****

**UNIVERSIDAD TÉCNICA DE MACHALA**

Maestría en Software

**Asignatura:**

Ciencia de Datos e Inteligencia de Negocios

**Tema:**

## Tarea Práctica 4. Técnicas de Minería de Datos

Docente: Ing. Bertha Mazón, Mg. Inf.

Estudiantes:

ESTEBAN FABRICIO GONZABAY JIMENEZ

JIMMY FERNANDO CASTILLO CRESPÍN

CESAR DAVID SANTILLAN VILLOTA

JORGE LUIS MIRANDA GALLEGOS

2021-2022

**¿Que es un archivo “csv”?**

De las siglas Comma-Separated values, es un archivo de texto que almacena los datos en forma de columnas, separadas por coma y las filas se distinguen por saltos de línea.

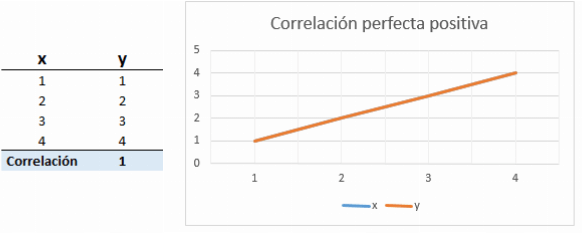
Estos archivos normalmente se usan para importar o exportar de base de datos de unas aplicaciones.

**Libreria “Readr”**

Readr es una componente del denominado “tidyverso”, un conjunto de librerías que todo usuario de R debería si no dominar, al menos conocer, para así resolver ciertas situaciones de la manera más sencilla posible.

**Análisis de Correlación**

El análisis de correlación es un enfoque estadístico que se utiliza para determinar la relación entre variables cuantitativas o categóricas.

****

**Packages (“corrplot”)**

El paquete corrplot es una representación gráfica de una matriz de una correlación, intervalo de confianza. También contiene algunos algoritmos para ordenar matrices. Ademas, corrplot es bueno para los detalles, incluida la eleccion del color, las etiquetas de texto, las etiquetas de color, el diseño, etc.

**Métodos de visualización “corrplot”**

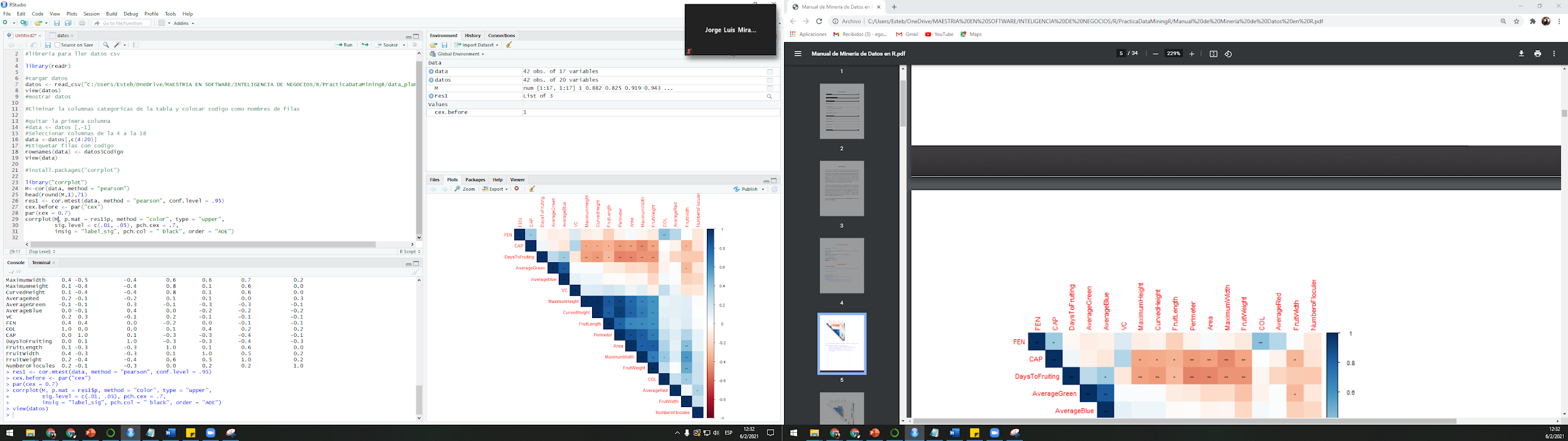
Hay siete modos de visualización (parámetro method) en el paquete corrplot, llamado “circle”, “square”, “ellipse”, “number”, “shade”, “color”, “pie”.

**Correlación método de pearson**

El coeficiente de correlación de Pearson es una prueba que mide la relación estadística entre dos variables continuas. Si la asociación entre los elementos no es lineal, entonces el coeficiente no se encuentra representado adecuadamente.

La escala de medida debe ser una escala de intervalo o relación.

* Las variables deben estar distribuidas de forma aproximada.
* La asociación debe ser lineal.
* No debe haber valores atípicos en los datos.

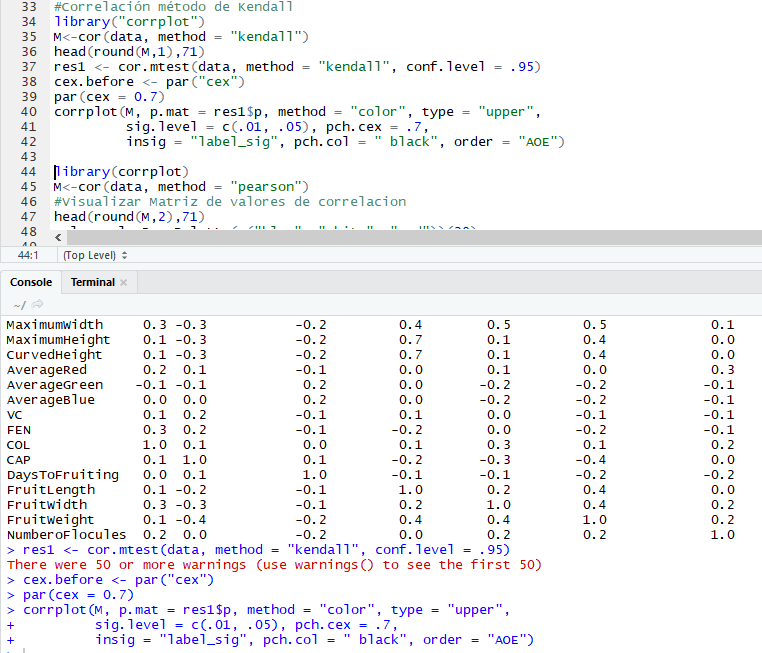
****

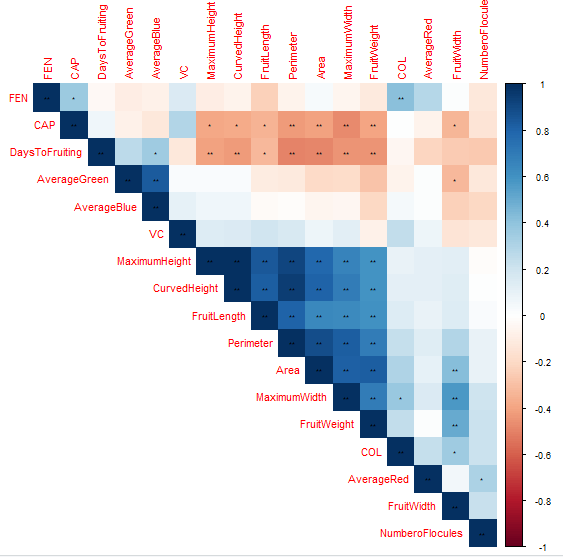
**Correlación método de kendall**

Mide el grado de asociación entre varios conjuntos (k) de N entidades. Es útil para determinar el grado de acuerdo entre varios jueces, o la asociación entre tres o más variables.

**Método de kendall en estadística**

En la prueba estadística el Coeficiente de Concordancia de Kendall (W), ofrece el valor que posibilita decidir el nivel de concordancia entre los expertos. El valor de W oscila entre 0 y 1. El valor de 1 significa una concordancia de acuerdos total y el valor de 0 un desacuerdo total. La tendencia a 1 es lo deseado pudiéndose realizar nuevas rondas si en la primera no es alcanzada significación en la concordancia.

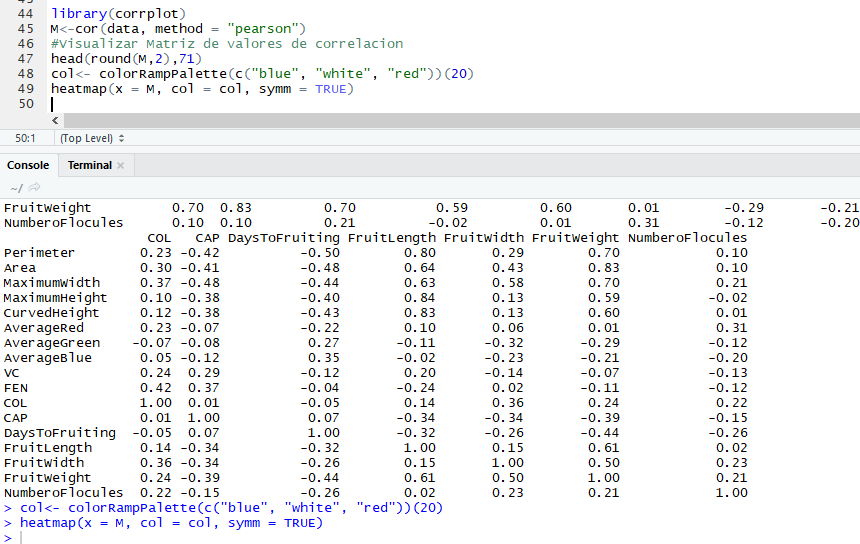
****

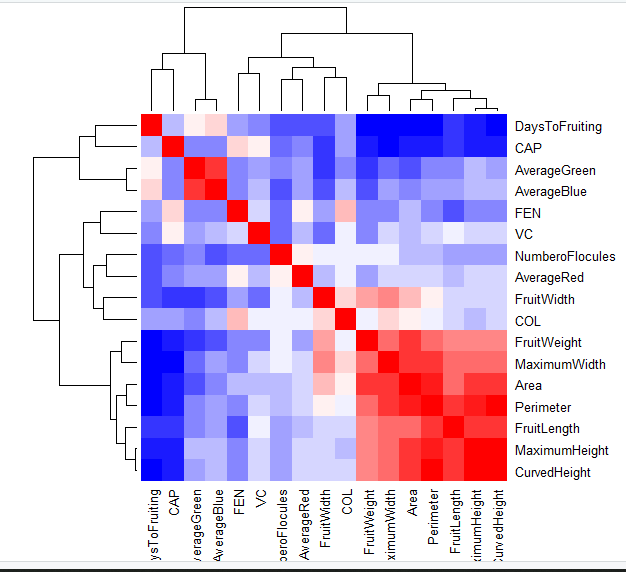
****

**Mapa de calor y Cluster en base a la correlación de pearson**

Los mapas de calor visualizan los datos a través de variaciones en el color. Cuando se aplican a un formato tabular, los mapas de calor son útiles para el examen cruzado de datos multivariados, mediante la colocación de variables en las filas y columnas, y coloreando las celdas dentro de la tabla.

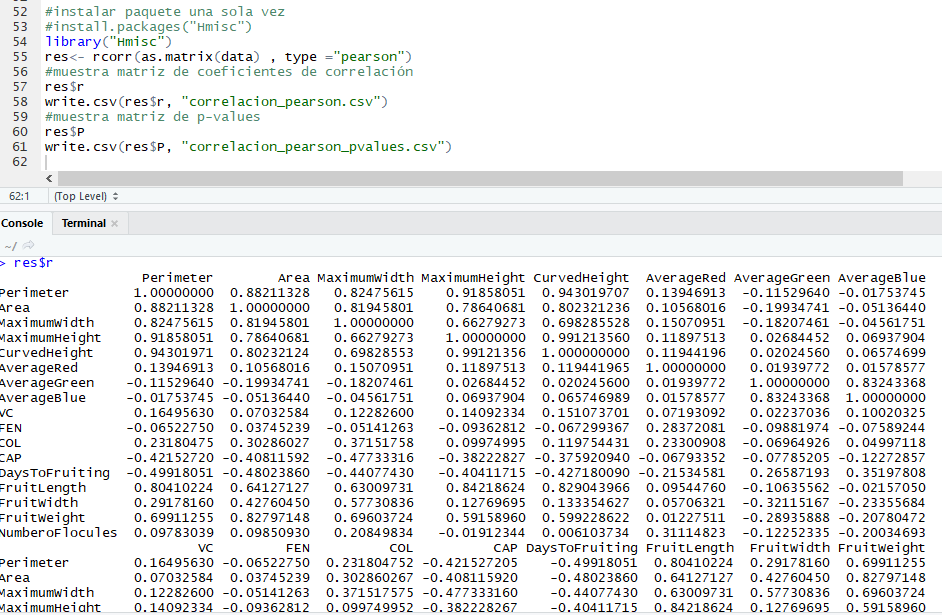
Los mapas de calor son buenos para mostrar la varianza a través de múltiples variables, revelando cualquier patrón, mostrando si las variables son similares entre sí y para detectar si existen correlaciones entre ellas.

****

****

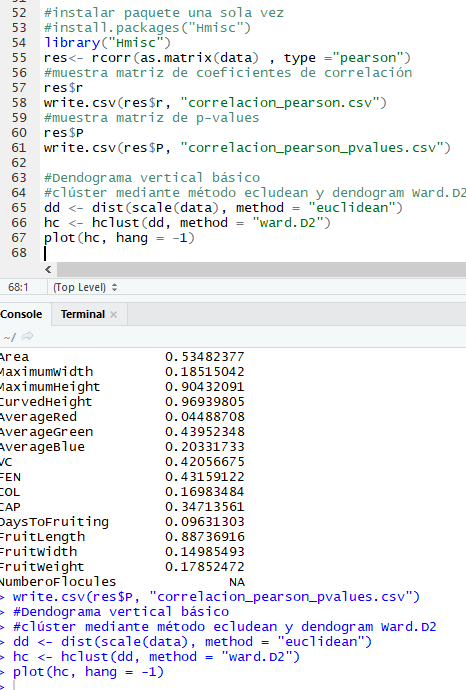
**Paquete “Hmisc”**

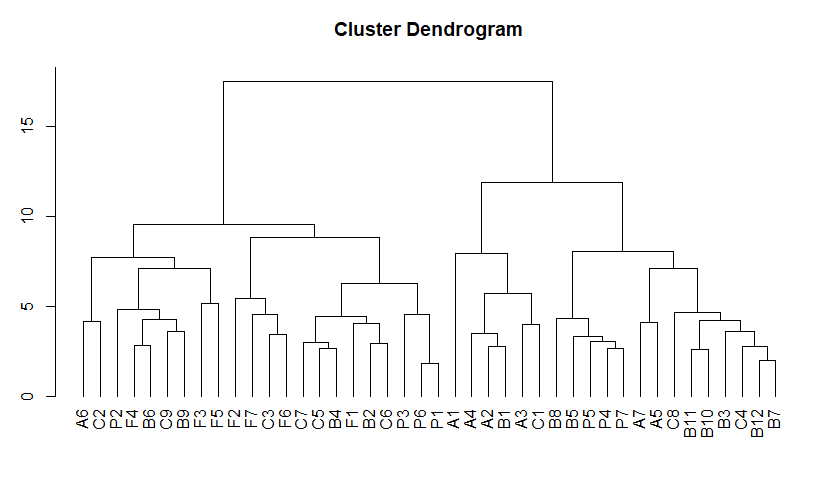
El lenguaje R funciona mediante la adición de paquetes elaborados por diferentes usuarios. Cada paquete realiza operaciones o cálculos específicos. La biblioteca Hmisc contiene funciones útiles para análisis de datos, como ofrecer una matriz de correlaciones de Pearson.

****

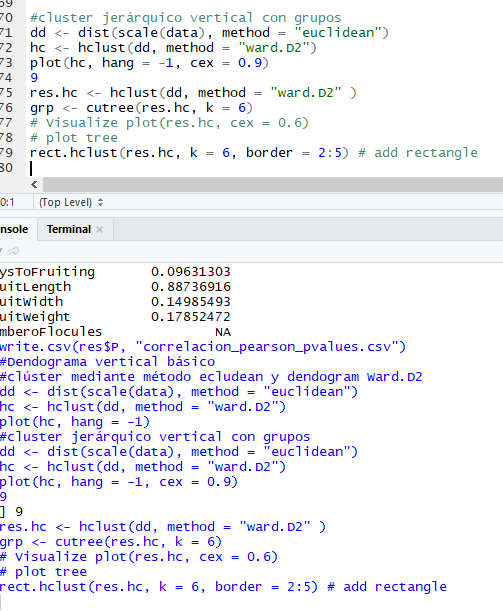
**Cluste Jerarquico**

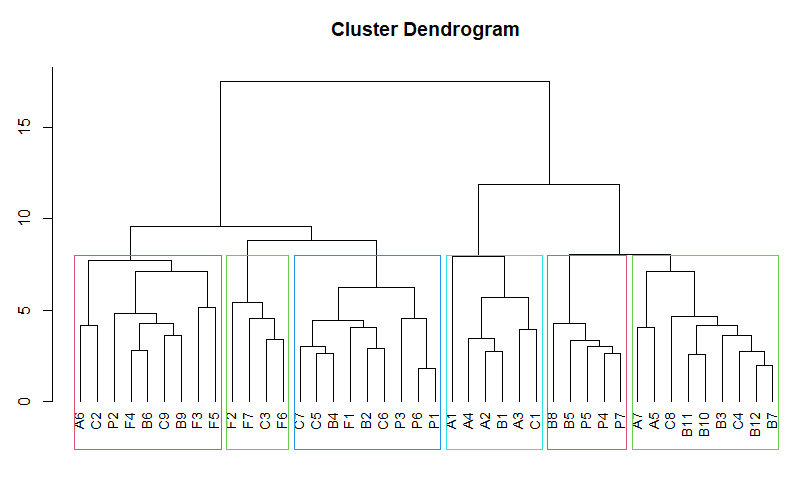
La representación de la jerarquia de cluster se representa por medio de un “dendograma”, en el que las sucesivas fusiones de las ramas a los distintos niveles nos informan de las sucesivas fusiones de los grupos en grupos de superior nivel(mayor tamaño, menos homogeneidad) sucesivamente.

****

****

**Cluster jerárquico vertical con grupos**

****

****

**Número óptimo de clusters**

La [agrupación en clúster](https://ligdigonzalez.com/algoritmos-de-clustering-agrupamiento-aprendizaje-no-supervisado/) es una parte importante del proceso de Machine Learning para empresas comerciales o científicas que utilizan la [Ciencia de Datos](https://ligdigonzalez.com/diferencias-entre-ciencia-de-datos-analista-de-datos-y-machine-learning/). Como su nombre lo indica, ayuda a identificar congregaciones de puntos de datos estrechamente relacionados, por alguna medida de distancia, en un conjunto de datos, los cuales, de otra manera, serían difíciles de entender.

Sin embargo, en la mayoría de los casos, el proceso de agrupación cae dentro del ámbito de [Aprendizaje no Supervisado](https://ligdigonzalez.com/aprendizaje-no-supervisado-machine-learning/). Acá no hay respuestas o etiquetas conocidas para guiar el proceso de optimización o para medir nuestro éxito. Estamos en el territorio inexplorado.

Por lo tanto, no es de extrañar que un método tan popular como la agrupación K Means no parezca proporcionar una respuesta completamente satisfactoria cuando nos hacemos la pregunta básica ¿cómo sabríamos el número real de grupos para empezar?

Esta cuestión es de importancia crítica debido al hecho de que el proceso de agrupamiento es a menudo un precursor del procesamiento posterior de los datos individuales de las agrupaciones y, por lo tanto, la cantidad de recursos computacionales puede depender de esta medición.

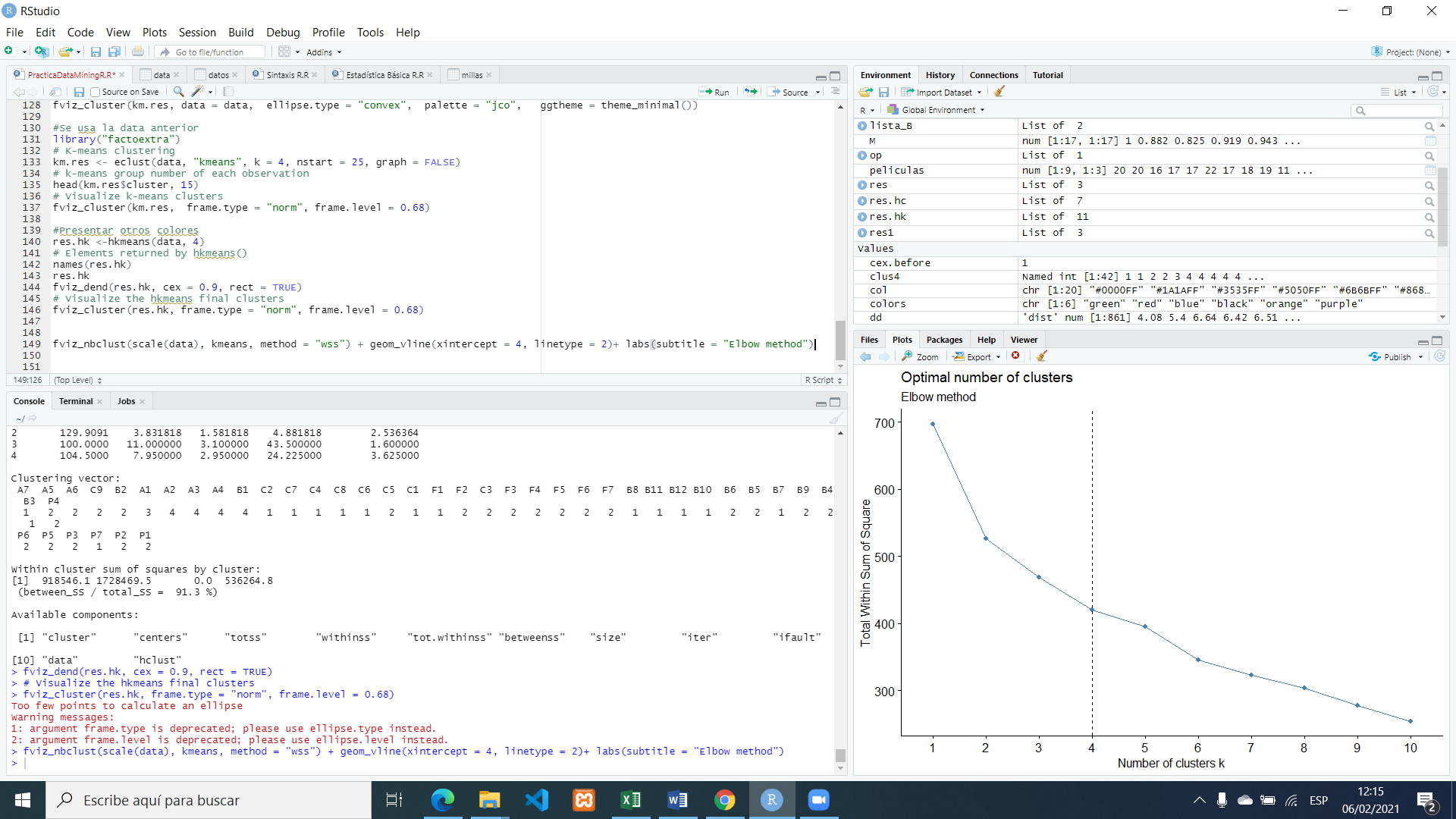
Hay múltiples métodos que puedes utilizar para determinar cuál es el número óptimo de clústeres para tus datos:

* Método del codo
* Método de la Silueta
* Método de estadística de la brecha

# Elbow method

Este método utiliza los valores de la inercia obtenidos tras aplicar el K-means a diferente número de Clusters (desde 1 a N Clusters), siendo la inercia la suma de las distancias al cuadrado de cada objeto del Cluster a su centroide:

**fviz\_nbclust(scale(data), kmeans, method = "wss") + geom\_vline(xintercept = 4, linetype = 2)+ labs(subtitle = "Elbow method")**

****

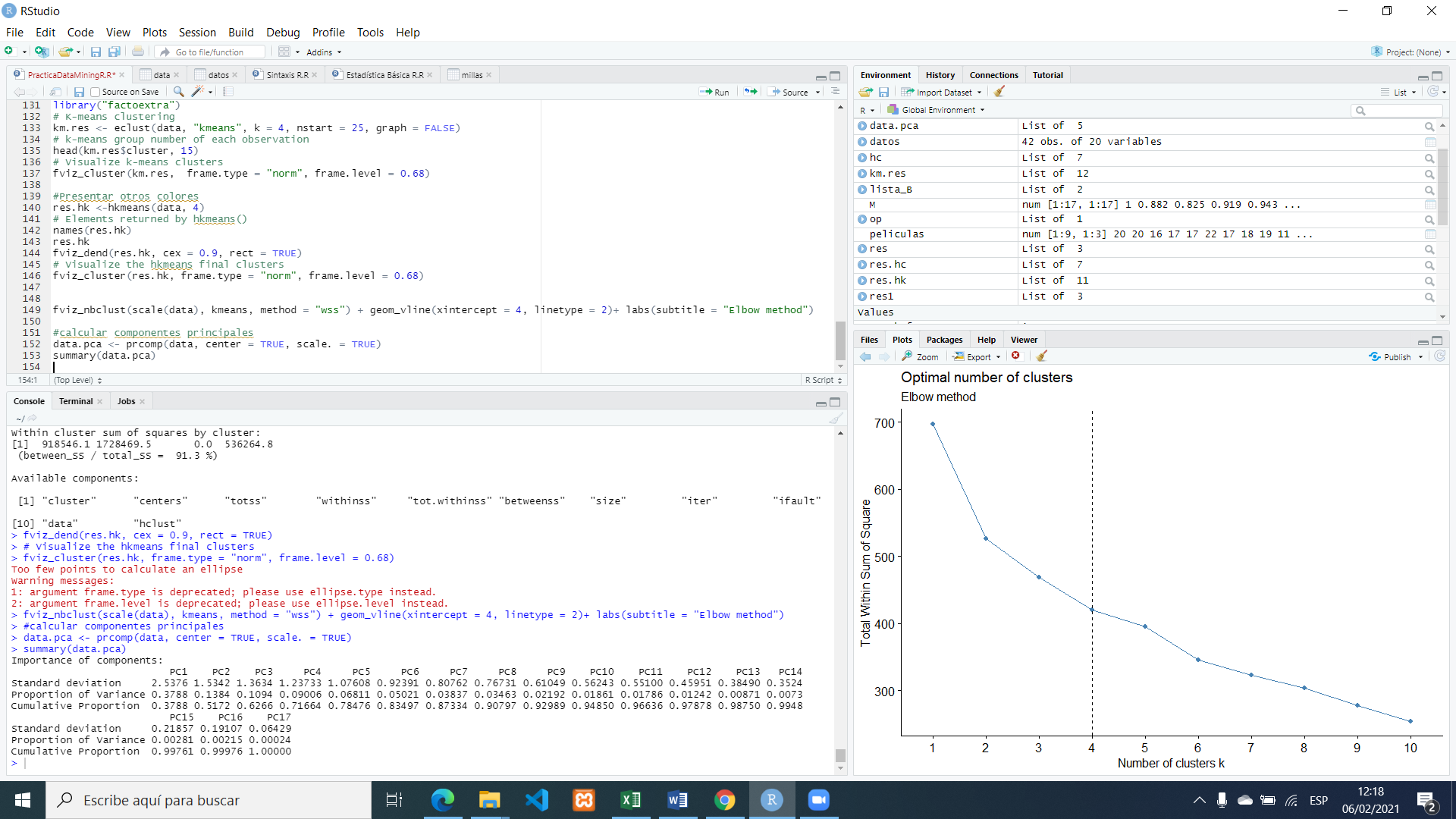
**Análisis de componentes**

El análisis de componentes principales (PCA) es una técnica útil para el análisis de datos exploratorios, que le permite visualizar mejor la variación presente en un conjunto de datos con muchas variables. Es particularmente útil en el caso de conjuntos de datos "amplios", donde tiene muchas variables para cada muestra.

**#calcular componentes principales**

**data.pca <- prcomp(data, center = TRUE, scale. = TRUE)**

**summary(data.pca)**

****

**#instalar una sola vez la siguiente libreria**

**install\_github("vqv/ggbiplot")**

**library(devtools)**

**El objetivo devtoolses facilitarle la vida como desarrollador de paquetes proporcionando funciones R que simplifican muchas tareas comunes**

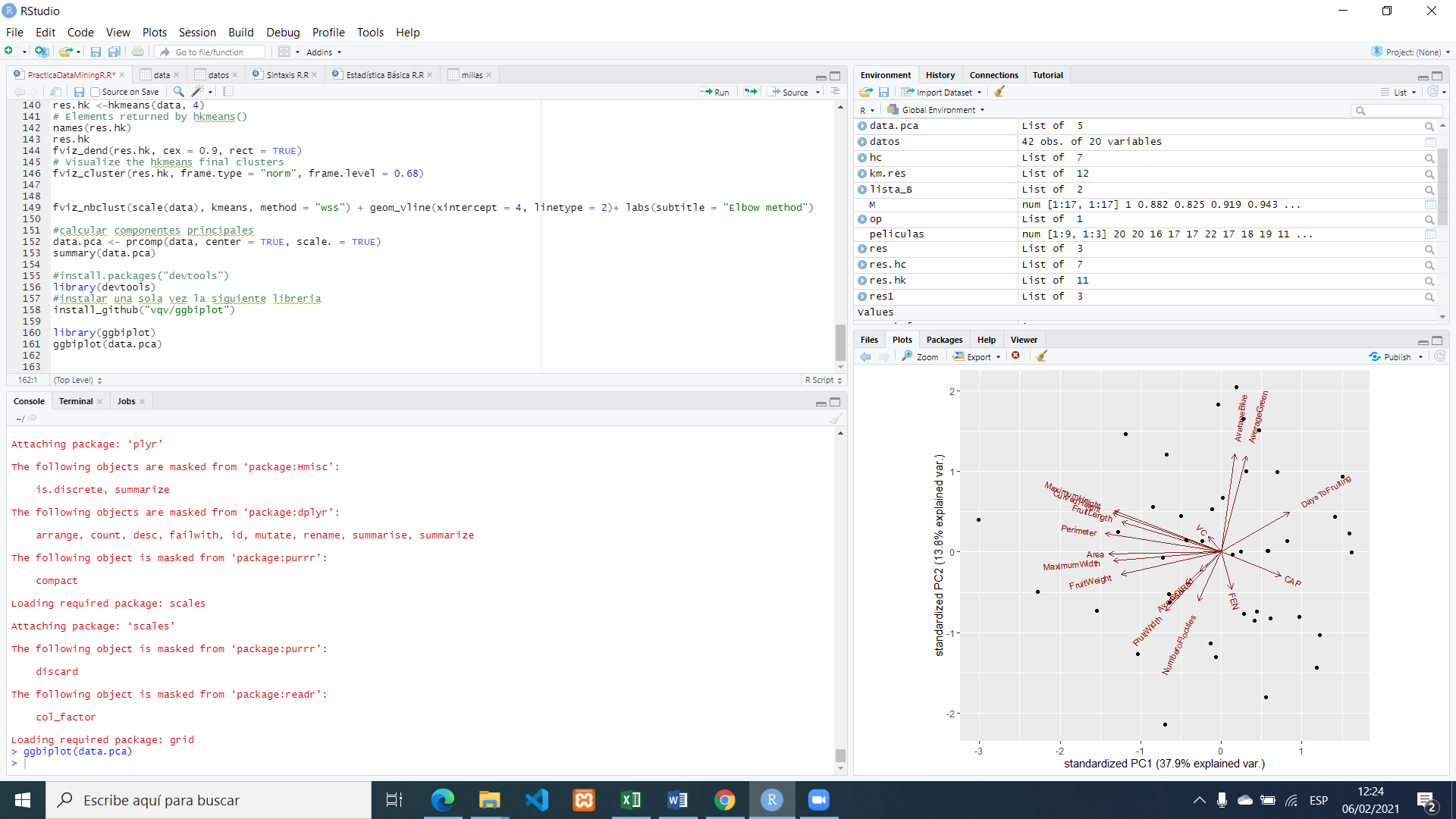
**#instalar una sola vez la siguiente libreria**

**#install\_github("vqv/ggbiplot")**

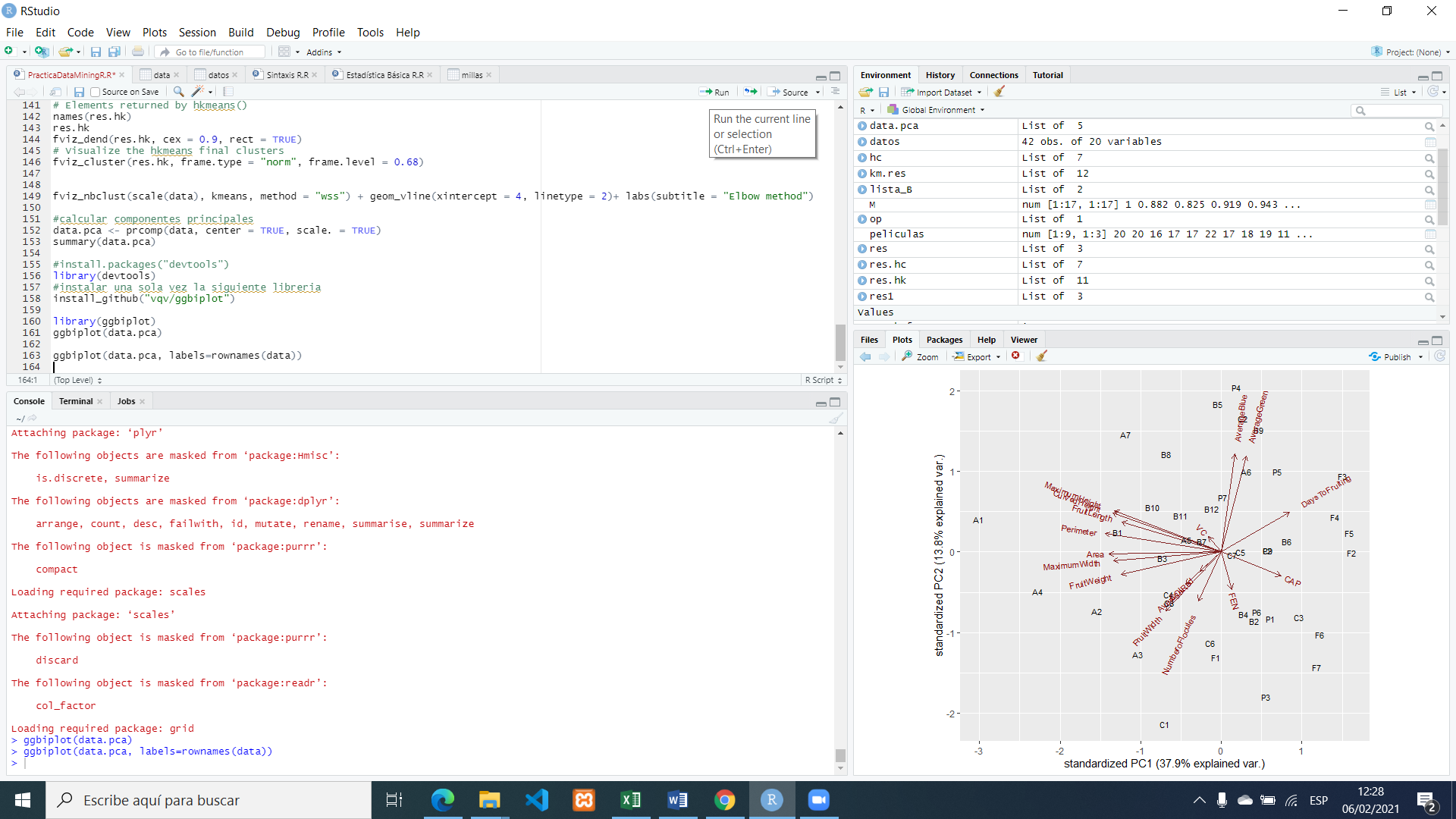
**library(ggbiplot)**

**es un paquete de visualización de datos para el lenguaje R que implementa lo que se conoce como la “Gramática de los Gráficos”**

**ggbiplot(data.pca)**

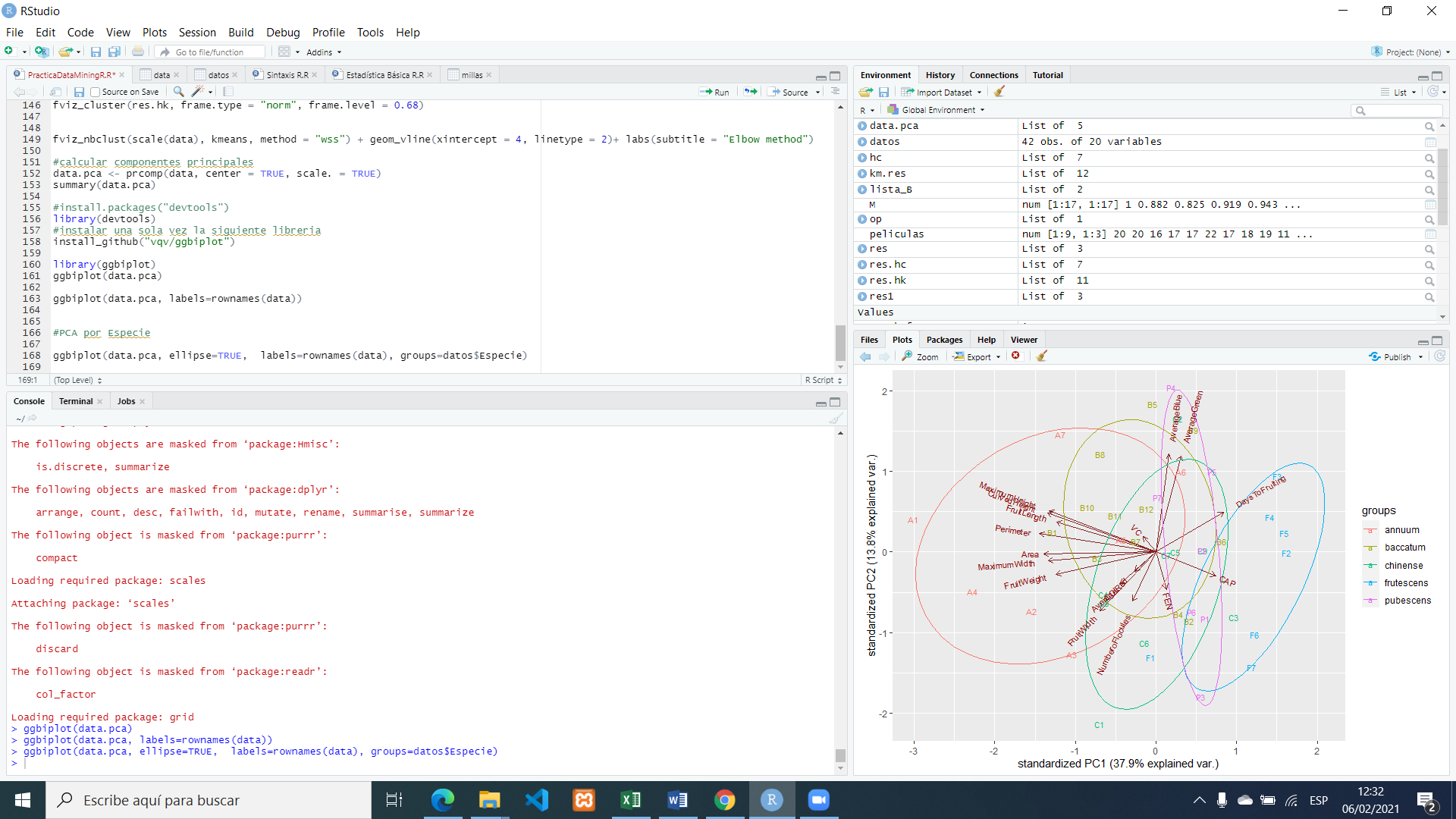
****

**ggbiplot(data.pca, labels=rownames(data))**

****

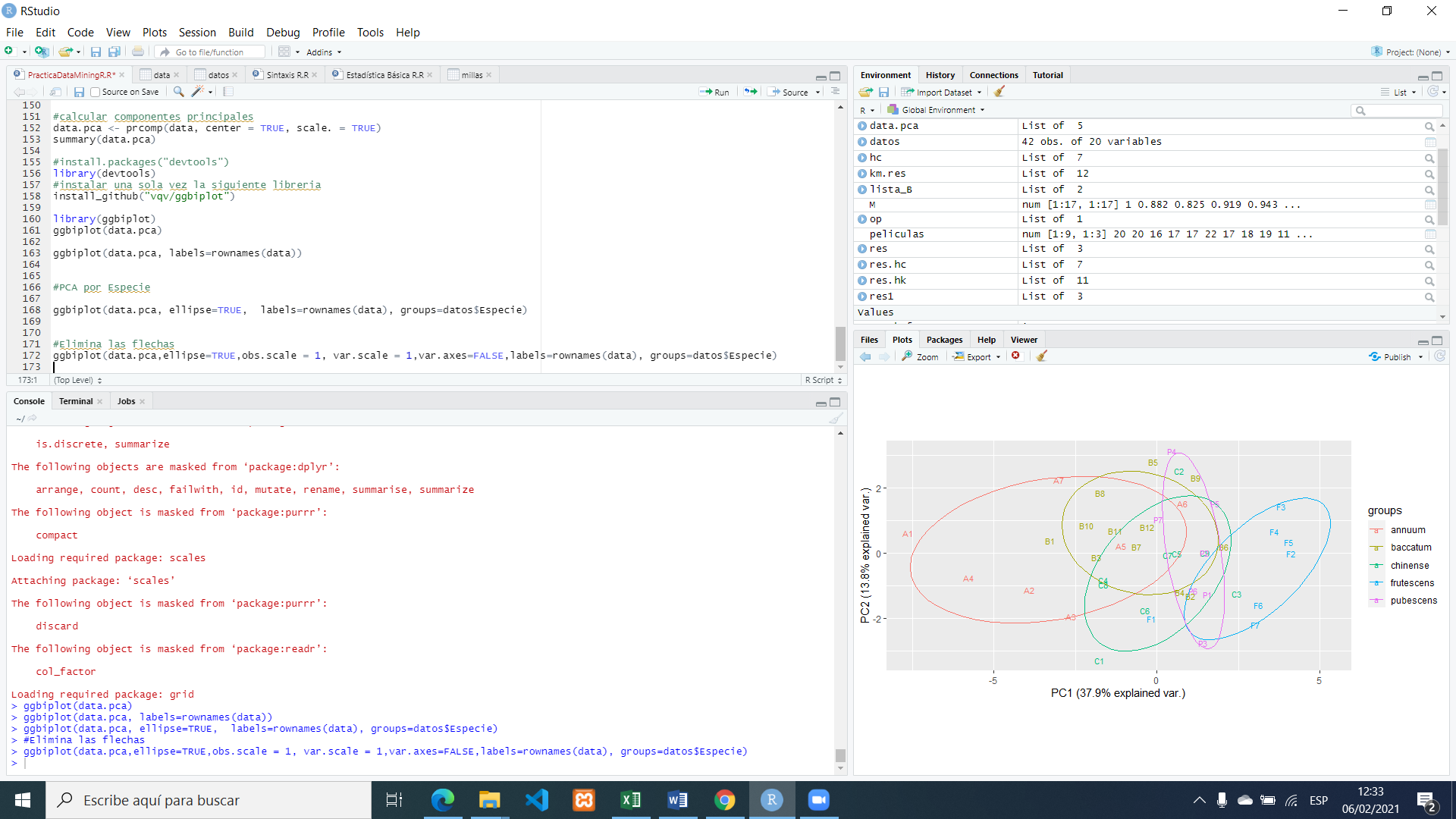
**#PCA por Especie**

**ggbiplot(data.pca, ellipse=TRUE, labels=rownames(data), groups=datos$Especie)**

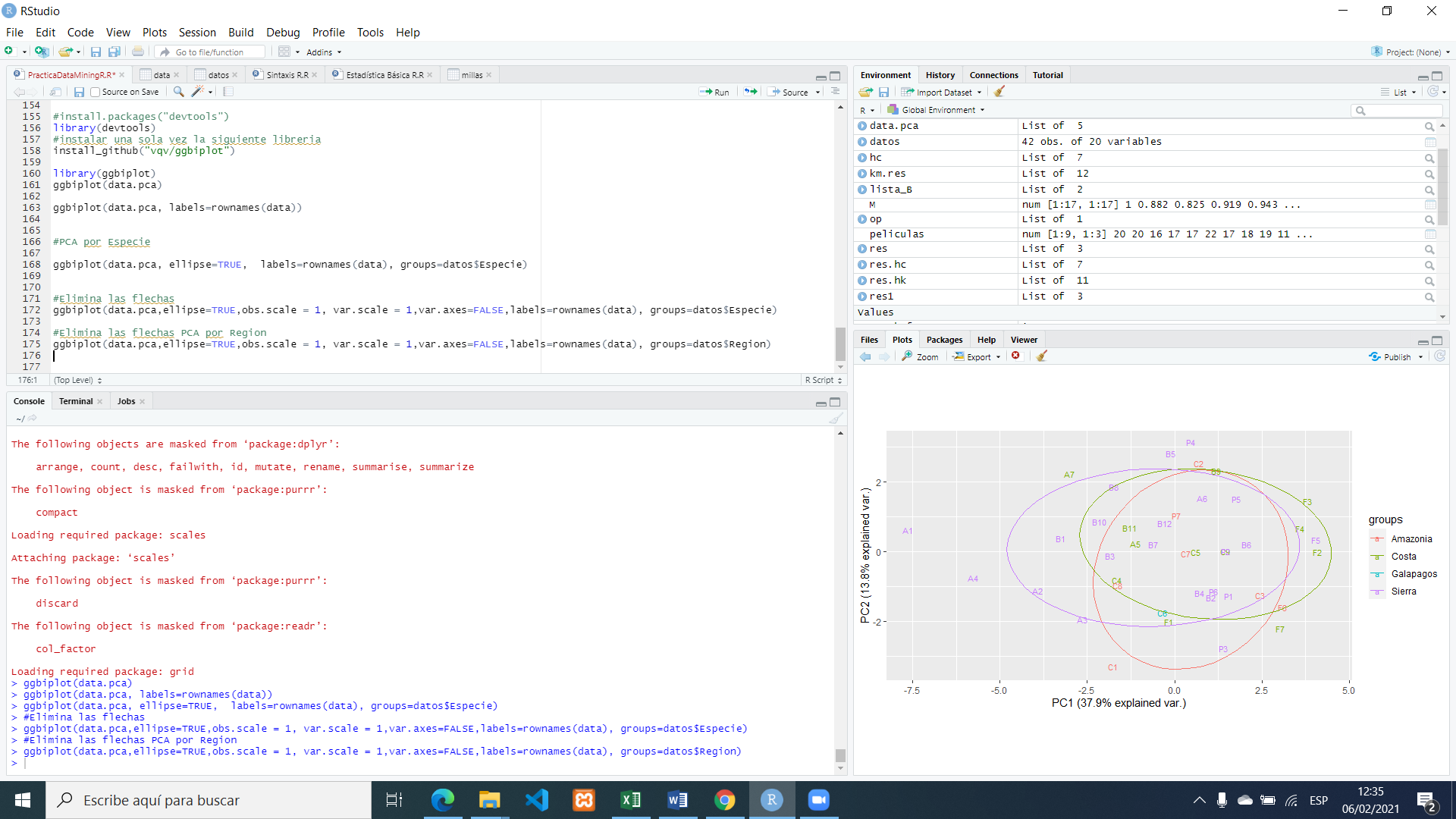
****

**#Elimina las flechas**

**ggbiplot(data.pca,ellipse=TRUE,obs.scale = 1, var.scale = 1,var.axes=FALSE,labels=rownames(data), groups=datos$Especie)**

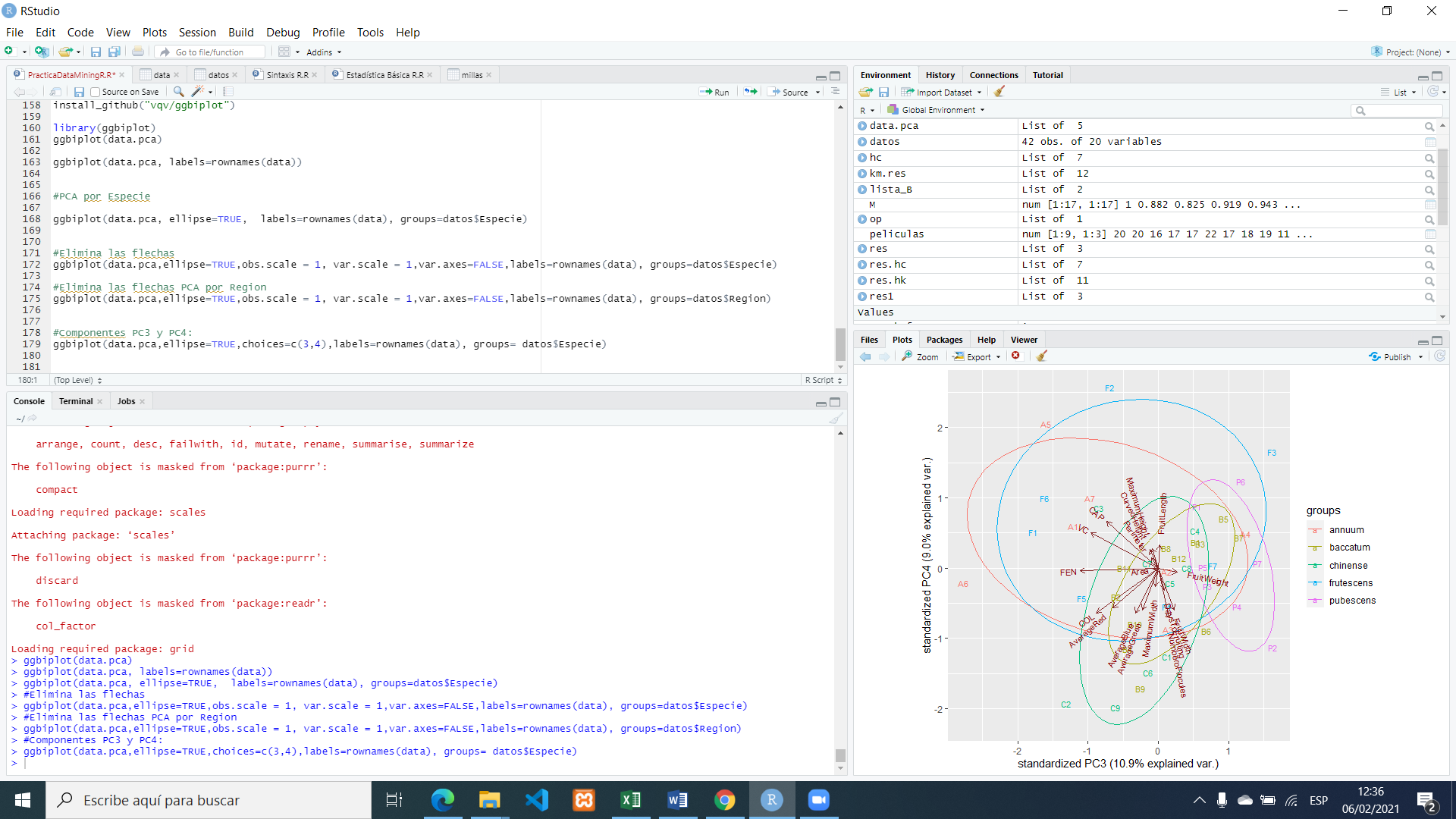
****

**#Elimina las flechas PCA por Region**

**ggbiplot(data.pca,ellipse=TRUE,obs.scale = 1, var.scale = 1,var.axes=FALSE,labels=rownames(data), groups=datos$Region)**

**#Componentes PC3 y PC4:**

**ggbiplot(data.pca,ellipse=TRUE,choices=c(3,4),labels=rownames(data), groups= datos$Especie)**

****

# Técnicas Predictivas

## Arboles de Decisión.\_Los árboles de decisión son un método usado en distintas disciplinas como modelo de predicción. Estos son similares a diagramas de flujo, en los que llegamos a puntos en los que se toman decisiones de acuerdo a una regla.

#librerías, instalar alguna si es necesario

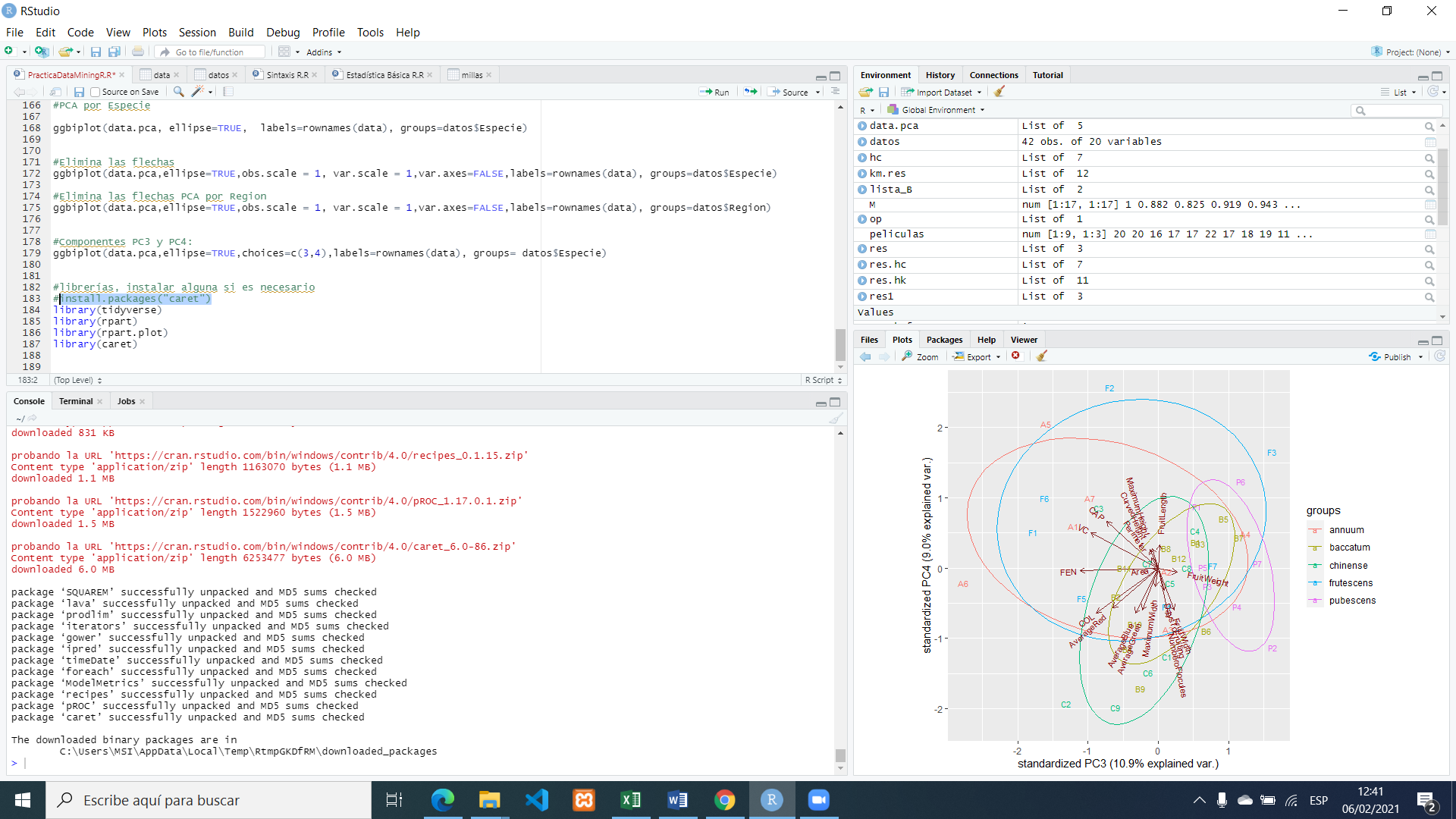
**library**(tidyverse)

**library**(rpart)

**library**(rpart.plot)

**library**(caret)

Instalamos librerias faltantes como se muestra a continuación

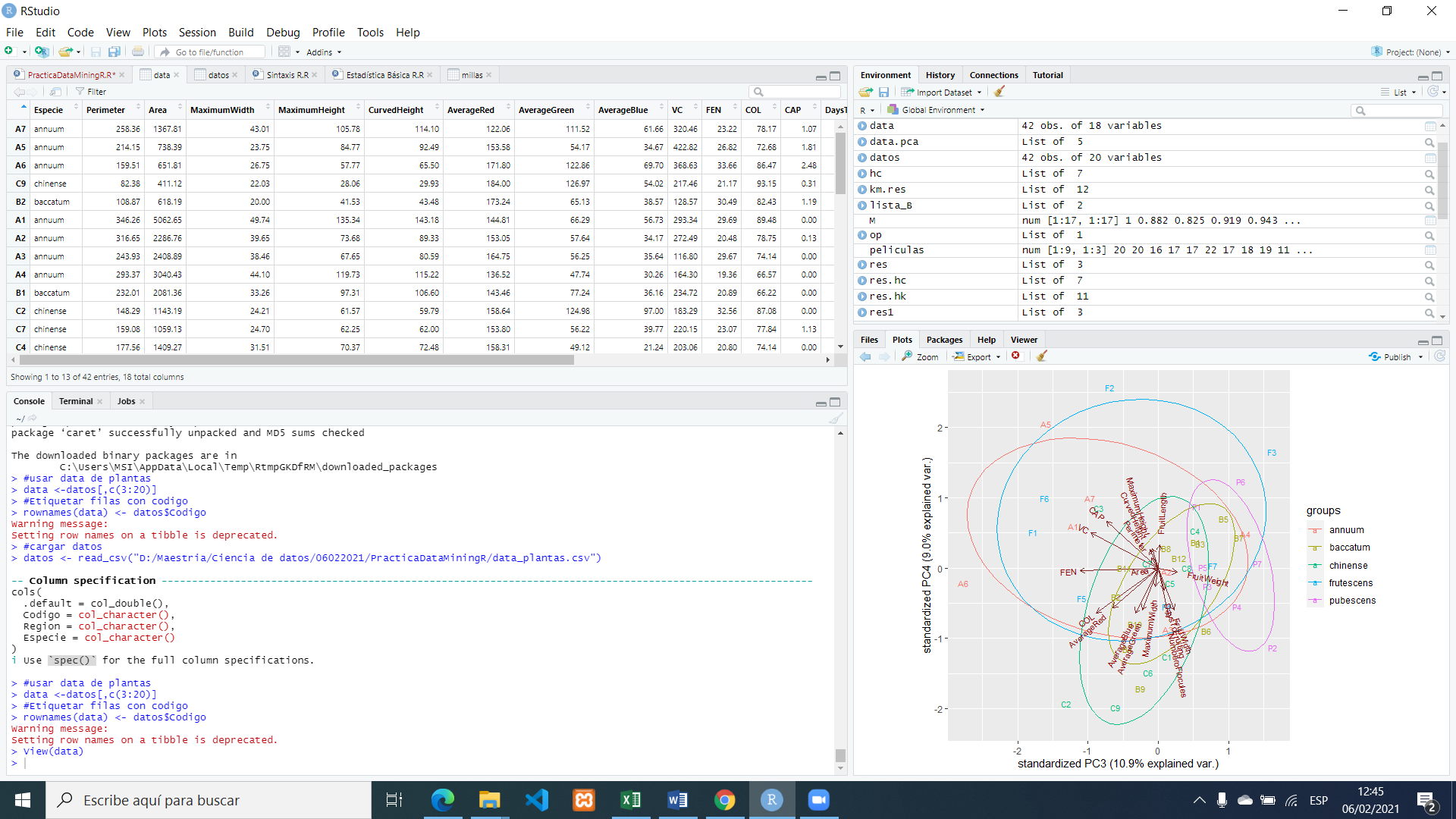
****

**#usar data de plantas**

**data <-datos[,c(3:20)]**

**#Etiquetar filas con codigo**

**rownames(data) <- datos$Codigo**

**View(data)**

**set.seed(1649)**

**train <- sample\_frac(data, .7)**

**test <- setdiff(data, train)**

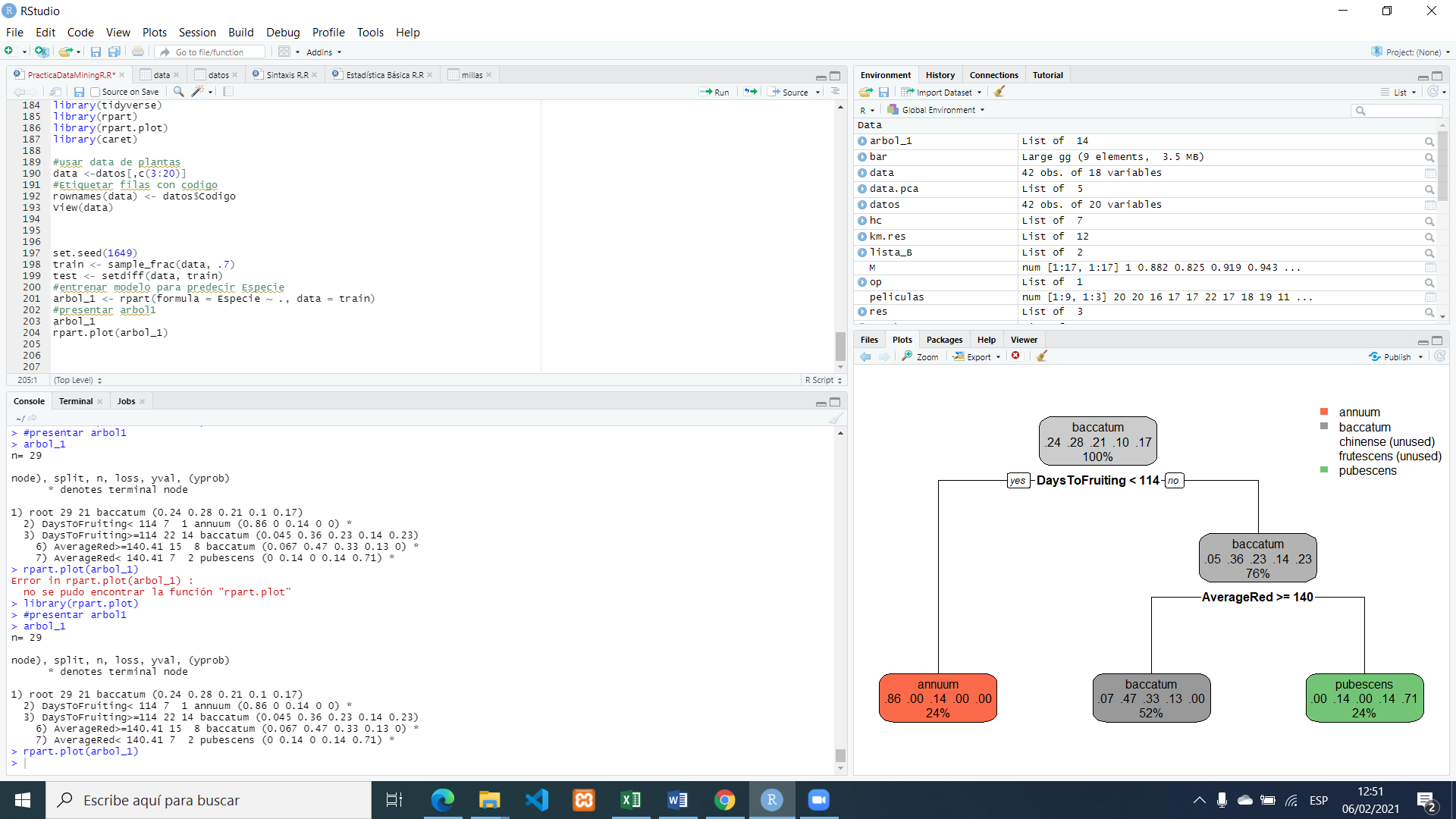
**#entrenar modelo para predecir Especie**

**arbol\_1 <- rpart(formula = Especie ~ ., data = train)**

**#presentar arbol1**

**arbol\_1**

**rpart.plot(arbol\_1)**

****

**#Generamos un segundo árbol, usando sets de entrenamiento y prueba diferentes.**

**set.seed(7439)**

**train2 <- sample\_frac(data, .7)**

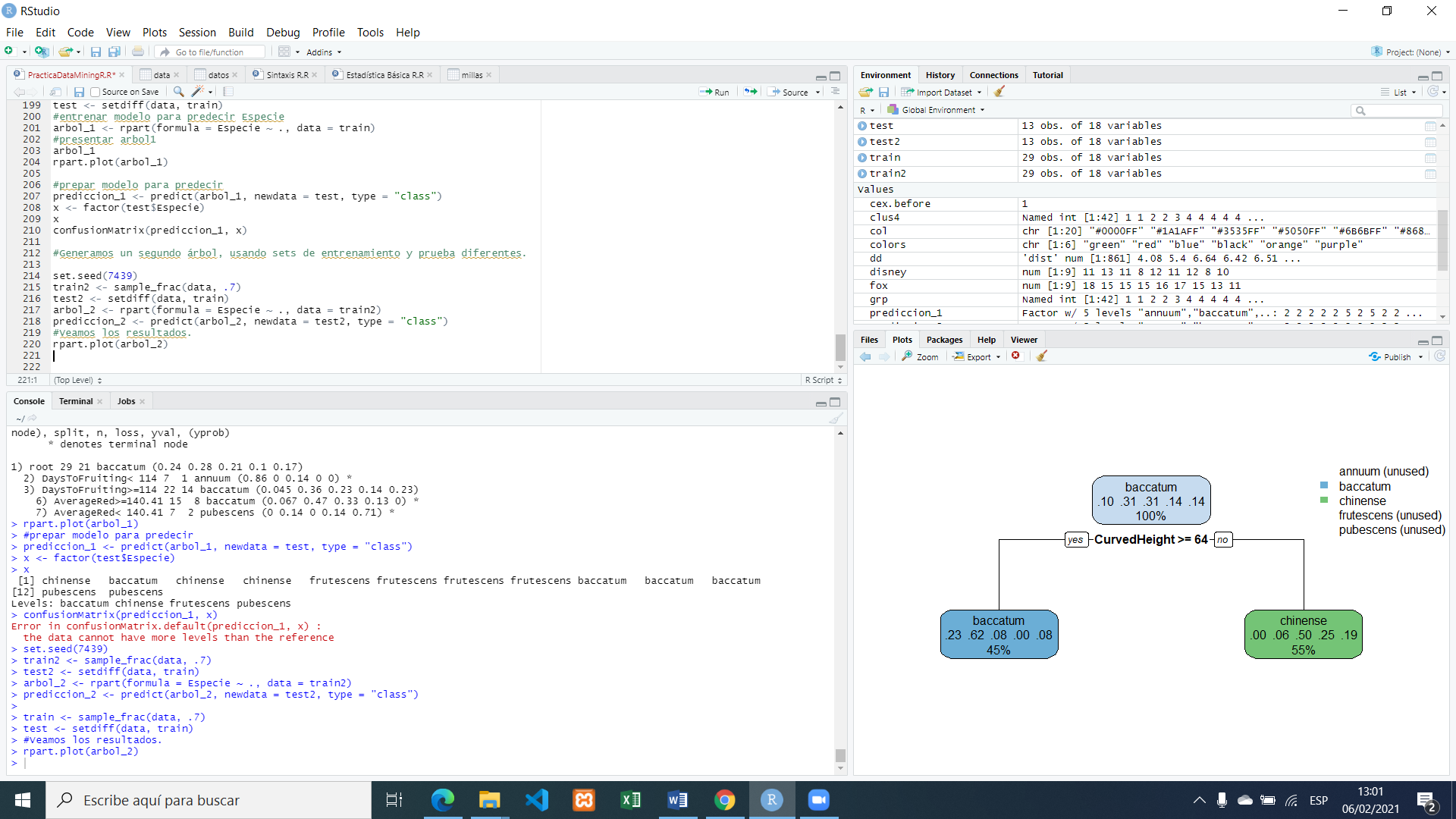
**test2 <- setdiff(data, train)**

**arbol\_2 <- rpart(formula = Especie ~ ., data = train2)**

**prediccion\_2 <- predict(arbol\_2, newdata = test2, type = "class")**

**#Veamos los resultados.**

**rpart.plot(arbol\_2)**

****

**#Generamos un tercer árbol, usando sets de entrenamiento y prueba diferentes.**

**set.seed(8476)**

**train3 <- sample\_frac(data, .7)**

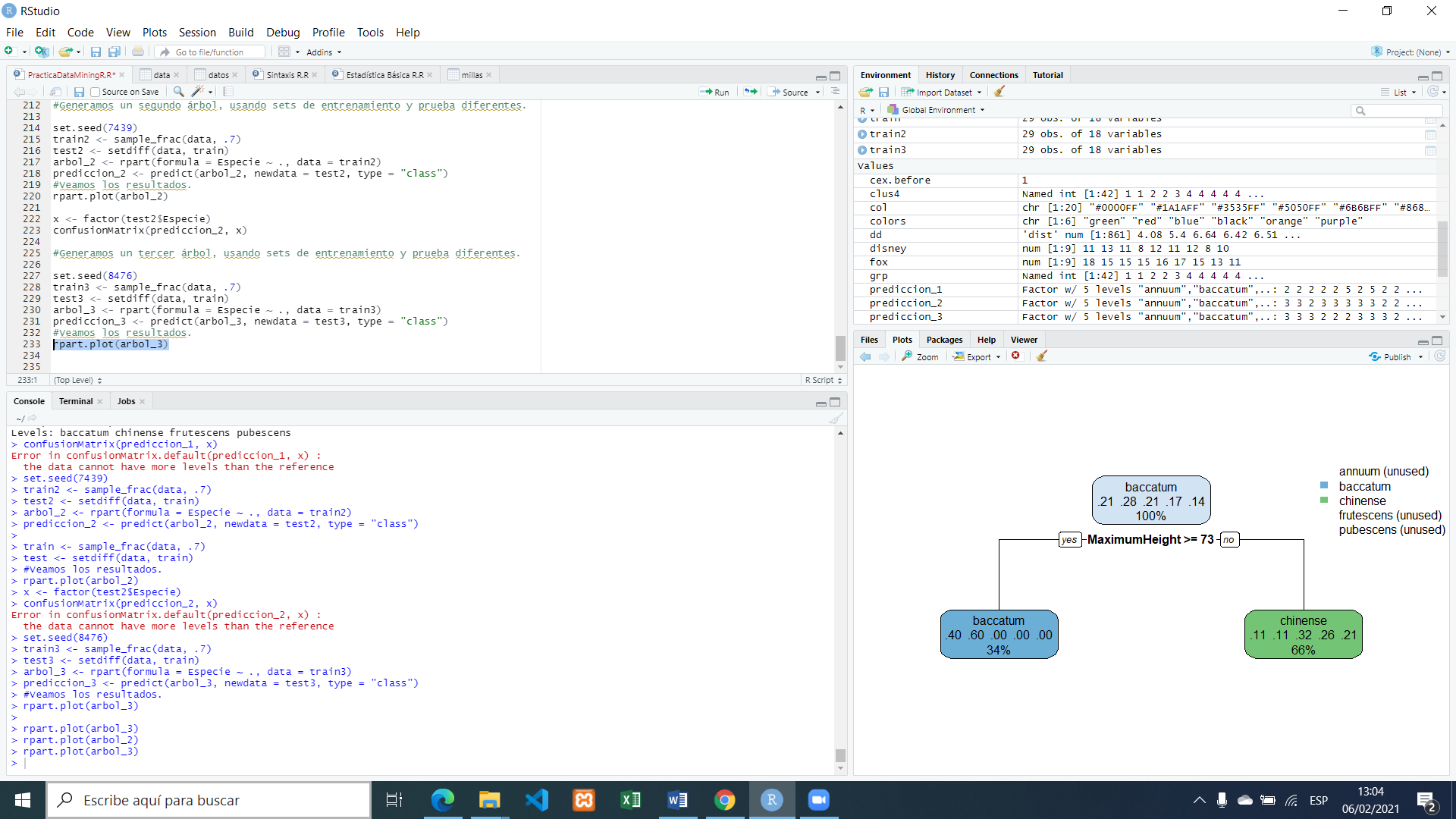
**test3 <- setdiff(data, train)**

**arbol\_3 <- rpart(formula = Especie ~ ., data = train3)**

**prediccion\_3 <- predict(arbol\_3, newdata = test3, type = "class")**

**#Veamos los resultados.**

**rpart.plot(arbol\_3)**

****

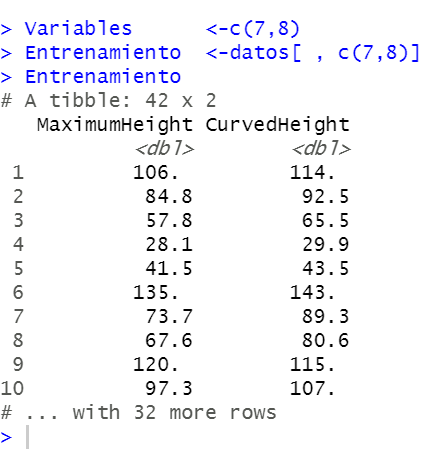
## Regresión lineal simple

La regresión lineal simple consiste en generar un modelo de regresión (ecuación de una recta) que permita explicar la relación lineal que existe entre dos variables. A la variable dependiente o respuesta se le identifica como “Y” y a la variable predictora o independiente como “X”.

**Ejemplo práctico en R**

**#Aplicamos la data de plantas, volver a cargar si es necesario**

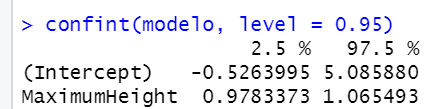
**# Elegir Variables CurvedHeight y MaximumHeight**

****

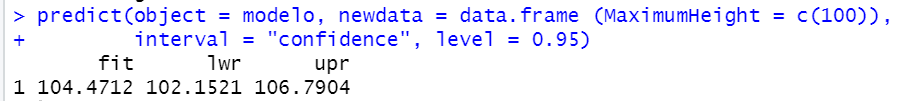
# **MaximumHeight** es la variable independiente y **CurvedHeight** la variable dependiente



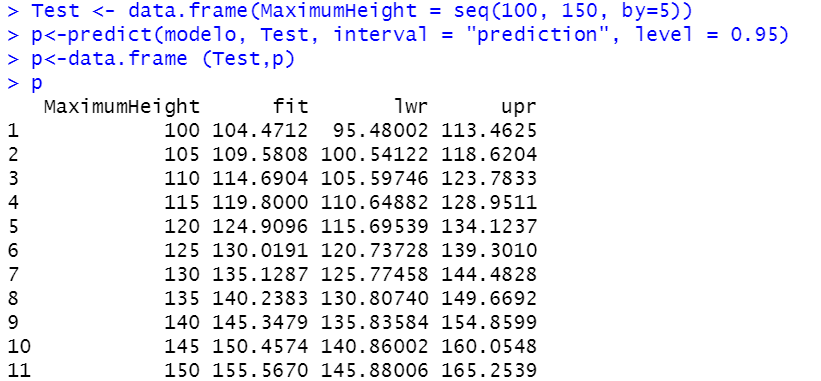
confint(modelo, level = 0.95)



#Ejemplo de predicción, asignar un valor a la variable independiente, **MaximumHeight** = 100

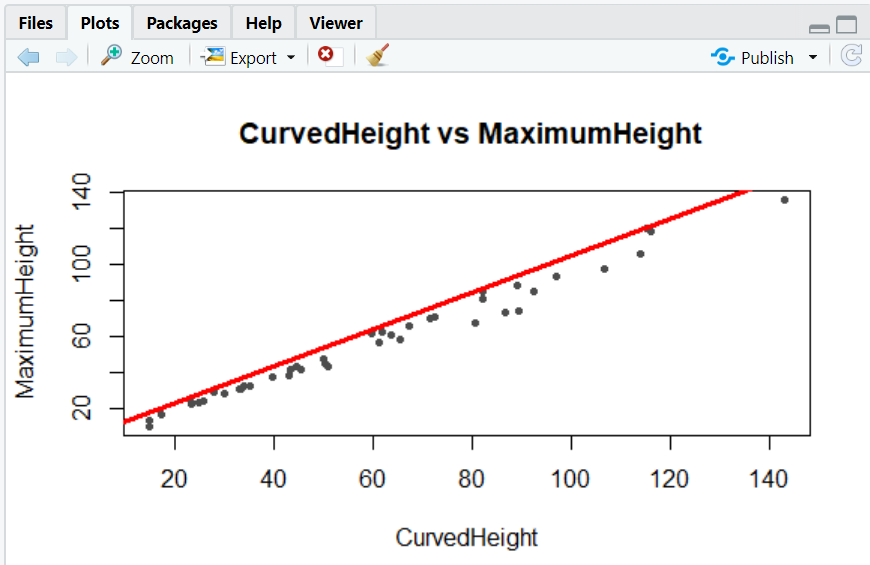


#Ejemplo de predicción para varios datos



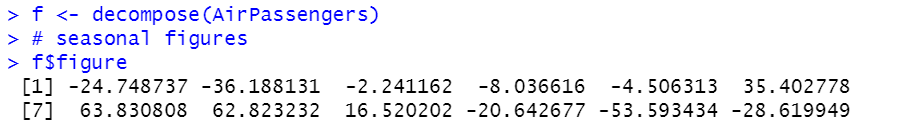
#Ejemplo de gráfico 1

plot(x = Entrenamiento$CurvedHeight, y = Entrenamiento$MaximumHeight, main = " CurvedHeight vs MaximumHeight", xlab = "CurvedHeight", ylab = "MaximumHeight", pch = 20, col = "grey30") abline(modelo, lwd = 3, col = "red")

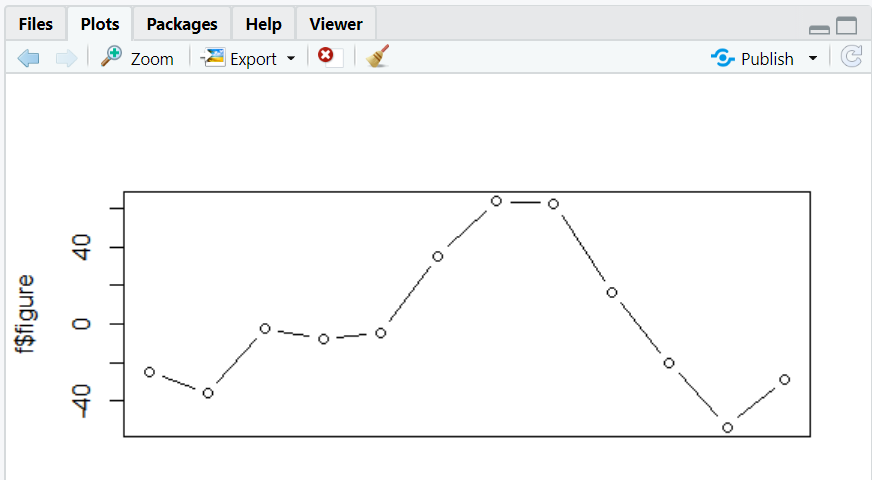


## Series temporales

Una serie temporal se define como una colección de observaciones de una variable recogidas secuencialmente en el tiempo. Estas observaciones se suelen recoger en instantes de tiempo equiespaciados. Si los datos se recogen en instantes temporales de forma continua, se debe o bien digitalizar la serie, es decir, recoger sólo los valores en instantes de tiempo equiespaciados, o bien acumular los valores sobre intervalos de tiempo.



plot(f$figure, type="b", xaxt="n", xlab="")



# get names of 12 months in English words

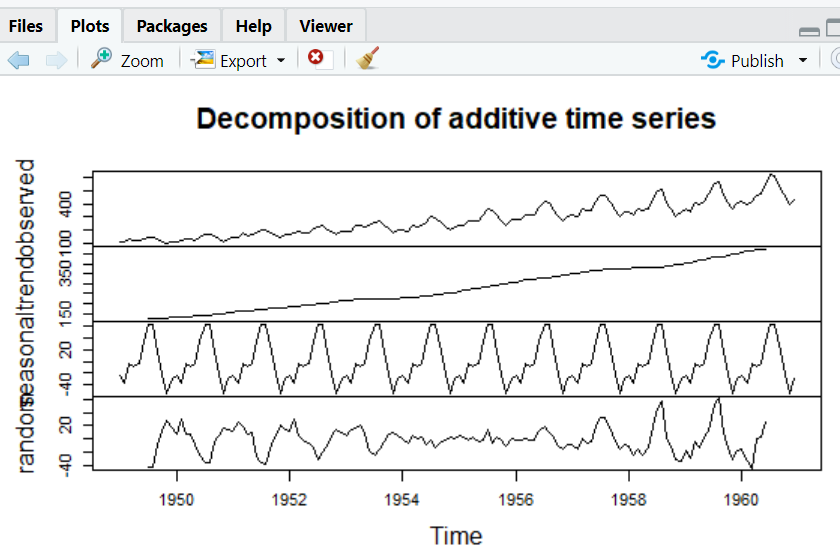
monthNames <- months(ISOdate(2011,1:12,1))

# label x-axis with month names

# las is set to 2 for vertical label orientation

axis(1, at=1:12, labels=monthNames, las=2)

plot(f)



#Predicción

fit <- arima(AirPassengers, order=c(1,0,0), list(order=c(2,1,0), period=12))

fore <- predict(fit, n.ahead=24)

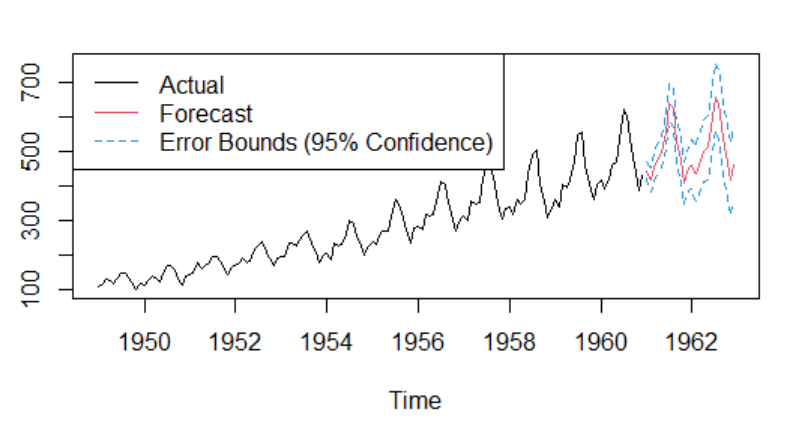
# error bounds at 95% confidence level

U <- fore$pred + 2\*fore$se

L <- fore$pred - 2\*fore$se

ts.plot(AirPassengers, fore$pred, U, L, col=c(1,2,4,4), lty = c(1,1,2,2))

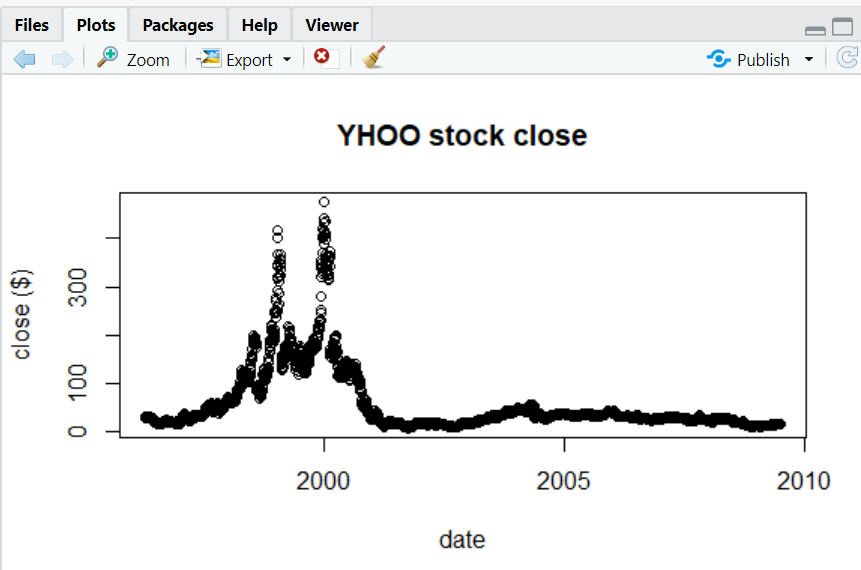
legend("topleft", c("Actual", "Forecast", "Error Bounds (95% Confidence)"), col = c(1,2,4), lty = c(1,1,2))



**Ejemplos # 2 yahoo\_series temporales.csv**

**#graficar la serie**

**plot(x=yahoo$date, y=yahoo$close, main='YHOO stock close', xlab='date', ylab='close ($)')**

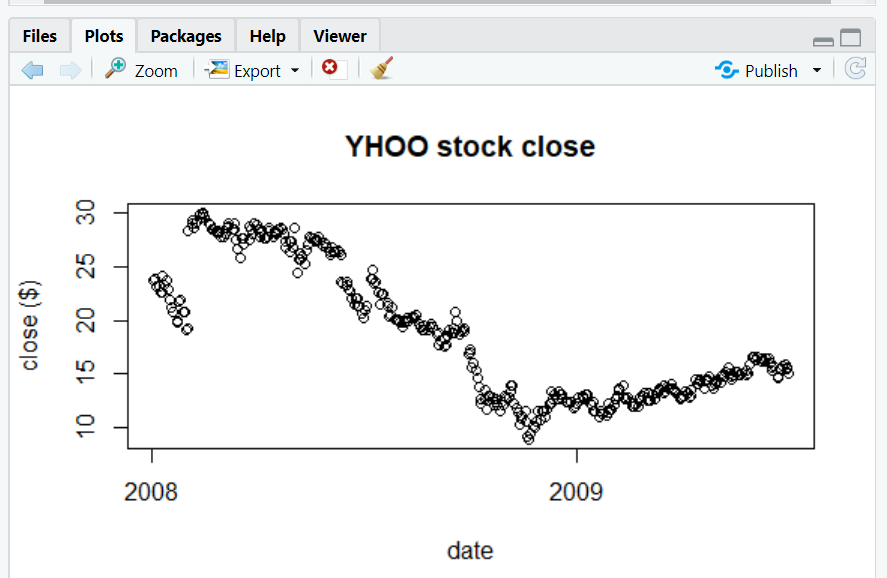
****

**#filtrar datos y graficar la serie**

**yahoo2 <- yahoo[ yahoo$date >= as.Date('2008-01-01'), ]**

**plot(x=yahoo2$date, y=yahoo2$close,**

**main='YHOO stock close', xlab='date', ylab='close ($)')**

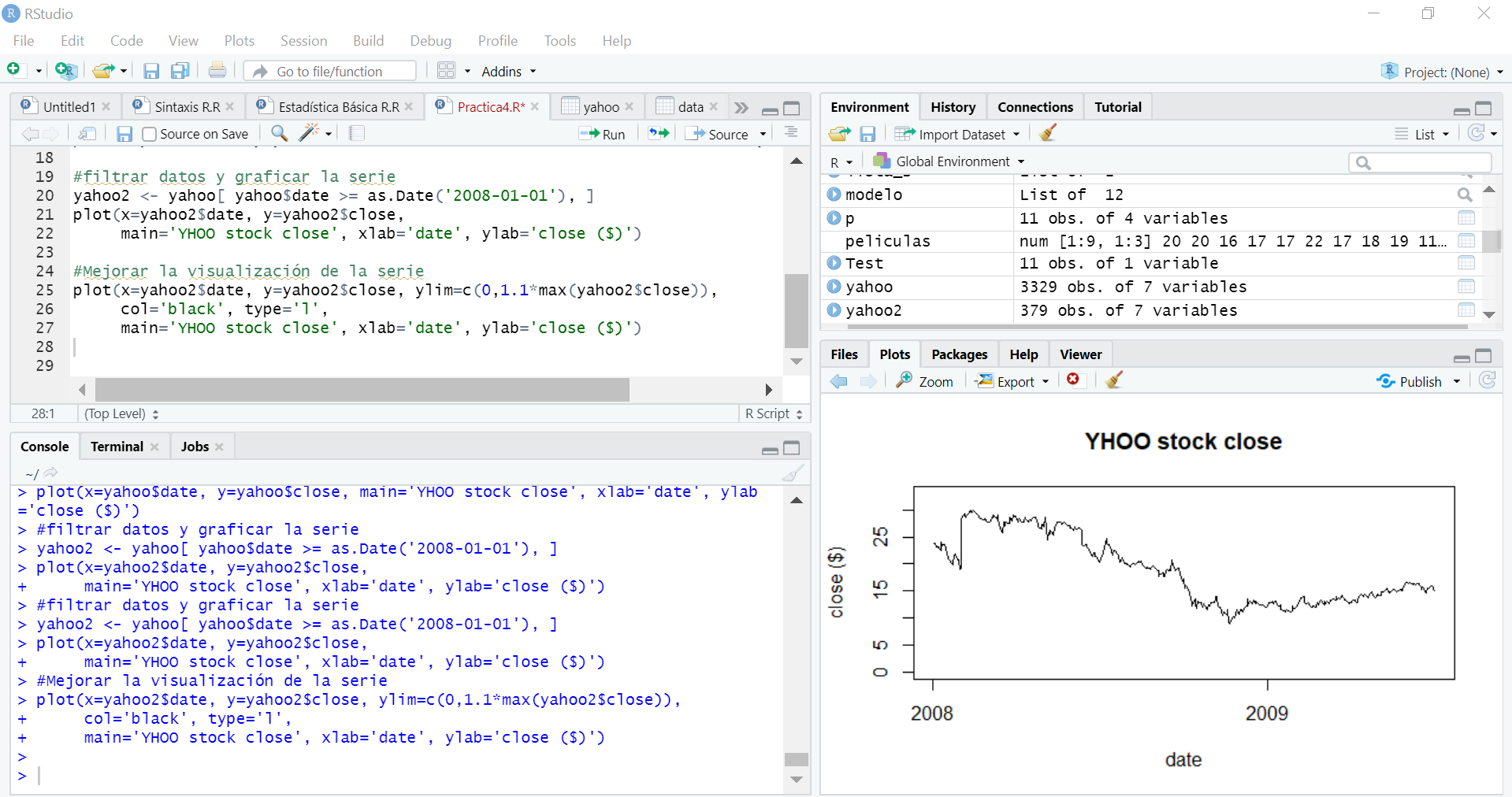
****

**#Mejorar la visualización de la serie**

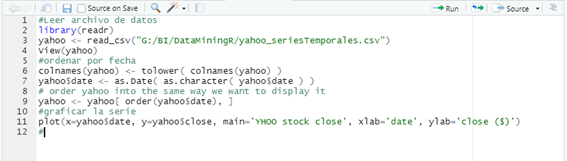
**plot(x=yahoo2$date, y=yahoo2$close, ylim=c(0,1.1\*max(yahoo2$close)),**

**col='black', type='l',**

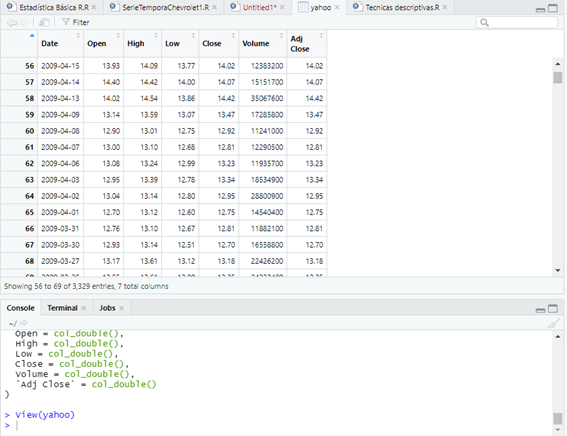
**main='YHOO stock close', xlab='date', ylab='close ($)')**

****

Ejemplo de Predicciones #2

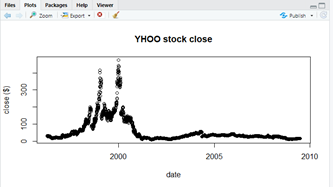


En la instrucción de la línea View(yahoo) – obtendremos este resultado.

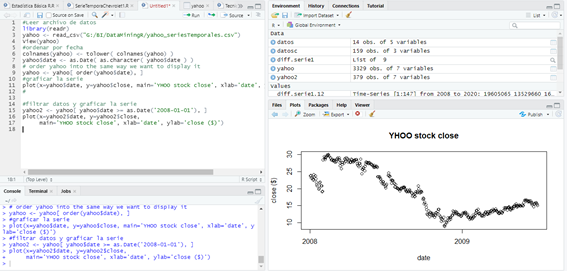


Al regresar a nuestra ventana de Codigo R continuar la ejecución – al ejecutar la línea

plot(x=yahoo$date, y=yahoo$close, main='YHOO stock close', xlab='date', ylab='close ($)') nos mostrara lo siguiente:



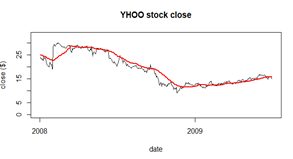
Filtrar los datos y graficar la serie:



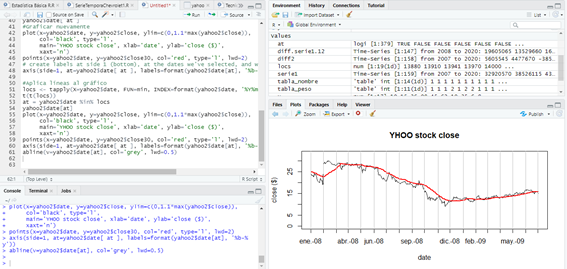
Mejorar la Visualización de la Serie



Graficar el promedio móvil.



Resultado final del Ejercicio #2



**Redes Neuronales**

Las redes neuronales artificiales son un modelo inspirado en el funcionamiento del cerebro humano. Está formado por un conjunto de nodos conocidos como neuronas artificiales que están conectadas y transmiten señales entre sí. Estas señales se transmiten desde la entrada hasta generar una salida.

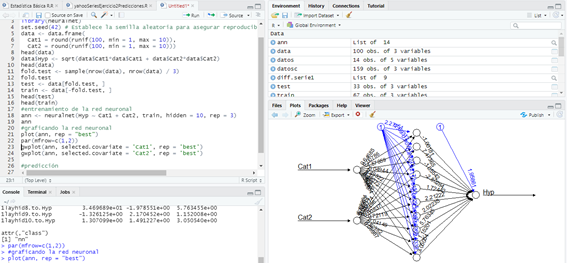
**¿Cómo funcionan las redes neuronales?**

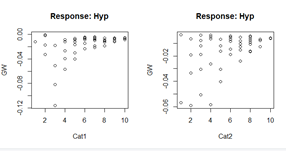
Como se ha mencionado el funcionamiento de las redes se asemeja al del cerebro humano. Las redes reciben una serie de valores de entrada y cada una de estas entradas llega a un nodo llamado neurona. Las neuronas de la red están a su vez agrupadas en capas que forman la red neuronal. Cada una de las neuronas de la red posee a su vez un peso, un valor numérico, con el que modifica la entrada recibida. Los nuevos valores obtenidos salen de las neuronas y continúan su camino por la red. Este funcionamiento puede observarse de forma esquemática en la siguiente imagen.



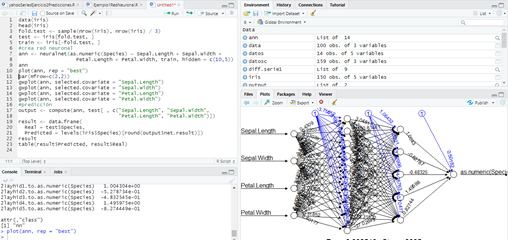
Una vez que se ha alcanzado el final de la red se obtiene una salida que será la predicción calculada por la red. Cuantas más capas posea la red y más compleja sea, también serán mas complejas las funciones que pueda realizar.

**Ejemplo # 1 – Usando lenguaje R**

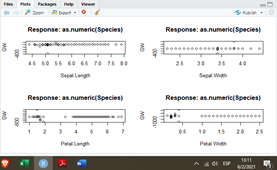
****

****

**Ejemplo#2 Red Neuronal.**

****

Predicción:



Ejemplo #3 Red Neuronal:

