Inclass Assignment (5)

Akshar Patel

3/16/2022

* If you don’t know the answer, leave it blank.
* If you are caught cheating, you will be given minus 50 points.

### Run this chunk first

if (!require('tidyverse')) install.packages('tidyverse'); library('tidyverse')

## Loading required package: tidyverse

## Warning: package 'tidyverse' was built under R version 4.1.3

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5 v purrr 0.3.4  
## v tibble 3.1.6 v dplyr 1.0.8  
## v tidyr 1.2.0 v stringr 1.4.0  
## v readr 2.1.2 v forcats 0.5.1

## Warning: package 'tidyr' was built under R version 4.1.3

## Warning: package 'readr' was built under R version 4.1.3

## Warning: package 'forcats' was built under R version 4.1.3

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

Q1. Update the author name with your name in YAML part above.

Q2. A line of code to import a productivity.csv file from your Desktop folder and assign it as a productivity variable.

productivity <- read\_csv("C:/Users/Akshar/Desktop/Inclass\_Assigment\_5/productivity.csv")

## Rows: 1192 Columns: 6  
## -- Column specification --------------------------------------------------------  
## Delimiter: ","  
## chr (2): sex, race  
## dbl (4): earn, height, ed, age  
##   
## i Use `spec()` to retrieve the full column specification for this data.  
## i Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

Q3. What is the number of rows and columns of productivity variable?

#Rows: 1192  
#Columns: 6

Run this chunk for Q4

ages <- data.frame(Name = c("Jane", "William", "Ben", "Jaewan"),  
 Age = c(31, 23, 12, 42))

Q4. A line of code to save ages dataframe to your Desktop folder as ages.csv. You don’t need to submit ages.csv to Canvas.

write\_csv(ages, "C:/Users/Akshar/Desktop/Inclass\_Assigment\_5/ages.csv")

Run this code for Q5.

if (!require('readxl')) install.packages('readxl'); library('readxl')

## Loading required package: readxl

## Warning: package 'readxl' was built under R version 4.1.3

Q5. A line of code to import a scale.xlsx file and store it as a scale variable from your Desktop folder.

scale <- read\_excel("C:/Users/Akshar/Desktop/Inclass\_Assigment\_5/scale.xlsx")

Q6. Run the following code and answer the reason why it is untidy.

table2

## # A tibble: 12 x 4  
## country year type count  
## <chr> <int> <chr> <int>  
## 1 Afghanistan 1999 cases 745  
## 2 Afghanistan 1999 population 19987071  
## 3 Afghanistan 2000 cases 2666  
## 4 Afghanistan 2000 population 20595360  
## 5 Brazil 1999 cases 37737  
## 6 Brazil 1999 population 172006362  
## 7 Brazil 2000 cases 80488  
## 8 Brazil 2000 population 174504898  
## 9 China 1999 cases 212258  
## 10 China 1999 population 1272915272  
## 11 China 2000 cases 213766  
## 12 China 2000 population 1280428583

#Explanation: each and every variable must have its own column

Q7. Run the following code and answer the reason why it is untidy.

table3

## # A tibble: 6 x 3  
## country year rate   
## \* <chr> <int> <chr>   
## 1 Afghanistan 1999 745/19987071   
## 2 Afghanistan 2000 2666/20595360   
## 3 Brazil 1999 37737/172006362   
## 4 Brazil 2000 80488/174504898   
## 5 China 1999 212258/1272915272  
## 6 China 2000 213766/1280428583

#Explanation: Each and every value must have its own cell

Q8. Run the following code and answer the reason why it is untidy.

table4a

## # A tibble: 3 x 3  
## country `1999` `2000`  
## \* <chr> <int> <int>  
## 1 Afghanistan 745 2666  
## 2 Brazil 37737 80488  
## 3 China 212258 213766

#Explanation: year variable does not exist

Q9. Two lines of code that pivots table4a as a tidy and longer dataset with the following information. - The name of the variable to move the column names to: year - The name of the variable to move the column values to: cases

table4a %>%  
 pivot\_longer(c('1999', '2000'), names\_to = "year", values\_to = "cases")

## # A tibble: 6 x 3  
## country year cases  
## <chr> <chr> <int>  
## 1 Afghanistan 1999 745  
## 2 Afghanistan 2000 2666  
## 3 Brazil 1999 37737  
## 4 Brazil 2000 80488  
## 5 China 1999 212258  
## 6 China 2000 213766

Q10. Two lines of code that pivots table2 as a tidy and wider dataset with the following information. - The column to take variable names from: type - The column to take values from: count

table2 %>%  
 pivot\_wider(names\_from = "type", values\_from = "count")

## # A tibble: 6 x 4  
## country year cases population  
## <chr> <int> <int> <int>  
## 1 Afghanistan 1999 745 19987071  
## 2 Afghanistan 2000 2666 20595360  
## 3 Brazil 1999 37737 172006362  
## 4 Brazil 2000 80488 174504898  
## 5 China 1999 212258 1272915272  
## 6 China 2000 213766 1280428583