```
---
title: 'Function Review — Unite () '
author: 'William David Hiott, Sr.'
date: 'April 11, 2024'
---
```

R Package Review
Due: Wed Apr 10, 2024 11:59pm
History 8510 at Clemson University

    Will — tidyr::unite()

R Function Review in .Qmd document:
History 8510 at Clemson University

SECTION I

1. Function and purpose

   Q: What function are you covering?

   A: My function review is unite() which is part of the tidyr package.

   Q: What does it do?

   A: Basically, the unite() function allows for combining separate columns into a new
composite column.

SECTION II

2. Scenarios & Tasks

   Q: What scenarios might this function be useful in?

   A: The function might be most useful in aiding in making some data more tidy where names,
dates, or other idenifiers are in miltiple columns.

   Q: What kinds of tasks and/or questions could you use it to address?

   A: For example, if you were using a data set that had Social Security numbers, not likely,
but for illustration purposes those might be separted into three columns from each
hyphenated section. Then if you desired to have those three numbers combined into one column
then a unite() function would complete the task.

SECTION III:

3. USE & EXAMPLES

How do you use it?

Explanation and examples of how this function works are below.
You can run a chunk to pull up the Help documentation.
```{r}
?unite
```
Or you can follow along on this copy.
WDH:
Now if we look at the R Help window, the topic will be described as follows:

unite {tidyr}   R Documentation

WDH:
The Main function is to:

Unite multiple columns into one by pasting strings together

WDH:
Unite is called a convenience function, so think of a convenience store as a quick and
simple task that makes life easier.

Description
Convenience function to paste together multiple columns into one.

WDH:
The usage formula is cited as:

Usage
unite(data, col, ..., sep = "_", remove = TRUE, na.rm = FALSE)

Arguments
data   A data frame.

col   The name of the new column, as a string or symbol.

Description:
This argument is passed by expression and supports quasiquotation (you can unquote strings
and symbols). The name is captured from the expression with rlang::ensym() (note that this
kind of interface where symbols do not represent actual objects is now discouraged in the
tidyverse; we support it here for backward compatibility).

...   <tidy-select> Columns to unite

sep   Separator to use between values.

remove
If TRUE, remove input columns from output data frame.

na.rm   If TRUE, missing values will be removed prior to uniting each value.

See Also   separate(), the complement.

Examples----------------------

WDH:
Run examples from R Studio.

First, the cited example creates a simple column chart.
```{r}
df <- expand_grid(x = c("a", NA), y = c("b", NA))
df
```
A Help example is shown as taking two columns above x and y and unite () or merge into z.
```{r}
df %>% unite("z", x:y, remove = FALSE)
# To remove missing values:
df %>% unite("z", x:y, na.rm = TRUE, remove = FALSE)
```
The oppostie of unite () is the separate () function as in the example below:
```{r}
# Separate is almost the complement of unite
df %>%
  unite("xy", x:y) %>%
  separate(xy, c("x", "y"))
# (but note `x` and `y` contain now "NA" not NA)
```

Examples for 'tidyr::unite'
Unite multiple columns into one by pasting strings together
Aliases: unite

Nest let's look a a more practical example from a previous worksheet relating to columns in
the BWV.

If you need to run any libraries, below is just a reminder from the recent topic modeling
worksheet.
```{r}
library(tidytext)
library(tidyverse)
library(tm)

```

A question from a previous worksheet asked to combine two columns from the Boston Women
Voters.

Since this is a rather large dataset, the print out below is just the head or six entries.

```{r}
data("BostonWomenVoters")
head(BostonWomenVoters)
```
WDH:
As some of you may recall, the data in the Boston Womens Voters had an odd arrangement of
two columns with Street.Number.on.April.1 & Street.of.Residence.on.April.1., but not an
entire street address in one column.

In the example below, to solve that issue of distinct columns using unite() one could, for
example from the Boston Women Voters, create a separate Mailing Address in this example that
combined the two number and street to make a cleaner and more efficient full mailing
address.

You may remember from class that the model used, sep = "_"
But to clean up those addresses, as pointed out in class, you can use the quotations with
nothing for, sep = " "

```{r}
BostonWomenVoters %>% unite("Mailing Address", "Street.Number.on.April.1" :
"Street.of.Residence.on.April.1", sep = " ", remove = FALSE)
```

And finally, just for fun! The next example utilizes all of you in a data frame, but your
names are in separate columns, by first and last name.

DON'T RUN THIS ONE!
```{r}
Pizza %>% unite("Name", "First":"Last", remove = FALSE)
```

Again, the example below is similar, but more improved because you did not have the
pizza.csv that was diagrammed in a Excel spreadsheet.

So, in this verion you all appear in a df below.

```{r}
#create data frame
df <-
data.frame(first=c("David","Alex","Candy","Jeff","Will","Addison","Megha","Page","Cynthia"),
                last=c("Askew", "Billinis", "Boatwright", "Gilbert", "Hiott", "Horton",
"Khanna", "Magoto", "Welborn"),
```

```
                           pizza=c("Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "Yes"))
#view data frame
df
```

As I showed in the in class presentation, I first pulled the head or first six line to view.

```{r}
head(df)
```

Now finally, to show the utility of the unite() function the next code chunk used the
function to combine both first and last into a new colume with full name.

```{r}
df %>% unite("Name", "first":"last", sep = " ", remove = FALSE)
```

WDH: Now if a full name, for example, was needed for a pizza order for Tiltonpalooza, one
could use unite to combine first and last names into one column.

THE END!

FINALLY TO SUBMIT-----------------------

Complete Submission:

To turn in your review:

     You should fork the 8510-FunctionReview

Links to an external site. repository from github.
Clone your copy of this repository
Contribute your function review. Title the document "Lastname, functionname.qmd"
Push the repository to GitHub.
Create a pull request.

     Links to an external site.
     Provide the link to your copy of the GitHub repository here to turn in the assignment.