PRÁCTICA 4. SPEARHEADS

Por Judit Praena García

#Ejercicio 1. Importar\_tabla\_de\_datos\_spearheads\_como\_data\_frame

#Install\_readxl

#Para importar una tabla de datos Excel a un data frame, lo primero que hay que hacer es instalar un paquete denominado “readxl”, que es encuentra en la gestión del directorio. El código que aparece abajo tiene que ponerse en el editor de códigos para que se pueda ejecutar.

installed.packages**(**"readxl"**)**

library**(**readxl**)**

#Ahora, hay que pegar el enlace de la ruta del documento Excel. Para que no haya problemas, la ruta no debe ser muy larga. Por eso, siempre se recomienda crear la carpeta en los archivos de Windows.

spear **<-** read\_excel**(**"C:/spearheads/spearheads.xlsx"**)**

View**(**spear**)** ##Muestra una tabla en el editor de registros

str**(**spear**)** ##Tipo de datos campos

class**(**spear**)** ##Tipo de estructura de datos del objeto

#Convertimos a data frame con la función de as.data.frame(spear)

spear **<-** as.data.frame**(**spear**)**

class**(**spear**)**

#Ejercicio 2 renombra\_las\_variables

#Ahora se cambian los valores a una abreviatura para que resulte más sencillo. En este caso, se ha decidido cambiar todas las denominaciones. Se ha decidido poner todos los nombres con la primera letra en mayúscula y sin tildes, para evitar confusiones en los próximos ejercicios.

#Con el doble igual == se comparan las variables

#Con la función View, se podrá ver el resultado en forma de tabla.

names**(**spear**)[**names**(**spear**)** **==** "Mat"**]** **<-** "Materiales"

names**(**spear**)[**names**(**spear**)** **==** "Con"**]** **<-** "Contexto"

names**(**spear**)[**names**(**spear**)** **==** "Cond"**]** **<-** "Conservacion"

names**(**spear**)[**names**(**spear**)** **==** "Loo"**]** **<-** "Loop"

names**(**spear**)[**names**(**spear**)** **==** "Peg"**]** **<-** "Remache"

names**(**spear**)[**names**(**spear**)** **==** "Date"**]** **<-** "Fecha"

names**(**spear**)[**names**(**spear**)** **==** "Maxle"**]** **<-** "Longitud\_max"

names**(**spear**)[**names**(**spear**)** **==** "Socle"**]** **<-** "Longitud\_encaje"

names**(**spear**)[**names**(**spear**)** **==** "Maxwi"**]** **<-** "Ancho\_max"

names**(**spear**)[**names**(**spear**)** **==** "Upsoc"**]** **<-** "Ancho\_encaje"

names**(**spear**)[**names**(**spear**)** **==** "Mawit"**]** **<-** "Ancho\_max\_encaje"

names**(**spear**)[**names**(**spear**)** **==** "Weight"**]** **<-** "Peso"

spear

View**(**spear**)**

#Para renombrar un valor, se debe poner la función "names" y

#seguir la fórmula que se refleja arriba. En cuanto al doble igual "==",

#esta es una función lógica que compara el data.frame.

#Las funciones "names", "==" y "<-" permiten que se pueda modificar el

#nombre que viene predeterminado en el excell para el dataframe del código.

#Ejercicio 3 asigna\_las\_etiquetas

#Ahora se han colocado cuatro etiquetas: Contexto, Conservación, Remache y Materiales. En base a estas cuatro etiquetas,

#se han ido colocando los valores asociados. No hay diferencia entre las dos comillas o la comilla única, algunas veces

#la ciencia es así (Galo, 2024).

#La razón por la cual en algunas ocasiones se ponen los valores #numéricos entre comillas y en otras ocasiones no se debe a procesos #informáticos que escapan de nuestro control (Galo, 2024).

spear**$**Contexto**=**factor**(**spear**$**Contexto, levels**=**c**(**'1','2','3'**)**, labels**=**c**(**"s/c", "Habitacional", "Funerario"**))**

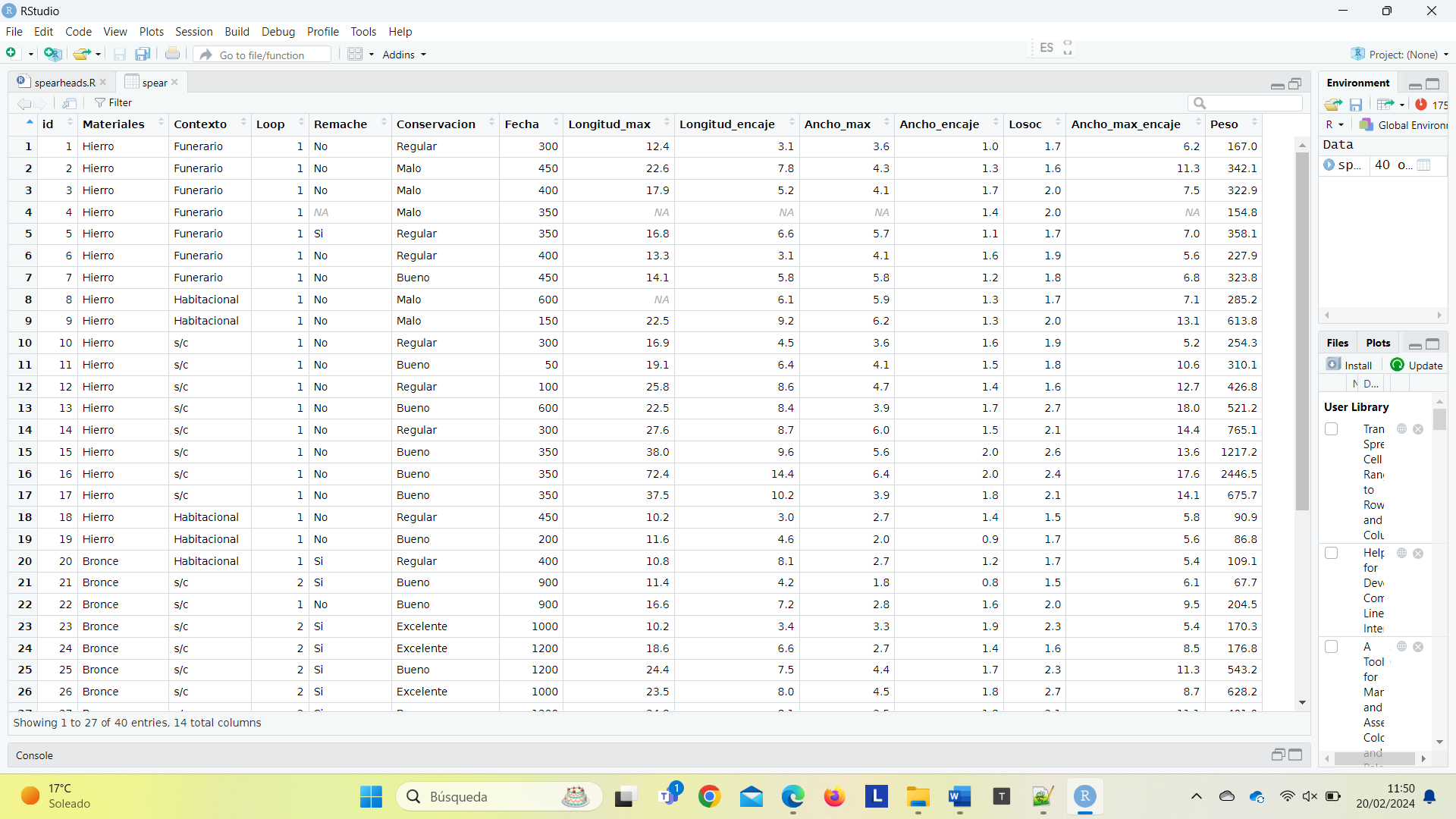
spear**$**Conservacion**=**factor**(**spear**$**Conservacion, levels**=**c**(**1,2,3,4**)**,

labels**=**c**(**"Excelente", "Bueno", "Regular", "Malo"**))**

spear**$**Remache**=**factor**(**spear**$**Remache, levels**=**c**(**1,2**)**, labels**=**c**(**'Si', 'No'**))**

spear**$**Materiales**=**factor**(**spear**$**Materiales, levels**=**c**(**1,2**)**, labels**=**c**(**'Bronce', 'Hierro'**))**

Interfaz de usuario gráfica, Aplicación, Tabla, Excel

Descripción generada automáticamenteView**(**spear**)**

#Ejercicio 4 tablas\_de\_frecuencias\_Materiales\_Contextos\_Conservación

#Esto se realizará a partir de la función table, que permite generar una tabla con los datos

#del data.set.

#Con la función $ se bloquean únicamente aquellos valores que queramos utilizar.

freq.mat**=**table**(**spear**$**Materiales**)**

View**(**freq.mat**)**

freq.con**=**table**(**spear**$**Contexto**)**

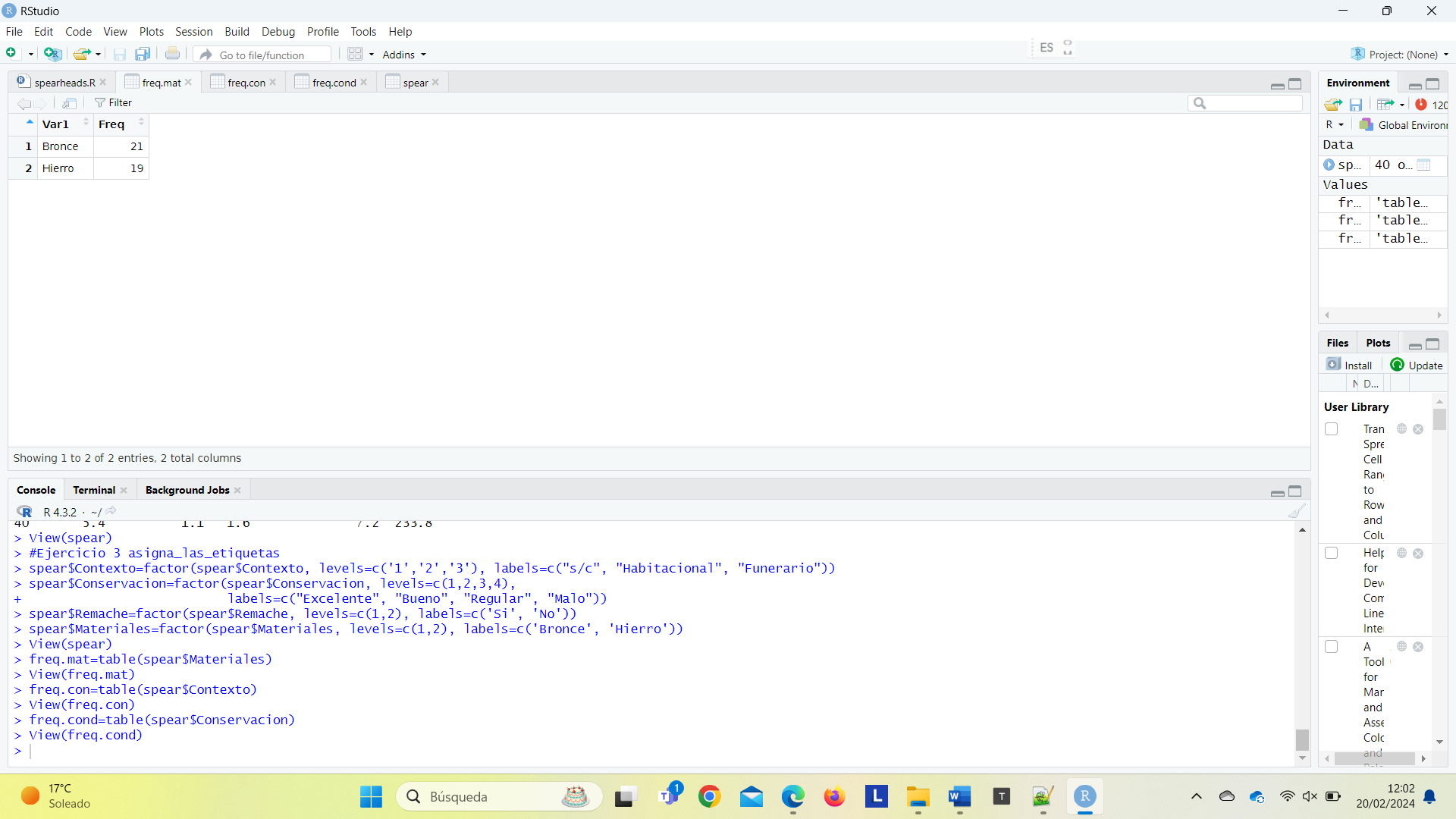
View**(**freq.con**)**

freq.cond**=**table**(**spear**$**Conservacion**)**

View**(**freq.cond**)**

Una captura de pantalla de una computadora

Descripción generada automáticamenteUna captura de pantalla de una computadora

Descripción generada automáticamente

Conservación

Contexto

Materiales

#Ejercicio 5 tablas\_cruzadas

#Se vuelve a emplear la función table para volver a generar otra tabla. Sin embargo,

#en esta ocasión se requiere que se generen tablas cruzadas y no de frecuencia.

#Por ello, se elimna la función freq. Además, en este caso se añaden dos valores

#según la función, y por eso, se pone la función $ entre las dos.

materiales\_contexto **<-** table**(**spear**$**Materiales, spear**$**Contexto**)**

View**(**materiales\_contexto**)**

materiales\_conservacion **<-**table**(**spear**$**Materiales, spear**$**Conservacion**)**

View**(**materiales\_conservacion**)**

Una captura de pantalla de una computadora

Descripción generada automáticamenteUna captura de pantalla de una computadora

Descripción generada automáticamente

Materiales y contexto

Materiales y conservación

#Ejercicio 6 tablas\_de\_porcentajes

#Para realizar tablas de porcentajes, se utiliza la función prop.table.

#Los datos vuelven a seleccionarse específicamente con la $.

#Para poder obtener el porcentaje, se multiplica por 100 cada función.

procentaje\_materiales **<-** prop.table**(**table**(**spear**$**Materiales**))** **\*** 100

View**(**procentaje\_materiales**)**

procentaje\_contexto **<-** prop.table**(**table**(**spear**$**Contexto**))** **\*** 100

View**(**procentaje\_contexto**)**

procentaje\_conservacion **<-** prop.table**(**table**(**spear**$**Conservacion**))** **\*** 100

View**(**procentaje\_conservacion**)**

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamenteInterfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamenteInterfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Conservación

Materiales

Contexto

#Ejercicio 7 tablas\_cruzadas\_de\_porcentajes

#Se vuelve a utilizar la función prop.table para hacer una tabla de frecuencia,

#en donde se vuelve a integrar la $ para identificar los valores que se quieren mantener.

#Como novedad, introducimos la función magin para especficar la dimensión sobre la

#cual se va a realizar el calculo del porcentaje.

porcentaje\_materiales\_contexto **<-** prop.table**(**table**(**spear**$**Materiales,

spear**$**Contexto**)**, margin **=** 1**)** **\*** 100

View**(**porcentaje\_materiales\_contexto**)**

porcentaje\_materiales\_conservacion **<-** prop.table**(**table**(**spear**$**Materiales,

spear**$**Conservacion**)**, margin **=** 1**)** **\*** 100

View**(**porcentaje\_materiales\_conservacion**)**

Una captura de pantalla de una computadora

Descripción generada automáticamenteUna captura de pantalla de una computadora

Descripción generada automáticamente

Materiales - Conservación

Materiales - Contexto

#Ejercicio 8 gráficos\_barras\_verticales

#Ahora se utiliza la función barplot para la elaboración de gráficos de barras verticales.

barras\_verticales\_conservacion **<-** barplot**(**table**(**spear**$**Conservacion**)**,

main **=** "Frecuencia de Conservacion",

xlab **=** "Conservación"**)**

barras\_verticales\_contexto **<-** barplot**(**table**(**spear**$**Contexto**)**,

main **=** "Frecuencia de Contexto",

xlab **=** "Contexto"**)**

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamenteInterfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

#Ejercicio 9 gráficos\_barras\_horizontales

#Ahora se añade la función horiz = True para que los gráficos sean horizontales.

barras\_horizontales\_materiales **<-** barplot**(**table**(**spear**$**Materiales**)**,

horiz **=** **TRUE**, main **=** "Frecuencia de Materiales",

ylab **=** "Materiales"**)**

barras\_horizontales\_remache **<-** barplot**(**table**(**spear**$**Remache**)**,

horiz **=** **TRUE**, main **=** "Frecuencia de Remache",

ylab **=** "Remache"**)**

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamenteInterfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

#Ejercicio 10 gráficos\_barras\_agrupados

#Ahora lo que hay que hacer es seleccionar los datos que se quieren agrupar: $

#Empleamos la función de gráficos de barras: barplot

#Incluimos leyenda con los valores otorgados a las categorías.

barras\_material\_conservacion **<-** barplot.default**(**table**(**spear**$**Conservacion,

spear**$**Materiales**)**, beside **=** **TRUE**,

legend **=** **TRUE**, main **=** "Conservacion por Material",

xlab **=** "Conservacion", col **=** c**(**"purple", "blue", "pink", "orange"**))**

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

#Ejercicio 11 gráfico\_sectores\_variable\_Conservación

#Para ello, se utiliza la función pie.

sectores\_conservacion **<-** pie**(**table**(**spear**$**Conservacion**)**, main **=** "Porcentaje de Conservacion"**)**

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

#Ejercicio 12 histograma\_probabilidad\_variables

#Para ello, se utiliza la función "hist".

#Las variables continuas son el ancho máx y la long máx.

hist**(**spear**$**Longitud\_max, prob **=** **TRUE**, main **=** "Histograma de Probabilidad de Longitud Max", xlab **=** "Longitud Max"**)**

hist**(**spear**$**Ancho\_max, prob **=** **TRUE**, main **=** "Histograma de Probabilidad de Ancho Max", xlab **=** "Ancho Max"**)**

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamenteInterfaz de usuario gráfica, Aplicación

Descripción generada automáticamente