Course › Section 2: Tidy Data › 2.1: Reshaping Data › Assessment Part 1: ...

Assessment Part 1: Reshaping Data

Part 1 consists of 8 questions are conceptual questions about tidy data and reshaping data. They do not necessarily require R, but you may benefit from checking your work on the console.

Part 2 consists of 7 questions which require you to write code in R to apply the new concepts about tidy data and reshaping data.

Question 1

0/1 point (graded)

A collaborator sends you a file containing data for three years of average race finish times.

```
age_group,2015,2016,2017
20,3:46,3:22,3:50
30,3:50,3:43,4:43
40,4:39,3:49,4:51
50,4:48,4:59,5:01
```

Are these data considered “tidy” in R? Why or why not?

○ Yes. These data are considered “tidy” because each row contains unique observations.

● Yes. These data are considered “tidy” because there are no missing data in the data frame. ✗

○ No. These data are not considered “tidy” because the variable “year” is stored in the header. ✓

○ No. These data are not considered “tidy” because there are not an equal number of columns and rows.

Answer

Incorrect:

Try again. Tidy data may have missing data represented as “NA”, but each row should contain a single observation. In this case, the race finish time represent the observations and all other information is a variable.

Explanation

These data are not tidy because year is a variable and should be stored as a column instead of across multiple columns in the header.

Submit

You have used 2 of 2 attempts

Answers are displayed within the problem

Question 2

1/1 point (graded)

Below are four versions of the same dataset. Which one is in a tidy format?

●

```
state      abb region  population total
Alabama     AL  South   4779736   135
Alaska      AK   West    710231    19
Arizona     AZ   West   6392017   232
Arkansas    AR  South   2915918    93
California  CA   West  37253956  1257
Colorado    CO   West   5029196    65
```

✓

○

```
state   abb region      var   people
Alabama  AL  South population 4779736
Alabama  AL  South      total     135
Alaska   AK   West population  710231
Alaska   AK   West      total      19
Arizona  AZ   West population 6392017
Arizona  AZ   West      total     232
```

○

```
state      abb Northeast South North Central West
Alabama     AL        NA 4779736    NA        NA
Alaska      AK        NA      NA    NA    710231
Arizona     AZ        NA      NA    NA   6392017
Arkansas    AR        NA 2915918    NA        NA
California  CA        NA      NA    NA  37253956
Colorado    CO        NA      NA    NA   5029196
```

○

```
state      abb region    rate
Alabama     AL  South 2.82e-05
Alaska      AK   West 2.68e-05
Arizona     AZ   West 3.63e-05
Arkansas    AR  South 3.19e-05
California  CA   West 3.37e-05
Colorado    CO   West 1.29e-05
```

Explanation

In tidy format, each observation has its own row, and each variable has its own column.

Submit You have used 1 of 2 attempts

ℹ Answers are displayed within the problem

Question 3

1/1 point (graded)

Your file called “times.csv” has age groups and average race finish times for three years of marathons.

```
age_group,2015,2016,2017
20,3:46,3:22,3:50
30,3:50,3:43,4:43
40,4:39,3:49,4:51
50,4:48,4:59,5:01
```

You read in the data file using the following command.

```
d <- read_csv("times.csv")
```

Which commands will help you “tidy” the data?

●
```
tidy_data <- d %>%
    gather(year, time, `2015`:`2017`)
```
✓

○
```
tidy_data <- d %>%
    spread(year, time, `2015`:`2017`)
```

○
```
tidy_data <- d %>%
    gather(age_group, year, time, `2015`:`2017`)
```

○
```
tidy_data <- d %>%
    gather(time, `2015`:`2017`)
```

Answer

Correct:

This code will gather the years from 2015 to 2017 into a single column and create a single column called “time” that contains the time for each age group and each year.

Explanation

```
tidy_data <- d %>%
    gather(year, time, `2015`:`2017`)
```

This code will gather the years from 2015 to 2017 into a single column and create a single column called “time” that contains the time for each age group and each year.

Submit

You have used 1 of 2 attempts

Answers are displayed within the problem

Question 4

1/1 point (graded)

You have a dataset on U.S. contagious diseases, but it is in the following wide format:

```
> head(dat_wide)
state year population HepatitisA Mumps Polio Rubella
Alabama 1990    4040587       86     19     76     1
Alabama 1991    4066003       39     14     65     0
Alabama 1992    4097169       35     12     24     0
Alabama 1993    4133242       40     22     67     0
Alabama 1994    4173361       72     12     39     0
Alabama 1995    4216645       75      2     38     0
```

You want to transform this into a tidy dataset, with each row representing an observation of the incidence of each specific disease (as shown below):

```
> head(dat_tidy)
state    year  population  disease  count
Alabama 1990    4040587 HepatitisA      86
Alabama 1991    4066003 HepatitisA      39
Alabama 1992    4097169 HepatitisA      35
Alabama 1993    4133242 HepatitisA      40
Alabama 1994    4173361 HepatitisA      72
Alabama 1995    4216645 HepatitisA      75
```

Which of the following commands would achieve this transformation to tidy the data?

Pay attention to the column names.

○

```
dat_tidy <- dat_wide %>%
    gather (key = count, value = disease, HepatitisA, Rubella)
```

○

```
 dat_tidy <- dat_wide %>%
    gather(key = count, value = disease, -state, -year, -population)
```

○

```
dat_tidy <- dat_wide %>%
    gather(key = disease, value = count, -state)
```

●

```
dat_tidy <- dat_wide %>%
    gather(key = disease, value = count, HepatitisA:Rubella)
```

✓

Answer

Correct:
In this command, you properly specified that the “key” column will be called “disease”, the value of each entry will be called “count”, and that the columns HepatitisA through Rubella will all be included in the gather command.

Submit You have used 1 of 2 attempts

Answers are displayed within the problem

Question 5

1/1 point (graded)
You have successfully formatted marathon finish times into a tidy object called `tidy_data`. The first few lines are shown below.

```
age_group year   time
20        2015   03:46
30        2015   03:50
40        2015   04:39
50        2015   04:48
20        2016   03:22
```

Select the code that converts these data back to the wide format, where each year has a separate column.

○ `tidy_data %>% spread(time, year)`

● `tidy_data %>% spread(year, time)` ✓

○ `tidy_data %>% spread(year, age_group)`

○ ``tidy_data %>% spread(time, year, `2015`:`2017`)``

Answer
Correct: This code tells the function to create new columns for each year and spread the time values over those cells.

Submit You have used 1 of 2 attempts

Answers are displayed within the problem

Question 6

1/1 point (graded)
You have the following dataset:

```
> head(dat)
state   abb region      var   people
Alabama  AL  South population 4779736
Alabama  AL  South      total     135
Alaska   AK   West population  710231
Alaska   AK   West      total      19
Arizona  AZ   West population 6392017
Arizona  AZ   West      total     232
```

You would like to transform it into a dataset where population and total are each their own column (shown below):

```
state      abb region population total
Alabama     AL  South    4779736   135
Alaska      AK   West     710231    19
Arizona     AZ   West    6392017   232
Arkansas    AR  South    2915918    93
California  CA   West   37253956  1257
Colorado    CO   West    5029196    65
```

Which code would best accomplish this?

● `dat_tidy <- dat %>% spread(key = var, value = people)` ✓

○ `dat_tidy <- dat %>% spread(key = state:region, value = people)`

○ `dat_tidy <- dat %>% spread(key = people, value = var)`

○ `dat_tidy <- dat %>% spread(key = region, value = people)`

Answer
Correct:
In this command, you properly specify that the column “var” will be used as the new column names, and that the column “people” should be spread into these two columns.

Submit

You have used 1 of 2 attempts

Answers are displayed within the problem

Question 7

1.0/1.0 point (graded)
A collaborator sends you a file containing data for two years of average race finish times, "times.csv":

```
age_group,2015_time,2015_participants,2016_time,2016_participants
20,3:46,54,3:22,62
30,3:50,60,3:43,58
40,4:39,29,3:49,33
50,4:48,10,4:59,14
```

You read in the data file:

```
d <- read_csv("times.csv")
```

Which of the answers below best makes the data tidy?

○

```
tidy_data <- d %>%
    gather(key = “key”, value = “value”, -age_group) %>%
    separate(col = key, into = c(“year”, “variable_name”), sep = “.”) %>%
    spread(key = variable_name, value = value)
```

●

```
tidy_data <- d %>%
    gather(key = “key”, value = “value”, -age_group) %>%
    separate(col = key, into = c(“year”, “variable_name”), sep = “_”) %>%
    spread(key = variable_name, value = value)
```

✓

○

```
tidy_data <- d %>%
    gather(key = “key”, value = “value”) %>%
    separate(col = key, into = c(“year”, “variable_name”), sep = “_”) %>%
    spread(key = variable_name, value = value)
```

○

```
tidy_data <- d %>%
    gather(key = “key”, value = “value”, -age_group) %>%
    separate(col = key, into = “year”, sep = “_”) %>%
    spread(key = year, value = value)
```

Answer

Correct:

This column gathers the column names 2015_time, 2015_participants, 2016_time, and 2016_participants into one column called “key”, with the values for each stored in the column “value.” The key column is then separated into two columns, “year” and “variable_name”. The two entries for “variable_name”, time and participants, are then spread into their own columns.

Submit

You have used 1 of 2 attempts

Answers are displayed within the problem

Question 8

1.0/1.0 point (graded)

You are in the process of tidying some data on heights, hand length, and wingspan for basketball players in the draft. Currently, you have the following:

```
> head(stats)
key               value
allen_height      75
allen_hand_length 8.25
allen_wingspan    79.25
bamba_height      83.25
bamba_hand_length 9.75
bamba_wingspan    94
```

Select all of the correct commands below that would turn this data into a “tidy” format.

☑

```
tidy_data <- stats %>%
    separate(col = key, into = c("player", "variable_name"), sep = "_", extra = "merge") %>%
    spread(key = variable_name, value = value)
```

✓

□

```
tidy_data <- stats %>%
    separate(col = key, into = c("player", "variable_name1", "variable_name2"), sep = "_", fill = "right") %>%
    unite(col = variable_name, variable_name1, variable_name2, sep = "_") %>%
    spread(key = variable_name, value = value)
```

□

```
tidy_data <- stats %>%
    separate(col = key, into = c("player", "variable_name"), sep = "_") %>%
    spread(key = variable_name, value = value)
```

✓

Answer

Correct:
This is an efficient way to separate the key column into two new columns, “player” and “variable_name”, while keeping the full variable names using the extra command.

Submit

You have used 1 of 2 attempts

Answers are displayed within the problem