

# Topic: Building Tables from Scratch in



Tricks for  **sas** Users



# Welcome to Today's Lecture!

## Today's Lecture

- SAS is very restrictive when it comes to data editing
- Data steps are not that hard for editing large datasets
- But SAS is not very agile when it comes to making datasets
- Compare SAS ODS to exporting a table of results from a SQL count query
- **Demonstration: Data are easy to edit in R, and you can use R to complement SAS**



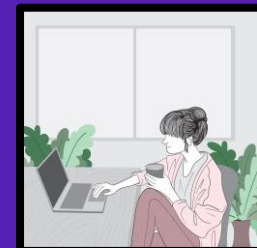
Try all of our career-building data science resources!



Join our group online data science mentoring program:  
<https://buff.ly/3Sw4Fsa>



Take our exclusive online data science courses aimed healthcare analysts:  
<https://monika-s-school-e295.thinkific.com>



Watch our tutorials on YouTube:  
<https://buff.ly/2W6ijuc>

Take our LinkedIn Learning courses: <https://buff.ly/2lhd4Rq>



## Today's Example

- In SAS, we use data steps and refer to columns by name (e.g., EDGROUP)
- In R, when I do data editing, I usually refer to columns using the name.
- R is very flexible for editing data.
- R makes it easy for you to use other objects to help you edit data (e.g., vectors).
- R let's you “manicure” data – edit one specific cell, for example

## Today's Resources

Blog post about this demonstration:

<https://buff.ly/3P3Ez0K>

Download demonstration **dataset**  
from GitHub: <https://buff.ly/3E4VzgV>

It's the last one in the folder named  
*BRFSS\_j.rds*

Download the demonstration **code**  
from GitHub:

<https://buff.ly/3Laxp8P>



# Summary of Today's Example

- I used the example today of making a blank summary table (“Table 1”) in R, then using *table* commands to generate frequencies to automatically replace 0’s in the table I made with correct values from my dataset.
- We created *dataframe* objects in R, so we were able to use data programming on them to copy values from frequency tables into the main table we were “manicuring”.
- We used the *row,column* references to cells in the dataframe we built with Table 1 as we populated it with values from the other dataframes we were generating with the *table* command.
- Once we filled in the columns and rows with frequencies, we export as a \*.csv, and can finish the post-processing (%’s) in Excel.



# Questions?

**Facilitator:** Monika M. Wahi, MPH, CPH

Email:  
[dethwench@gmail.com](mailto:dethwench@gmail.com)

Blog:  
<https://dethwench.com/blog/>

LinkedIn:  
<https://www.linkedin.com/in/dethwench/>



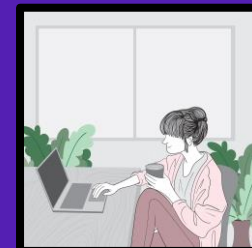
Try all of our career-building data science resources!



Join our group online data science mentoring program:  
<https://buff.ly/3Sw4Fsa>



Take our exclusive online data science courses aimed healthcare analysts:  
<https://monika-s-school-e295.thinkific.com>



Watch our tutorials on YouTube:  
<https://buff.ly/2W6ijuc>

Take our LinkedIn Learning courses: <https://buff.ly/2lhd4Rq>