Practice Questions for Prelim 1

AEM 2850 / AEM 5850 – Fall 2025

Answer Key

READ THESE NOTES FIRST:

- Prelim 1 will cover all content we covered in weeks 1 through 5
- These practice questions are intended as a study resource, not a comprehensive guide
- These practice questions are not exhaustive in terms of topics and question types
- These practice questions are not necessarily representative of the weight that different topics and question types will receive on Prelim 1

Preface

The goal of this prelim is to assess your familiarity with programming concepts, ability to approach programming tasks, and facility with key data wrangling tasks we covered in weeks 1 through 5 of the course.

Instructions

- You must complete Prelim 1 in person
- Prelim 1 is a closed-book paper prelim
- Manage your time carefully
- If you get stuck, move on and come back later as time allows

Additional notes

- There are X questions worth a total of 100 points. The total number of points per question is stated with each question
- We will give partial credit if your answers are incomplete, especially if you outline the logic of what you *would* do if you had more time

Multiple Choice: circle only one answer per question

Q. [X points] What does the `mutate()` function do in the tidyverse?

a. Filters rows
b. Creates new variables
c. Changes column names
d. Removes duplicates

Solution

b. Creates new variables

Q. [X points] Which join keeps all rows from the left table and only matching rows from the right?

a. `inner_join()`
b. `right_join()`
c. `left_join()`
d. `full_join()`

Solution

c. `left_join()`

Q. [X points] What does `%in%` do in R?

a. It checks if a vector is contained in another vector
b. It adds elements to a vector
c. The same thing as `==`
d. It checks whether elements of one vector are contained in another vector

Solution

d. It checks whether elements of one vector are contained in another vector

Q. [X points] Suppose you want to keep only rows where the value of `price` is greater than 100. Which code would you use?

a. `filter(price < 100)`
b. `select(price > 100)`
c. `filter(price > 100)`
d. `arrange(price > 100)`

Solution

c. `filter(price > 100)`

Q. [X points] Which operator checks for exact equality in R?

a. `=`
b. `==`
c. `:=`
d. `is_equal()`

Solution

b. `==`

Multiple Choices: circle any number of answers per question

Q. [X points] Which of the following expressions return `TRUE`?

a.

```
5!=4
```

b.

```
!FALSE
```

c.

```
TRUE & FALSE
```

d.

```
2 > 1 | NA
```

e.

```
!(FALSE | TRUE)
```

Solution

a, b, and d are correct.

Short Answer

Q. [X points] When using a tidyverse join function to combine data frames, how does R determine which columns to use as join keys if you don't provide explicit instructions?

Solution

R uses all columns with matching names in both data frames as the join keys. This is referred to as a "natural join," based on the intersection of column names.

Q. [X points] What is the purpose of `read_csv()` and, at a high level, what does it do?

Solution

It reads in a comma separated values file and builds up a data frame. In order to do this, it has to parse the data types stored in each column of the original plain-text file and then create columns of that type in the data frame. This is because plain-text files do not store information about the types of data in each column.

Q. [X points] Will this expression return 5? Why or why not?

```
"3" + "2"
```

Solution

No, it will return Error in "3" + "2" : non-numeric argument to binary operator.

Q. [X points] A marketing analyst is asked to analyze promotional campaign performance by combining data on promotions (e.g., `Buy1Get1`) across each season, and then analyzing each promotion separately. They receive the following data frame of results from the campaigns:

```
# A tibble: 28 x 5
  campaign        impressions clicks conversions spend
  <chr>                 <dbl>  <dbl>       <dbl> <dbl>
1 Spring-Save20         12000    800          75   300
2 Spring-Buy1Get1       18000   1200         130   500
3 Fall-Save20           15000    900          95   400
4 Winter-Save20         16000   1000         110   450
# i 24 more rows
```

Are the data tidy? Why or why not?

Solution

The data are not tidy because each cell in the column `campaign` contains two measurements: the `season` and the `promotion`.

If the data are not tidy, what (if anything) would you do to make them tidy? Explain what your conceptual approach would be, name the function(s) you would use, and describe any important argument(s) you would include.

Note: If the data are already tidy, you can leave this question blank or restate that here. We will award credit based on the logic of your approach first and foremost, followed by your understanding of key functions needed to implement the approach. You are not expected to write a complete code snippet that will run without errors (and will not receive any extra credit if you do), though you are welcome to do so if it helps you to explain your answer.

Solution

To tidy the data, we would need to separate the column `campaign` into two separate columns. One way to do this would be to use `separate_wider_delim`, using a hyphen (`-`) as the delimiter at which to separate `campaign` into two columns. We would also need to specify the names of the two new columns to create.

```
promo_data |>
  separate_wider_delim(campaign, delim = "-", names = c("season", "promo"))
```

Q. [X points] Consider the following data frame `stocks`:

```
# A tibble: 6 x 3
  date       stock price
  <date>     <chr> <dbl>
1 2025-10-01 AAPL   234.
2 2025-10-01 GOOG   199.
3 2025-10-01 MSFT   412.
4 2025-09-30 AAPL   236.
5 2025-09-30 GOOG   195.
6 2025-09-30 MSFT   364.
```

If you ran the following code, how many rows and columns will the result contain? What are the column names?

```
stocks |>
  pivot_wider(names_from = stock, values_from = price)
```

Solution

The resulting data frame will have 2 rows and 4 columns: `date`, `AAPL`, `GOOG`, and `MSFT`.

```
# A tibble: 2 x 4
  date        AAPL  GOOG  MSFT
  <date>     <dbl> <dbl> <dbl>
1 2025-10-01  234.  199.  412.
2 2025-09-30  236.  195.  364.
```

Q. [X points] The `coffee_sales` dataset below contains sales data from March 2024 to February 2025:

```
# A tibble: 3,071 x 6
   year month date                payment_type money coffee_name
  <dbl> <dbl> <dttm>              <chr>        <dbl> <chr>
1  2024     3 2024-03-01 00:00:00 card          38.7 Latte
2  2024     3 2024-03-01 00:00:00 card          38.7 Hot Chocolate
3  2024     3 2024-03-01 00:00:00 card          38.7 Hot Chocolate
4  2024     3 2024-03-01 00:00:00 card          28.9 Americano
5  2024     3 2024-03-01 00:00:00 card          NA   Latte
# i 3,066 more rows
```

You wrote code to compute the total monthly sales for each available month in 2024:

```
coffee_sales |>
  filter(year == 2024) |>
  group_by(month) |>
  summarise(total_sales = sum(money))
```

Do you think the above code will produce the correct total sales for every month? Why or why not?

Solution

No, there is at least one instance of an `NA` value in `money` in the rows of the data frame above. As a result, the output will contain some `NA` values.

If not, how would you revise your approach to do so?

Solution

Acceptable answers include:

1. Investigate the cause of `NA` values, rectify them, and then proceed with the analysis.
2. Replace `mean(money)` with `mean(money, na.rm = TRUE)` to remove missing values before computing the mean. Full credit only requires a qualitative explanation of the solution, not perfect syntax. Other approaches that would achieve the same objective, such as filtering out `NA` values first, are also acceptable answers.

Q. [X points] You've been asked to analyze purchase behavior for an e-commerce company. The company stores customer details in one table, and tracks each order in another:

```
customers
```

```
# A tibble: 3 x 3
  customer_id name         segment
        <dbl> <chr>        <chr>
1         101 Alice Kim    Premium
2         102 Brian Chen   Standard
3         103 Carlos Lopez Standard
```

```
orders
```

```
# A tibble: 4 x 4
  order_id customer_id order_date amount
     <dbl>       <dbl> <date>      <dbl>
1      201         101 2023-10-01    120
2      202         102 2023-10-02     75
3      203         101 2023-10-05     90
4      204         104 2023-10-07     60
```

The company asked you to analyze all the purchases by customer type. Your manager told you to start by merging the tables using `customer_id`, preserving all purchases without introducing unnecessary information. Write a brief code snippet to share with your manager.

Solution

```
orders |>
  left_join(customers, by = join_by(customer_id))
```

How many rows will the resulting data frame contain?

Solution

There should be four rows, one corresponding to each order. The result should not contain a row for Carols Lopez.