Congratulations! You passed!
TO PASS 75% or higher

Keep Learning

GRADE
100%

Manipulating Variables and Creating Summaries Quiz

LATEST SUBMISSION GRADE
100%

1. Say I have an tibble called "cities" with columns including population ("population"), a measure of economic activity ("gdp"), and the state in which the city is located ("state").

   Which of these commands would select rows from the dataset where the value for population is more than 3,000, the value for economic activity is less than 120,000 and where the city is not located in Alabama?

   1 / 1 point

   - ○ filter(cities,population>3000, gdp<120000, state!="Alabama")
   - ◉ filter(cities,population>3000 & gdp<120000 & state!="Alabama")
   - ○ filter(cities,population>3000 & gdp<120000 | state!="Alabama")
   - ○ select(cities,population>3000 & gdp<120000 & state!="Alabama")

   Correct
   You got it!

2. Let's say you have a dataset that looks like this:

   1 / 1 point

```
colors <- c("red","green","yellow")
speeds <- c("slow","fast","medium")
my_dat <- data.frame(colors,speeds)

```

   What is the correct code to recode the "colors" column so that red equals 0, green equals 2, and yellow equals 1?

   - ○ my_dat$colors<-recode(my_dat$colors,"red"==0,"green"==2,"yellow"==1)
   - ◉ my_dat$colors<-recode(my_dat$colors,"red"=0,"green"=2,"yellow"=1)
   - ○ my_dat$colors<-recode(my_dat$colors,0=red,2=green,1=yellow)

   Correct
   Correct!

3. Let's say you have a tibble named "dat" that has a time, date, employee, and sales column.

   1 / 1 point

   You are reviewing someone's R code and see the following lines:

   my_time <- filter(dat,time==1)

   my_time_and_date <- filter(my_time,date>5)

   group_by_employee_my_time_and_date <- group_by(my_time_and_date,employee)

   summarise(group_by_employee_my_time_and_date ,average=mean(sales))

   Which of these would do the same thing using piping?

   - ○ dat %>% filter(time=1 & date >5) %>% group_by(employee) %>% summarise(average=mean(sales))
   - ◉ dat %>% filter(time==1 & date >5) %>% group_by(employee) %>% summarise(average=mean(sales))
   - ○ dat %>% filter(time==1 & date >5) %>% group_by(employee) %>% summary(average=mean(sales))
   - ○ summarise(average=mean(sales)%>%dat %>% filter(time==1 & date >5) %>% group_by(employee)

   Correct
   Yes! This is the more compact method.

4. You have a data frame named "dat" with two numeric columns, value1 and value2. You want to add a third column called my_value where the value in each row is the product of multiplying the other two values in the row by one another. Which is the correct line of code?

   1 / 1 point

   - ○ add_column(dat,my_value=value1*value2)
   - ○ new_column(dat,my_value=value1*value2)

- (●) dat<- mutate(dat,my_value=value1*value2)
- ( ) mutate(dat,my_value=value1*value2)

✓ **Correct**
Correct!

5. Say you have a tibble saved into your R environment as "my_dat" with two columns named "alpha" and "beta". You want to rename the "beta" column and call it "gamma". Which of these will create a new tibble with the renamed column? **1 / 1 point**

- ( ) recode(my_dat,gamma=beta)
- ( ) rename(my_dat,beta=gamma)
- (●) rename(my_dat,gamma=beta)
- ( ) mutate(my_dat,gamma=beta)

✓ **Correct**
Correct!

6. Say you have a tibble saved to the object my_dat with two columns, alpha and beta. These are filled with numeric data. Which of these will arrange the data in descending order by alpha? **1 / 1 point**

- [ ] my_dat %>% arrange(alpha)
- [ ] desc(arrange(alpha,my_dat))
- [x] arrange(my_dat,desc(alpha))

✓ **Correct**
Correct! This version does it without piping.

- [x] my_dat %>% arrange(desc(alpha))

✓ **Correct**
Correct! This version uses piping.

7. Which of these accurately describes piping? **1 / 1 point**

- [x] Piping allows you to write cleaner, more readable code.

✓ **Correct**
Correct! Piping makes for easier to read code.

- [x] Piping will pass data on the left hand side of the pipe to the function on the right hand side of the pipe.

✓ **Correct**
Correct! Piping allows you to move data from left to right in a series of functions.

- [ ] Piping will pass data on the right hand side of the pipe to the function on the left hand side of the pipe.
- [ ] Piping is an advanced technique that only experience programmers should use.

8. What's the point of using group_by()? **1 / 1 point**

- [ ] Grouping your data will create a new column in the data that allows you to manage groups.
- [x] Grouping your data allows you to use those groups when summarizing data.

✓ **Correct**
Correct!

- [ ] Grouping your data allows you to keep track of large data sets in the R environment.
- [ ] Grouping your data create several new tibbles by group.