­­ **Homework Assignment 6: Managing data in R**

Due date: Wednesday, 12/18/2024 by midnight

**Description**In this assignment, you will be working with datasets in R. You are being given datasets that need to be cleaned, labeled, merged, and reshaped for analysis. With these instructions, you will find 3 R datasets needed for this assignment. They are:  
 - wws.csv : working women survey data   
 - cardio\_wide1.csv : First batch of pulse and blood pressure data for 5 time points.  
 - cardio\_wide2.csv : Second batch of pulse and blood pressure data for 5 time points.  
 - assignment6\_template.R : This is a sample R file template for you to start with and modify.   
 - Assignment 6 report.rmd : This is a sample R Markdown file that will produce a sample report.   
  
Your task is to create multiple R script files to simulate the actual data management process over time in a typical research study. Each R script file will modify the previous version of the data file and then save a newly named version. This process allows you to easily recreate your work and to re-run your data management steps to an exact endpoint.

**Goals**After this assignment, you should be able to the following tasks:

- Create organized R script files that can be run sequentially to reproduce your work.  
- Correct values in a dataset  
- Generate and recode variables  
- Merge data  
**-** Run a sample R Markdown report

**Notes**

* It is good practice not to overwrite the original data as you make changes. Save the data to a new name and this will allow you to re-run your scripts any time starting with the original data.
* The R script for the current generation of the data should encompass the natural endpoint for the current activities with the data. Often you will do as much as you currently can with the data that you have and then save that as the current version. Later, you will start another script to continue with the last version of the data and make more changes, add more new data, etc.
* It is good practice to number each of the script files with a two-digit serial number so that I know the order in which they must be run. I use two-digits so I can go beyond 9 iterations and still sort the files by name correctly. For example, you can name this first R script file as **wws01.R** and the next one will be **wws02.R**, etc. You can use your own naming system but be consistent.
* In the data that is created by my script file, I usually incorporate the serial number that I gave to the script file. The **wws01.R** file creates an output file named **wws01.rdata**

**Steps to complete the exercise**

1. Setup a project in RStudio for this assignment.

Save all the files included with this assignment to their own folder. In R Studio, create a new project by selecting File, New Project, and choose Existing Directory to the folder where you unzipped the files.

1. **Part 1 – Initial Data**

For this first part, you have been sent two files: **wws.csv** and **cardio\_wide1.csv**. The study is still collecting data, but you must begin assembling these files into a dataset for analysis. The **wws.csv** is all the study participants. You must merge the **cardio\_wide1.csv** file with **wws.csv** and save the merged dataset. These steps should all be stored in your first R script file. You can use the included **assignment6\_template.R** file as the starting point and rename it. In the steps below, when it says to “load the xxx package”, you can confirm that it is loaded at the start of the script with a “library(xxx)” command. The **assignment6\_template.R** already has the commands to load the required packages for this assignment.

*Task*: Merge Datasets

To complete this first step:

* Run all of the library() lines under the comment “# Load the packages” at the beginning of the script. This will load all of the packages used in the following steps.
* Import **wws.csv** and **cardio\_wide1.csv** into the workspace.
  + Using the **rio** package, you can import the **wws.csv** data with the command like:

df\_name <- import("filename")

with “filename” replaced by the name of the file you want to read in, enclosed by double-quotes, and df\_name replaced by the name of the newly created dataframe.

You must also add the command to import **cardio\_wide1.csv**

You should have two lines to import the two files and they should now be listed as dataframes in your environment.

* Merge **wws with** **cardio\_wide1** on variable **idcode** and save the merged data to a new data frame named wws**01**
  + This step uses the **tidyverse** package which includes **dplyr**. Use the **left\_join()** function to merge the two data frames.
  + The command will look similar to this:

dfname <- df1 %>% left\_join(df2, by = "fieldname")

dfname is the resulting data frame

df1 is the first data frame.

df2 is the second data frame which you are joining to df1

fieldname is the field to join the dataframes

* You can verify the outcome of the merge by viewing the new dataframe wws01. In the Environment tab, click the table icon to the right of the line that lists the wws01 dataframe.
* Save the R script as **wws01**.**R**
* Save your workspace which includes the data frames with the command:

save.image(file="wws01.RData")

1. **Part 2 – Additional Variables**

Now imagine some time has passed since we made the dataset in part 1. We’ve given out copies of the dataset to study team members and now there is a request to create some additional summary variables.

*Tasks*: Create summary variables and clean data:

Average of pulse readings

Average of blood pressure readings

Recode average pulse to hi/lo

Here are the steps:

* If you closed the project that you created in step 1, re-open it.
* Create a new R script file and name it **wws02.R**. (Select File, New file, R Script.) You can also start with the **assignment6\_template.R** and save it as **wws02.R** and make edits to it.
* Just after the “Load the packages” lines in the script, begin by loading the R script file we saved at the end of step 1 using the command:

load(file="wws01.RData")

* Copy the **wws01** data frame to a new data frame named **wws02** to match the R script file that we are creating.
* Create a new variable named **wws02$plavg**. This should be the average of the 5 pulse readings (pl1:pl5). *See the R lecture file from class for an example of “average”.*
* Create a new variable named **wws02$bpavg**. This is the average of the 5 blood pressure readings (bp1:bp5).
* Create a new variable named **wws02$plavg\_hilo.** This will be the average pulse recoded as either low (1) or high(2). You will need to use the **cut()** function so that (min/89=1) and (90/max=2). *See the R lecture file from class for an example.*

Add these arguments to the cut() function:  
Set the “breaks = c(0,90,Inf)” argument which will create two categories.  
Label the values by including “labels=c("low","high")”

After you create the variable, you can check it with the **summary(wws02$plavg\_hilo)** and **str(wws02$plavg\_hilo)**

> summary(wws02$plavg\_hilo)

low high NA's

487 513 1246

> str(wws02$plavg\_hilo)

Factor w/ 2 levels "low","high": 1 1 2 1 1 1 1 1 2 2 ...

* Make the following corrections to the data. These types of corrections often occur after you receive the data and start reviewing it OR sometimes other members of the study will send corrections that they have discovered. Include a comment in the R script that indicates that these changes were “changes confirmed by the site” to document them. *See the R lecture file from class for an example using replace().*
  + collgrad=0 if idcode==15
  + age=38 if idcode==227
  + married = 1 if idcode==354
  + age=40 if idcode == 492
  + collgrad=0 if idcode == 714
* Save the R script as **wws02**.**R**
* Save your workspace as **wws02.RData**

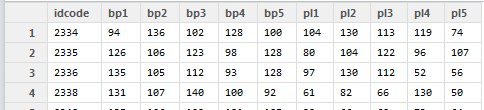
1. **Part 3 – Updated Data**

More time has passed and you are finally sent the last data with the final pulse and blood pressure data, called **cardio\_wide2.csv**.

*Tasks*: Update previous data with new data, update summary variables, and clean data.

Here are the steps:

* If you closed the project that you created in step 1, re-open it.
* Create a new R script file and name it **wws03.R**. (Select File, New file, R Script.) It should be located in the project folder with the wws01.R and wws02.R.
* Copy the **wws02** data frame to a new data frame named **wws03** to match the R script file that we are creating.
* Import **cardio\_wide2.csv** into the workspace. You can use the “Import Dataset” tool in the Environment tab.
* Now we are ready to update **wws03** with values from **cardio\_wide2**
* The **cardio\_wide2** data frame has the variables shown below:



We need to update **wws03** with these values. To do this we’ll use the **match()** function. We’ll use this function to find the rows in both data frames that have a matching **idcode** variable and then update the corresponding value. We will match and update the variables one by one. Here is the first command to update **bp1** :

wws03$bp1[match(cardio\_wide2$idcode,wws03$idcode)] <- cardio\_wide2$bp1

Now repeat the command above for **bp2,bp3,bp4,bp5,pl1..pl5.** You can verify that they have updated the data by running **tail(wws03)** which will show the last 6 records of the data frame. You should now see values for **bp1** through **pl5**

**Note:** The match command given above could also be written using piping, mutate and replace as shown below. You may use whichever you prefer.

wws03 <- wws03 |> mutate(bp1 = replace(bp1, match(cardio\_wide2$idcode,idcode), cardio\_wide2$bp1))

* Now that we have all the bp and pl data, we need to re-run the commands for the 3 new variables created earlier in wws02 (**plavg, bpavg, plavg\_hilo)** so that all participants with bp and pl data will have values for these new variables. You can simply copy those 3 lines of code from **wws02** and update them so that they now assign their calculated value to **wws03** data frame.
* Make the following corrections to the data. Include a comment that indicates that these changes were “confirmed by the site” to document them.
  + married=1 if idcode==2377
  + age=27 if idcode==2394
  + collgrad = 0 if idcode==2886
  + age=26 if idcode == 4102
  + collgrad=0 if idcode == 4248
* Save the R script as **wws03**.**R**
* Save your workspace as **wws03.RData**

You should now have 3 R script files that each produces a separate RData file. With the included csv data files, you should be able to run all three R script files in sequence and end up with a final data set **wws03.RData** which has all the data merged, new variables created, and all corrections made to existing data.

1. **Part 4** *–* Modify an R Markdown report

Included in the files for this assignment is a sample R Markdown file named **Assignment 6 report.Rmd**. By creating an R Markdown file like this one, you could create a “pushbutton” report that could easily be run to report on your data. The RMD file contains text and “chunks” of R code that are weaved together when you compile it. Note also that you can change the **Knit HTML** button by clicking the down arrow next to it and choose a different output option such as PDF or Word.

**Before running it**, copy the lines below to the RStudio console and run them to make sure you have the necessary packages:

required.packages <- c("descr","pander","dplyr","knitr")

packages.to.install <- setdiff(required.packages, installed.packages())

if(length(packages.to.install) > 0) install.packages(packages.to.install)

Also note that this report begins by loading “wws03.RData” workspace which you should have created already in step 3. If that file does not exist, it will not run. You can either modify the R Markdown to call your saved workspace from step 3 OR you can change the save.image step to save it as “wws03.RData”.

In the lower, right window of RStudio, click the Files tab and locate the R Markdown file named **Assignment 6 report.Rmd**. Click on it to open it in the Source window in the upper left window. At the top of the window, click on **Knit HTML** and the R Markdown program should be run and window should pop-up with an HTML page created from the program. Make sure that you can successfully “knit” it before modifying the file.

*Task*: Add another bar plot for “Years of school” which will use the “yrsschool” field.

Here are the steps:

* Since there is already a bar plot for Age Distribution, you can copy that “chunk” of code and then modify it to display “Years of school” instead.
* In the R Markdown file, copy the lines of code from line 32-37. Move to line 38 and paste the copied lines.
* Make edits to the new code on lines from 38-42.
  + In line 39, give the R chunk a different name like “Barplot School Years Distribution”. You cannot have 2 chunks with the same name.
  + The counts table should now be based on wws03$yrschool
  + The main title and the x-axis label need to be changed to correctly identify the data.
* Knit the R Markdown file again and it should now have the added barplot for “Years of school”.

**Submitting Assignment**

The R script files that you turn in should be able to run without error and create the final analytic dataset needed. **Before you submit the files,** be sure to close the R files, exit RStudio completely, and then re-open RStudio and try running the scripts again.

To submit this assignment, please zip the files below into one zip file and upload to CoursePlus:  
- 3 R script files   
- 3 RData workspace files  
- Assignment 6 report R Markdown file that you modified in step 4

If you are unsure about how to zip the 6 files into a single file, please send the files attached to a single email to us directly.