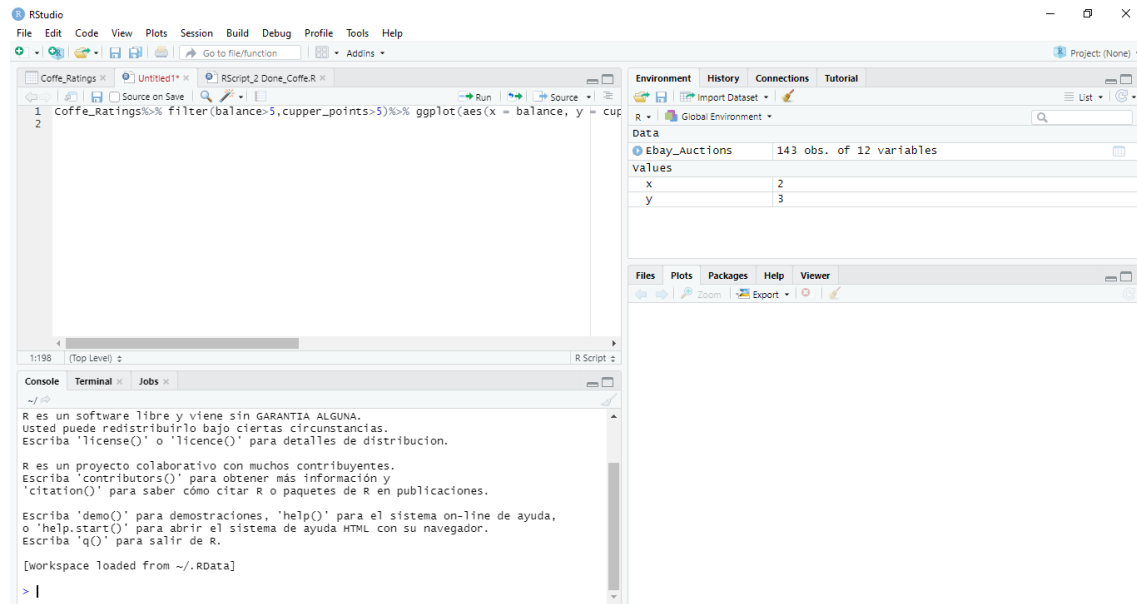


INSTRUCTIONS 1: LET'S SEE

INSTRUCTIONS 1: LET'S SEE

We have to be careful, **everything**, almost, in R is written in NOT capital letters, so we have to write it as I have posted in the examples here

Interface



When you open the interface, you have divided in four different spaces:

- ❖ **1st Quadrant:** Where all the formulas, view of the data and coding is done and **saved**. If you write something here it will get saved.
- ❖ **2nd Quadrant:** Is where you can see the history and environment you have uploaded to R. Everything that has been done in the 1st quadrant will be reflected here in the history.
- ❖ **3rd Quadrant:** On the bottom left is just a reflection of what has been done on the 1st quadrant. However, you can write as well here coding, formulas... but everything will be a **draft**, it won't be saved.
- ❖ **4th Quadrant:** The charts, plots and whatever you process will be reflected in this quadrant. Here is where we are going to take the charts for the projects.

Every Coding

First step

We have to put the **Data Base. Ex.:** Coffee_Ratings

Second step

We have to put this sign: `%>%`

Formula: Coffee_Ratings `%>%`

Third step

Decide if we want to take some outliers out or not with this code: **filter(variable > "Number")**

And again, put the sign: `%>%`

INSTRUCTIONS 1: LET'S SEE

Formula: `Coffee_Ratings %>% filter(sweetness > 5) %>%`

Fourth step

We have to determine which is going to be the variable of x, and other variables if there is more **(see the cheat sheet to know which charts are for more than 1 variable)**.

This is the part which is called **mapping**.

In order to do so, we will put this coding: `ggplot(aes(x = variable 1, y = variable 2))`

Formula: `Coffee_Ratings %>% filter(sweetness > 5) %>% ggplot(aes(x = sweetness, y = flavor))`

Fifth step

We have to decide if we want to divide even more the charts we are going to build with another variable resembling the color or not.

We have to take the coding; **color = variable 3** (preferable a quality variable to do this)

This code has to be inside the ggplot.

Formula: `Coffee_Ratings %>% filter(sweetness > 5) %>% ggplot(aes(x = sweetness, y = flavor, color = species))`

Sixth step

Decide which chart do you want, look at the **cheat sheet in order to know the charts**.

Just remember how the charts start: **geom_**

Formula: `Coffee_Ratings %>% filter(sweetness > 5) %>% ggplot(aes(x = sweetness, y = flavor, color = species) + geom_point() + geom_smooth())`

Note: While the geom point is a graphic, the geom smooth is a correlation line.

Seventh step

Decide if you want to have a chart for every of the numbers, quality variables types you have.

The coding for this is: **face_wrap (~ variable 4)**, (Better a quality variable)

Formula: `Coffee_Ratings %>% filter(sweetness > 5) %>% ggplot(aes(x = sweetness, y = flavor, color = species) + geom_point() + geom_smooth() + face_wrap (~processing_method))`

Other stuff**Line of correlation straight without standard error**

We need to use these two codes inside the geom_smooth formula:

Method = "lm" To have a straight line.

Se = FALSE To not have represented the standard error

Example: `geom_smooth(method = "lm", se = FALSE)`