

Explore data and R

- Video: Data in R  
1 min
- Video: R data frames  
4 min
- Video: Working with data frames  
4 min
- Practice Quiz: Hands-On Activity: Create your own data frame  
1 question
- Reading: More about tibbles  
10 min
- Reading: Data-import basics  
10 min
- Practice Quiz: Hands-On Activity: Importing and working with data  
1 question
- Discussion Prompt: Data in R versus SQL  
10 min
- Practice Quiz: Test your knowledge on R data frames  
4 questions

Cleaning data

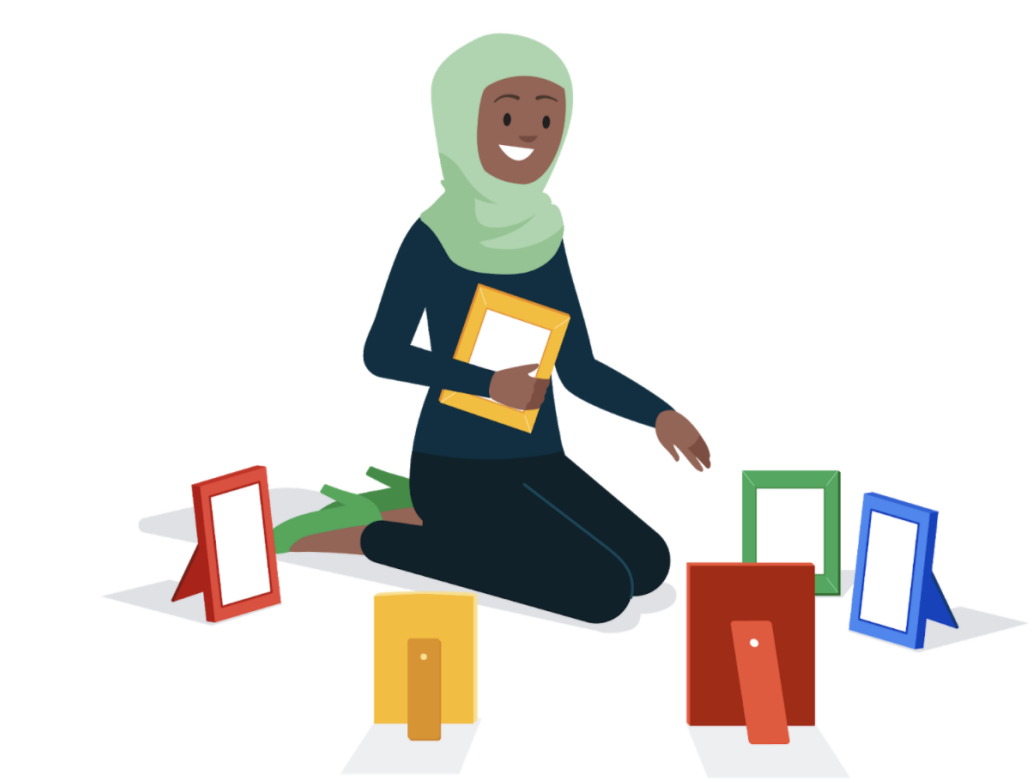
Take a closer look at the data

Weekly challenge 3

# More about tibbles

In this reading, you will learn about tibbles, which are a super useful tool for organizing data in R. You will get a review of what tibbles are, how they differ from standard data frames, and how to create them in R.

## Tibbles



Tibbles are a little different from standard data frames. A data frame is a collection of columns, like a spreadsheet or a SQL table. Tibbles are like streamlined data frames that are automatically set to pull up only the first 10 rows of a dataset, and only as many columns as can fit on the screen. This is really useful when you're working with large sets of data. Unlike data frames, tibbles never change the names of your variables, or the data types of your inputs. Overall, you can make more changes to data frames, but tibbles are easier to use. The tibble package is part of the core tidyverse. So, if you've already installed the tidyverse, you have what you need to start working with tibbles.

### Creating tibbles

Now, let's go through an example of how to create a tibble in R. You can use the pre-loaded *diamonds* dataset that you're familiar with from earlier videos. As a reminder, the *diamonds* dataset includes information about different diamond qualities, like carat, cut, color, clarity, and more.

You can load the dataset with the **data()** function using the following code:

```
library(tidyverse)
```

```
data(diamonds)
```

Then, let's add the data frame to our data viewer in RStudio with the **View()** function.

```
View(diamonds)
```

The dataset has 10 columns and thousands of rows. This image displays part of the data frame:

	carat	cut	color	clarity	depth	table	price	x	y	z
1	0.23	Ideal	E	SI2	61.5	55.0	326	3.95	3.98	2.43
2	0.21	Premium	E	SI1	59.8	61.0	326	3.89	3.84	2.31
3	0.23	Good	E	VS1	56.9	65.0	327	4.05	4.07	2.31
4	0.29	Premium	I	VS2	62.4	58.0	334	4.20	4.23	2.63
5	0.31	Good	J	SI2	63.3	58.0	335	4.34	4.35	2.75
6	0.24	Very Good	J	VVS2	62.8	57.0	336	3.94	3.96	2.48
7	0.24	Very Good	I	VVS1	62.3	57.0	336	3.95	3.98	2.47
8	0.26	Very Good	H	SI1	61.9	55.0	337	4.07	4.11	2.53
9	0.22	Fair	E	VS2	65.1	61.0	337	3.87	3.78	2.49
10	0.23	Very Good	H	VS1	59.4	61.0	338	4.00	4.05	2.39
11	0.30	Good	J	SI1	64.0	55.0	339	4.25	4.28	2.73
12	0.23	Ideal	J	VS1	62.8	56.0	340	3.93	3.90	2.46
13	0.22	Premium	F	SI1	60.4	61.0	342	3.88	3.84	2.33
14	0.31	Ideal	J	SI2	62.2	54.0	344	4.35	4.37	2.71
15	0.20	Premium	E	SI2	60.2	62.0	345	3.79	3.75	2.27
16	0.32	Premium	E	I1	60.9	58.0	345	4.38	4.42	2.68
17	0.30	Ideal	I	SI2	62.0	54.0	348	4.31	4.34	2.68
18	0.30	Good	J	SI1	63.4	54.0	351	4.23	4.29	2.70
19	0.30	Good	J	SI1	63.8	56.0	351	4.23	4.26	2.71
20	0.30	Very Good	J	SI1	62.7	59.0	351	4.21	4.27	2.66
21	0.30	Good	I	SI2	63.3	56.0	351	4.26	4.30	2.71

Now let's create a tibble from the same dataset. You can create a tibble from existing data with the **as\_tibble()** function. Indicate what data you'd like to use in the parentheses of the function. In this case, you will put the word "diamonds."

```
as_tibble(diamonds)
```

### Results

When you run the function, you get a tibble of the *diamonds* dataset.

```
# A tibble: 53,940 x 10
  carat cut    color clarity depth table price    x    y    z
<dbl> <ord> <ord> <ord> <dbl> <dbl> <int> <dbl> <dbl> <dbl>
1 0.23 Ideal E      SI2     61.5  55    326  3.95  3.98  2.43
2 0.21 Prem... E      SI1     59.8  61    326  3.89  3.84  2.31
3 0.23 Good E      VS1     56.9  65    327  4.05  4.07  2.31
4 0.290 Prem... I      VS2     62.4  58    334  4.2  4.23  2.63
5 0.31 Good J      SI2     63.3  58    335  4.34  4.35  2.75
6 0.24 Very... J      VVS2    62.8  57    336  3.94  3.96  2.48
7 0.24 Very... I      VVS1    62.3  57    336  3.95  3.98  2.47
8 0.26 Very... H      SI1     61.9  55    337  4.07  4.11  2.53
9 0.22 Fair E      VS2     65.1  61    337  3.87  3.78  2.49
10 0.23 Very... H      VS1     59.4  61    338  4  4.05  2.39
# ... with 53,930 more rows
```

While RStudio's built-in data frame tool returns thousands of rows in the *diamonds* dataset, the tibble only returns the first 10 rows in a neatly organized table. That makes it easier to view and print.

### Additional resources

For more information on tibbles, check out the following resources:

- The entry for [Tibble](#) in the tidyverse documentation summarizes what a tibble is and how it works in R code. If you want a quick overview of the essentials, this is the place to go.
- The [Tidy chapter](#) in "A Tidyverse Cookbook" is a great resource if you want to learn more about how to work with tibbles using R code. The chapter explores a variety of R functions that can help you create and transform tibbles to organize and tidy your data.

### Mark as completed