

How can pivot tables be quickly created in R?

Authored by
stats writer

April 25, 2024

RECOMMENDED CITATION

stats writer (2024). *How can pivot tables be quickly created in R?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=139349>

Pivot tables in R can be quickly created by utilizing the "reshape2" package, which allows for efficient data manipulation and organization. The process involves specifying the desired variables to be used as rows and columns, as well as the function to be applied to the data. This allows for easy and quick summarization and visualization of large datasets. Additionally, R offers a user-friendly interface with customizable options for creating pivot tables, making it a convenient and efficient tool for data analysis.

Quickly Create Pivot Tables in R

In Excel, pivot tables offer an easy way to group and summarize data.

For example, if we have the following dataset in Excel then we can use a pivot table to quickly summarize the total sales by region:

	A	B	C	D	E	F
1	region	device	sales			
2	A	X	12		Row Labels	Sum of sales
3	A	X	18		A	51
4	A	Y	21		B	85
5	B	X	22		C	140
6	B	Y	34		Grand Total	276
7	B	Y	29			
8	C	X	38			
9	C	X	36			
10	C	Y	34			
11	C	Y	32			
12						
13						
14						
15						
16						
17						
18						

This tells us:

**Region A had 51 total sales
Region B had 85 total sales
Region C had 140 total sales**

Or we could summarize by another metric such as the average sales by region:

	A	B	C	D	E	F	G
1	region	device	sales				
2	A	X	12		Row Labels	Average of sales	
3	A	X	18		A	17	
4	A	Y	21		B	28.33	
5	B	X	22		C	35	
6	B	Y	34		Grand Total	27.6	
7	B	Y	29				
8	C	X	38				
9	C	X	36				
10	C	Y	34				
11	C	Y	32				
12							
13							
14							
15							
16							
17							
18							
19							
20							

It turns out that we can quickly create similar pivot tables in R by using the `group_by()` and `summarize()` functions from the package.

This tutorial provides several examples of how to do so.

Example: Create Pivot Tables in R

First, let's create the same dataset in R that we used in the previous examples from Excel:

```
#create data frame
```

```
df <- data.frame(region=c('A', 'A', 'A', 'B', 'B', 'B', 'C', 'C',  
'C', 'C'),  
device=c('X', 'X', 'Y', 'X', 'Y', 'Y', 'X', 'X', 'Y', 'Y'),  
sales=c(12, 18, 21, 22, 34, 29, 38, 36, 34, 32))
```

```
#view data frame
```

```
df
```

```
region device sales
```

```
1 A X 12
```

```
2 A X 18
```

```
3 A Y 21
```

```
4 B X 22
```

```
5 B Y 34
```

```
6 B Y 29
```

```
7 C X 38
```

```
8 C X 36
```

```
9 C Y 34
```

```
10 C Y 32
```

Next, let's load the `dplyr` package and use the `group_by()` and `summarize()` functions to group by region and find the sum of sales by region:

```
library(dplyr)#find sum of sales by region
```

```
df %>%
```

```
group_by(region) %>%
```

```
summarize(sum_sales = sum(sales))
```

```
# A tibble: 3 x 2
```

```
region sum_sales
```

```
1 A 51
```

```
2 B 85
```

```
3 C 140
```

We can see that these numbers match the numbers shown in the introductory Excel example.

We can also calculate the average sales by region:

```
#find average sales by region
```

```
df %>%
```

```
group_by(region) %>%
```

```
summarize(mean_sales = mean(sales))
```

```
# A tibble: 3 x 2
region mean_sales
```

```
1 A 17
2 B 28.3
3 C 35
```

Note that we can also group by multiple variables. For example, we could find the sum of sales grouped by region *and* device type:

```
#find sum of sales by region and device type
df %>%
group_by(region, device) %>%
summarize(sum_sales = sum(sales))
```

```
# A tibble: 6 x 3
# Groups:   region
region device sum_sales
```

```
1 A X 30
2 A Y 21
3 B X 22
4 B Y 63
5 C X 74
```

6 C Y 66

ARABPSYCHOLOGY.COM