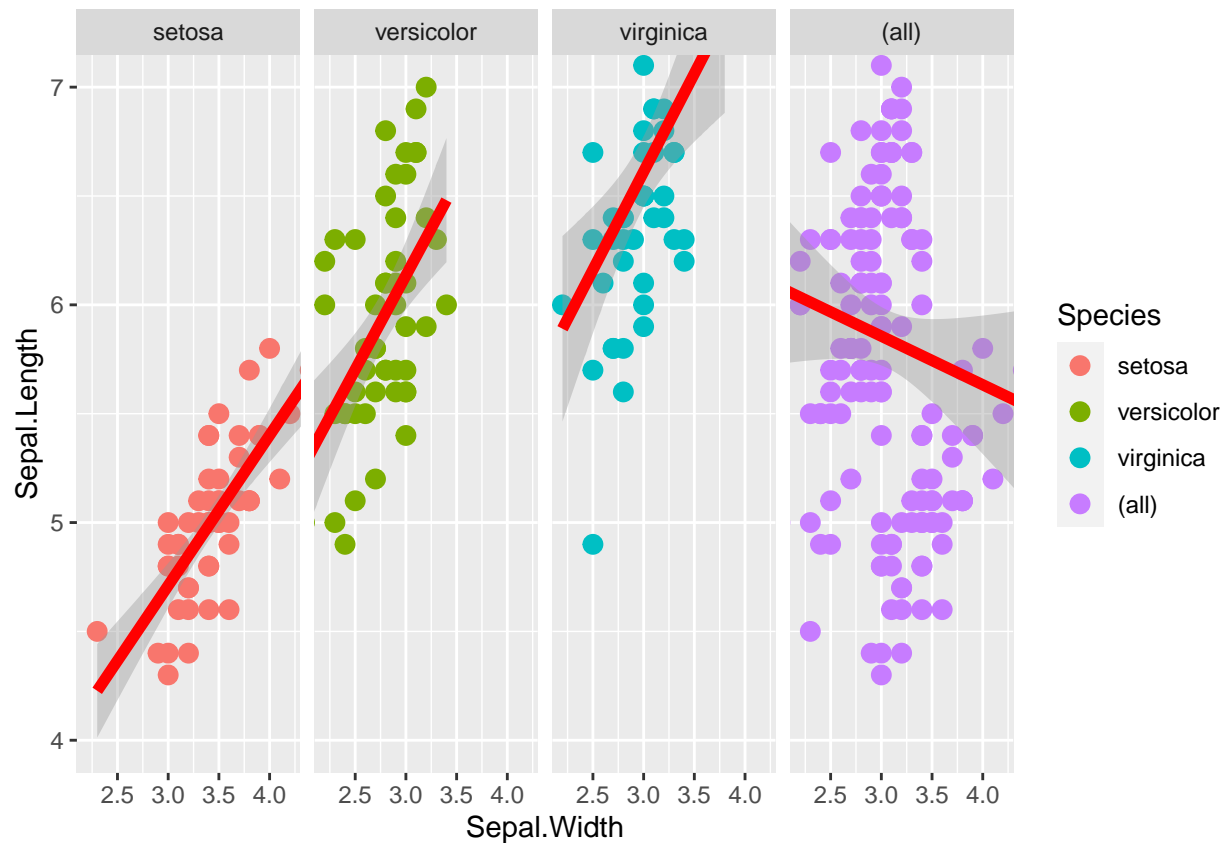
```
## 'geom_smooth()' using formula 'y ~ x'
```



```
ggplot(iris, aes(x=Sepal.Width, y=Sepal.Length)) +
  geom_point(aes(col=Species), size=3) +
  geom_smooth(method="lm", col="red", size=2) +
  coord_cartesian(xlim=c(2.2, 4.2), ylim=c(4, 7)) +
  # Add Facet Grid
  facet_grid(~ Species, margin=TRUE)
```

```
## 'geom_smooth()' using formula 'y ~ x'
```





```
ggplot(iris, aes(x=Sepal.Width, y=Sepal.Length)) +
  geom_point(aes(col=Species), size=3) +
  geom_smooth(method="lm", col="red", size=2) +
  coord_cartesian(xlim=c(2.2, 4.2), ylim=c(4, 7)) +
  #Add Facet Wrap
  facet_wrap(~ Species, nrow=2)
```

```
## 'geom_smooth()' using formula 'y ~ x'
```

