## Congratulations! You passed!

Grade received 100% To pass 80% or higher

Go to next item

# Weekly challenge 3

Latest Submission Grade 100%

**1.** 1 / 1 point

A data analyst is working with a dataset in R that has more than 50,000 observations. Why might they choose to use a tibble instead of the standard data frame? Select all that apply.

- [ ] Tibbles can automatically change the names of variables
- [x] Tibbles automatically only preview as many columns as fit on screen

Correct
Tibbles make printing in R easier. They won't accidentally overload the data analyst's console because they're automatically set to pull up only the first 10 rows and as many columns as fit on screen.

- [x] Tibbles automatically only preview the first 10 rows of data

Correct
Tibbles make printing in R easier. They won't accidentally overload the data analyst's console because they're automatically set to pull up only the first 10 rows and as many columns as fit on screen.

- [ ] Tibbles can create row names

**2.** 1 / 1 point

A data analyst is working with a large data frame. It contains so many columns that they don't all fit on the screen at once. The analyst wants a quick list of all of the column names to get a better idea of what is in their data. What function should they use?

- (•) colnames()
- ( ) str()
- ( ) mutate()
- ( ) head()

Correct
The colnames() function will return a list of all the column names in a data frame for easy reference.

**3.** You are working with the ToothGrowth dataset. You want to use the head() function to get a preview of the dataset. Write the code chunk that will give you this preview. 1 / 1 point

```
1  head(ToothGrowth)
```

Run
Reset

```
   len supp dose
1  4.2   VC  0.5
2 11.5   VC  0.5
3  7.3   VC  0.5
4  5.8   VC  0.5
5  6.4   VC  0.5
6 10.0   VC  0.5
```

What are the names of the columns in the ToothGrowth dataset?

- ○ len, supp, VC
- ○ len, VC, dose
- ◉ len, supp, dose
- ○ VC, supp, dose

**Correct**
The code chunk `head(ToothGrowth)` gives you a preview of the dataset. Inside the parentheses of the head() function is the name of the dataset you want to preview. The code returns a view of the column names and the first few rows of the dataset. The names of the columns in the ToothGrowth dataset are len, supp, dose.

**4.** A data analyst is working with a data frame named *cars*. The analyst notices that all the column names in the data frame are capitalized. What code chunk lets the analyst change all the column names to lowercase? **1 / 1 point**

- ○ `rename_with(cars, toupper)`
- ◉ `rename_with(cars, tolower)`
- ○ `rename_with(toupper, cars)`
- ○ `rename_with(tolower, cars)`

**Correct**
The code chunk is `rename_with(cars, tolower)`. The rename_with() function will enable the analyst to easily change the case of the column names to lowercase. Including the tolower argument indicates that all column names will be changed to lowercase.

**5.** A data analyst is working with the penguins data. They write the following code: **1 / 1 point**

```
penguins %>%
```

The variable *species* includes three penguin species: Adelie, Chinstrap, and Gentoo. What code chunk does the analyst add to create a data frame that only includes the Gentoo species?

- ○ `filter(species == "Adelie")`
- ◉ `filter(species == "Gentoo")`
- ○ `filter(Gentoo == species)`
- ○ `filter(species <- "Gentoo")`

**Correct**
The code chunk is `filter(species == "Gentoo")`. The filter function allows the data analyst to specify which part of the data they want to view. Two equal signs in an argument mean "exactly equal to." Using this operator instead of the assignment operator <- calls only the data about Gentoo penguins to the dataset.

**6.** You are working with the penguins dataset. You want to use the summarize() and mean() functions to find the mean value for the variable *body_mass_g*. You write the following code: **1 / 1 point**

```
penguins %>%

  drop_na() %>%

  group_by(species) %>%
```

Add the code chunk that lets you find the mean value for the variable *body_mass_g*.

```
1
2  penguins %>%
3    group_by(species) %>%
4    drop_na() %>%
5    summarize(mean_body_mass_g = mean(body_mass_g))
6
```

Run

Reset

```
Warning in install.packages(., "tidyverse") :
  'lib = "tidyverse"' is not writable
Error in install.packages(., "tidyverse") : unable to install packages
```

```
Calls: %>% ... freduce -> withVisible -> <Anonymous> -> install.packages
```

What is the mean body mass in g for the Adelie species?

- ○ 4207.433
- ◉ 3706.164
- ○ 5092.437
- ○ 3733.088

**Correct**
The code chunk `summarize(mean(body_mass_g))` lets you find the mean value for the variable body_mass_g. The correct code is `penguins %>% drop_na() %>% group_by(species) %>% summarize(mean(body_mass_g))`. The summarize() function displays summary statistics. You can use the summarize() function in combination with other functions -- such as mean(), max(), and min() -- to calculate specific statistics. In this case, you use mean() to calculate the mean value for body mass. The mean body mass for the Adelie species is 3706.164g.

**7.** A data analyst is working with a data frame named *salary_data*. They want to create a new column named *wages* that includes data from the *rate* column multiplied by 40. What code chunk lets the analyst create the *wages* column? **1 / 1 point**

- ○ `mutate(salary_data, wages = rate + 40)`
- ○ `mutate(wages = rate * 40)`
- ○ `mutate(salary_data, rate = wages * 40)`
- ◉ `mutate(salary_data, wages = rate * 40)`

**Correct**
The code chunk is `mutate(salary_data, wages = rate * 40)`. The analyst can use the mutate() function to create a new column called wages that includes data from the rate column multiplied by 40. The mutate() function can create a new column without affecting any existing columns.

**8.** A data analyst is working with a data frame named *stores*. It has separate columns for city (*city*) and state (*state*). The analyst wants to combine the two columns into a single column named *location*, with the city and state separated by a comma. What code chunk lets the analyst create the *location* column? **1 / 1 point**

- ◉ `unite(stores, "location", city, state, sep=",")`
- ○ `unite(stores, city, state, sep=",")`
- ○ `unite(stores, "location", city, sep=",")`
- ○ `unite(stores, "location", city, state)`

**Correct**
The code chunk `unite(stores, "location", city, state, sep=",")` lets the analyst create the *location* column. The unite() function lets the analyst combine the city and state data into a single column. In the parentheses of the function, the analyst writes the name of the data frame, then the name of the new column in quotation marks, followed by the names of the two columns they want to combine. Finally, the argument `sep="," ` places a comma between the city and state data in the *location* column.

**9.** **1 / 1 point**

In R, which statistical measure demonstrates how strong the relationship is between two variables?

- ○ Maximum
- ◉ Correlation
- ○ Standard deviation
- ○ Average

Correct
Correlation measures how strong the relationship between two variables is. This is represented by the cor() function.

**10.** A data analyst wants to find out how much the predicted outcome and the actual outcome of their data model differ. What function can they use to quickly measure this? **1 / 1 point**

- (●) bias()
- ( ) cor()
- ( ) sd()
- ( ) mean()

Correct
The bias() function can be used to calculate the average amount a predicted outcome and actual outcome differ in order to determine if the data model is biased.