

ICA 2.2 Filtering, summarizing, and grouping (solutions)

```
library(tidyverse)
polyps <- read_csv("data/polyps.csv")
```

Task - Filter a data frame

1. Consider the values assigned to `weights` below. Write a logical expression that checks whether each value is greater than 150. Predict the outcome. Then modify the expression to check weights above and including 200, again making and checking a prediction.

```
weights <- c(264, 356, 155, 280, 175)
```

```
# add your code below
```

2. Use `dplyr::filter()` to retrieve the elder half of participants (mean age is 24). Note that you will need to use a logical expression to do this.

```
# P
# your code here
```

3. Write the complement of the previous query, which should return participants aged 24 and younger. Confirm that the row counts of both queries add up to the total rows in the data frame. (You can see a data frame's row count in the Environment pane, by printing it to console, or running `nrow()` on the data frame.)
4. Suppose you did not know the mean age (and didn't want to calculate it). Rewrite the previous query so that it compares `age` to `mean(age)`. Visually confirm that you have the same result.
5. Isolate placebo and treated subjects into two separate data frames. Assign each filter result into a new object, one called `placebo` and the other called `treated`. This will take two+ lines of code. Confirm the results by printing a representation of each data frame to the console and examining values. (Hints: the comparators you want are `==` and `!=`. Make sure that string values always have quotes around them, e.g., `name == "Lee"` for finding rows whose `name` is Lee).

Answer - Filter a data frame

```
# 1
weights > 150
weights > 200
# 2
polyps %>% filter(age > 24)
# 3
polyps %>% filter(age <= 24)
# 4
polyps %>% filter(age <= mean(age))
# 5
placebo <- polyps %>%
  filter(treatment == "placebo")
placebo
```

```
treated <- polyps %>%  
  filter(treatment != "placebo")
```

Answer commentary.

Task - Summarize a data frame

1. Use `dplyr::summarize()` to produce one row showing the mean age, the mean baseline, the median baseline, and the mean 3-month polyp count.

```
# your code here
```


2. Copy and modify the previous code to round each number to one decimal place. (You may need to scroll right-to-left to see all of the result columns.)

Answer - Summarize a data frame

```
# 1  
polyps %>%  
  summarize(mean(age), mean(baseline), mean(number3m))  
# 2  
polyps %>%  
  summarize(round(mean(age), 1),  
            round(mean(baseline), 1),  
            round(mean(number3m), 1))
```

Task - Group (and summarize) a data frame

1. Using `dplyr::group_by()`, group `polyps` by treatment, and then summarize the groups according to their number of members (row count), average baseline, their average 3-month polyp count, and their maximum 3-month polyp count. (Hint: `n()` gives a row count for a group.) Do you notice differences between the groups?

```
#   
# your code here
```

2. Repeat the previous query, but add a grouping for sex within each treatment group. How does the result change? How about changing the grouping order?

Answer - Group (and summarize) a data frame

```
# 1  
polyps %>%  
  group_by(treatment) %>%  
  summarize(n(), mean(baseline), mean(number3m), max(number3m))  
# 2  
polyps %>%  
  group_by(treatment, sex) %>%  
  summarize(n(), mean(baseline), mean(number3m), max(number3m))
```

