

Intro to R

Data Visualization with Esquisse

Esquisse Package

```
# install.packages("esquisse")  
library(esquisse)
```

Esquisse Package

The [esquisse package](#) is helpful for getting used to creating plots in R.

It is an interactive tool to help you in RStudio.

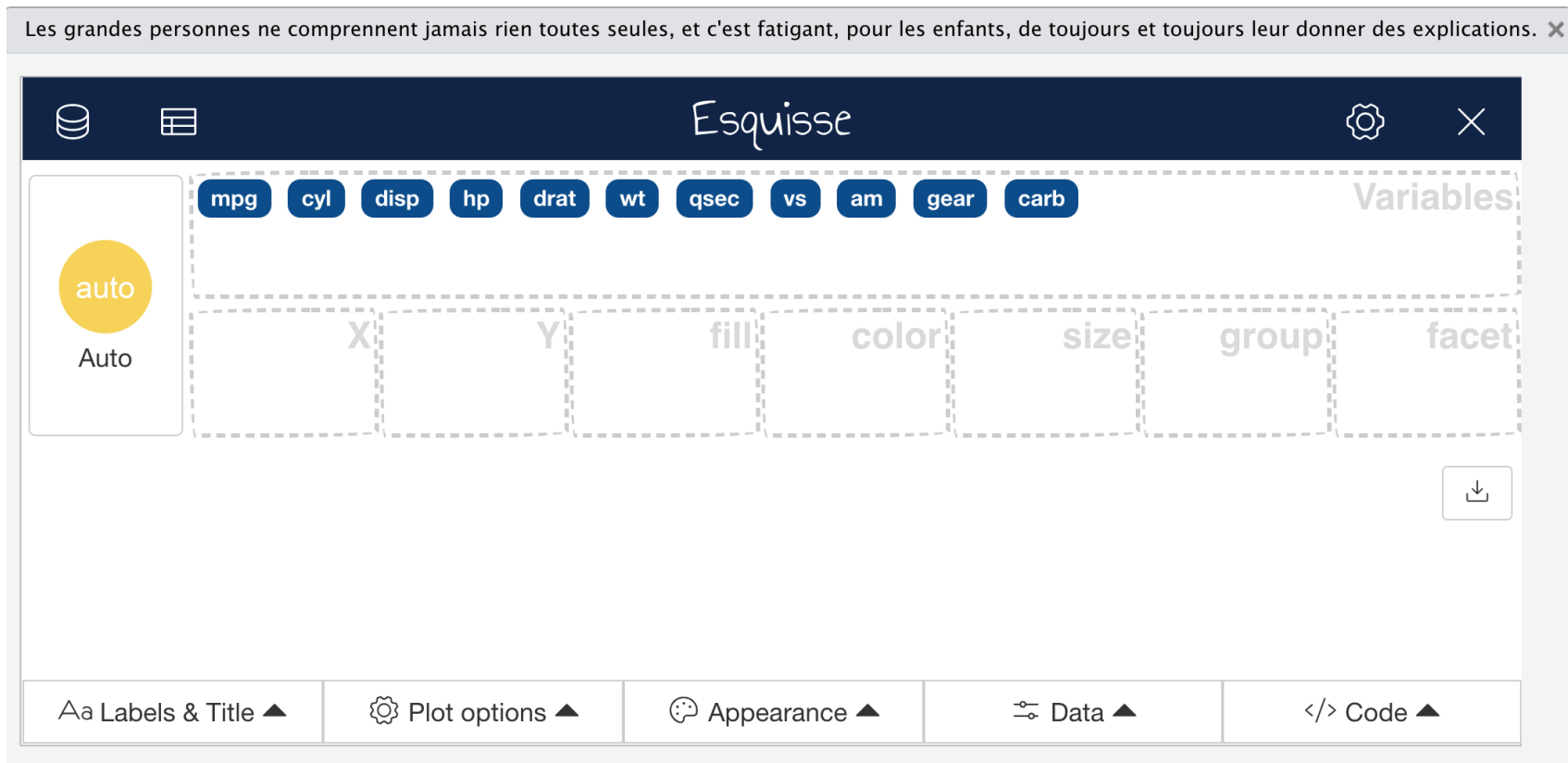
It's super **nifty**!



Starting a plot

Using the `esquisser()` function you can start creating a plot for a `data.frame` or `tibble`. That's it!

```
esquisser(mtcars)
```



Show the plot in the browser

```
esquisse::esquisser(iris, viewer = "browser")
```

Select Variables

To select variables you can drag and drop variables to the respective axis that you would like the variable to be plotted on.

The screenshot shows a software interface for creating a plot. At the top, there is a title bar with the text "Esquisse" and a close button. Below the title bar, a list of variables is displayed: mpg, cyl, disp, hp, drat, wt, qsec, vs, am, gear, and carb. These variables are arranged in a row, each in a blue button. To the right of this list is a "Variables" label. Below the list of variables, there are several dashed boxes representing plot axes and options: X, Y, fill, color, size, group, and facet. A yellow circle with the text "auto" and "Auto" is visible on the left side of the interface. At the bottom, there is a navigation bar with five tabs: "Aa Labels & Title", "Plot options", "Appearance", "Data", and "Code".

Find code

To select variables you can drag and drop variables to the respective axis that you would like the variable to be plotted on.

The screenshot shows the Esquisse interface with a window titled "Esquisse". At the top, there is a header bar with a hamburger menu icon, a list icon, the title "Esquisse", a settings gear icon, and a close "X" icon. Below the header, there is a "Variables" section containing a row of variable buttons: mpg, cyl, disp, hp, drat, wt, qsec, vs, am, gear, and carb. Below this, there are two rows of plot configuration options: the first row has buttons for "mpg" (under X), "cyl" (under Y), "fill", "color", "size", "group", and "facet"; the second row is empty. Below the configuration options is a plot area with a grid and a single data point. At the bottom, there is a navigation bar with five tabs: "Aa Labels & Title", "Plot options", "Appearance", "Data", and "Code". The "Code" tab is highlighted and has a mouse cursor pointing to it.

Change plot type

esquisse automatically assumes a plot type, but you might want to change this.



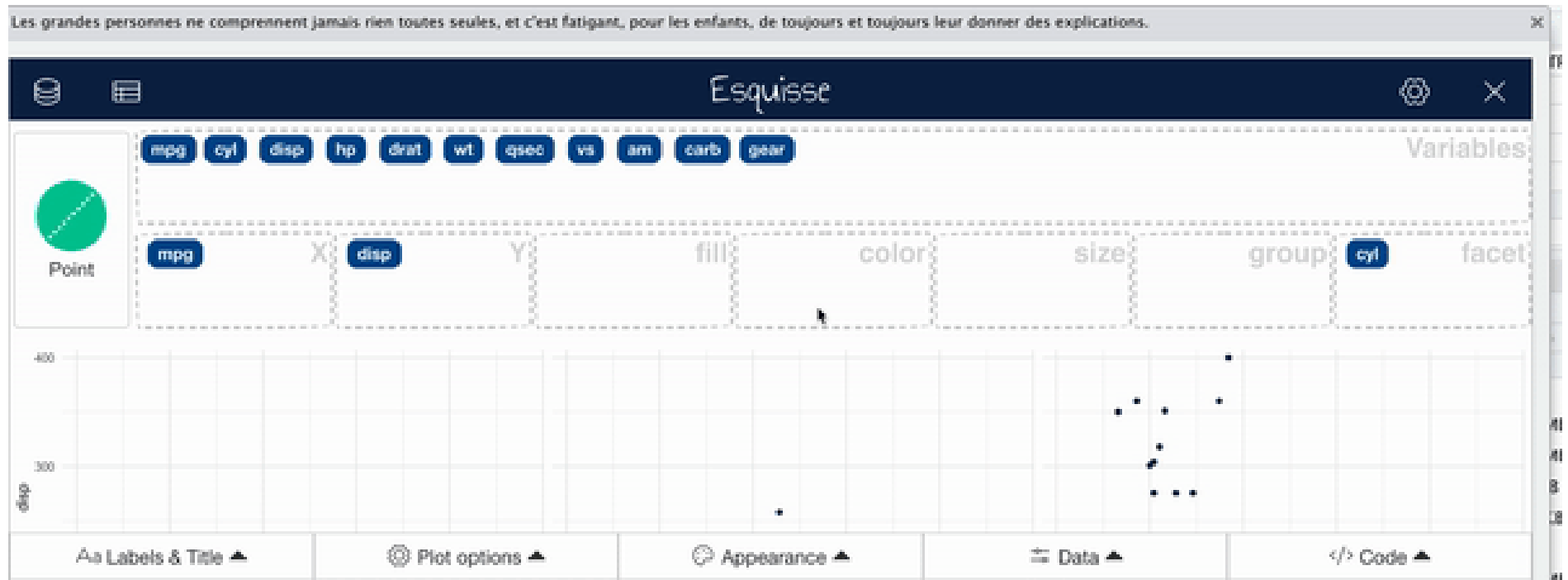
Add Facets

Facets create multiple plots based on the different values of a variable.



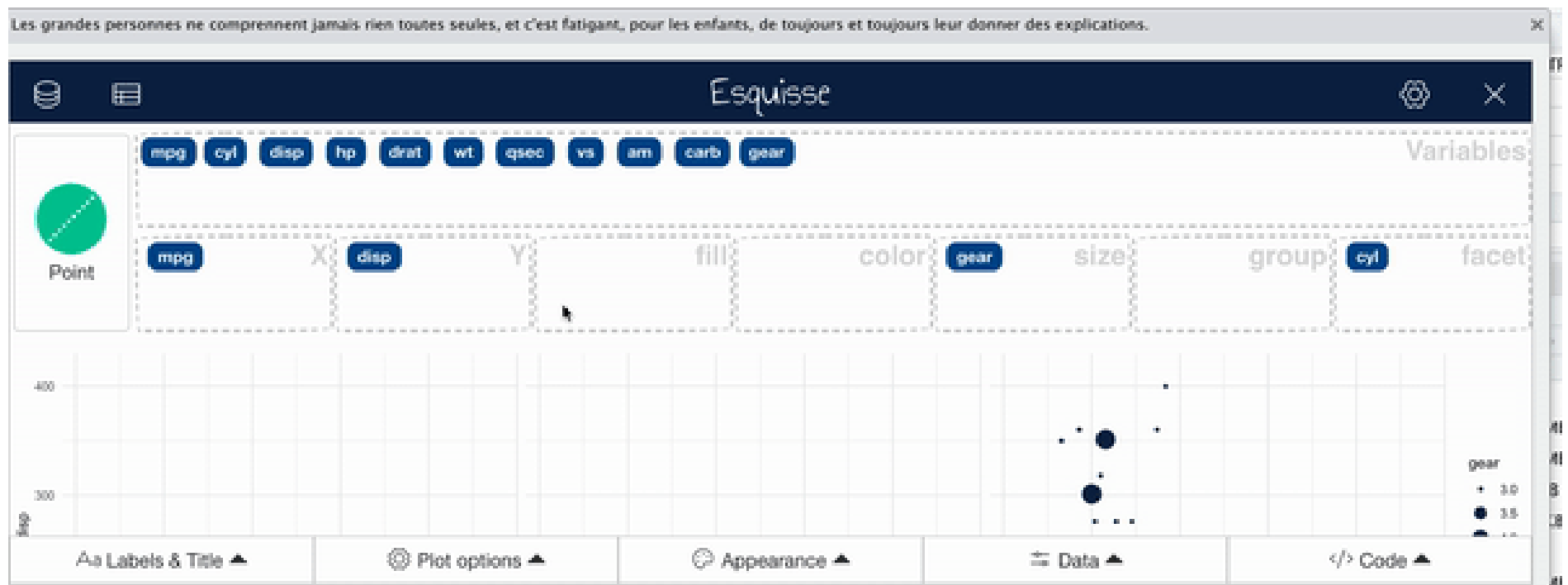
Add size

Sometimes it is useful to change the way points are plotted so that size represents a variable. This can especially be helpful if you need your plot to be black and white.



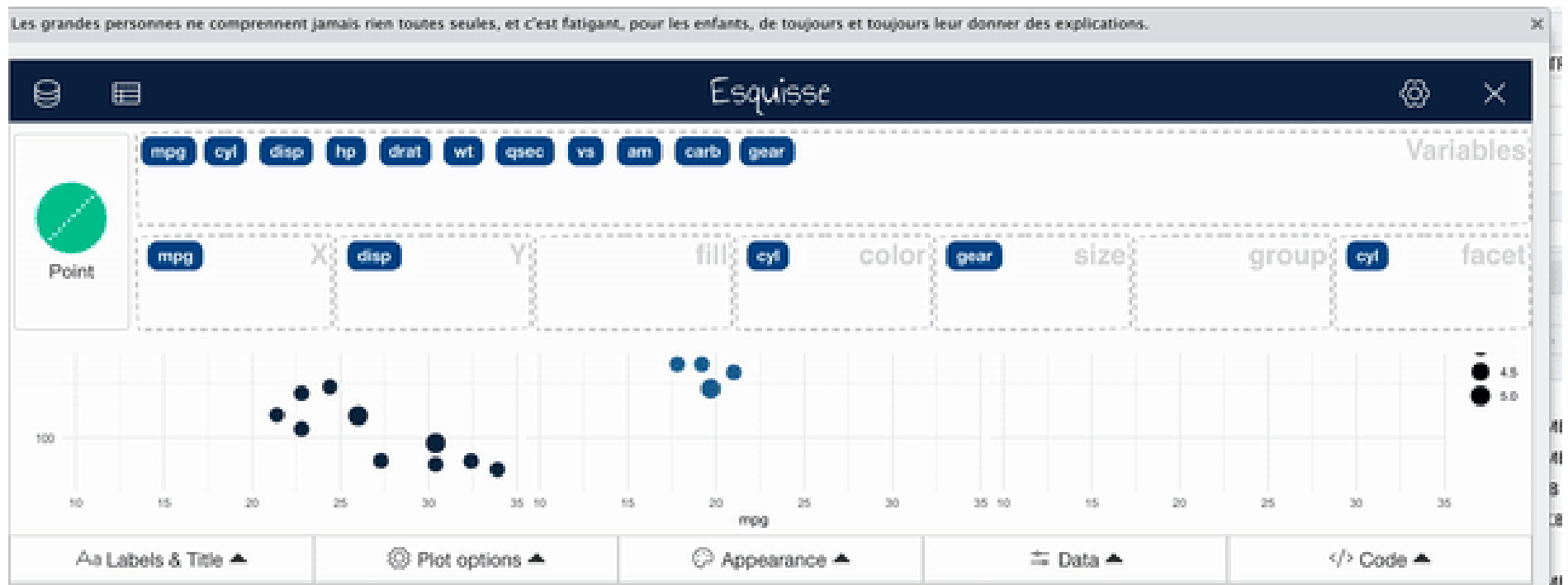
Add color

For plots with points use the color region to change coloring according to a variable. (use "fill" for bar plots)



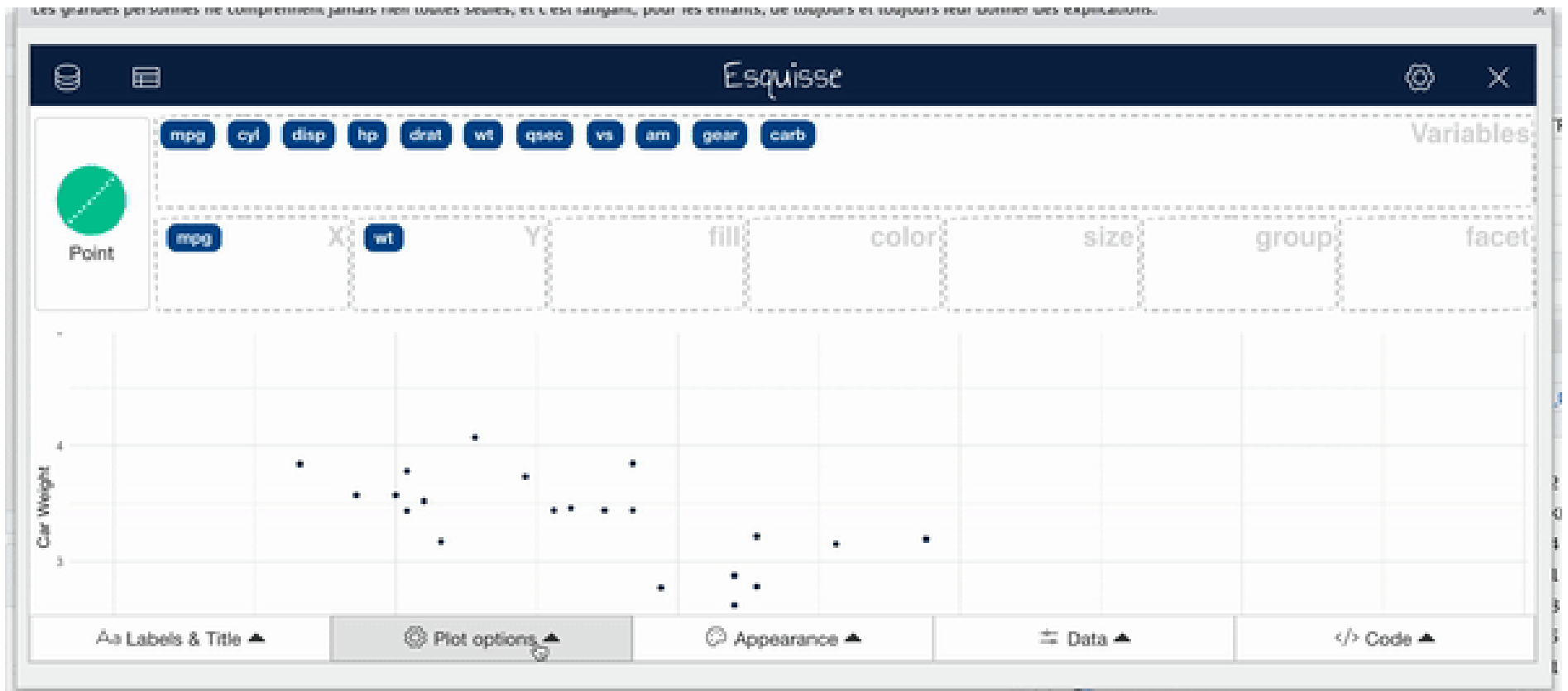
Appearance

You can change the overall appearance with the appearance tab.



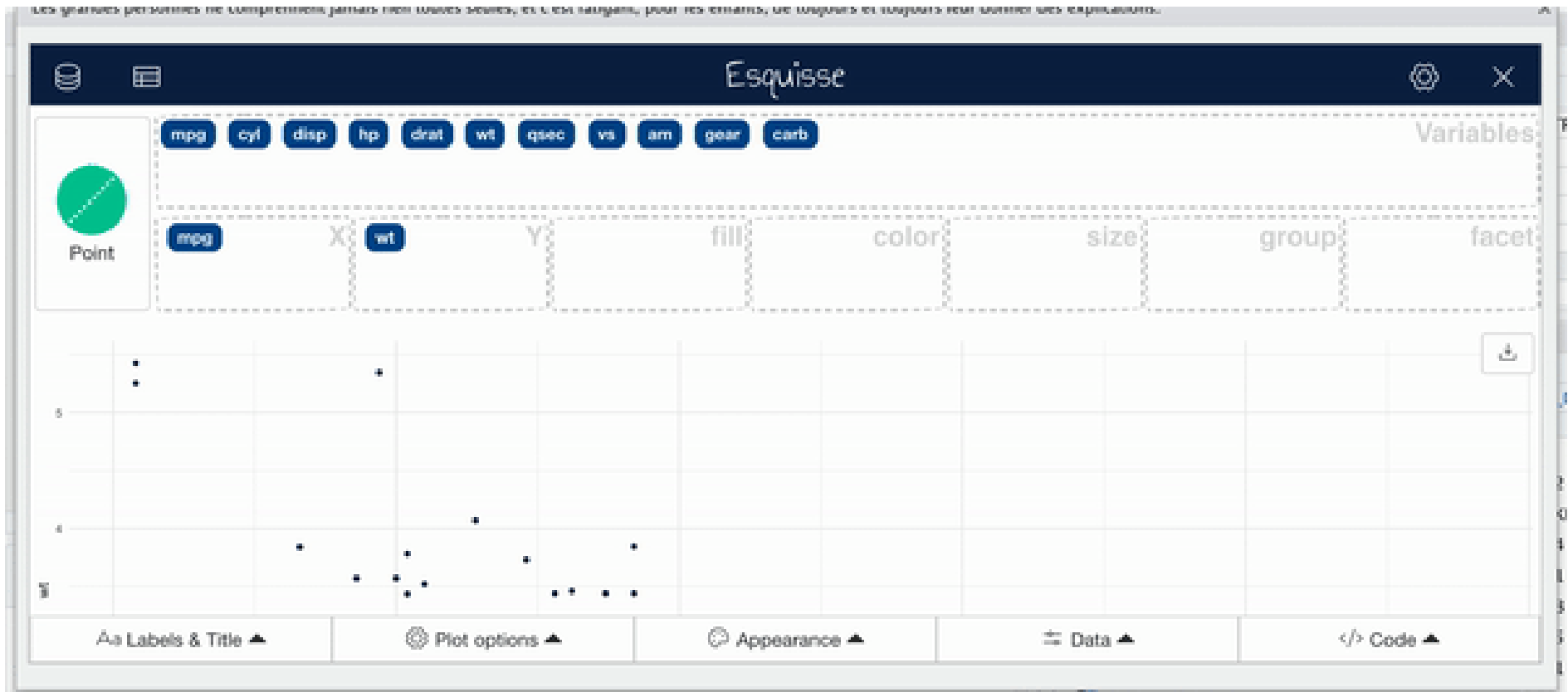
Smooth Lines

Especially when you have a scatter plot, it can be helpful to add a smooth/trend line.



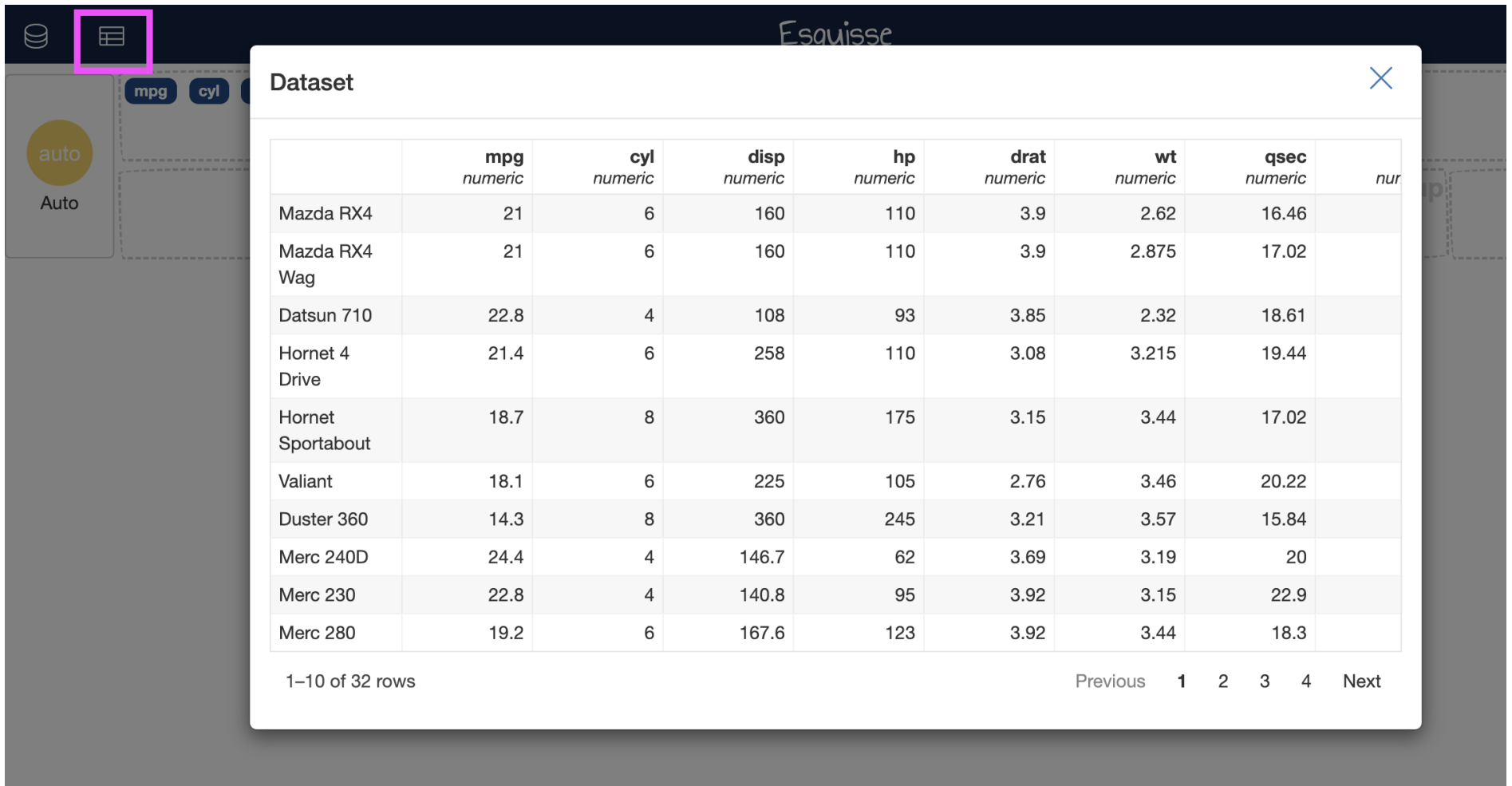
Change titles

To change titles on your plot, use the titles tab.



View data

You can also easily view data



The screenshot shows a data visualization interface with a dark header. A pink box highlights a menu icon in the top left. Below it, there are filters for 'mpg' and 'cyl'. A circular button labeled 'auto' is also visible. The main content area is a 'Dataset' modal window with a close button (X) in the top right. The modal contains a table with 10 rows and 9 columns. The columns are labeled with car specifications and their data types. The rows list various car models and their corresponding values for each specification. At the bottom of the modal, there is a pagination control showing '1-10 of 32 rows' and navigation buttons for 'Previous', '1', '2', '3', '4', and 'Next'.

	mpg <i>numeric</i>	cyl <i>numeric</i>	disp <i>numeric</i>	hp <i>numeric</i>	drat <i>numeric</i>	wt <i>numeric</i>	qsec <i>numeric</i>	nur
Mazda RX4	21	6	160	110	3.9	2.62	16.46	
Mazda RX4 Wag	21	6	160	110	3.9	2.875	17.02	
Datsun 710	22.8	4	108	93	3.85	2.32	18.61	
Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	
Hornet Sportabout	18.7	8	360	175	3.15	3.44	17.02	
Valiant	18.1	6	225	105	2.76	3.46	20.22	
Duster 360	14.3	8	360	245	3.21	3.57	15.84	
Merc 240D	24.4	4	146.7	62	3.69	3.19	20	
Merc 230	22.8	4	140.8	95	3.92	3.15	22.9	
Merc 280	19.2	6	167.6	123	3.92	3.44	18.3	

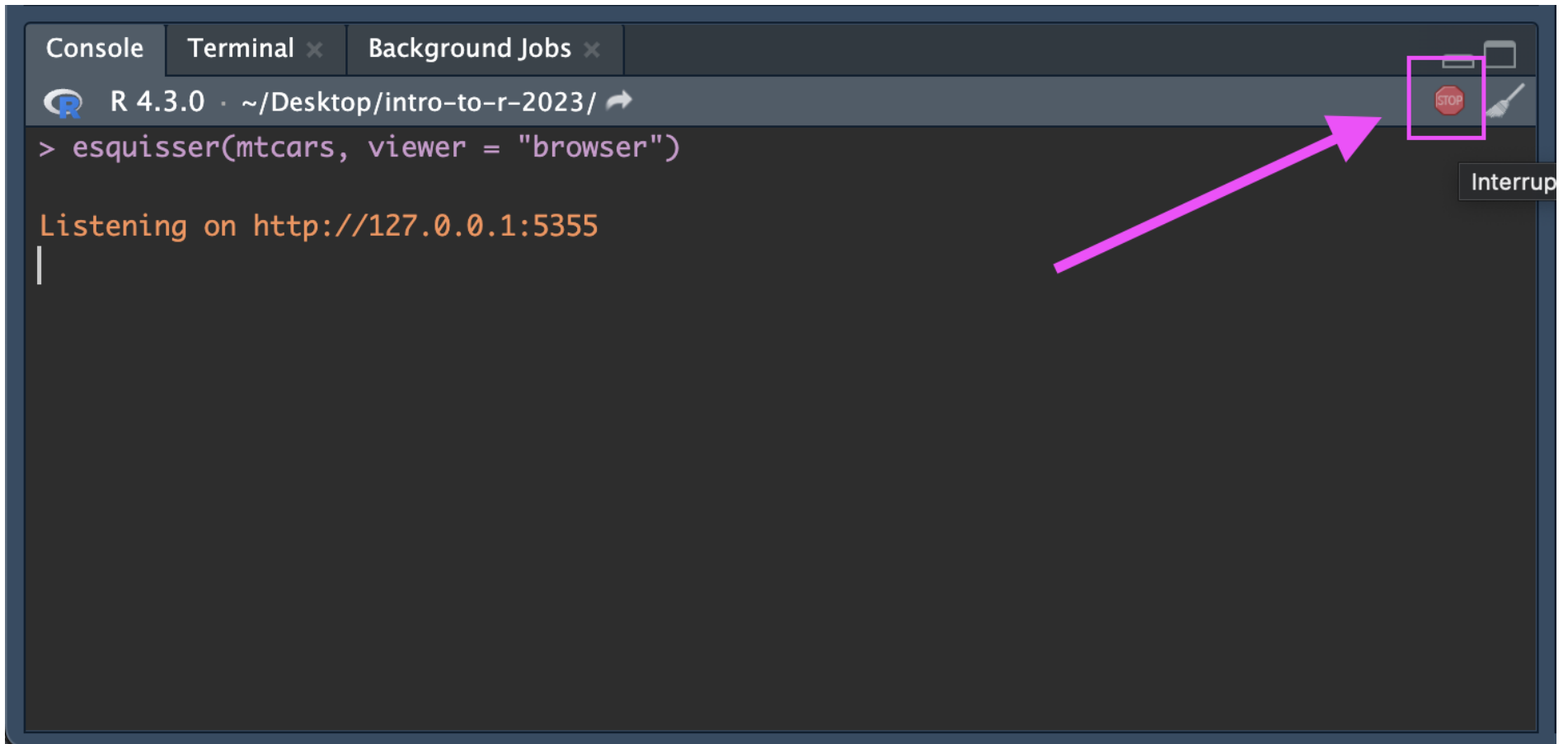
1-10 of 32 rows

Previous 1 2 3 4 Next

Interrupting Esquisse

You'll need to "interrupt" Esquisse to launch it with a new dataset.

Use the stop button or press ctrl+c to stop the Esquisse app.



The screenshot shows an R console window with the following content:

```
Console Terminal x Background Jobs x  
R 4.3.0 · ~/Desktop/intro-to-r-2023/ ↵  
> esquisser(mtcars, viewer = "browser")  
Listening on http://127.0.0.1:5355  
|
```

A pink arrow points to a red octagonal "STOP" button in the top right corner of the console window. A tooltip labeled "Interrupt" is visible next to the button.

Wide & Long Data Example

```
wide_circ <- read_csv("https://hutchdatascience.org/SeattleStatSummer_R/data/C

## Rows: 1146 Columns: 15
## — Column specification _____
## Delimiter: ","
## chr (2): day, date
## dbl (13): orangeBoardings, orangeAlightings, orangeAverage, purpleBoardings
##
## □ Use `spec()` to retrieve the full column specification for this data.
## □ Specify the column types or set `show_col_types = FALSE` to quiet this me
```

Wide Data

```
library(dplyr)
glimpse(wide_circ)
```

```
## Rows: 1,146
## Columns: 15
## $ day <chr> "Monday", "Tuesday", "Wednesday", "Thursday", "Fri
## $ date <chr> "01/11/2010", "01/12/2010", "01/13/2010", "01/14/2
## $ orangeBoardings <dbl> 877, 777, 1203, 1194, 1645, 1457, 839, 999, 1023,
## $ orangeAlightings <dbl> 1027, 815, 1220, 1233, 1643, 1524, 938, 1000, 1047
## $ orangeAverage <dbl> 952.0, 796.0, 1211.5, 1213.5, 1644.0, 1490.5, 888.
## $ purpleBoardings <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA
## $ purpleAlightings <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA
## $ purpleAverage <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA
## $ greenBoardings <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA
## $ greenAlightings <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA
## $ greenAverage <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA
## $ bannerBoardings <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA
## $ bannerAlightings <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA
## $ bannerAverage <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA
## $ daily <dbl> 952.0, 796.0, 1211.5, 1213.5, 1644.0, 1490.5, 888.
```

Long Data

```
library(tidyr)
long_circ <- wide_circ %>%
  pivot_longer(
    cols = contains(c("boarding")),
    names_to = "Route",
    values_to = "Boardings"
  )
```

Long Data

```
glimpse(long_circ)
```

```
## Rows: 4,584
## Columns: 13
## $ day <chr> "Monday", "Monday", "Monday", "Monday", "Tuesday",
## $ date <chr> "01/11/2010", "01/11/2010", "01/11/2010", "01/11/2010",
## $ orangeAlightings <dbl> 1027, 1027, 1027, 1027, 815, 815, 815, 815, 1220,
## $ orangeAverage <dbl> 952.0, 952.0, 952.0, 952.0, 796.0, 796.0, 796.0, 796.0,
## $ purpleAlightings <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
## $ purpleAverage <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
## $ greenAlightings <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
## $ greenAverage <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
## $ bannerAlightings <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
## $ bannerAverage <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
## $ daily <dbl> 952.0, 952.0, 952.0, 952.0, 796.0, 796.0, 796.0, 796.0,
## $ Route <chr> "orangeBoardings", "purpleBoardings", "greenBoardings",
## $ Boardings <dbl> 877, NA, NA, NA, 777, NA, NA, NA, 1203, NA, NA, NA, NA, NA, NA,
```

Make a plot of boardings by day for different routes

```
esquisser(wide_circ) # days as x...? Tricky!  
esquisser(long_circ) # day as x, Boardings as y, Route as fill
```

Summary

- Use the `esquisser()` function on a dataset
- Use the `viewer = "browser"` argument to launch in your browser.
- Code from Esquisse can be copied into code chunks to be generated in the "Plots" pane
- It's easier if your code is in "long" form!

[Workshop Website](#)