

Comprehension Check due May 9, 2021 06:59 +03

In this part of the assessment, you will answer several multiple choice questions that review the concepts of data import. You can answer these questions without using R, although you may find it helpful to experiment with commands in your console.

In the second part of the assessment on the next page, you will import real datasets and learn more about useful arguments to **readr** functions. The second part of the assessment will require you to program in R.

---

## Question 1

1/1 point (graded)

Which of the following is NOT part of the data wrangling process?

- ☐ Importing data into R
- ☐ Formatting dates/times
- ☒ Checking correlations between your variables
- ☐ Tidying data



### Answer

Correct:

Data analyses, such as checking correlations or creating visualizations, are done AFTER the data has been processed into a tidy format.

### Explanation

Importing data, formatting dates and times, and tidying data are all parts of the data wrangling process. Checking correlations between your variables is a data analysis that is completed after the data has been processed into a tidy format.

Submit

You have used 1 of 2 attempts

---

**i** Answers are displayed within the problem

---

## Question 2

1/1 point (graded)

Which files could be opened in a basic text editor?

Select ALL that apply.

☒ data.txt

☒ data.csv

☐ data.xlsx

☒ data.tsv



### Explanation

.txt, .csv, and .tsv files can all be opened by any basic text editor. Microsoft Excel files (.xlsx) use a proprietary format that can't be viewed with a basic text editor.

Submit

You have used 1 of 2 attempts

---

**i** Answers are displayed within the problem

---

## Question 3

1/1 point (graded)

You want to analyze a file containing race finish times for a recent marathon. You open the file in a basic text editor and see lines that look like the following:

```
initials,state,age,time  
vib,MA,61,6:01  
adc,TX,45,5:45  
kme,CT,50,4:19
```

What type of file is this?

☐ A comma-delimited file without a header

☐ A tab-delimited file with a header

☐ A white space-delimited file without a header

☒ A comma-delimited file with a header



### Answer

Correct:

This file has a header that describes the contents of each column. Values are separated by commas.

## Explanation

This file has a header that describes the contents of each column. Values are separated by commas.

Submit

You have used 1 of 2 attempts

**i** Answers are displayed within the problem

## Question 4

1/1 point (graded)

Assume the following is the full path to the directory that a student wants to use as their working directory in R: `"/Users/student/Documents/projects/"`

Which of the following lines of code CANNOT set the working directory to the desired "projects" directory?

- ☐ `setwd("~/Documents/projects/")`
- ☐ `setwd("/Users/student/Documents/projects/")`
- ☒ `setwd(/Users/student/Documents/projects/)`
- ☐ `dir <- "/Users/student/Documents/projects"`  
`setwd(dir)`



## Answer

Correct:

Correct: this code will not work. You need to use quotation marks when providing a full path to the working directory.

Submit

You have used 1 of 2 attempts

✓ Correct (1/1 point)

## Question 5

1/1 point (graded)

We want to copy the "murders.csv" file from the dslabs package into an existing folder "data", which is located in our HarvardX-Wrangling projects folder. We first enter the code below into our RStudio console.

```
> getwd()
[1] "C:/Users/UNIVERSITY/Documents/Analyses/HarvardX-Wrangling"
> filename <- "murders.csv"
> path <- system.file("extdata", package = "dslabs")
```

Which of the following commands would NOT successfully copy “murders.csv” into the folder “data”?



```
file.copy(file.path(path, "murders.csv"), getwd())
```



```
setwd("data")
file.copy(file.path(path, filename), getwd())
```



```
file.copy(file.path(path, "murders.csv"), file.path(getwd(), "data"))
```



```
file.location <- file.path(system.file("extdata", package = "dslabs"), "murders.csv")
file.destination <- file.path(getwd(), "data")
file.copy(file.location, file.destination)
```



### Answer

Correct:

This command does NOT copy the “murders.csv” file into your “data” folder; instead it copies it into the parent directory, “HarvardX-Wrangling”. You need to specify the “data” folder, either by changing the working directory or by adding it to the file path.

### Explanation

`file.copy(file.path(path, "murders.csv"), getwd())` does NOT copy the “murders.csv” file into your “data” folder; instead it copies it into the parent directory, “HarvardX-Wrangling”. You need to specify the “data” folder, either by changing the working directory or by adding it to the file path.

The other commands all correctly copy the "murders.csv" file into your "data" folder.

Submit

You have used 1 of 2 attempts

---

**i** Answers are displayed within the problem

---

## Question 6

1/1 point (graded)

You are not sure whether the murders.csv file has a header row. How could you check this?  
Select ALL that apply.

- ☒ Open the file in a basic text editor.
- ☒ In the RStudio "Files" pane, click on your file, then select "View File".
- ☒ Use the command `read_lines` (remembering to specify the number of rows with the `n_max` argument).



### Explanation

All three choices are ways to inspect the header.

Submit

You have used 1 of 2 attempts

---

**i** Answers are displayed within the problem

---

## Question 7

1/1 point (graded)

What is one difference between `read_excel()` and `read_xlsx()` ?

- ☐ `read_excel()` also reads meta-data from the excel file, such as sheet names, while `read_xlsx()` only reads the first sheet in a file.
- ☒ `read_excel()` reads both .xls and .xlsx files by detecting the file format from its extension, while `read_xlsx()` only reads .xlsx files.
- ☐ `read_excel()` is part of the **readr** package, while `read_xlsx()` is part of the **readxl** package and has more options.
- ☐ `read_xlsx()` has been replaced by `read_excel()` in a recent **readxl** package update.



**Answer**

Correct:

You can use `read_excel` for both `.xls` and `.xlsx` files, while `read_xlsx` only reads `.xlsx` files.

### Explanation

The difference between these functions is that you can use `read_excel()` for both `.xls` and `.xlsx` files, while `read_xlsx()` only reads `.xlsx` files.

Submit

You have used 1 of 2 attempts

**i** Answers are displayed within the problem

## Question 8

1/1 point (graded)

You have a file called "times.txt" that contains race finish times for a marathon. The first four lines of the file look like this:

```
initials,state,age,time
vib,MA,61,6:01
adc,TX,45,5:45
kme,CT,50,4:19
```

Which line of code will NOT produce a tibble with column names "initials", "state", "age", and "time"?

☐ `race_times <- read_csv("times.txt")`

☒ `race_times <- read.csv("times.txt")`

☐ `race_times <- read_csv("times.txt", col_names = TRUE)`

☐ `race_times <- read_delim("times.txt", delim = ",")`



### Answer

Correct:

This code will import the comma-separated values table called "times.txt", but the base R function `read.csv` does not produce a tibble. It creates a data frame.

### Explanation

`race_times <- read.csv("times.txt")` will import the comma-separated values table called "times.txt", but the base R function `read.csv` does not produce a tibble. It creates a data frame.

Submit

You have used 1 of 2 attempts

---

**i** Answers are displayed within the problem

---

## Question 9

1/1 point (graded)

You also have access to marathon finish times in the form of an Excel document named "times.xlsx". In the Excel document, different sheets contain race information for different years. The first sheet is named "2015", the second is named "2016", and the third is named "2017".

Which line of code will NOT import the data contained in the "2016" tab of this Excel sheet?

☐ `times_2016 <- read_excel("times.xlsx", sheet = 2)`

☒ `times_2016 <- read_xlsx("times.xlsx", sheet = "2")`

☐ `times_2016 <- read_excel("times.xlsx", sheet = "2016")`

☐ `times_2016 <- read_xlsx("times.xlsx", sheet = 2)`



### Answer

Correct:

When the sheet argument is contained within quotes, the function expects a sheet name. There is no sheet named "2" in this spreadsheet, so the code will not work.

Submit

You have used 1 of 2 attempts

---

✓ Correct (1/1 point)

---

## Question 10

1/1 point (graded)

You have a comma-separated values file that contains the initials, home states, ages, and race finish times for marathon runners. The runners' initials contain three characters for the runners' first, middle, and last names (for example, "KME").

You read in the file using the following code.

```
race_times <- read.csv("times.csv")
```

What is the data type of the initials in the object `race_times` ?

☐ integers

☒ characters

☐ factors ✓

☐ logical



### Answer

Correct: In R4.0, this is now true.

### Explanation

As of R4.0, the data type is now `characters` .

Submit

You have used 1 of 2 attempts

---

**i** Answers are displayed within the problem

---

## Question 11

1/1 point (graded)

Which of the following is NOT a real difference between the readr import functions and the base R import functions?

☐ The import functions in the readr package all start as `read_` , while the import functions for base R all start with `read.`

☐ Base R import functions automatically convert character columns to factors. ✓

☒ The base R import functions can read .csv files, but cannot read files with other delimiters, such as .tsv files, or fixed-width files.

☐ Base R functions import data as a data frame, while readr functions import data as a tibble.



### Answer

Correct:

This statement is false. The base R import functions can read files with other delimiters like .tsv using `read.delim` and can read fixed-width files using `read.fwf` .

Submit

You have used 1 of 2 attempts



---

**i** Answers are displayed within the problem

---

## Question 12

1/1 point (graded)

You read in a file containing runner information and marathon finish times using the following code.

```
race_times <- read.csv("times.csv", stringsAsFactors = F)
```

What is the class of the object `race_times` ?

☒ data frame

☐ tibble

☐ matrix

☐ vector



### Answer

Correct: The `read.csv` function generates a data frame containing these data.

Submit

You have used 1 of 2 attempts

---

✓ Correct (1/1 point)

---

## Question 13

1/1 point (graded)

Select the answer choice that summarizes all of the actions that the following lines of code can perform. Please note that the url below is an example and does not lead to data.

```
url <- "https://raw.githubusercontent.com/MyUserName/MyProject/master/MyData.csv "
dat <- read_csv(url)
download.file(url, "MyData.csv")
```

☐ Create a tibble in R called `dat` that contains the information contained in the csv file stored on Github and save that tibble to the working directory.

- ☐ Create a matrix in R called `dat` that contains the information contained in the csv file stored on Github. Download the csv file to the working directory and name the downloaded file "MyData.csv".
- ☐ Create a tibble in R called `dat` that contains the information contained in the csv file stored on Github. Download the csv file to the working directory and randomly assign it a temporary name that is very likely to be unique.
- ☒ Create a tibble in R called `dat` that contains the information contained in the csv file stored on Github. Download the csv file to the working directory and name the downloaded file "MyData.csv".



### Answer

Correct:

The `read_csv` command creates the tibble in R and the `download.file` command downloads the csv file from the internet to the working directory with the specified name.

Submit

You have used 1 of 2 attempts

---

**i** Answers are displayed within the problem