



Tratamiento Estadístico de Datos



Analytics AoZ

Preprocesamiento → Métodos de Remuestreo → Evaluación de las Métricas

Preprocesamiento

Analizaremos las diferentes técnicas que tenemos a nuestra disposición para realizar un buen preprocesamiento de datos y, así, mejorar el rendimiento de los modelos que generaremos posteriormente. Se debe tener en cuenta si en el modelado supervisado realizado el problema es de regresión o clasificación.

Data Wrangling with dplyr and tidyr

Cheat Sheet

Studio

Syntax - Helpful conventions for wrangling

dplyr::tbl_df(iris)
Converts data to tbl class. tbl's are easier to examine than data frames. R displays only the data that fits onscreen.

dplyr::glimpse(iris)
Information dense summary of tbl data.

utils::View(iris)
View data set in spreadsheet-like display (note capital V).

dplyr::%>%
Pastes object on left hand side as first argument (or, argument) of function on righthand side.

%>%
x %>% f(x) is the same as f(x, y)
y %>% f(x, ..., z) is the same as f(x, y, z)

"Piping" with %>% makes code more readable, e.g.

```
iris %>%  
  group_by(Species) %>%  
  summarise(mv = mean(Sepal.Length)) %>%  
  arrange(mv)
```

tidyR is a trademark of tidyR, Inc. • © 2019 tidyR, Inc. • 800-498-0212 • tidyR.com

Tidy Data - A foundation for wrangling in R

In a tidy data set:

- Each variable is saved in its own column
- Each observation is saved in its own row

Tidy data complements R's vectorized operations. R will automatically preserve observations as you manipulate variables. No other format works as intuitively with R.

Reshaping Data - Change the layout of a data set

dplyr::gather(cases, "year", "n", 2:4)
Gather columns into rows.

tidyr::spread(pollution, size, amount)
Spread rows into columns.

dplyr::separate(storms, date, c("y", "m", "d"))
Separate one column into several.

tidyr::unite(data, col, ..., sep)
Unite several columns into one.

Subset Observations (Rows)

dplyr::filter(iris, Sepal.Length > 7)
Extract rows that meet logical criteria.

dplyr::distinct(iris)
Remove duplicate rows.

dplyr::sample_frac(iris, 0.5, replace = TRUE)
Randomly select fraction of rows.

dplyr::sample_n(iris, 10, replace = TRUE)
Randomly select n rows.

dplyr::slice(iris, 10:15)
Select rows by position.

dplyr::top_n(storms, 2, date)
Select and order top n entries (by group if grouped data).

Subset Variables (Columns)

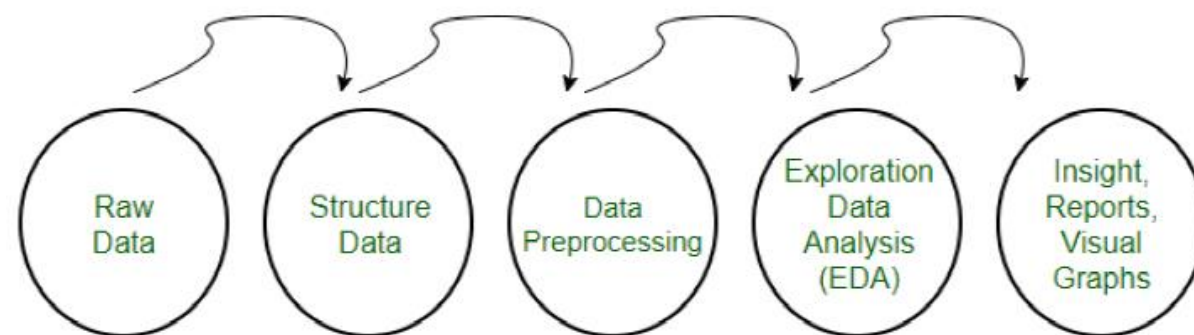
dplyr::select(iris, Sepal.Length, Species)
Select columns by name or helper function.

Helper functions for select - select

- select(iris, contains("year"))**
Select columns whose name contains a character string.
- select(iris, ends_with("Length"))**
Select columns whose name ends with a character string.
- select(iris, everything())**
Select every column.
- select(iris, matches(".*"))**
Select columns whose name matches a regular expression.
- select(iris, num_range("x", 1:3))**
Select columns named x1, x2, x3, x4, x5.
- select(iris, one_of("Species", "Sepal"))**
Select columns whose name are in a group of names.
- select(iris, starts_with("Sepal"))**
Select columns whose name starts with a character string.
- select(iris, Sepal.Length:Petal.Length)**
Select all columns between Sepal.Length and Petal.Length (inclusive).
- select(iris, -Species)**
Select all columns except Species.
- select(iris, !Species)**
Select all columns except Species.

Revised: 10/10/2019, 10/10/2019 for tidyR v1.0.0. Learn more with [tidyR::get_tidyR_version\(\)](#) for data sets. Learn more with [tidyR::get_tidyR_version\(\)](#) for data sets. tidyR v1.0.0 - tidyR 1.0.0 - updated 1/1/20

Data Wrangling

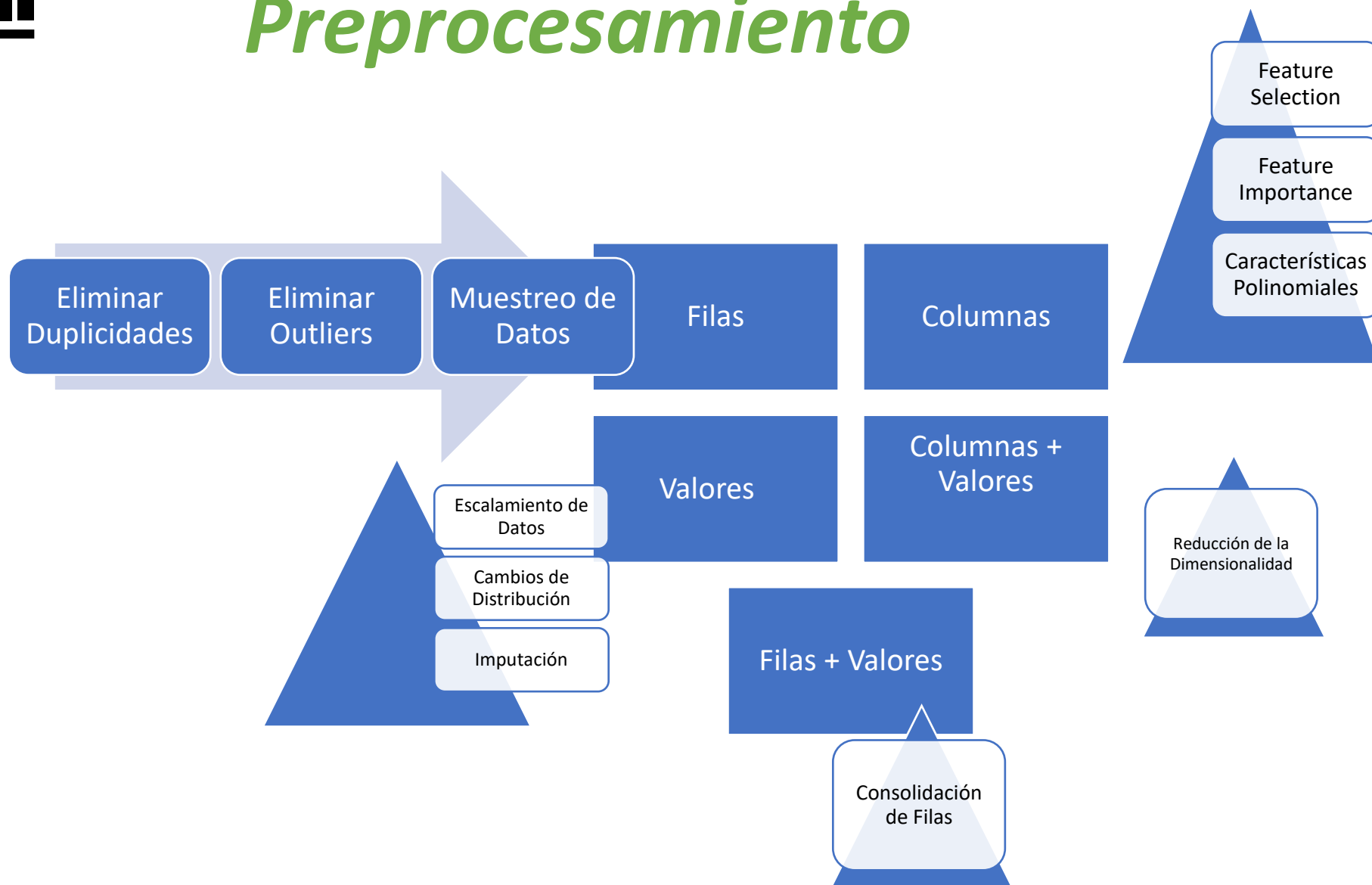




Preprocesamiento



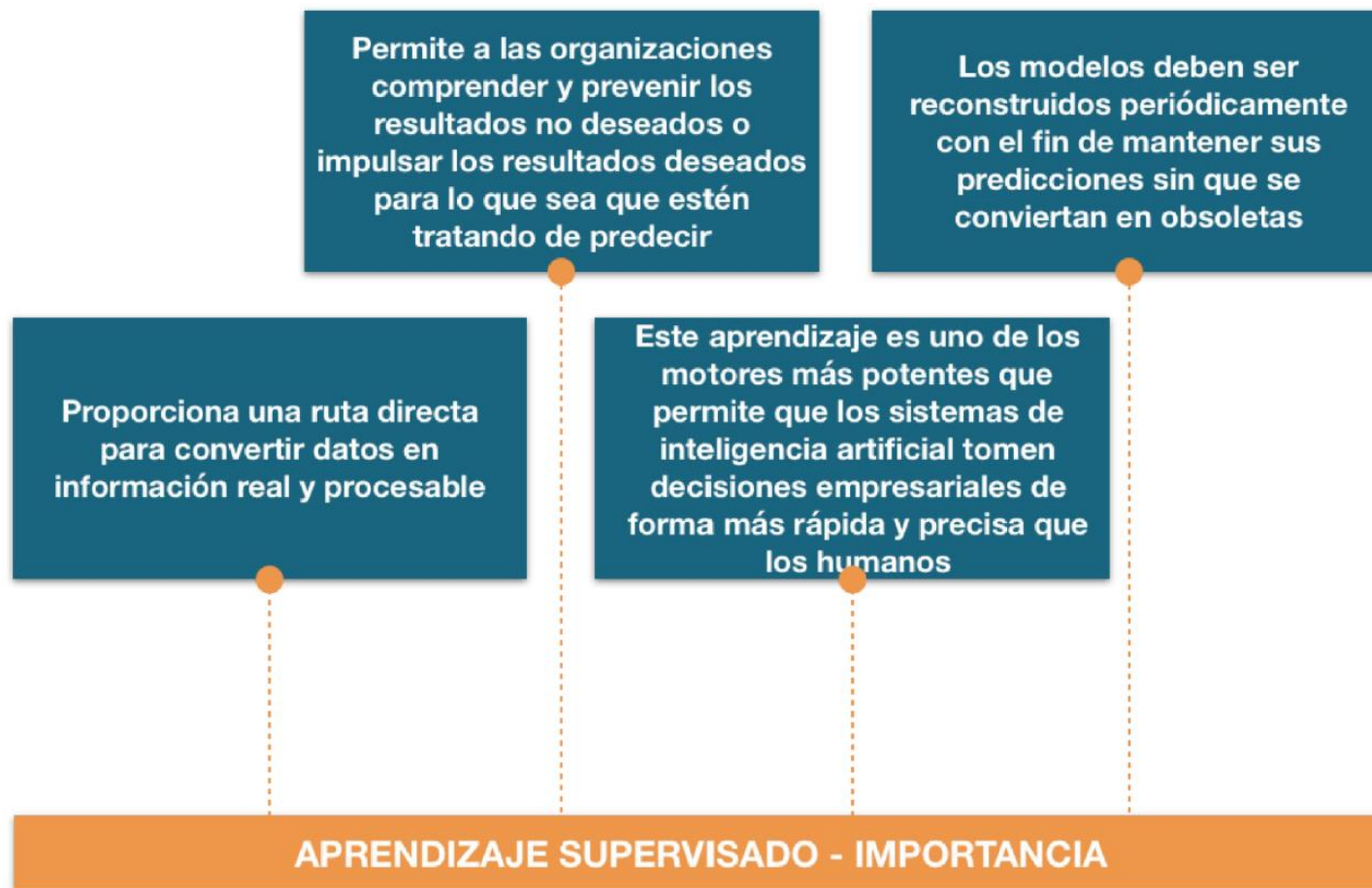
Analytics AoZ

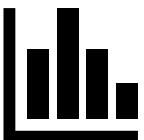




Analytics AoZ

APRENDIZAJE SUPERVISADO

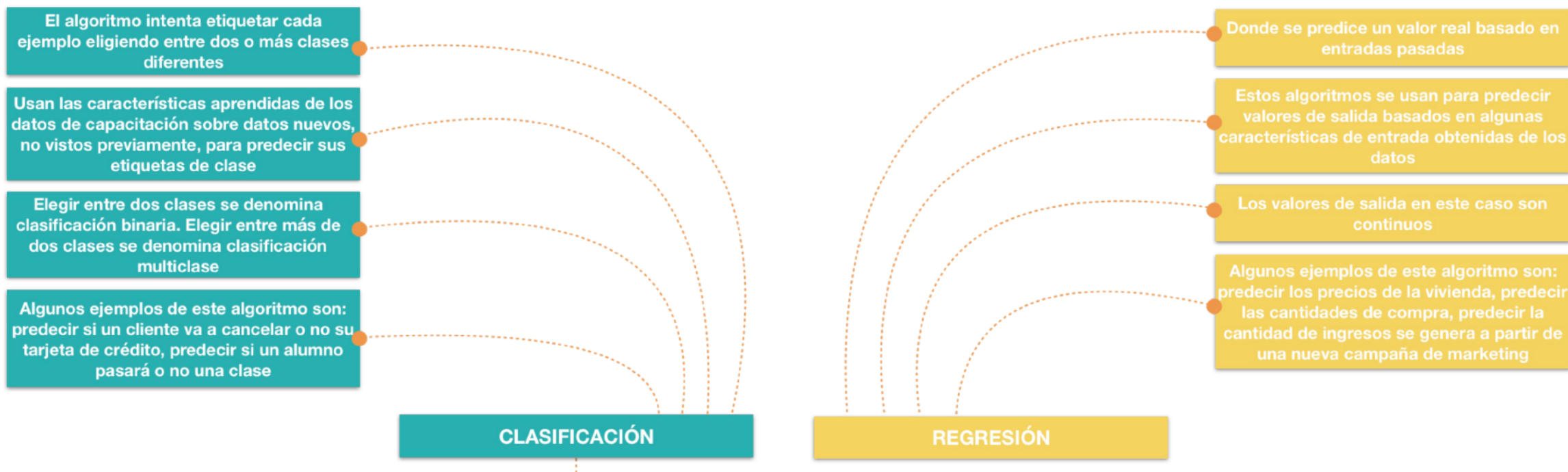




PROCESO DE MACHINE LEARNING SINTETIZADO



Analytics AoZ





TRATAMIENTO DE DATOS



(a) Analogía con la vida diaria.



(b) Proceso completo de machine learning.



Necesidad de Preprocesamiento

- Los datos en bruto, sin analizar de antemano, muy difícilmente nos va a dar **precisiones robustas a la hora de realizar modelos de machine learning** ya que, entre otros aspectos, muchos de ellos requieren los datos en una forma específica, por lo que debemos realizar transformación del conjunto de datos. Por otro lado, otros algoritmos pueden funcionar mejor si los datos se preparan de una manera específica, por ejemplo, los algoritmos *tipo árbol* con atributos características tipo nominal. Por lo tanto, es de vital importancia realizar un preprocesamiento de nuestros datos como parte fundamental de un proyecto de Machine Learning.



Consideraciones a tomar

La aplicación de algoritmos de Machine Learning para la predicción es una **ciencia empírica**, a pesar de las guías generales que podemos tener, la única forma de saber que algoritmos usar para nuestro conjunto de datos es a través de nuestra experiencia y conocimiento conjunto de nuestros datos. Debemos probar una variedad de transformaciones de datos con un conjunto de algoritmos diferentes que nos ayudará a descubrir qué algoritmos pueden ser los **mejores candidatos** para un conjunto determinado de datos. A grande rasgos:

- Los métodos basados en **instancias (filas)** son más efectivos si los atributos de entrada tienen la misma **escala**.
- Los métodos de regresión pueden funcionar mejor si los atributos de entrada están **estandarizados**.



¿Por qué transformar?

En algunos casos al realizar el modelo predictivo de Machine Learning y al momento de cuantificar categorías el modelo le da *mayor relevancia a una que otra*, cuando tendría que haber ponderado a ambas por igual. Por ejemplo 3 variables analizadas (edad, salario que aspira el candidato, contratado). Donde Edad es la edad del postulante a una oferta de trabajo, el Salario al que aspira es lo que pide por realizar ese trabajo y donde contratado indica si el postulante fue contratado o no (Si-No). En ese caso la **edad** es de 2 dígitos y una categoría de **salario** posiblemente de 4, 5 o más dígitos. En otros casos la diferencia puede ser mayor, con lo que el modelo podría otorgar más importancia al salario que la edad por esa razón necesitamos realizar el **escalamiento/transformación** a nuestro conjunto de datos.

Cabe resaltar que si todas las características están dentro de un rango similar entre sí, entonces no hay necesidad real de transformar.



Transformación en R

Librería ``caret`` de R.

Técnicas principales para transformar los datos:

❖ **Autónomo:** Las transformaciones se pueden modelar a partir de los *datos de entrenamiento* y se pueden aplicar a *múltiples conjunto de datos*, se usará lo siguiente:

- ❖ `PreProcess()` : Entrena los datos
- ❖ `predict()` : Aplica la transformación

❖ **Entrenamiento:** Las transformaciones se preparan y aplican automáticamente durante a evaluación del modelo, se usará lo siguiente:

- ❖ `preProcess()` y dentro de `preProces(..., train(),...)`.

Debemos considerar que lo que se va a realizar es para variables o datos numéricos, R omite los datos no numéricos sin error. Existe una probabilidad de que los algoritmos para métodos basados en instancias tales como regresión, métodos basados en instancias (como k-NN y LVQ), Support Vector Machine (SVM) y redes neuronales; y menos útiles para métodos basados en reglas y en árboles, aunque no olvidemos es **empírico**.

Lin para revisar información sobre caret: <https://cran.r-project.org/web/packages/caret/caret.pdf> ,

Link para revisar información sobre Pre-Processing: <http://topepo.github.io/caret/pre-processing.html>



Métodos de Transformación



Analytics AoZ

Veremos los más conocidos métodos de transformación admitidos dentro de la función `preProcess()` de la librería `caret` que se aplica a la data de entrenamiento.

Posibles valores "**Box-Cox**", "**YeoJohnson**", "**expoTrans**", "**center**", "**scale**", "**range**", "**knnImpute**", "**bag-Impute**", "**medianImpute**", "**pca**", "**ica**", "**spatialSign**", "**corr**", "**zv**", "**nzv**", y "**conditionalX**".

- **BoxCox**: aplica una transformada de Box-Cox, los valores deben ser distintos de cero y positivos.
- **YeoJohnson**: aplica una transformada de Yeo-Johnson, como un BoxCox, pero los valores pueden ser negativos o cero.
- **expoTrans**: aplica una transformada de potencia como BoxCox y YeoJohnson.
- **zv**: elimina los atributos con una varianza cero (todos con el mismo valor).
- **nzv**: elimina los atributos con una varianza cercana a cero (cerca del mismo valor).
- **center**: resta la media de los valores.
- **scale**: dividir los valores mediante la desviación estándar.
- **range**: normaliza los valores.
- **pca**: transformación de datos mediante componentes principales.
- **ica**: transformación de datos mediante componentes independientes.
- **spatialSign**: proyecta datos en un círculo unitario.



Escalamiento de Datos



Analytics AoZ

- La transformación `scale()` calcula la desviación estándar de un atributo y divide cada valor por esa desviación estándar. Este escalamiento es útil principalmente para los datos que provienen de una distribución gaussiana lo hace de manera consistente. Los algoritmos de optimización: **Gradient Descending**. Algoritmos ponderan entradas: **Regression** y **Neural Network (NN)**. Usan medidas de distancia: **K-Nearest Neighbor (K-NN)**.

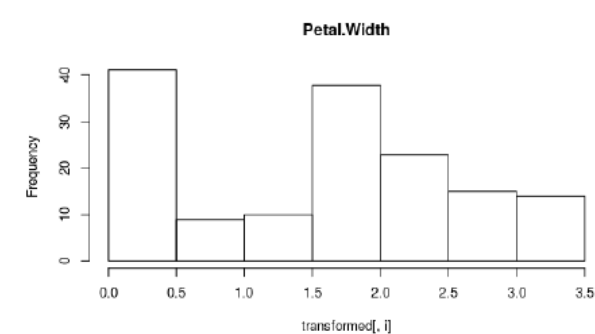
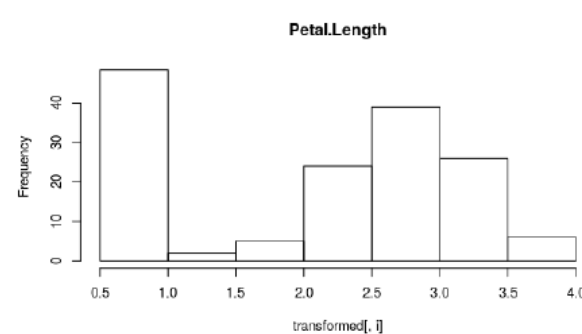
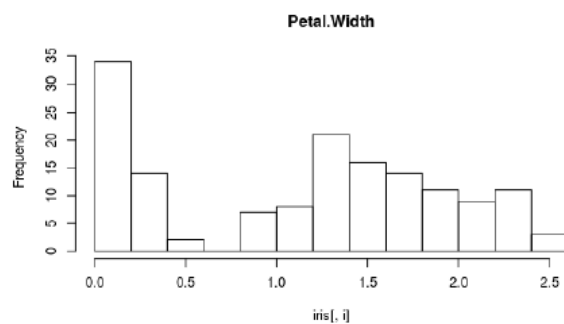
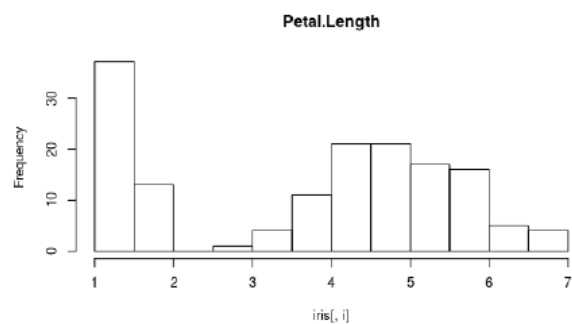
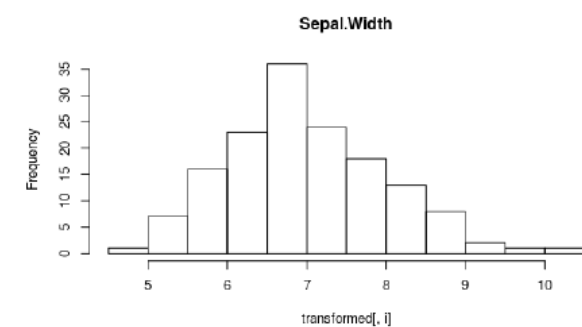
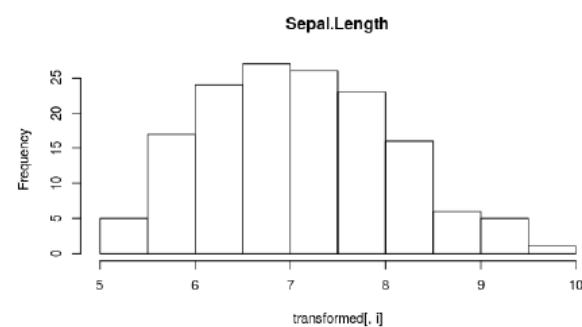
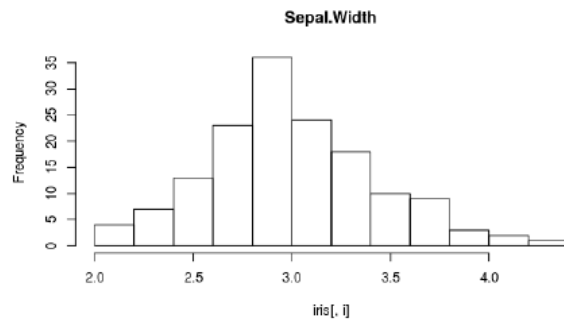
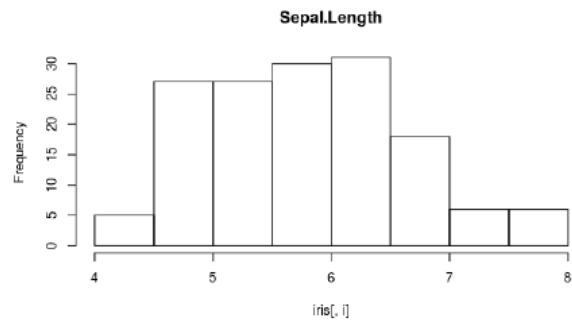
```
1 library(lattice)
2 library(ggplot2)
3 library(caret)
4 data("iris")
5 summary(iris[,1:4])
6 preProcessParams <- preProcess(iris[,1:4], method = c("scale"))
7 print(preProcessParams)
8 transformed <- predict(preProcessParams, iris[,1:4])
9 summary(transformed)
```



Escalamiento de Datos (HISTOGRAMAS)



Analytics AoZ



ANTES

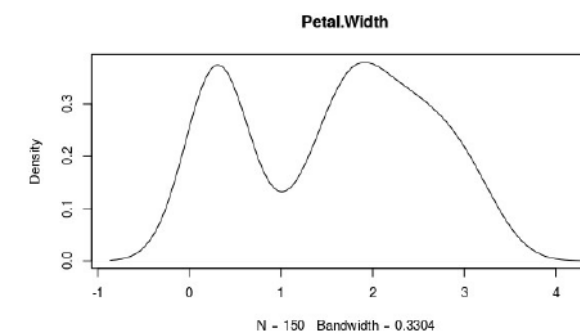
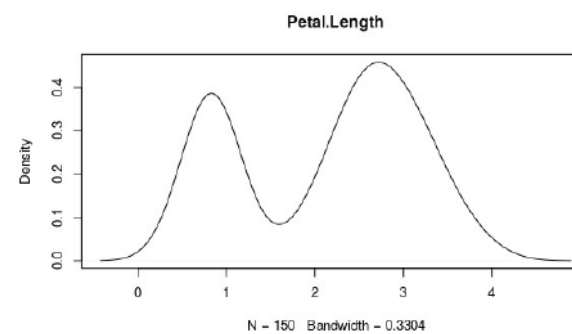
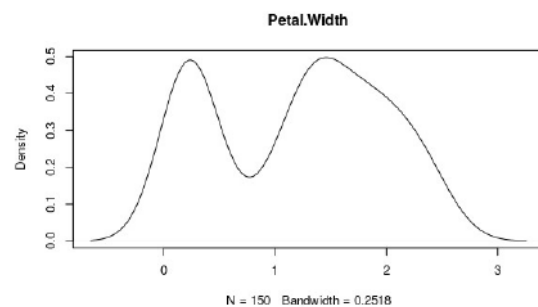
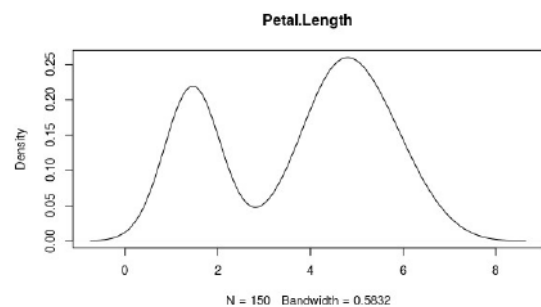
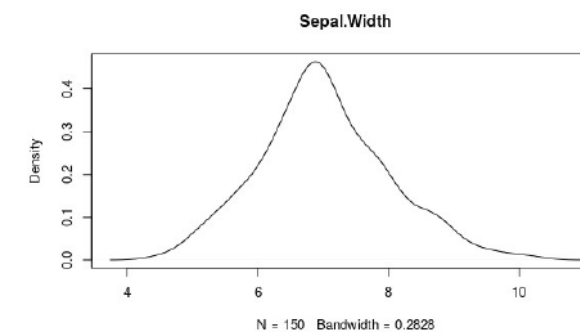
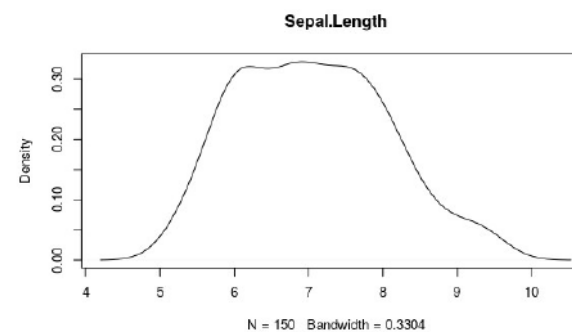
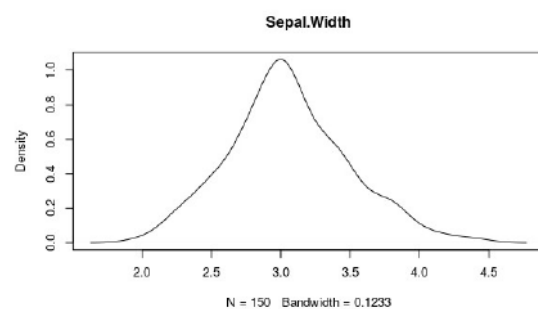
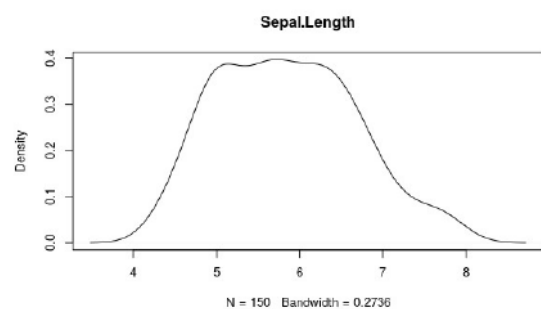
DESPUÉS



Escalamiento de Datos (DENSIDAD)

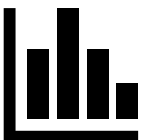


Analytics AoZ



ANTES

DESPUÉS



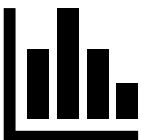
Centrado de Datos



Analytics AoZ

- La transformación ``center`` calcula la media de un atributo y, posteriormente, se lo resta de cada valor.

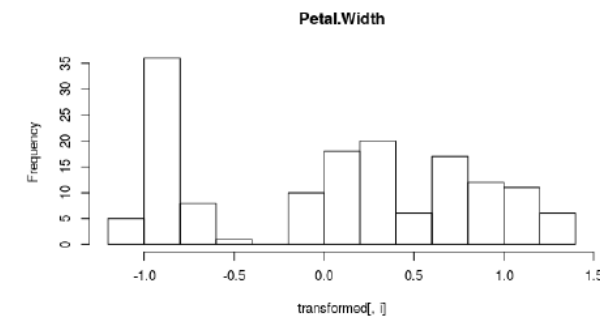
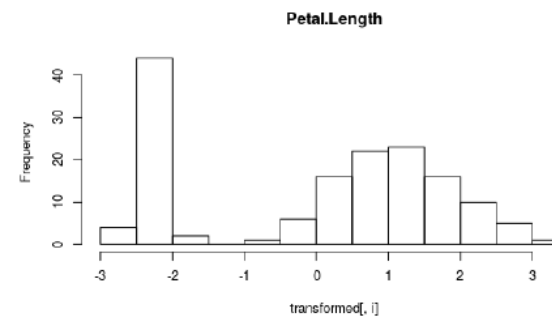
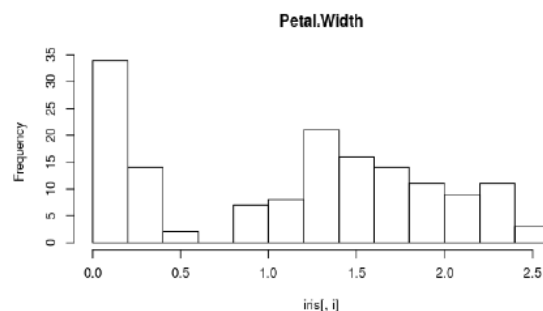
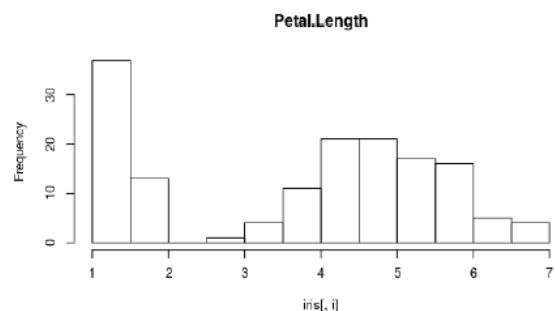
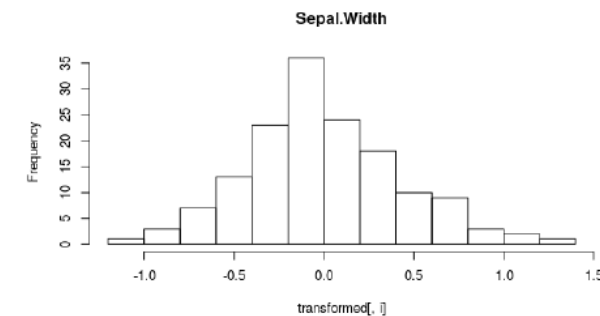
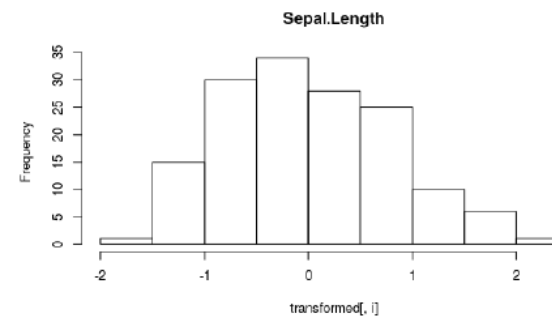
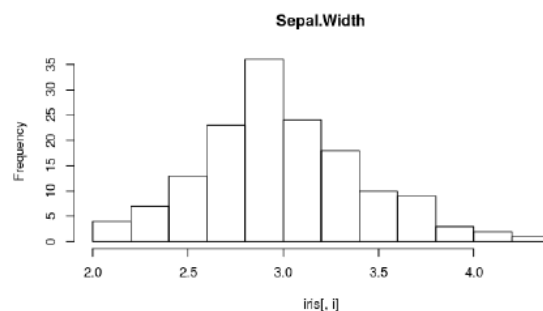
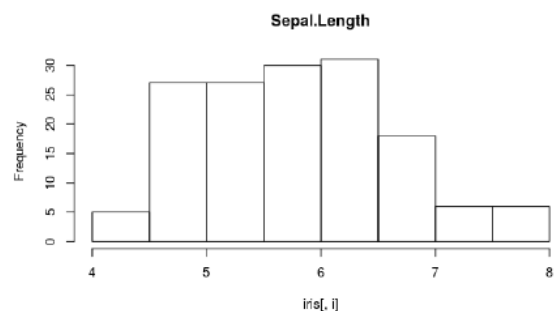
```
25 summary(iris[,1:4])
26
27 preProcessParams <- preProcess(iris[,1:4], method=c("center"))
28 print(preProcessParams)
29 transformed <- predict(preProcessParams, iris[,1:4])
30 summary(transformed)
31 View(transformed)
```



Centrado de Datos (HISTOGRAMAS)



Analytics AoZ



ANTES

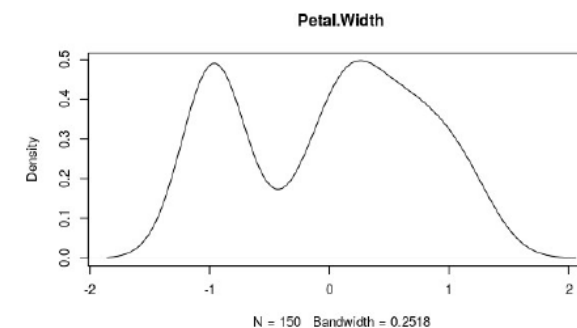
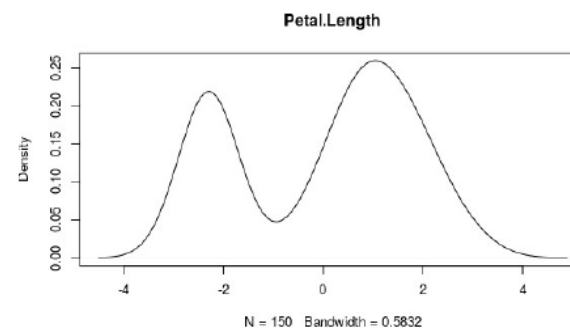
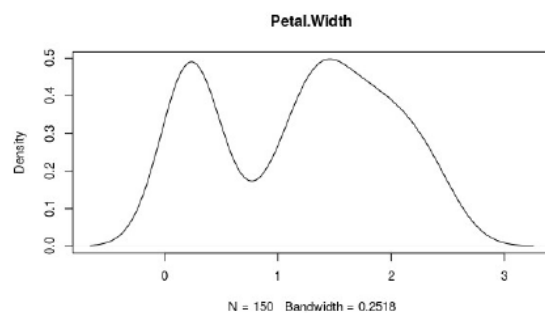
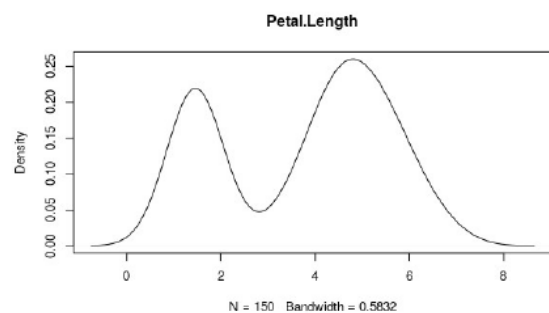
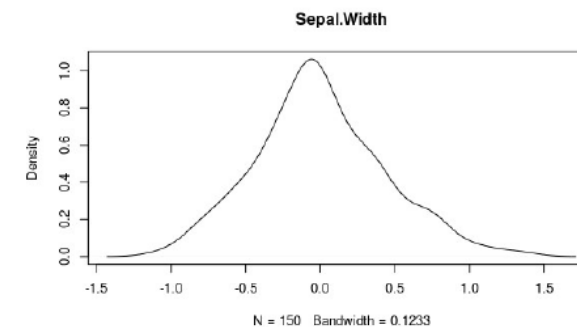
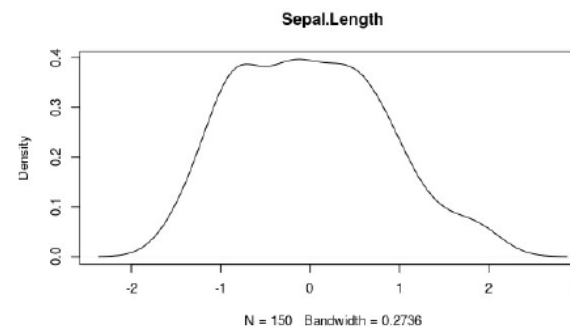
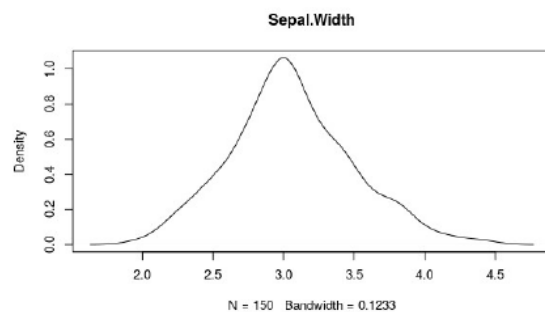
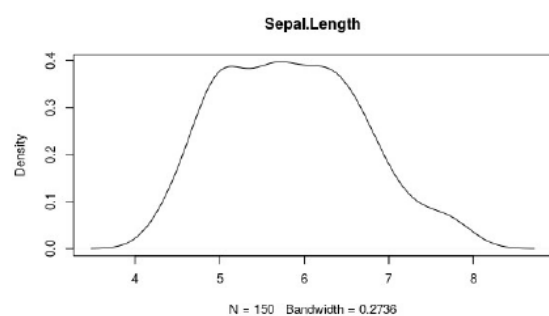
DESPUÉS



Centrado de Datos (DENSIDAD)



Analytics AoZ



ANTES

DESPUÉS



Estandarización de Datos



Analytics AoZ

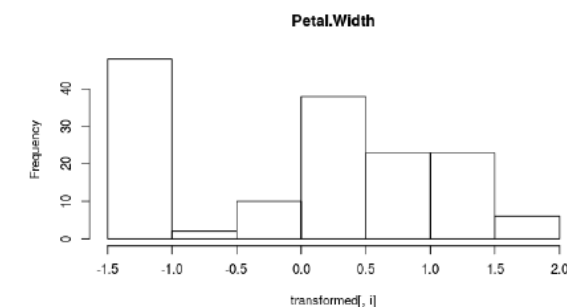
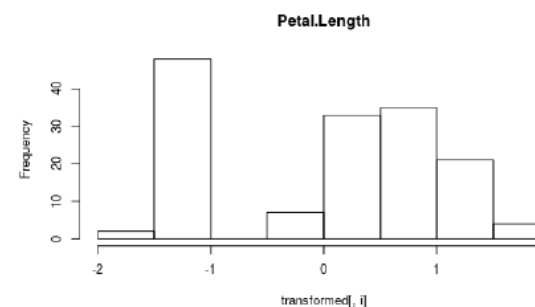
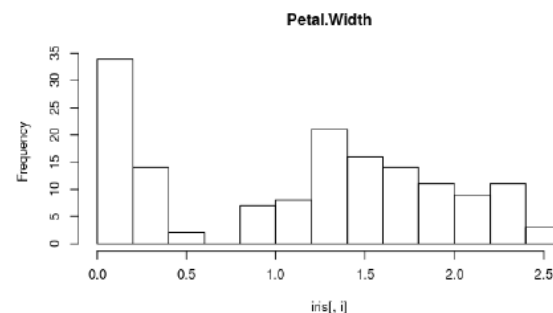
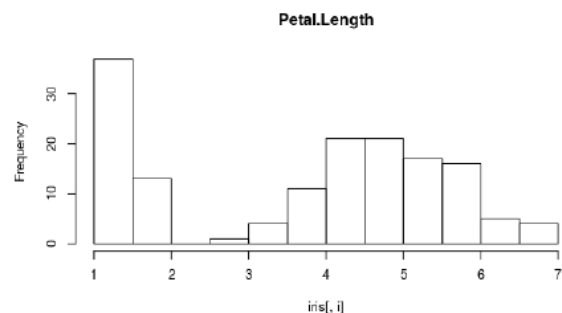
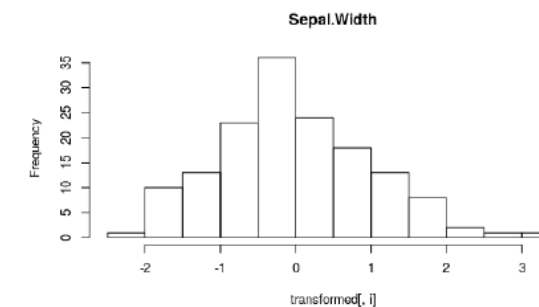
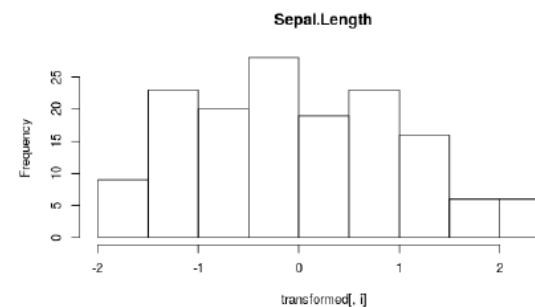
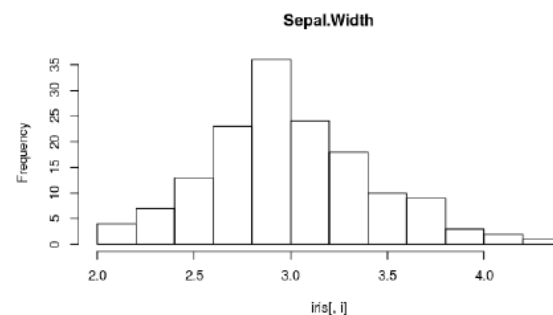
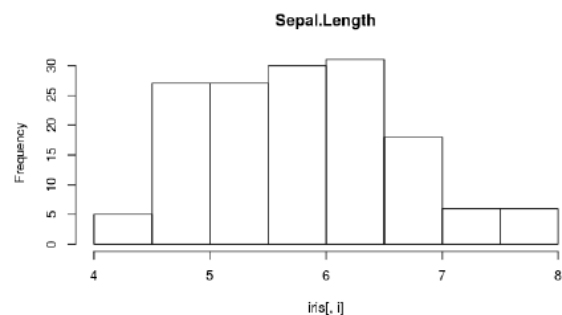
- La transformación ``scale`` y ``center`` realizan la estandarización de nuestros datos. En este caso, los atributos obtendrán un **valor medio de 0** y una **desviación estándar de 1**. Para poder la realizar la estandarización de nuestros datos tenemos que especificar el parámetro ``method`` de la función ``preProcess()`` ambas funciones para la transformación ``train()``. Es bueno para algoritmos **Linear Regression (LiR)**, **Logistic Regression (LoR)** y **Linear Discriminant Analysis (LDA)**, entre otros.

```
34 summary(iris[,1:4])
35 preProcessParams <- preProcess(iris[,1:4], method=c("center", "scale"))
36 print(preProcessParams)
37 transformed <- predict(preProcessParams, iris[,1:4])
38 summary(transformed)
```



Analytics AoZ

Estandarización de Datos (HISTOGRAMAS)



ANTES

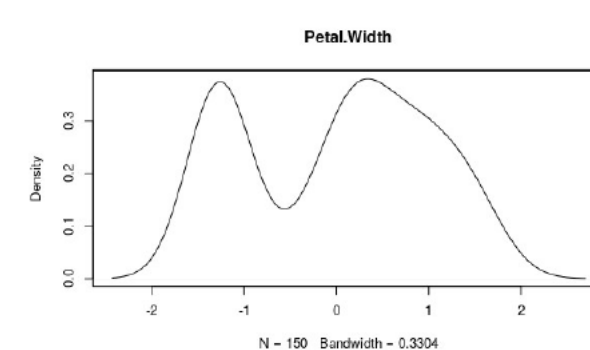
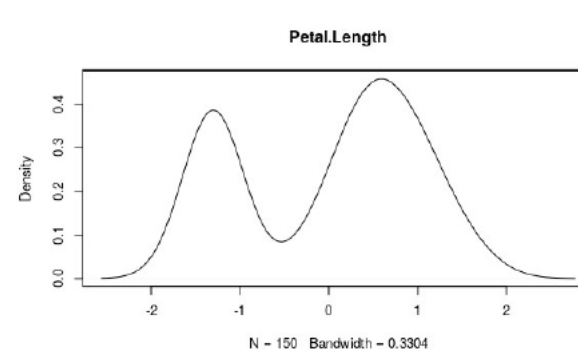
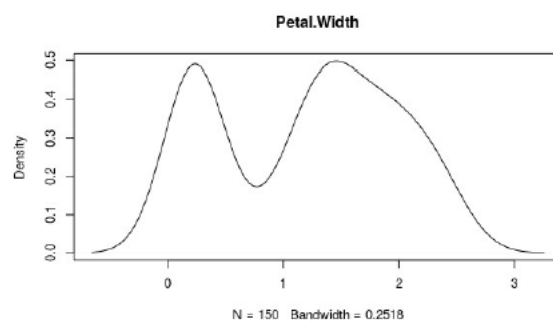
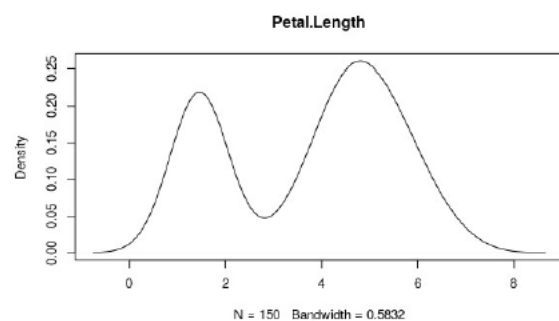
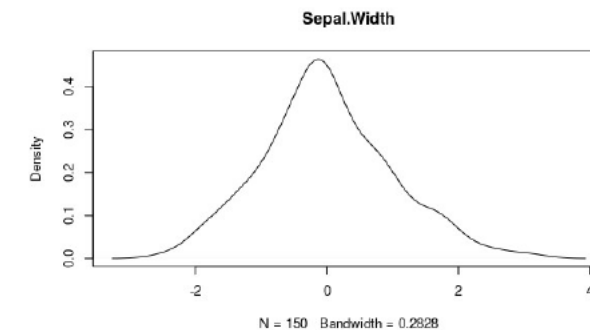
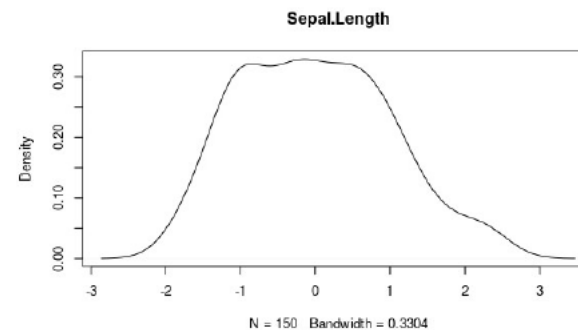
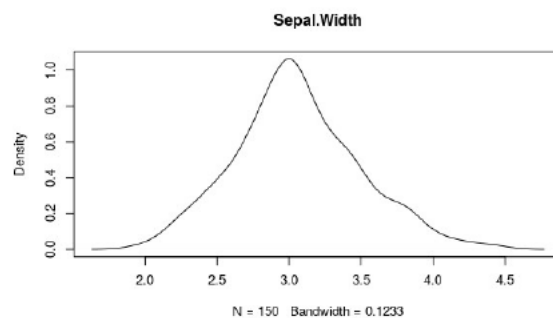
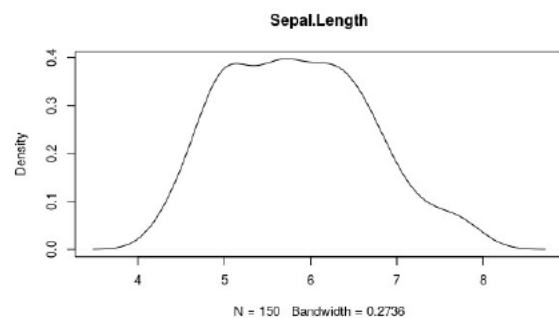
DESPUÉS



Estandarización de Datos (DENSIDAD)



Analytics AoZ



ANTES

DESPUÉS



Normalización de Datos



Analytics AoZ

- Los valores de los datos se pueden escalar en el rango de $[0,1]$ en otras palabras con ***un valor mínimo a 0*** y el ***valor máximo a 1***. A este proceso se le denomina normalización. En general la mayoría de algoritmos se beneficia más de una estandarización que de una normalización. Como ejemplo tenemos los análisis de agrupamiento, la normalización puede ser especialmente crucial al momento de comparar similitudes entre las características en base a ciertas medidas de distancia. Otro ejemplo es en el análisis de componentes principales, en el que generalmente preferimos la estandarización a la escala mínima, ya que estamos interesados en maximizar la varianza (depende de la pregunta y si la PCA calcula los componentes mediante la matriz de correlación en lugar de la matriz de covarianza). La escala **Min-Max** es útil para la aplicación popular del procesamiento de imágenes, donde la intensidad de los pixels deben normalizarse para ajustarse dentro de un cierto rango (es decir, de 0 a 255 para el rango de color RGB). Además los algoritmos de redes neuronales requieren datos en una escala de 0 a 1. Útil en algoritmos **NN y K-NN**.



Normalización de Datos



Analytics AoZ

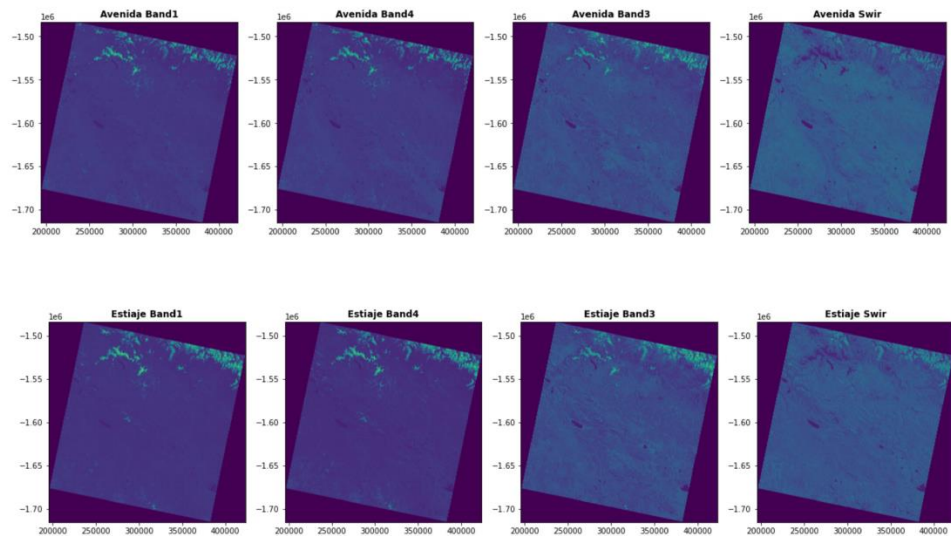


Figure 3: Bandas Obtenidas del Landsat8

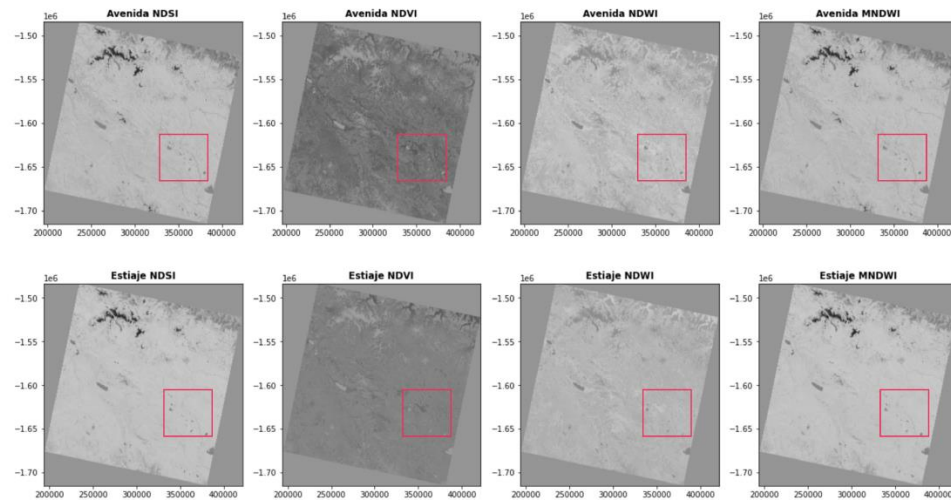


Figure 4: Índices Obtenidos del Procesamiento

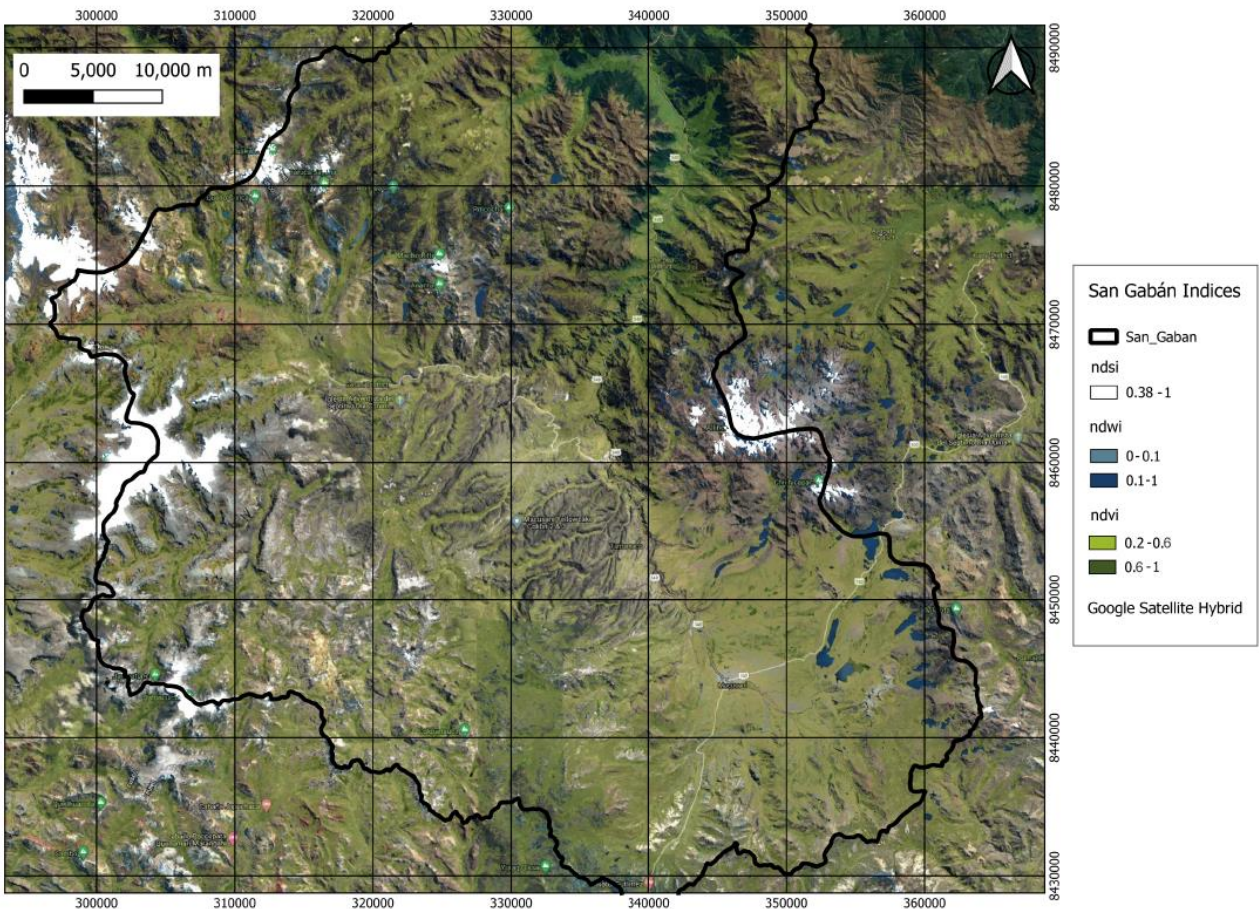
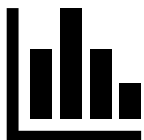


Figure 5: Combinación de Índices para Identificación de aguas, vegetación y nevado - Estiaje



Normalización Código en R

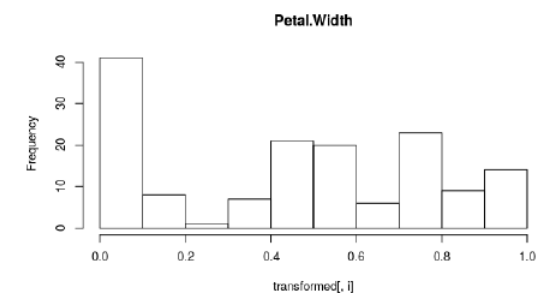
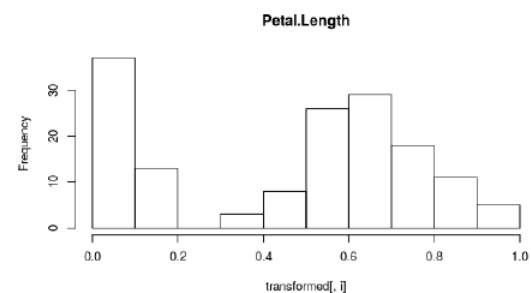
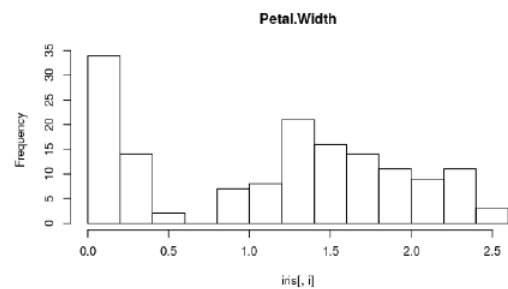
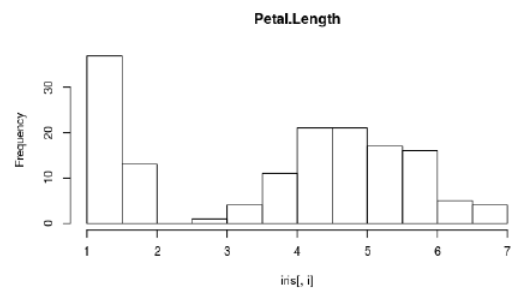
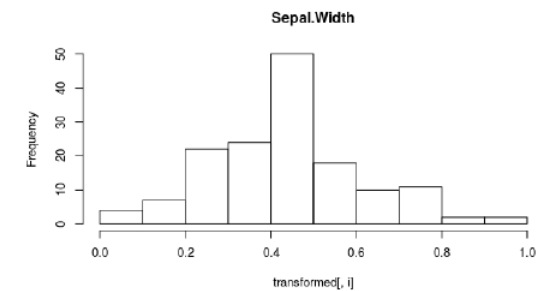
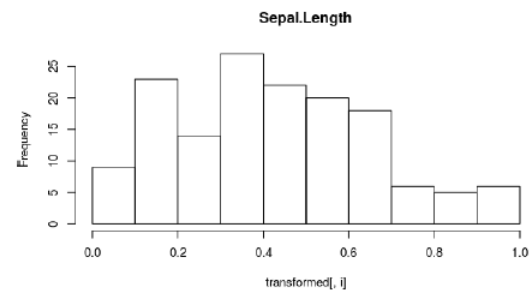
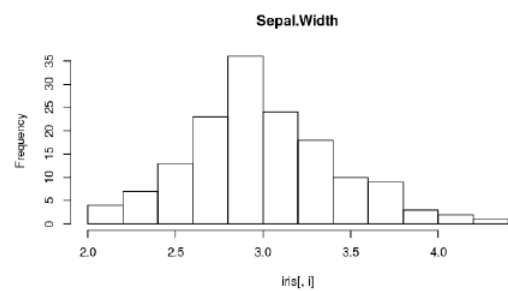
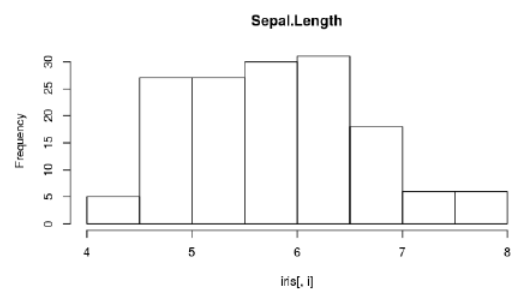
```
42 summary(iris[,1:4])
43 preProcessParams <- preProcess(iris[,1:4], method=c("range"))
44 print(preProcessParams)
45 transformed <- predict(preProcessParams, iris[,1:4])
46 summary(transformed)
```




Normalización de Datos (HISTOGRAMAS)



Analytics AoZ



ANTES

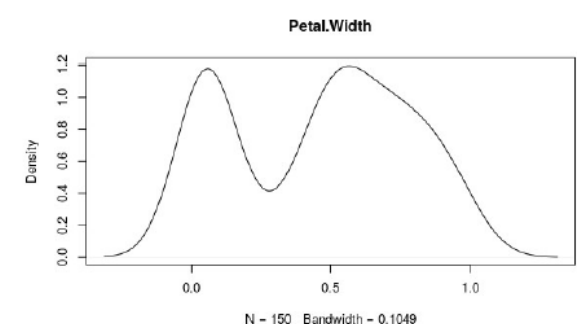
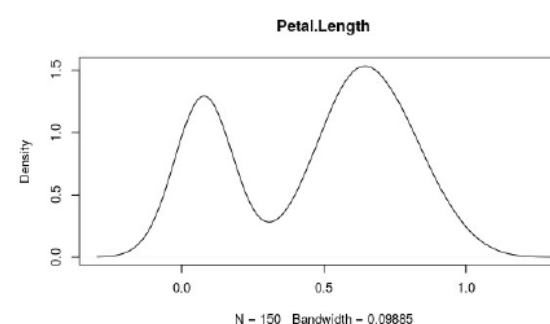
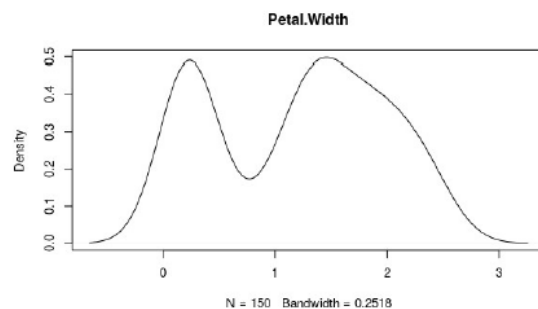
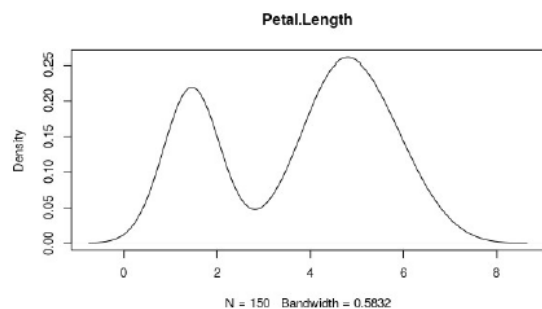
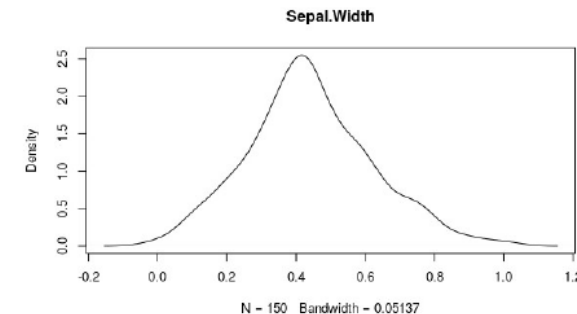
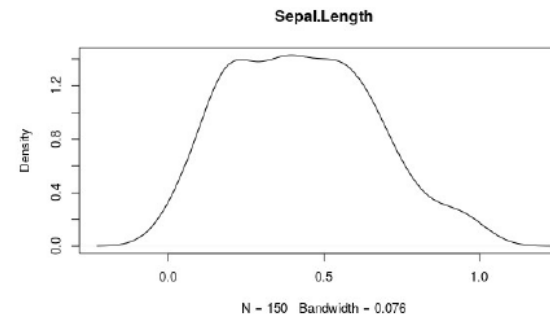
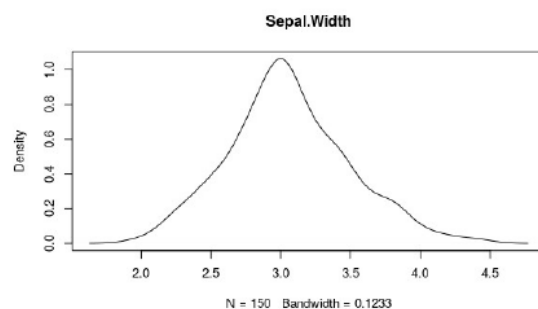
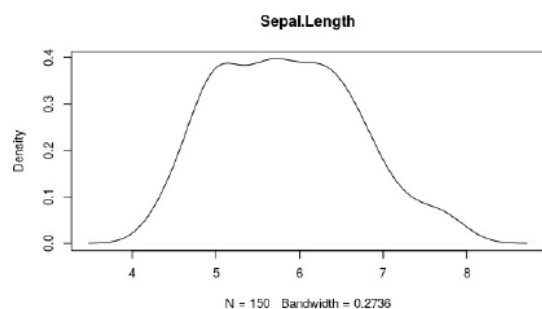
DESPUÉS



Normalización de Datos (DENSIDAD)



Analytics AoZ



ANTES

DESPUÉS



Transformación Box-Cox



Analytics AoZ

- La transformación Box-Cox son una familia de transformaciones potenciales usadas en estadística para corregir sesgos en la distribución de errores, para corregir varianzas desiguales (para diferentes valores de la variable predictora) y principalmente para corregir la no linealidad en la relación (mejora la correlación lineal entre variables). En muchos casos los atributos tienen sesgo o inclinación, que es una distribución Gaussiana pero desplazada. Para estos casos se usa la transformación Box-Cox. Se aplicará por recomendación en el modelado solo a atributos que tienen sesgo. Box-Cox asume todos los atributos positivos.

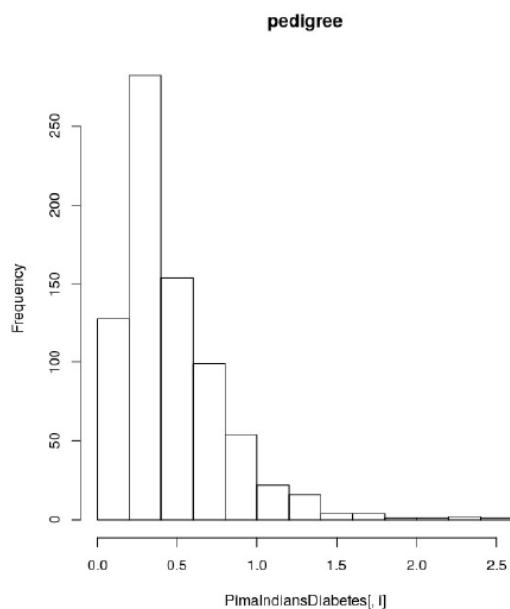
```
49 library(mlbench)
50 data("PimaIndiansDiabetes")
51 summary(PimaIndiansDiabetes[,7:8])
52 preProcessParams <- preProcess(PimaIndiansDiabetes[,7:8],
53                               method=c("BoxCox"))
54 print(preProcessParams)
55 transformed <- predict(preProcessParams, PimaIndiansDiabetes[,7:8])
56 summary(transformed)
```



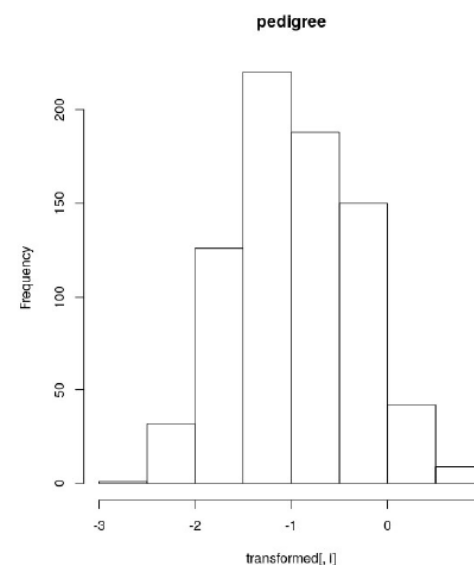
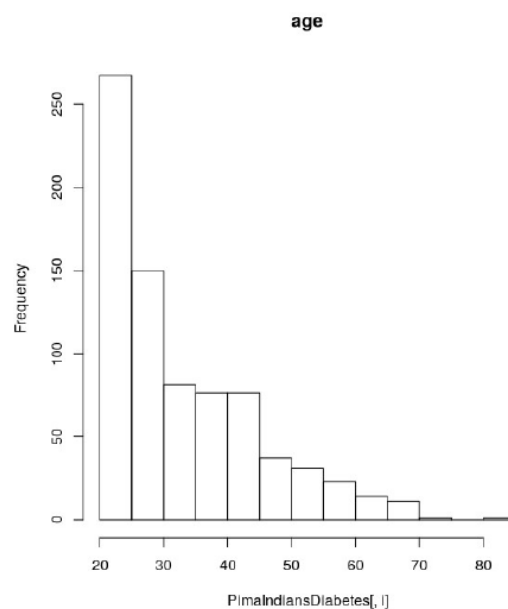
Transformación Box-Cox de Datos (HISTOGRAMAS)



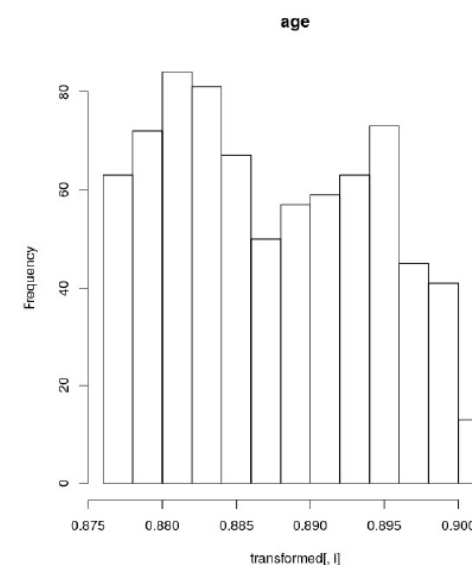
Analytics AoZ



ANTES



DESPUÉS

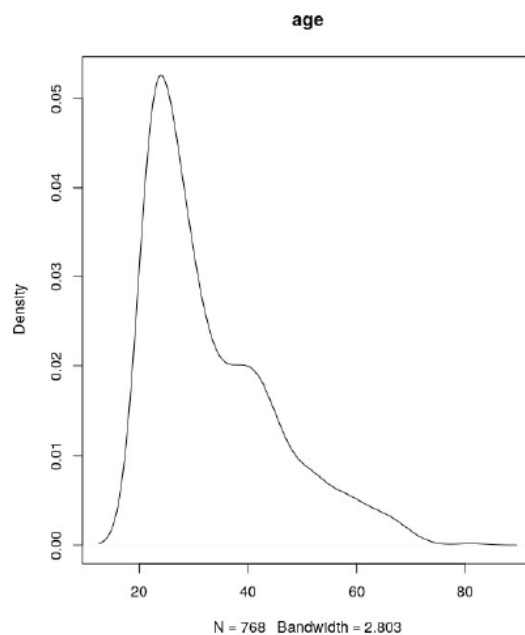
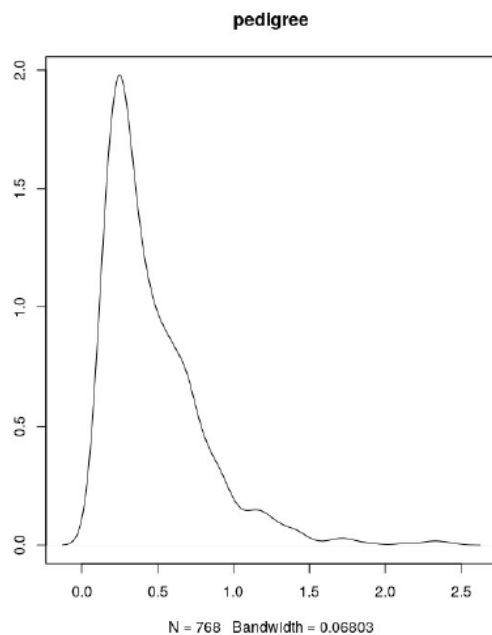




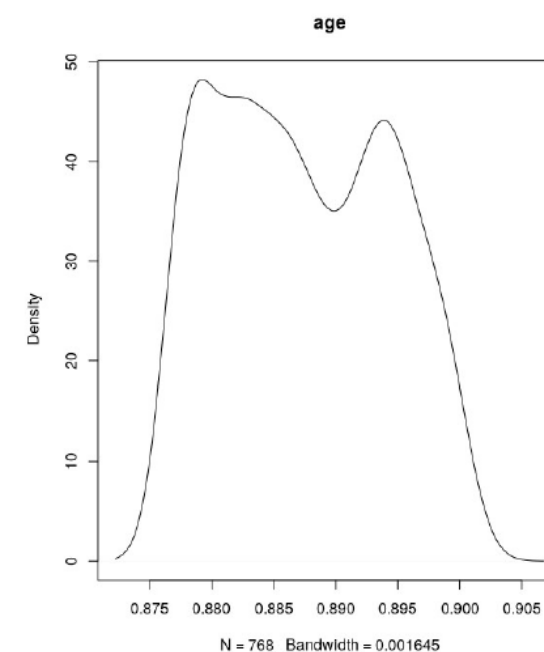
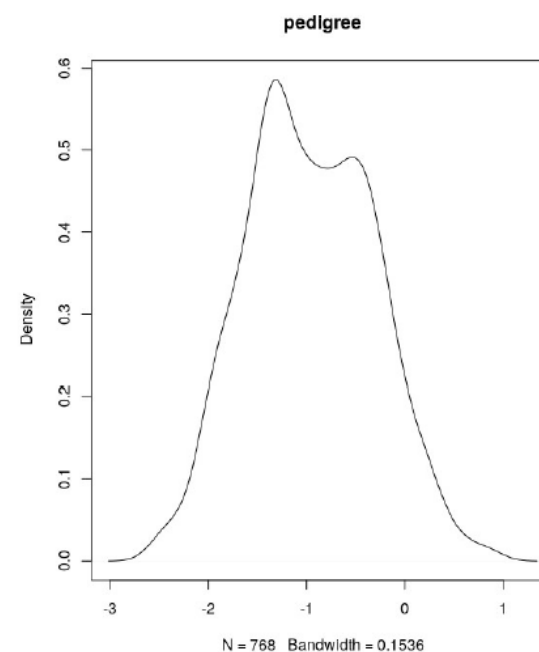
Transformación Box-Cox de Datos (DENSIDAD)



Analytics AoZ



ANTES



DESPUÉS



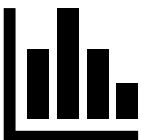
Transformación Yeo-Jhonson



Analytics AoZ

- La transformación Yeo-Jhonson es otra transformación de potencia como Box-Cox, pero soporta valores en bruto (sin procesar) que son iguales a cero o negativos.

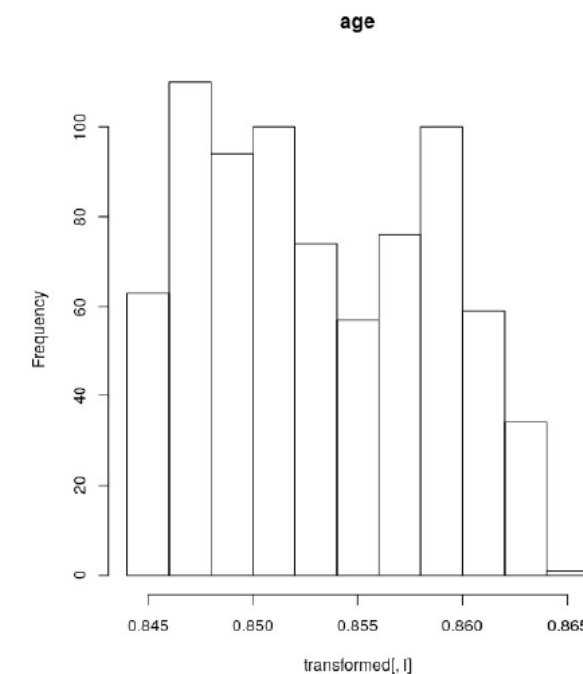
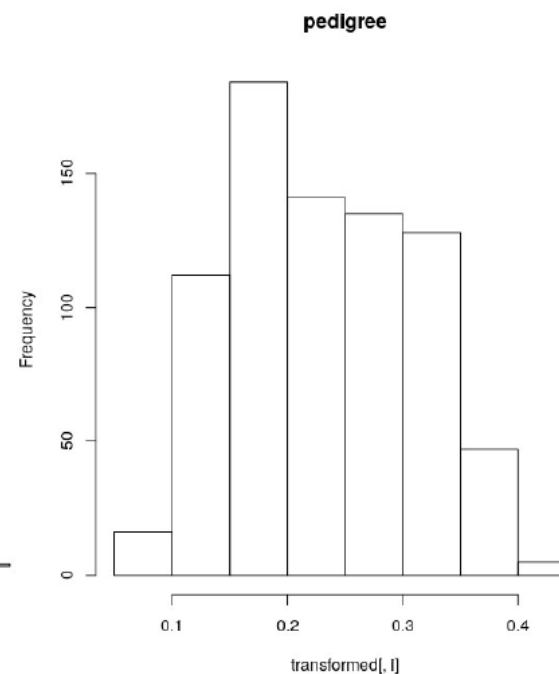
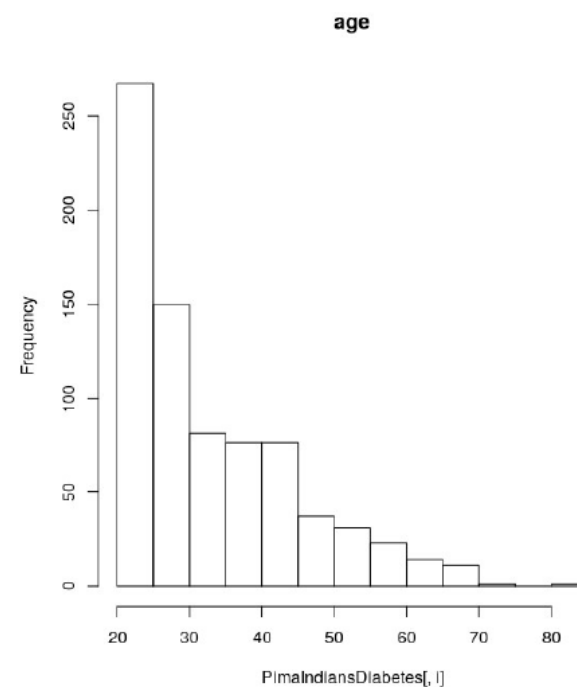
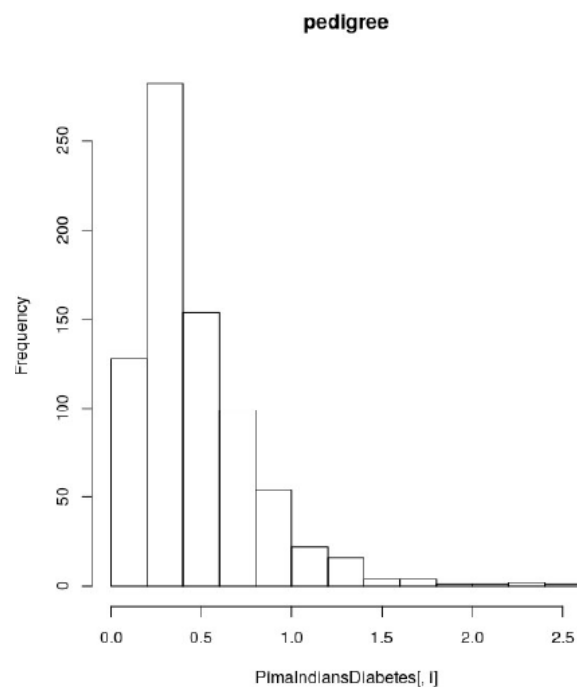
```
68 data("PimaIndiansDiabetes")
69 summary(PimaIndiansDiabetes[,7:8])
70 preProcessParams <- preProcess(PimaIndiansDiabetes[,7:8],
71                                method=c("YeoJohnson"))
72 print(preProcessParams)
73 transformed <- predict(preProcessParams, PimaIndiansDiabetes[,7:8])
74 summary(transformed)
```



Transformación Yeo-Jhonson de Datos (HISTOGRAMAS)



Analytics AoZ



ANTES

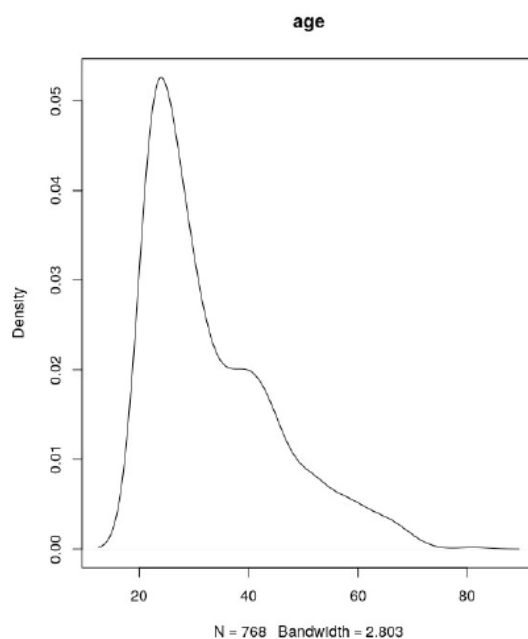
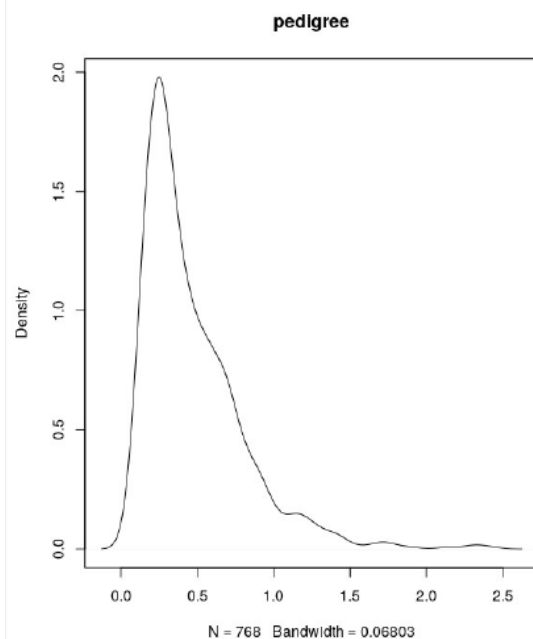
DESPUÉS



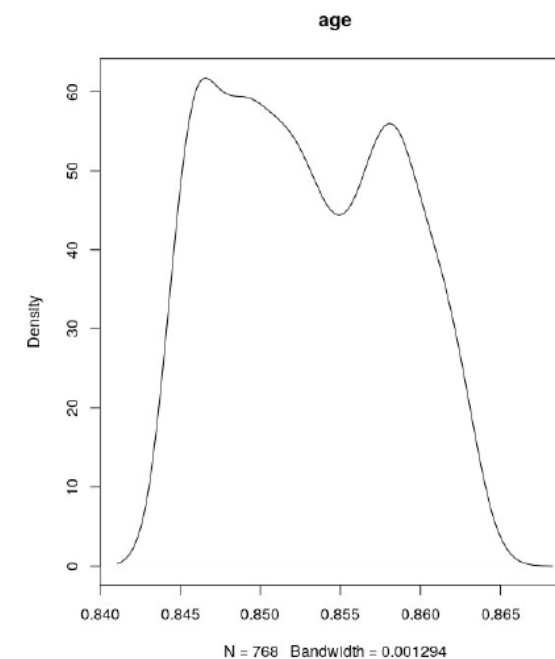
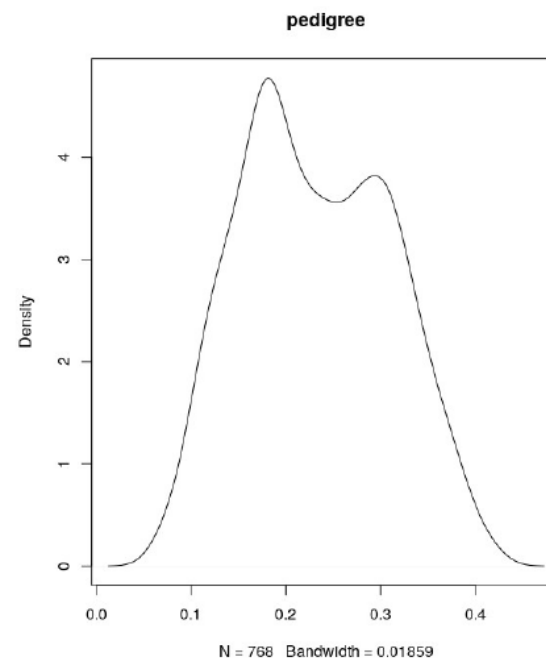
Transformación Yeo-Jhonson de Datos (DENSIDAD)



Analytics AoZ



ANTES



DESPUÉS



PCA (Principal Component Analysis)



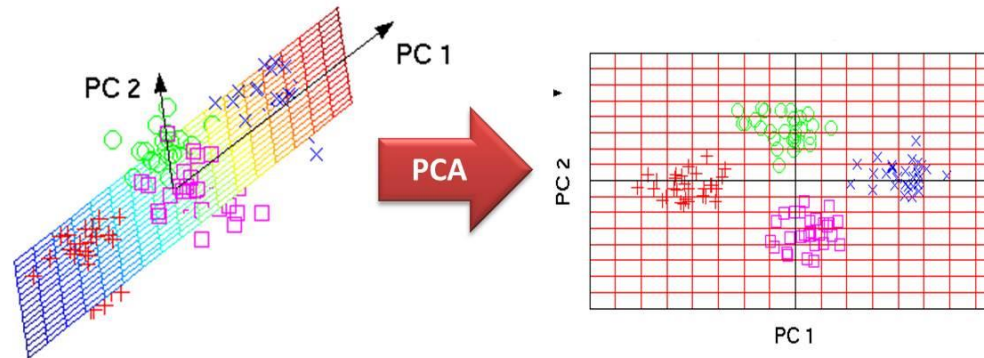
Analytics AoZ

- Busca explotar la estructura existente de manera no supervisada para simplificar los datos y reducirlos o comprimirlos. Son útiles para visualizar datos o para simplificar un conjunto de variables que luego puede usar el algoritmo supervisado. El PCA es un procedimiento estadístico que utiliza una transformación ortogonal que convierte el conjunto de variables correlacionada en un conjunto de variables no correlacionadas. El PCA identifica pautas en nuestros datos basados en las correlaciones entre características. Esta correlación implica redundancia en nuestros datos, en otras palabras, que hay parte de los datos que se pueden explicar por relaciones con otras partes de los mismos. Estos datos correlacionados no son necesarios para el aprendizaje correcto del modelo, y por tanto pueden ser eliminados. Se puede eliminar borrando directamente ciertas columnas (características) o combinando un número de ellos y obteniendo nuevos datos que contengan la mayoría de la información.
- Eliminar problema de dimensionalidad y overfitting.



- El PCA transforma los datos para devolver sólo los componentes principales, una técnica de estadística multivariada y álgebra lineal. La transformación mantiene esos componente por encima del umbral de varianza (valor predeterminado = 0.95) o se puede especificar el número de componentes (`pcaComp`). El resultados son atributos no correlacionados, útiles para algoritmos como *Linear Regression* y *Generalized Linear Regression*, entre otros.

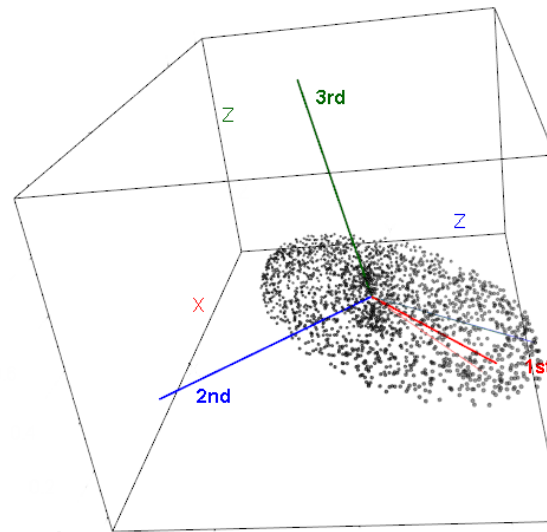
Dimensionality Reduction & Principal Component Analysis





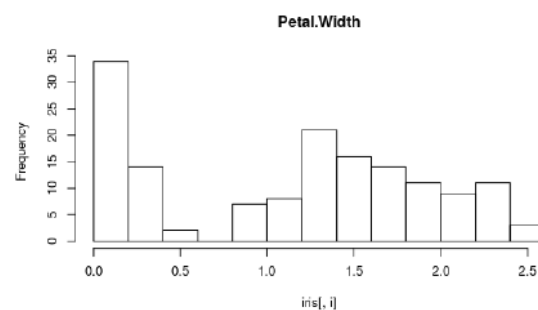
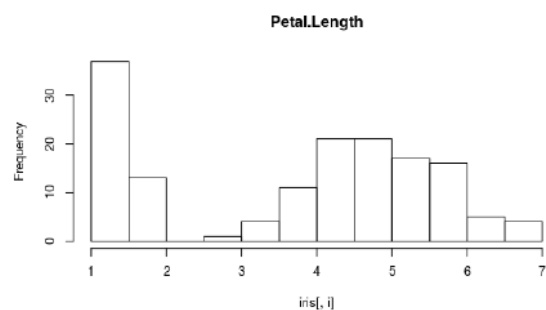
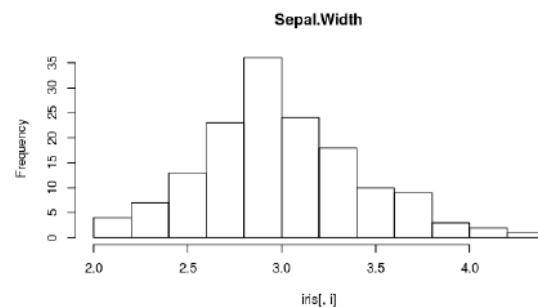
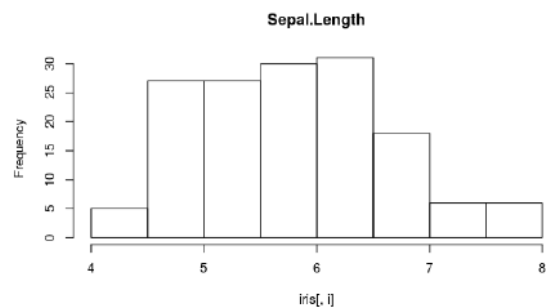
```
88 data("iris")
89 summary(iris)
90 preProcessParams <- preProcess(iris, method=c("center", "scale", "pca"))
91 print(preProcessParams)
92 transformed <- predict(preProcessParams, iris)
93 summary(transformed)
```

PCA applied to an ellipsoidally shaped point cloud

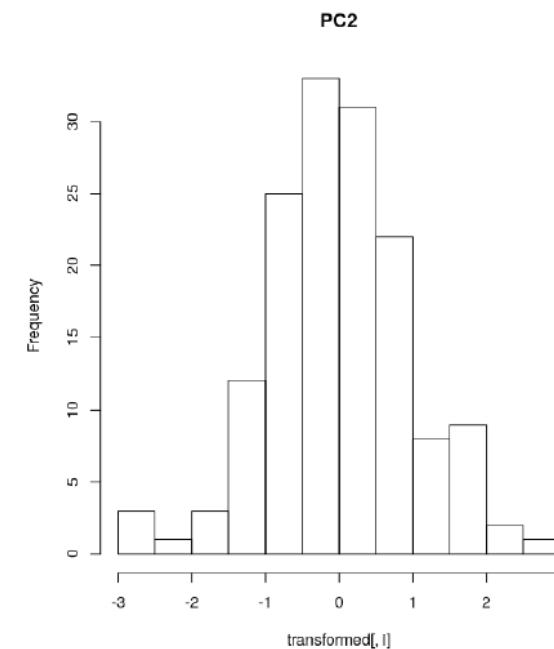
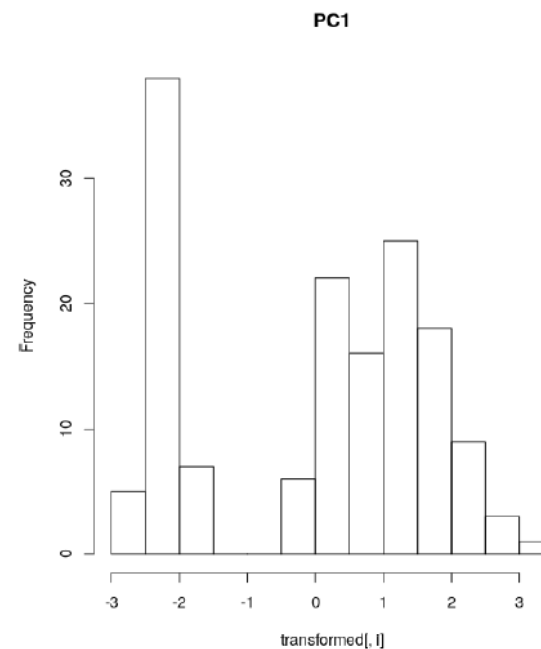




PCA (HISTOGRAMAS)



ANTES



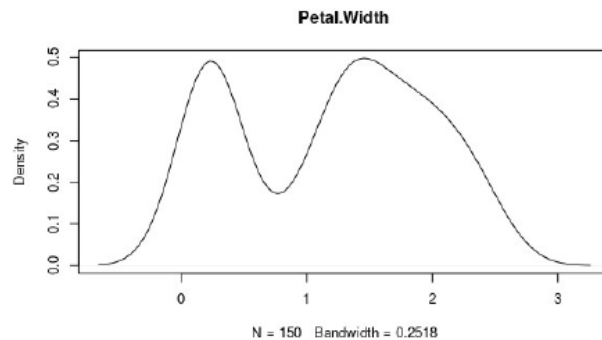
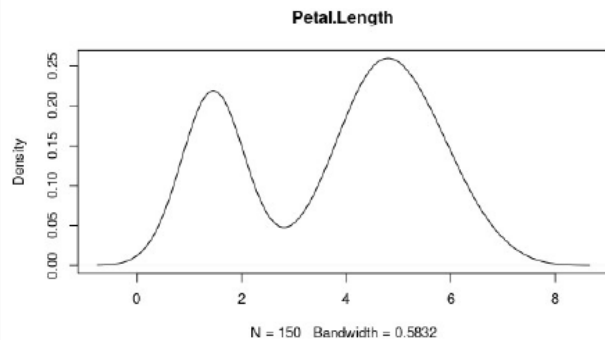
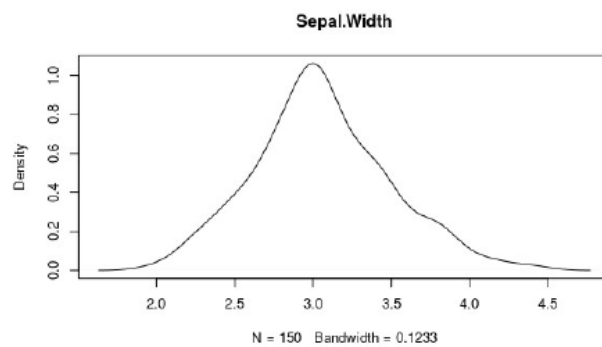
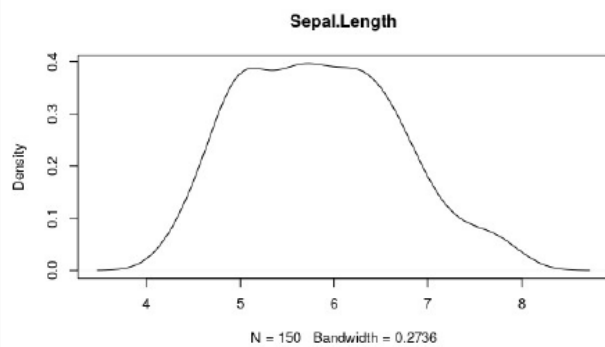
DESPUÉS



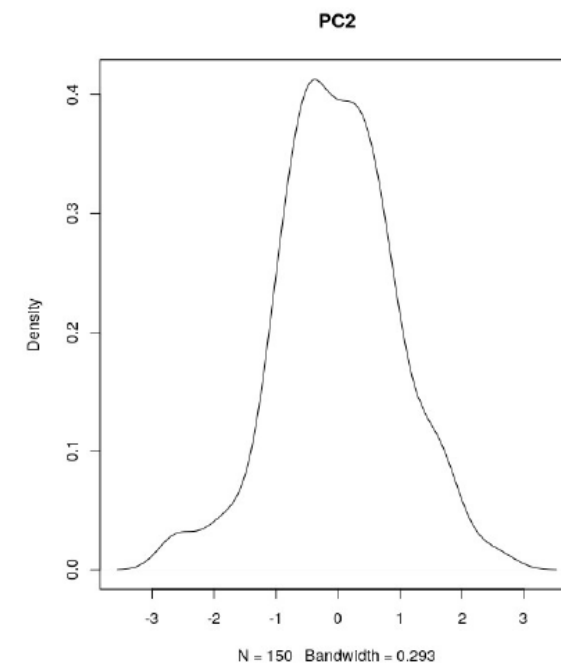
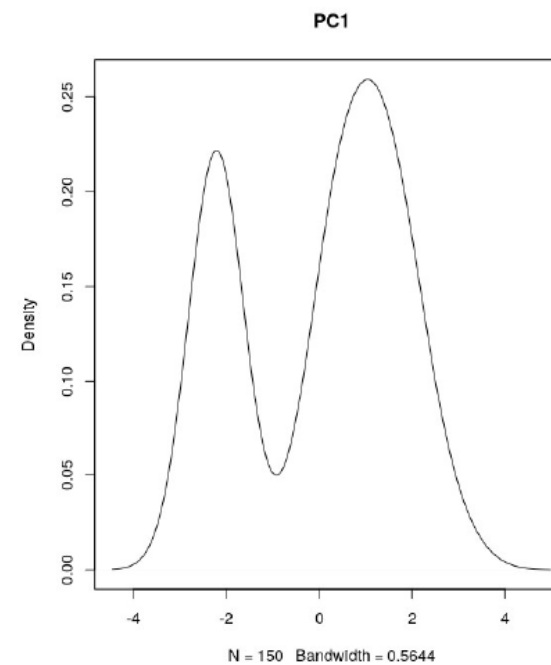
PCA (DENSIDAD)



Analytics AoZ



ANTES



DESPUÉS



ICA (Independent Component Analysis)



Analytics AoZ

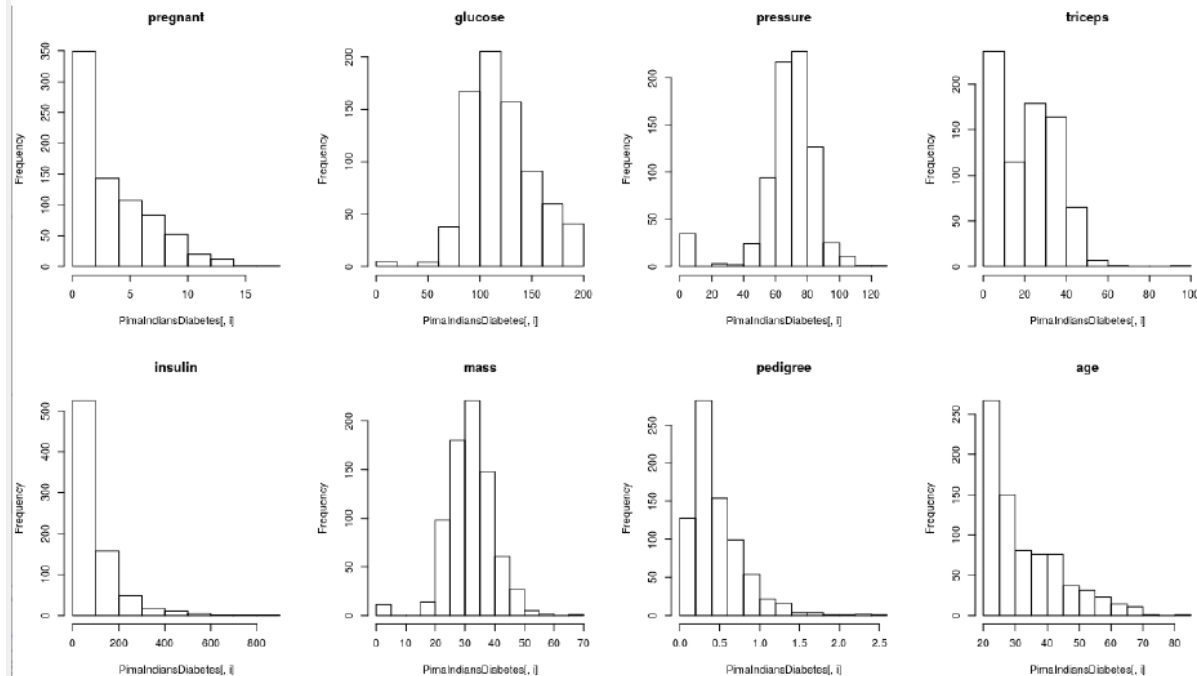
- El ICA es un método computacional que sirve para separar una señal multivariante en subcomponentes aditivos suponiendo que la señal de origen tienen una **independencia estadística y es no-Gaussiana**. En este contexto, ICA es una generalización de PCA, en ambos casos se practica una transformación lineal de los datos originales, aunque la diferencia básica es que el ICA no requiere que las variables tengan una distribución gaussiana. ICA conserva aquellos componentes que son independientes. Debe especificar el número de componentes independientes deseados con el argumento `n.comp`. Esta transformación puede ser útil para algoritmos como *Naive Bayes*.

```
97 data(PimaIndiansDiabetes)
98 summary(PimaIndiansDiabetes[,1:8])
99 preProcessParams <- preProcess(PimaIndiansDiabetes[,1:8],
100                                method=c("center", "scale", "ica"), n.comp=5)
101 transformed <- predict(preProcessParams, PimaIndiansDiabetes[,1:8])
102 summary(transformed)
```

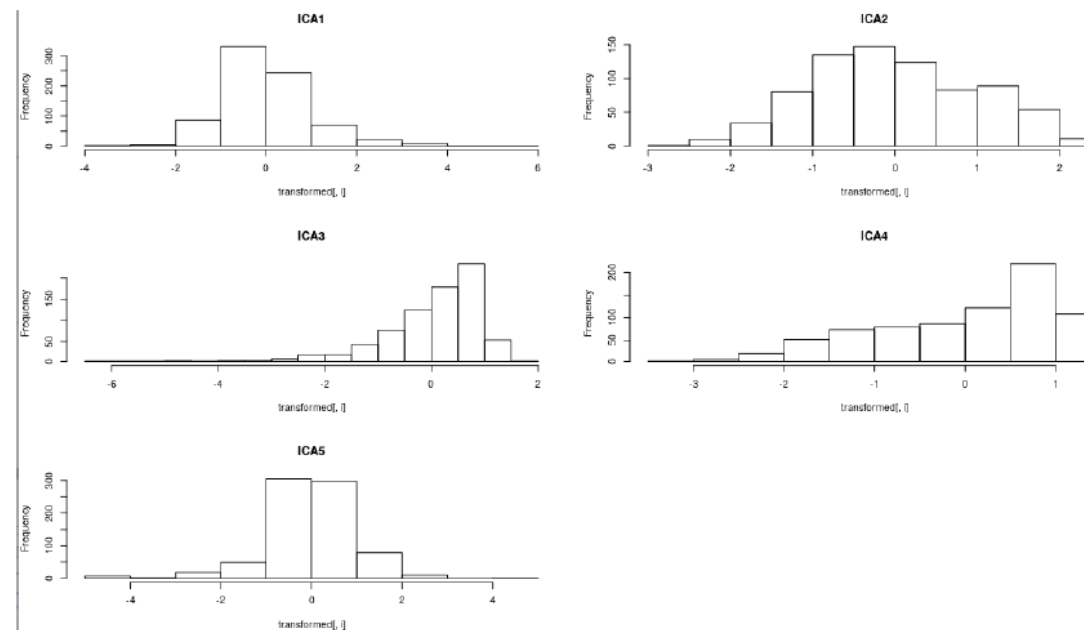



Analytics AoZ

ICA (HISTOGRAMAS)



ANTES

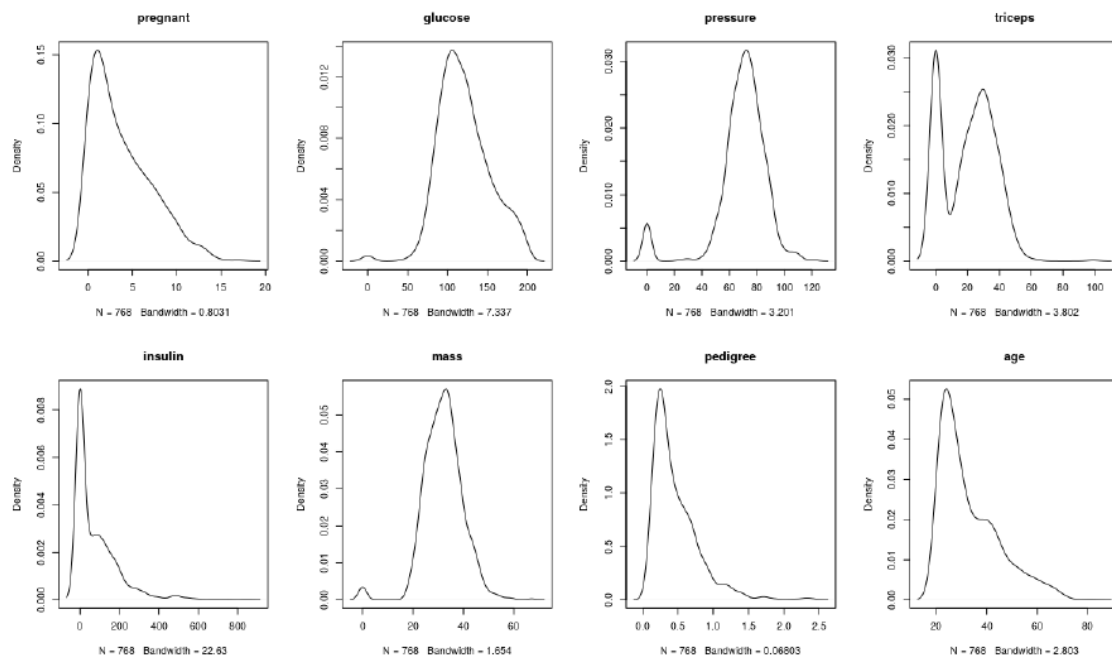


DESPUÉS

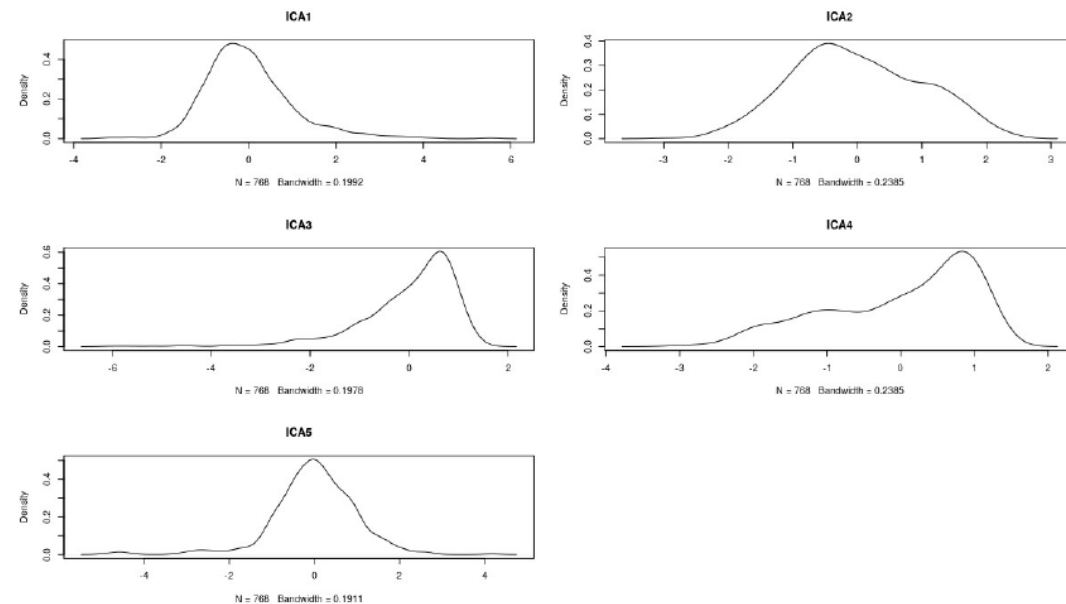


Analytics AoZ

ICA (DENSIDAD)



ANTES



DESPUÉS



Consejos

- Las transformaciones enseñadas en general tienen un gran impacto en la precisión de modelos finales por eso ***debemos usarlas***.
- En los problemas de “modelado” debemos probar diferentes transformaciones de los datos con diferentes algoritmos de machine learning por eso ***usemos en variedad***.
- Es una recomendación muy importante resumir y visualizar los datos ***antes y después*** de aplicar una transformación lo podemos hacer numérica o gráficamente (*summary()*, *hist()*, *density()*).

BIBLIOGRAFÍA:



Analytics AoZ

Preprocesado de datos:

<https://www.diegocalvo.es/preprocesado-de-datos-en-r/>

Tratamiento de NA:

<https://www.guru99.com/r-replace-missing-values.html>

PCA:

<https://www.aprendemachinelearning.com/comprende-principal-component-analysis/>

<https://towardsdatascience.com/principal-component-analysis-for-dimensionality-reduction-115a3d157bad>

<https://medium.com/towards-artificial-intelligence/machine-learning-dimensionality-reduction-via-principal-component-analysis-1bdc77462831>

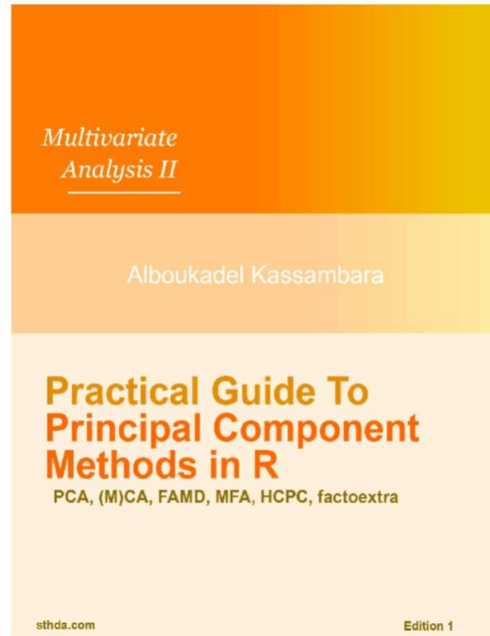
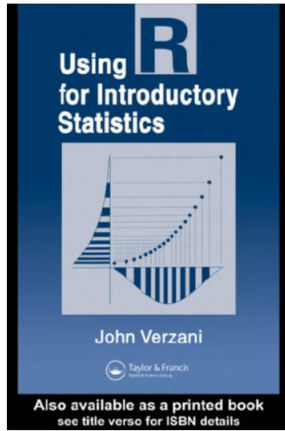
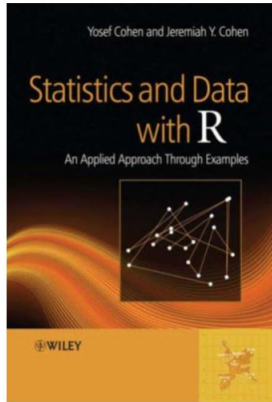
Curso de R con ejemplos:

<http://www.estadistica-dma.ulpgc.es/cursoR4ULPGC/>

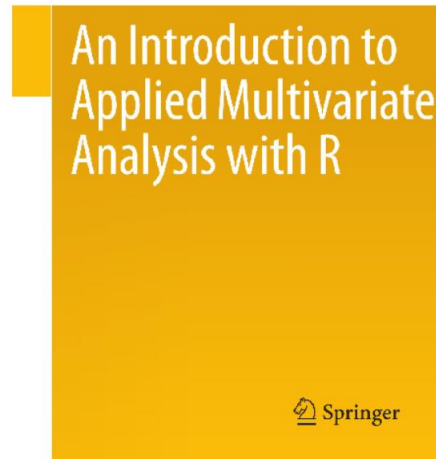
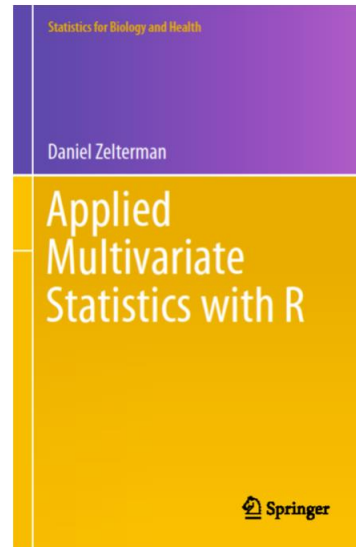
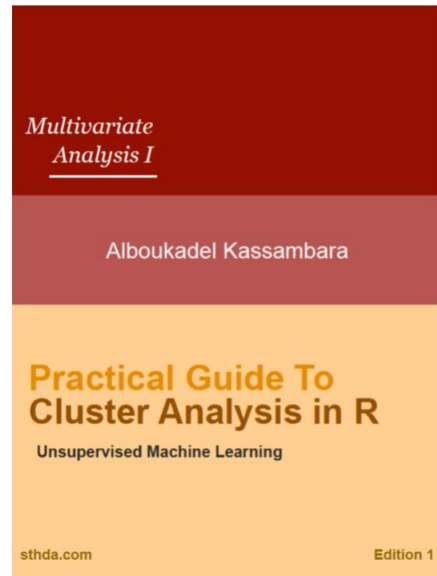
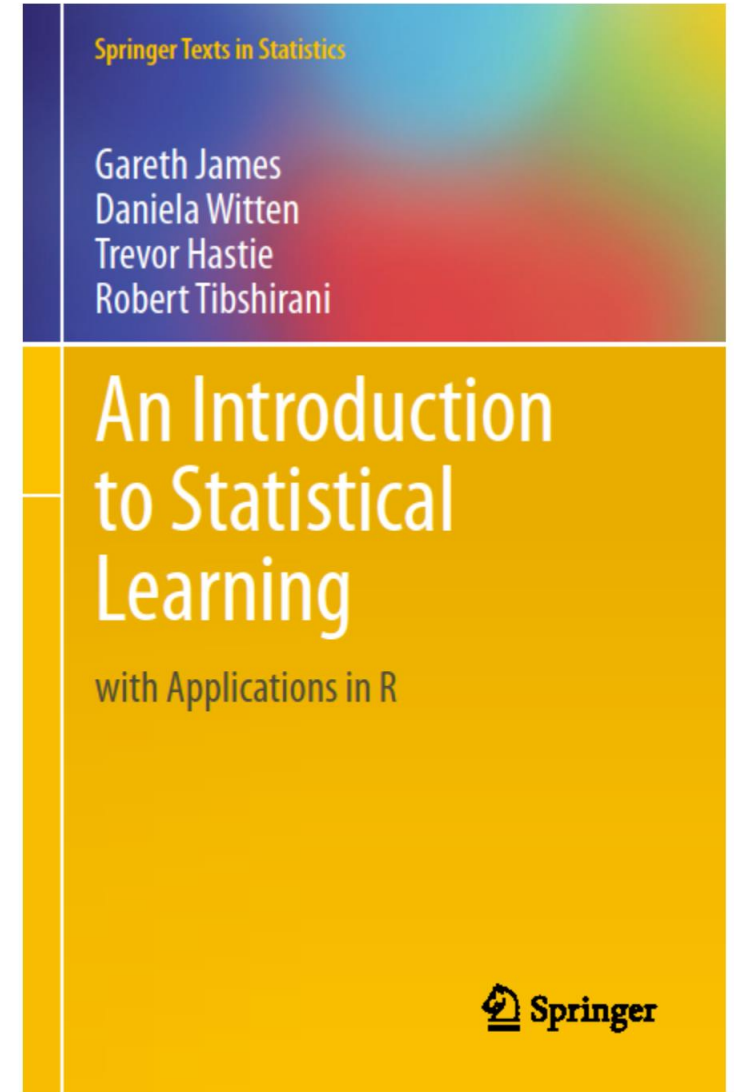
Cluster:



Cluster.txt



Use R!





Practical Methods for Data Analysis (US EPA QA/G-9, 2000)

Helsel, D. R., & Hirsch, R. M. (2002). *Statistical methods in water resources* (Vol. 323). Reston, VA: US Geological Survey.

Salvador Figu

eras, M y Gargallo, P. (2003): "Análisis Exploratorio de Datos, 5campus.com, Estadística <<http://www.5campus.com/leccion/aed>>

Ramalle-Gómara, E., & De Llano, J. A. (2003). Utilización de métodos robustos en la estadística inferencial. *Atención Primaria*, 32(3), 177-182.

Verzani, J. (2005). *Using R for introductory statistics*. CRC press.

Cohen, Y., & Cohen, J. Y. (2008). *Statistics and Data with R: An applied approach through examples*. John Wiley & Sons.

Arnaldo Mangeaud (2014). *Estadística aplicada a las Ciencias Geológicas*. Universidad nacional de Córdoba.

Helsel, D.R., Hirsch, R.M., Ryberg, K.R., Archfield, S.A., and Gilroy, E.J., 2020, *Statistical methods in water resources: U.S. Geological Survey Techniques and Methods*, book 4, chapter A3, 458 p.



**“LO QUE ESCUCHO LO OLVIDO. LO
QUE VEO LO RECUERDO. PERO
LO QUE HAGO, LO ENTIENDO.”
AUTOR: ANÓNIMO**

VAMOS A RESOLVER UN EJERCICIO

