R Tooklit Digest

...dose of hidden gems, packages, and practical R tricks...

Reshaping Survey Data in R

by Roy Mwavita
#RToolkitDigest

Why Does Data Structure Matter?

Survey datasets are often collected in a wide format.

But many R functions — especially ggplot2 — work better with long format data.

Understanding how to reshape data helps us:

- clean survey datasets
- summarize responses easily
- create better visualizations
- prepare data for dashboards and reports

The Two Main Functions

`pivot_longer()`

Converts **wide → long**

`pivot_wider()`

Converts **long → wide**

Both functions come from the **tidyr** package.

Example Survey Dataset

Suppose respondents answered three questions.

respondent	marrital_status	country	job_type	reliigon
1	Singal	Kenya	Employed	Christian
2	Married	Tanzania	Self Employed	Muslim
3	Divorced	Kenya	Freelancing	Christian
4	Widowed	Uganda	Unemployed	Hindu

What do we notice? Each question has its own column: `marital_status`, `country`, `job_type`, `religion`

This structure is common in KoboToolbox, ODK, and survey exports. Basically, the data is in wide format.

Understanding the `pivot_longer()`

```
pivot_longer(
  cols = ___,
  names_to = ___,
  values_to = ___
)
```

What happens?

- `cols` selects columns to reshape (or exclude using -)
- `names_to` creates a new column containing question names
- `values_to` stores the responses

Advanced `cols` selection options

You can use tidyselect helpers:

```
# Select all columns whose names begin with "q"
cols = starts_with("q")
```

```
# Select all columns ending in "_score"
cols = ends_with("_score")
```

```
# Select columns that contain the word "age" anywhere
cols = contains("age")
```

```
# Uses regex to select columns with numeric names
cols = matches("^[0-9]+$")
```

```
# Manually select specific columns
cols = -c(respondent, id)
```

Converting Wide → Long

Using `pivot_longer()`

```r
long_data <- survey_data %>%
  pivot_longer(
    cols = -respondent,
    names_to = "question",
    values_to = "answer"
  )
```

respondent	question	answer
1	marrital_status	Singal
1	country	Kenya
1	job_type	Employed
1	reliigon	Christian
2	marrital_status	Married
2	country	Tanzania
2	job_type	Self Employed
2	reliigon	Muslim
3	marrital_status	Divorced
3	country	Kenya
3	job_type	Freelancing
3	reliigon	Christian
4	marrital_status	Widowed
4	country	Uganda
4	job_type	Unemployed
4	reliigon	Hindu

Why Long Format is Useful

Instead of multiple question columns, we now have:

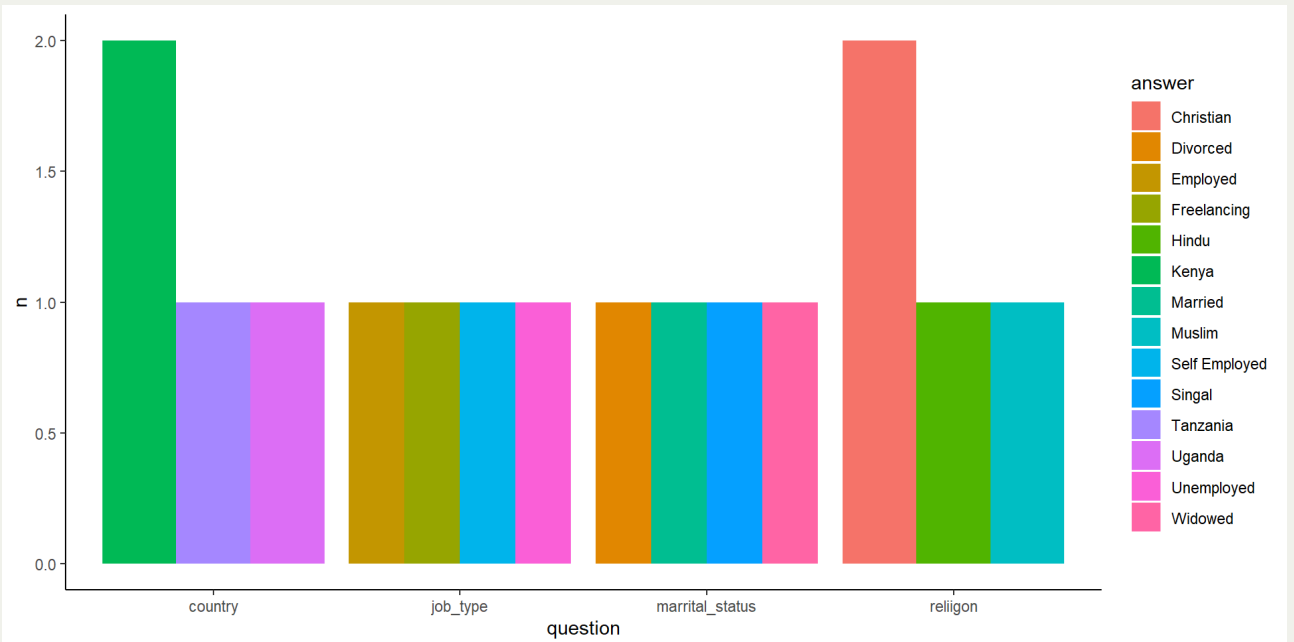
- one question column
- one answer column

Long format works especially well for:

- ggplot2 visualizations
- grouped summaries
- dashboards
- filtering responses
- survey analysis

Example Visualization

```r
long_data %>%
  count(question, answer) %>%
  ggplot(aes(
    x = question,
    y = n,
    fill = answer
  )) +
  geom_col(position = "dodge") +
  theme_classic()
```



Converting Long → Wide

Sometimes we need to return data back to wide format.

That's where pivot_wider() helps.

```
1
2 wide_data <- long_data %>%
3   pivot_wider(
4     names_from = question,
5     values_from = answer
6   )
```

```
# A tibble: 4 × 5
  respondent marrital_status country  job_type      reliigon
       <dbl> <chr>           <chr>    <chr>         <chr>
1          1 Singal          Kenya    Employed      Christian
2          2 Married         Tanzania Self Employed Muslim
3          3 Divorced        Kenya    Freelancing   Christian
4          4 Widowed         Uganda   Unemployed    Hindu
```

Understanding the `pivot_wider()`

```
pivot_wider(
  names_from = ___,
  values_from = ___
)
```

What happens?

- `names_from` creates new column names
- `values_from` fills those columns with values

Practical Applications

These functions are extremely useful for:

- KoboToolbox survey data
- ODK exports
- household assessments
- monitoring tools
- beneficiary tracking datasets

Why This Matters

- Good data structure makes analysis easier.
- Sometimes the challenge is not the visualization itself — it's organizing the data into the right format first.
- Learning to reshape data is one of the most useful R skills for survey analysis.

Practice

Copy and run the code below in R. Explore the dataset, then practice reshaping it using`pivot_longer()` and converting it back with `pivot_wider()`.

```r
library(tidyverse)

stack_overflow_survey <- read_csv(
  "https://github.com/StackExchange/Survey/raw/refs/heads/main/packages/archive/2024/results.csv"
  )
```

```
# A tibble: 2 × 114
  ResponseId MainBranch       Age   Employment RemoteWork Check CodingActivities
       <dbl> <chr>            <chr> <chr>      <chr>      <chr> <chr>
1          1 I am a develope… Unde… Employed,… Remote     Appl… Hobby
2          2 I am a develope… 35-4… Employed,… Remote     Appl… Hobby;Contribut…
# ℹ 107 more variables: EdLevel <chr>, LearnCode <chr>, LearnCodeOnline <chr>,
#   TechDoc <chr>, YearsCode <chr>, YearsCodePro <chr>, DevType <chr>,
#   OrgSize <chr>, PurchaseInfluence <chr>, BuyNewTool <chr>, BuildvsBuy <chr>,
#   TechEndorse <chr>, Country <chr>, Currency <chr>, CompTotal <dbl>,
#   LanguageHaveWorkedWith <chr>, LanguageWantToWorkWith <chr>,
#   LanguageAdmired <chr>, DatabaseHaveWorkedWith <chr>,
#   DatabaseWantToWorkWith <chr>, DatabaseAdmired <chr>, …
```

 Further Reading

- **Pivot data from long to wide**. Link
- **Pivoting Data with tidyr**. Link
- **tidyr official site**: https://tidyr.tidyverse.org/

More sample data to work with:

- **Stack Overflow Annual Developer Survey**. https://survey.stackoverflow.co/