EDS 221 Day 2 PM

Topic: *Indexing and 2d data*

Subgoal Labeled Worked Examples

Instructions

Problem-solving in programming, and data science in general, requires you to **recognize and apply procedures**. The biggest difference between a new programmer and an expert programmer is that experts have **mental models** to see complex procedures as a single step. **Worked examples** show learners how to use a procedure to accomplish a **goal**, such as accessing data (today's topic) or making choices based on a condition (tomorrow's topic). Research shows that pairing worked examples with **subgoal labels** helps learners' build the mental models they need to become experts (Margulieux, Morrison, & Decker 2020; Morrison, Margulieux, & Guzdial 2015).

In these exercises, you will see how an overall goal (e.g., evaluating indexing expressions) is composed of multiple subgoals. You will first be presented with a worked example of how to recognize the subgoals in sample code. Then you will be presented with a prompt (in plain language or code) to solve using the subgoals.

References

Margulieux, L. E., Morrison, B. B., & Decker, A. (2020). Reducing withdrawal and failure rates in introductory programming with subgoal labeled worked examples. *International Journal of STEM Education*, 7(1), 19.

Morrison, B. B., Margulieux, L. E., & Guzdial, M. (2015). Subgoals, context, and worked examples in learning computing problem solving. In *Proceedings of the Eleventh Annual International Conference on International Computing Education Research* (pp. 21-29).

EDS 221 Day 2 PM

Topic: *Indexing and 2d data*

Evaluating indexing expressions

Subgoals

1. Identify the object being indexed (vector, list, matrix, etc)
2. Identify the index (inside [ ] or following $)
3. Determine the type of the index (positive integers, negative integers, conditions, names, blanks) and evaluate it
4. Apply the index to the object and determine the result

Worked example

```
coral_cover_pct <- c(34.53, 22.26, 11.66, 9.25, 14.05, 22.30, 30.35,
16.35, 5.74)
coral_cover_pct[8:9]
```

1. **Identify the object being indexed (vector, list, matrix, etc)**

coral_cover_pct is a vector of doubles

2. **Identify the index (inside [ ] or following $)**

The index 8:9 is inside [ ]

3. **Determine the type of the index (positive integers, negative integers, conditions, names, blanks) and evaluate it**

8:9 is a positive integer index, evaluating to 8, 9

4. **Apply the index to the object and determine the result**

The 8th and 9th values are 16.35 and 5.74

EDS 221 Day 2 PM

Topic: *Indexing and 2d data*

Your turn 1

```
coral_cover_pct[c(1, length(coral_cover_pct))]
```

1. Identify the object being indexed (vector, list, matrix, etc)

2. Identify the index (inside [ ] or following $)

3. Determine the type of the index (positive integers, negative integers, conditions, names, blanks) and evaluate it

4. Apply the index to the object and determine the result

Your turn 2

```
n_observations <- c(Acropora = 13882, Pocillopora = 13988, Porites =
3192)
coral_name <- "Pocillopora"
n_observations[coral_name]
```

1. Identify the object being indexed (vector, list, matrix, etc)

2. Identify the index (inside [ ] or following $)

3. Determine the type of the index (positive integers, negative integers, conditions, names, blanks) and evaluate it

4. Apply the index to the object and determine the result

EDS 221 Day 2 PM

Topic: *Indexing and 2d data*

Your turn 3

```
songbird_survey <- data.frame(
 species = c("California Towhee", "House Finch", "House Sparrow"),
 song_freq_khz = c(4.99, 2.66, 4.08)
)
songbird_survey$species
```

1. Identify the object being indexed (vector, list, matrix, etc)

2. Identify the index (inside [ ] or following $)

3. Determine the type of the index (positive integers, negative integers, conditions, names, blanks) and evaluate it

4. Apply the index to the object and determine the result

EDS 221 Day 2 PM

Topic: *Indexing and 2d data*

Writing indexing expressions

Subgoals

1. Identify the object to index into (vector, list, matrix, etc)
2. Determine the appropriate method for indexing (by position, by exclusion, by condition, or by name)
3. Create the index (colon operator, seq(), c(), etc)
4. Apply the index ([] or $)

Worked example

```
survey_years <- seq(2008, 2024, by = 2)
```

Goal: drop the first two elements from `survey_years`

1. **Identify the object to index into (vector, list, matrix, etc)**

We want to index into survey_years

2. **Determine the appropriate method for indexing (by position, by exclusion, by condition, or by name)**

To drop elements we'll use negative integers

3. **Create the index (colon operator, seq(), c(), etc)**

We want -1 and -2, so our index is -(1:2)

4. **Apply the index ([] or $)**

survey_years[-(1:2)]

EDS 221 Day 2 PM

Topic: *Indexing and 2d data*

Your turn 1

```
survey_years <- seq(2008, 2024, by = 2)
coral_cover_pct <- c(34.53, 22.26, 11.66, 9.25, 14.05, 22.30, 30.35,
16.35, 5.74)
```

Goal: what were the survey years before 2015?

1. Identify the object being indexed (vector, list, matrix, etc)

2. Identify the index (inside [ ] or following $)

3. Determine the type of the index (positive integers, negative integers, conditions, names, blanks) and evaluate it

4. Apply the index to the object and determine the result

Your turn 2

Goal: what was the coral cover percentage in 2010?

1. Identify the object being indexed (vector, list, matrix, etc)

2. Identify the index (inside [ ] or following $)

3. Determine the type of the index (positive integers, negative integers, conditions, names, blanks) and evaluate it

4. Apply the index to the object and determine the result

EDS 221 Day 2 PM

Topic: *Indexing and 2d data*

Your turn 3

```
songbird_survey <- data.frame(
 species = c("California Towhee", "House Finch", "House Sparrow"),
 song_freq_khz = c(4.99, 2.66, 4.08)
)
```

Goal: what song frequencies were recorded?

1. Identify the object being indexed (vector, list, matrix, etc)

2. Identify the index (inside [ ] or following $)

3. Determine the type of the index (positive integers, negative integers, conditions, names, blanks) and evaluate it

4. Apply the index to the object and determine the result