Comprehension Check due May 18, 2021 14:59 +03 Use the following libraries for these questions:

library(tidyverse) library(dslabs)

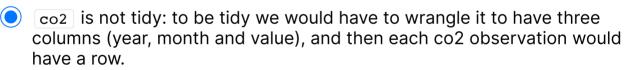
Question 9

1/1 point (graded)

Examine the built-in dataset co2. This dataset comes with base R, not **dslabs** - just type co2 to access the dataset.

Is co2 tidy? Why or why not?

| © co2 is tidy data: it has one year for each row. | |
|--|--|
| © co2 is tidy data: each column is a different month. | |
| © co2 is not tidy: there are multiple observations per column. | |





Explanation

These data are not tidy because month is a variable and should be stored as a column instead of across multiple columns in the header. There are also multiple observations per row, and each observation should be a different row.

Submit You have used 1 of 2 attempts

1 Answers are displayed within the problem

Question 10

1/1 point (graded)

Run the following command to define the co2_wide object:

```
co2_wide <- data.frame(matrix(co2, ncol = 12, byrow = TRUE)) %>%
  setNames(1:12) %>%
mutate(year = as.character(1959:1997))
```

Use the <code>gather()</code> function to make this dataset tidy. Call the column with the CO2 measurements <code>co2</code> and call the month column <code>month</code>. Name the resulting object <code>co2_tidy</code>.

Which code would return the correct tidy format?

| <pre>co2_tidy <- gather(co2_wide,month,co2,year)</pre> |
|--|
| <pre>co2_tidy <- gather(co2_wide,co2,month,-year)</pre> |
| <pre>co2_tidy <- gather(co2_wide,co2,month,year)</pre> |
| co2_tidy <- gather(co2_wide,month,co2,-year) |
| |

Explanation

gather takes 4 arguments: (1) the data, (2) the name of the key column, (3) the name of the value column, and optionally (4) the names of any columns not to gather. We want to collect the months as the key and CO2 as the value while keeping the year column as-is.

Submit You have used 1 of 2 attempts

Answers are displayed within the problem

Question 11

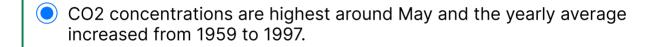
1/1 point (graded)

Use co2 tidy to plot CO2 versus month with a different curve for each year:

co2_tidy %>% ggplot(aes(as.numeric(month), co2, color = year)) + geom_line

What can be concluded from this plot?

| CO2 concentrations increased monotonically (never decreased) fro | m |
|--|---|
| 1959 to 1997. | |



| CO2 concentrations are highest around Oc | ctober and the yearly averag | e |
|--|------------------------------|---|
| increased from 1959 to 1997. | | |

| O Ye | early average | CO2 | concentrations | have | remained | constant | over | time. |
|------|---------------|-----|----------------|------|----------|----------|------|-------|
|------|---------------|-----|----------------|------|----------|----------|------|-------|

| CO2 | concentrations | do | not have | а | seasonal | trend |
|-----|----------------|----|----------|---|----------|--------|
| COZ | Concentiations | uО | HUL Have | а | Seasonai | u ena. |



Explanation

In every year, the highest CO2 level is around May. The line legend shows that earlier years had lower CO2 levels and that CO2 levels generally increase relative to the previous year. The most recent years have the highest CO2.

Submit You have used 1 of 2 attempts

1 Answers are displayed within the problem

Question 12

1/1 point (graded)

Load the admissions dataset from **dslabs**, which contains college admission information for men and women across six majors, and remove the applicants percentage column:

```
library(dslabs)
data(admissions)
dat <- admissions %>% select(-applicants)
```

Your goal is to get the data in the shape that has one row for each major, like this:

```
major men
              women
Α
       62
              82
В
       63
              68
С
       37
              34
       33
              35
D
       28
              24
Ε
               7
F
        6
```

Which command could help you to wrangle the data into the desired format?

```
dat_tidy <- spread(dat, major, admitted)

dat_tidy <- spread(dat, gender, major)

dat_tidy <- spread(dat, gender, admitted)

dat_tidy <- spread(dat, admitted, gender)</pre>
```

Explanation

spread takes 3 arguments: (1) the data frame, (2) the key to spread across columns, and (3) the value to put in individual cells of the table.

Submit You have used 1 of 2 attempts

1 Answers are displayed within the problem

Question 13

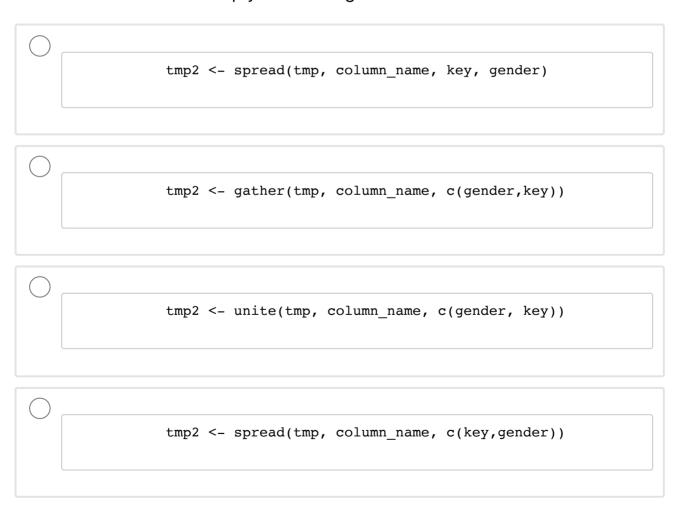
1/1 point (graded)

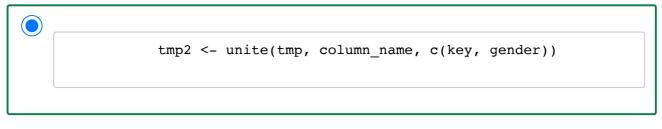
Now use the admissions dataset to create the object tmp, which has columns major, gender, key and value:

```
tmp <- gather(admissions, key, value, admitted:applicants)
tmp</pre>
```

Combine the key and gender and create a new column called <code>column_name</code> to get a variable with the following values: <code>admitted_men</code>, <code>admitted_women</code>, <code>applicants_men</code> and <code>applicants_women</code>. Save the new data as <code>tmp2</code>.

Which command could help you to wrangle the data into the desired format?







Explanation

unite takes 3 arguments: (1) the data frame, (2) the name of the new column to create, and (3) a vector of the columns to unite with an underscore, in order.

Submit You have used 1 of 2 attempts

Answers are displayed within the problem

Question 14

1/1 point (graded)

Which function can reshape <code>tmp2</code> to a table with six rows and five columns named <code>major</code>, <code>admitted_men</code>, <code>admitted_women</code>, <code>applicants_men</code> and <code>applicants_women</code>?

| gather() | |
|----------|--|
| spread() | |

| () | enarate() |
|-----|-----------|

| () unite() |
|------------|
| |
| |



Explanation

```
spread(tmp2, column_name, value)
```

Submit You have used 1 of 2 attempts

• Answers are displayed within the problem