How can the View() function be used in R and what are some examples?

Authored by

stats writer

July 1, 2024

The View() function in R is a useful tool for quickly viewing data frames and matrices. It allows users to interactively explore the data and make modifications if needed. This function provides a spreadsheet-like interface where users can easily scroll through the data, sort columns, and filter rows. It also allows for the addition of new variables and the editing of existing ones. Overall, the View() function is ideal for visually inspecting data and performing simple data manipulations.

One example of using the View() function is to examine a large dataset with multiple variables. By calling the View() function, users can easily scroll through the data and identify any outliers or patterns. Additionally, it can be used to merge or join data frames by visually comparing them side by side.

Another example is when performing data cleaning or data wrangling tasks. The View() function allows users to quickly identify missing values or incorrect data entries, and make necessary changes directly in the interface.

In summary, the View() function in R is a versatile tool that provides a user-friendly interface for data exploration and manipulation, making it an essential function for data analysis and management tasks.

Use the View() Function in R (With Examples)

The View() function in R can be used to invoke a spreadsheet-style data viewer within RStudio.

This function uses the following syntax:

View(df)

Note: Make sure you type a capital "V" when using this function.

The following example shows how to use this syntax in practice.

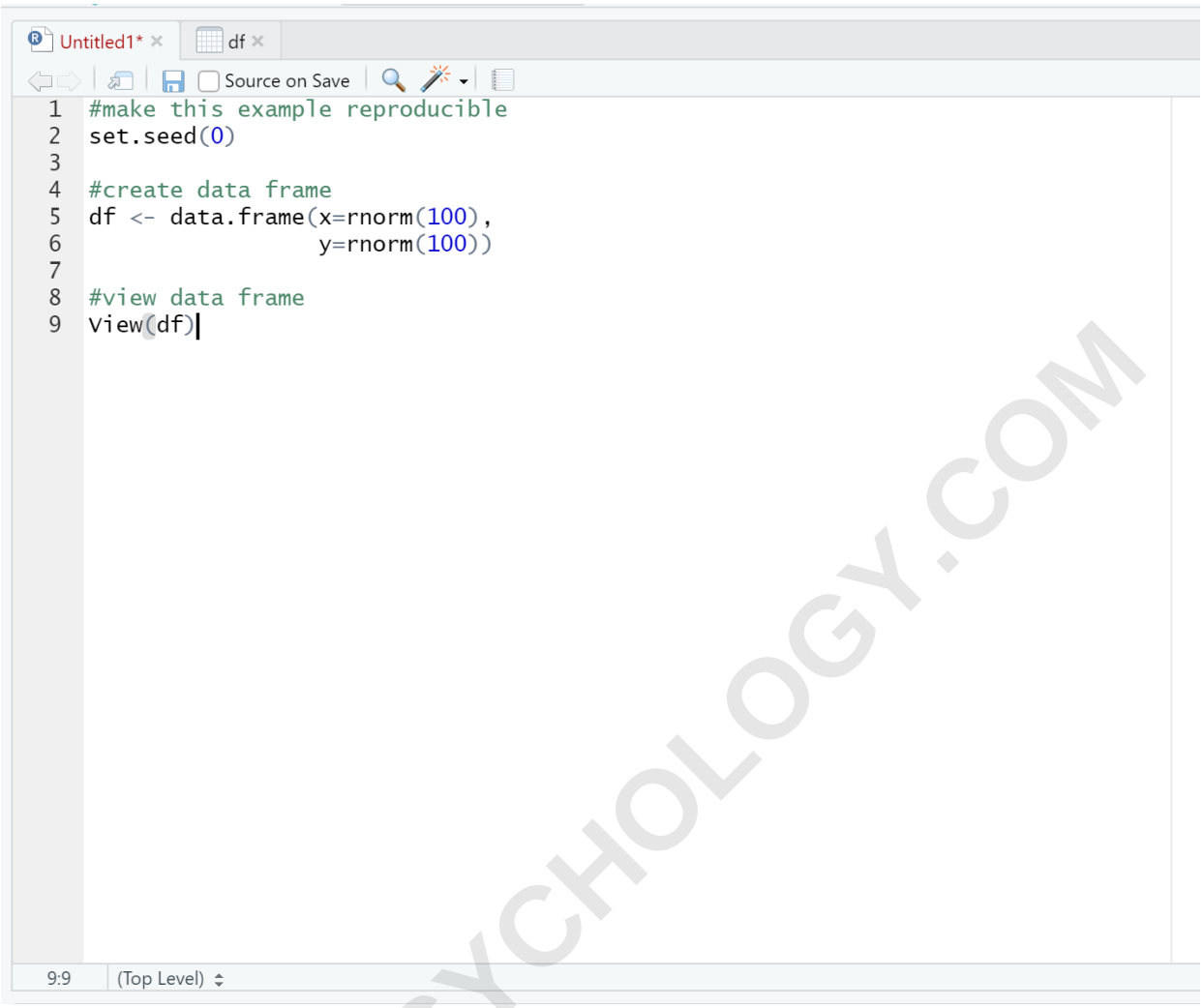
How to Use the View() Function

We can use the following code to create a data frame in R with 100 rows and 2 columns:

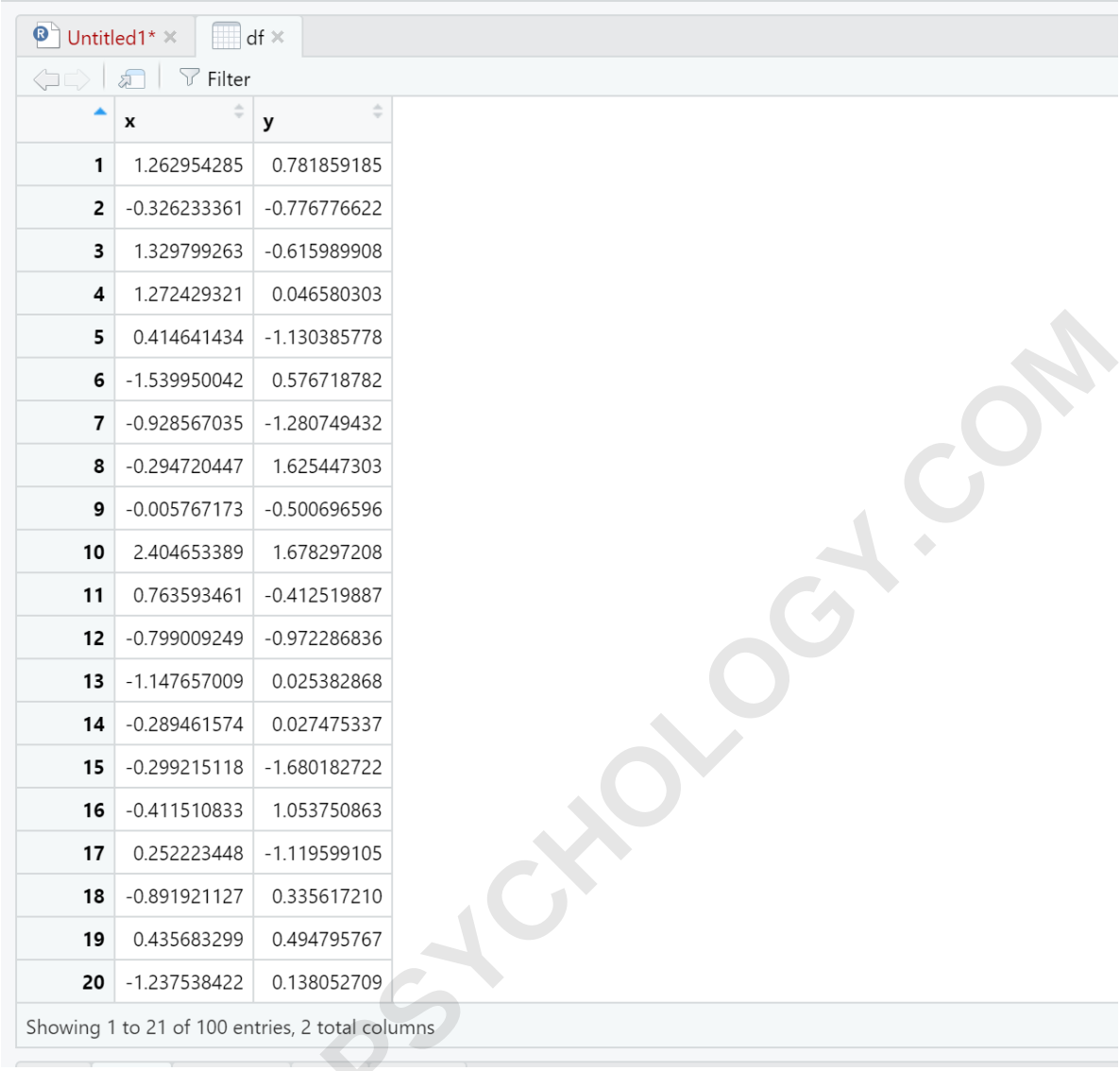
```
#make this example reproducible
set.seed(0)

#create data frame
df <- data.frame(x=rnorm(100),
y=rnorm(100))
```

We can then use the View() function to invoke a spreadsheet-style data viewer within RStudio:



Notice that a new tab appears in Rstudio that provides an interactive display of the data frame we just created:



	x	y
1	1.262954285	0.781859185
2	-0.326233361	-0.776776622
3	1.329799263	-0.615989908
4	1.272429321	0.046580303
5	0.414641434	-1.130385778
6	-1.539950042	0.576718782
7	-0.928567035	-1.280749432
8	-0.294720447	1.625447303
9	-0.005767173	-0.500696596
10	2.404653389	1.678297208
11	0.763593461	-0.412519887
12	-0.799009249	-0.972286836
13	-1.147657009	0.025382868
14	-0.289461574	0.027475337
15	-0.299215118	-1.680182722
16	-0.411510833	1.053750863
17	0.252223448	-1.119599105
18	-0.891921127	0.335617210
19	0.435683299	0.494795767
20	-1.237538422	0.138052709

Showing 1 to 21 of 100 entries, 2 total columns

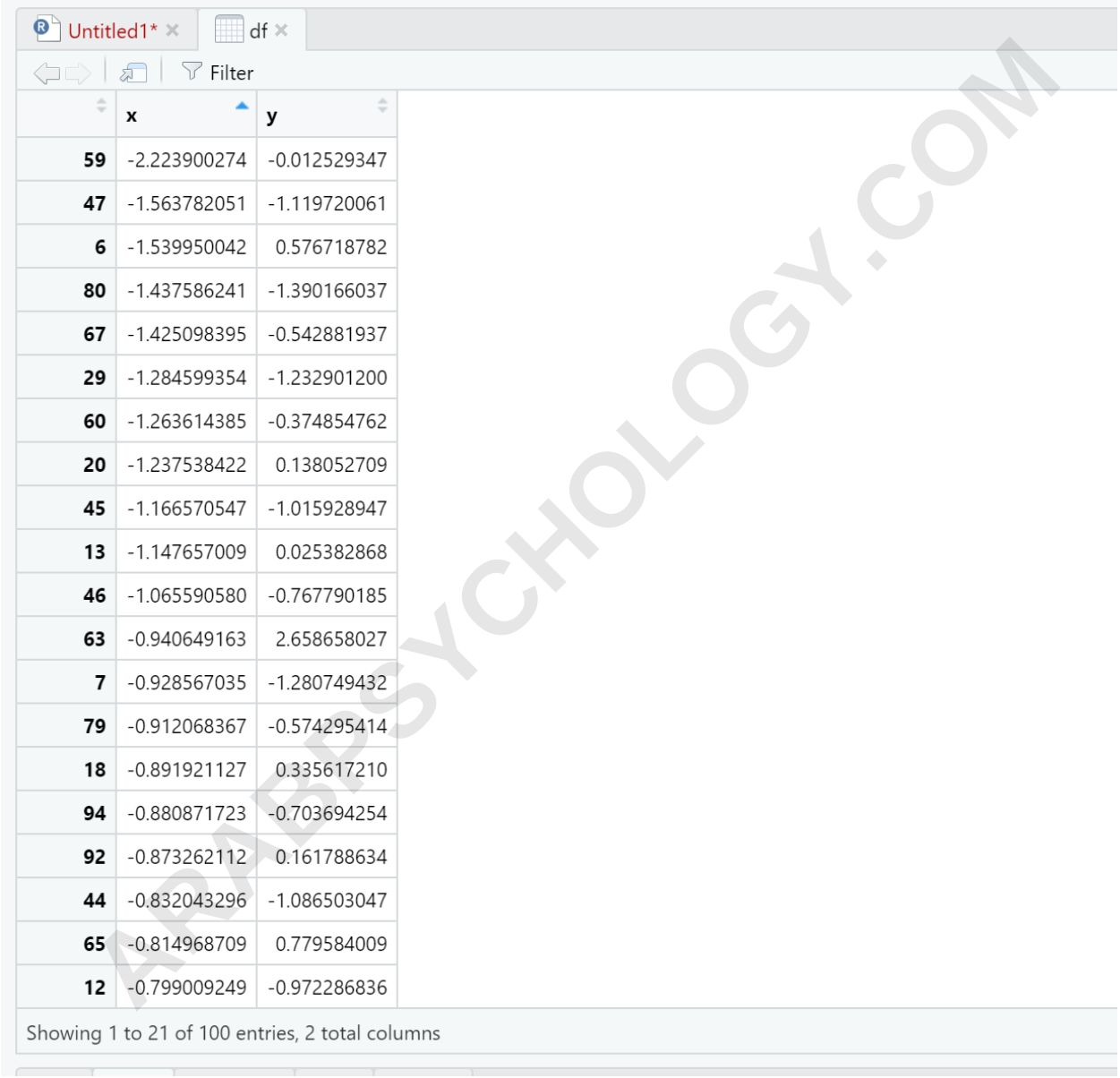
At the bottom of the viewer, we can see the size of the data frame: 100 entries (i.e. rows) and 2 columns.

How to Sort Data Using the View() Function

We can also quickly sort the data frame by clicking on one of the columns.

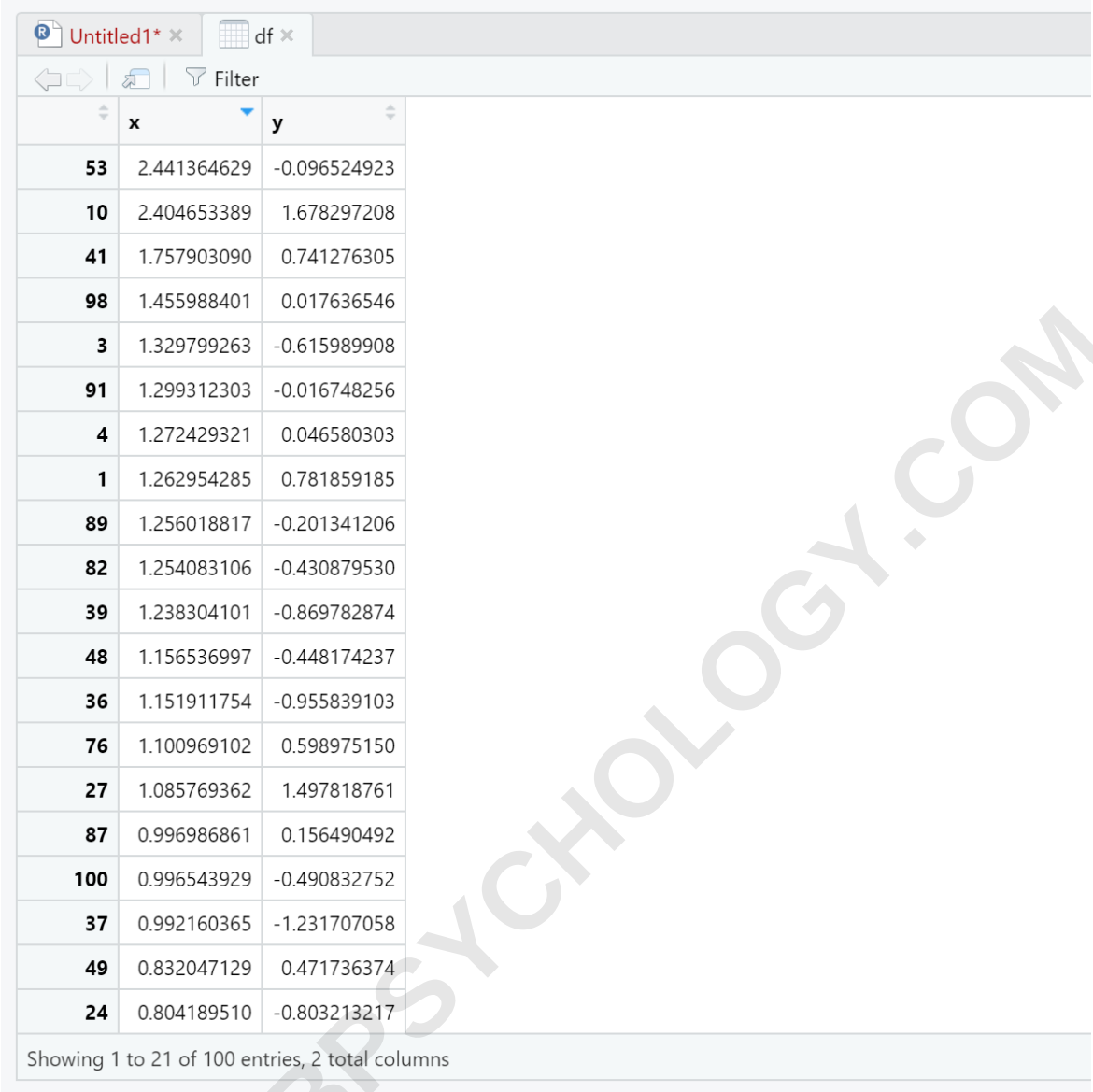
For example, if I click on the header for column x then

the rows of the data frame will automatically be sorted from smallest to largest based on the values in column x:



	x	y
59	-2.223900274	-0.012529347
47	-1.563782051	-1.119720061
6	-1.539950042	0.576718782
80	-1.437586241	-1.390166037
67	-1.425098395	-0.542881937
29	-1.284599354	-1.232901200
60	-1.263614385	-0.374854762
20	-1.237538422	0.138052709
45	-1.166570547	-1.015928947
13	-1.147657009	0.025382868
46	-1.065590580	-0.767790185
63	-0.940649163	2.658658027
7	-0.928567035	-1.280749432
79	-0.912068367	-0.574295414
18	-0.891921127	0.335617210
94	-0.880871723	-0.703694254
92	-0.873262112	0.161788634
44	-0.832043296	-1.086503047
65	-0.814968709	0.779584009
12	-0.799009249	-0.972286836

Showing 1 to 21 of 100 entries, 2 total columns



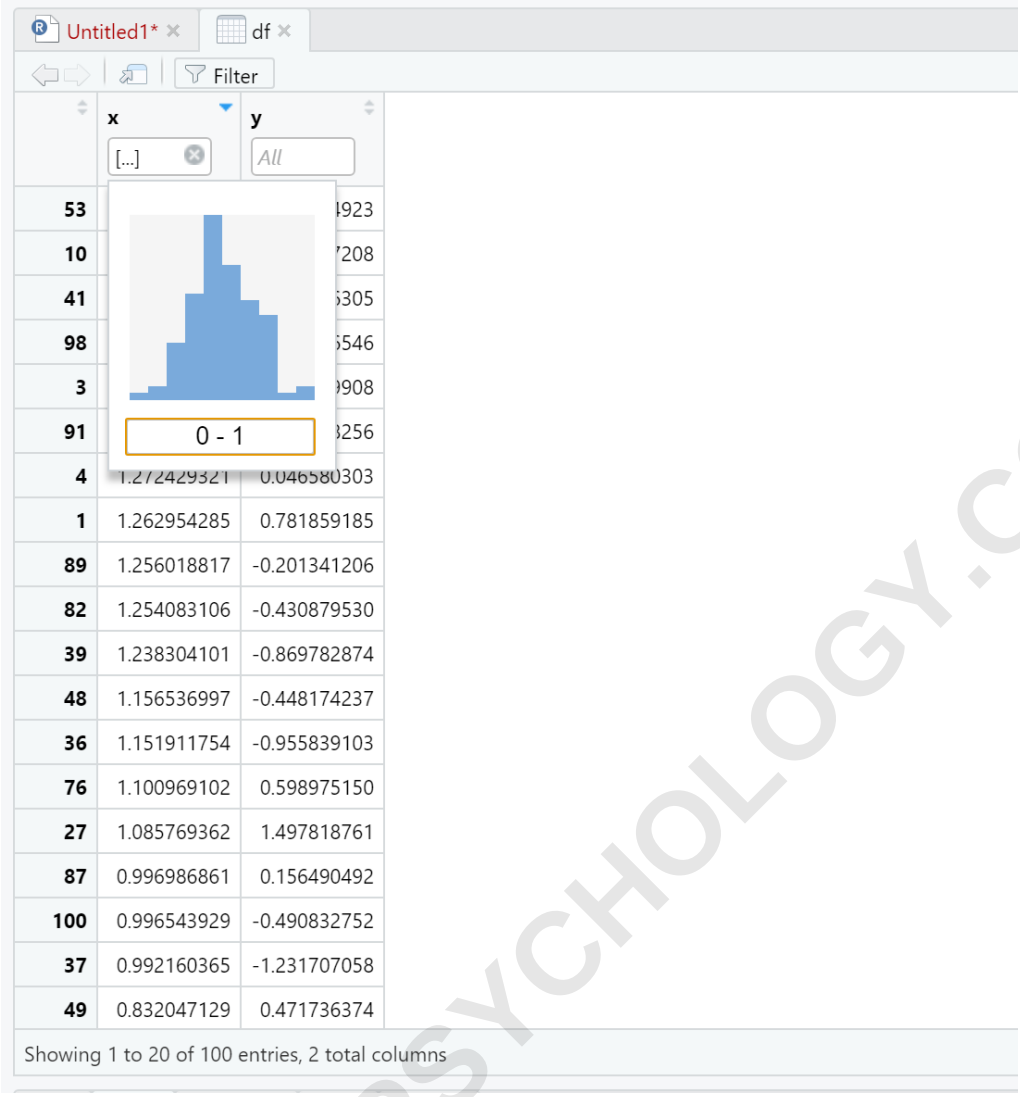
	x	y
53	2.441364629	-0.096524923
10	2.404653389	1.678297208
41	1.757903090	0.741276305
98	1.455988401	0.017636546
3	1.329799263	-0.615989908
91	1.299312303	-0.016748256
4	1.272429321	0.046580303
1	1.262954285	0.781859185
89	1.256018817	-0.201341206
82	1.254083106	-0.430879530
39	1.238304101	-0.869782874
48	1.156536997	-0.448174237
36	1.151911754	-0.955839103
76	1.100969102	0.598975150
27	1.085769362	1.497818761
87	0.996986861	0.156490492
100	0.996543929	-0.490832752
37	0.992160365	-1.231707058
49	0.832047129	0.471736374
24	0.804189510	-0.803213217

Showing 1 to 21 of 100 entries, 2 total columns

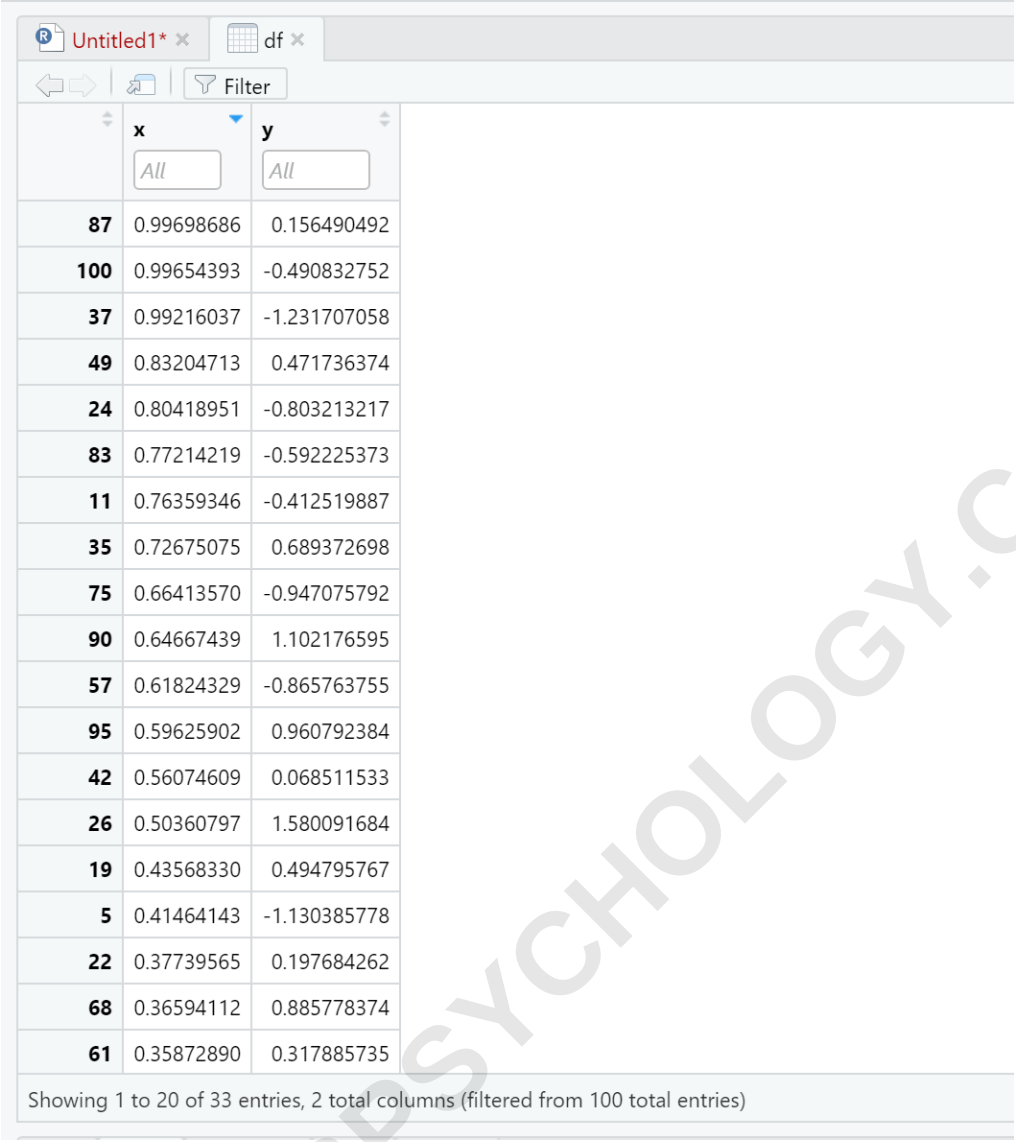
How to Filter Data Using the View() Function

I can also quickly filter the data frame by clicking the Filter icon, then clicking one of the column names, then typing in a range of values.

For example, I may choose to filter the data frame to only show the rows where x is between 0 and 1:



Once I press enter, the data frame will automatically be filtered:



	x	y
	All	All
87	0.99698686	0.156490492
100	0.99654393	-0.490832752
37	0.99216037	-1.231707058
49	0.83204713	0.471736374
24	0.80418951	-0.803213217
83	0.77214219	-0.592225373
11	0.76359346	-0.412519887
35	0.72675075	0.689372698
75	0.66413570	-0.947075792
90	0.64667439	1.102176595
57	0.61824329	-0.865763755
95	0.59625902	0.960792384
42	0.56074609	0.068511533
26	0.50360797	1.580091684
19	0.43568330	0.494795767
5	0.41464143	-1.130385778
22	0.37739565	0.197684262
68	0.36594112	0.885778374
61	0.35872890	0.317885735

Showing 1 to 20 of 33 entries, 2 total columns (filtered from 100 total entries)

At the bottom of the screen we can see that 33 rows have values in the x column between 0 and 1.

Note that I can also add a filter in the y column to filter by specific values in both x and y.

Additional Resources