COLLEGE OF SOCIAL & BEHAVIORAL SCIENCES
Women in Science & Engineering

The Bio/Diversity Project
Lesson Title: Coding Lesson #2

Teacher: Grace and Lily
Grade Level: 9th
Lesson Length: 50 minutes

This lesson has been adapted from: How can you use data visualization to enhance scientific communication?

AZ Science Standard:	• U2: The knowledge produced by science is used in engineering and technologies to solve problems and/or create products.
Learning Objective:	• Students will be able to use R to write code that creates a visualization of the wildlife camera footage we collected. • Students will be able to explain how data science contributes to science communication. • Students will be able to print their graphs to put on their posters next week.

Scientist of the Week:	• Student's Choice - 4 minute duration
	• Who is a scientist in your life that you want to showcase?
	• Scientist: someone who studies natural (physics, biology, chemistry) or social (politics, cultural, economics) sciences.
	• Nurse, teacher, chef, baker, gardener, researcher, zoo keeper, yourself, etc
	• In your journals write:
	  ○ Their name
	  ○ Your relationship
	  ○ Why you picked them
	  ○ A short description or story about their scientific work
	  ○ How their work made an impacted on you!
	• 
	• Conduct a Neighbor Share in which students will discuss with a table-mate who they picked, their scientist's background, and their work

Vocabulary	Materials
• Science communication • Scientist • Data visualization	• *Coding Lesson 2 Slides - Lesson 7* • *Coding lesson #2: Also in posit.cloud* • *Access to computers for coding*
Seasonality: Any	

COLLEGE OF SOCIAL & BEHAVIORAL SCIENCES
Women in Science & Engineering

Monsoons July-Sept.	*Autumn* Oct.-Nov.	*Winter* Dec.- Feb.	*Spring* Mar.-Apr.	*Dry Summer* May-June

Guiding Questions:

- What are data visualizations?
- How can data visualizations help scientists share their research and findings?
- How can we create a data visualization on the computer?

Engagement/Introductory Activity:

- Students will come up with their own scientist of the week
- They will pick someone in their lives who they want to showcase. It could be someone who studies natural (physics, biology, chemistry) or social (politics, cultural, economics) sciences, like a nurse, teacher, chef, baker, gardener, researcher, zookeeper, or themselves.
- They will be given 4 minutes to write in their journals, with the following prompts:
    - Their name
    - Your relationship
    - Why you picked them
    - A short description or story about their scientific work
    - How their work made an impact on you!

Exploratory Activity:

- Students will raise their hands and share one type of data visualization with the class. The interns will write the class's extensive list on the whiteboard. Examples are bar plots, pie charts, timelines, tables, sketches, etc.
- Students will learn why data visualization matters and how we can use computer science to enhance our science communication skills by:
    - ❖ Presenting complex data in a more accessible, engaging, and persuasive way
    - ❖ Capturing people's attention
    - ❖ Identifying the important points quickly
    - ❖ Comparing and contrasting scenarios
    - ❖ Highlighting patterns
    - ❖ Evoking emotions

Explain:

- Teachers will briefly explain what data visualization is and why it matters before heading to the computer lab to create one!
    - Data science is the study of collecting, organizing, and analyzing large amounts of data to uncover helpful information and insights. It involves using computer programming and statistical techniques to understand patterns and trends in data, which helps people answer questions.

    - We can use data science and computers to enhance our science communication skills by:
        - ■ Presenting complex data in a more accessible, engaging, and persuasive way
        - ■ Capturing people's attention
        - ■ Identifying the important points quickly
        - ■ Comparing and contrasting scenarios
        - ■ Highlighting patterns
        - ■ Evoking emotions

Extension Activity/Questions:

- Students will follow along with the code in posit.cloud to create their histograms, plotting the occurrences of birds found at CHS at different times of day.

Coding lesson #2

What are we doing?

Our wildlife cameras were busy capturing birds coming and going from Catalina High School. Each time a bird is spotted, the camera records the time of the day. We've saved all these timestamps in a file named valid_timestamps.csv.

But here's the challenge: how do we find out the most popular times for bird visits?

With a few lines of R code, we can create visual graphs that show us exactly when our feathered friends prefer to say hello!

Setting Up Our Toolbox

Before we can dive into our data, we need to make sure we have the right tools. In the world of coding, these tools are called "packages," and they help us do special tasks without writing a lot of code ourselves. For our adventure, we'll need three packages. Install them by running the following code. **You only have to run this once.**

```{r}
install.packages("ggplot2")
install.packages("dplyr")
install.packages("readr")
```

Now that the packages are installed, load them in using the following code:

```{r}
library(ggplot2)
library(dplyr)
library(readr)
```

Reading the Bird Log

Our next step is to read in our data, like we did last lesson. It is saved in the **`valid_timestamps.csv`** file. We'll run this code:

```{r}
data <- read_csv("valid_timestamps.csv")
```

Preparing the Data

Now, we have a data set full of times, but we want to break it up into categories so it is easier to understand. We need to organize it! We'll create categories for the time of day and format the hours in a way that's easier to read.

Don't worry about understanding every detail of this code; just know it helps us get our data ready for graphing:

```{r}
prepare_time_data <- function(data, time_column) {
  data %>%
    mutate(
      Hour = factor(substr(!!sym(time_column), 1, 2), levels = sprintf("%02d", 0:23)),
      TimeOfDay = cut(as.numeric(Hour), breaks = c(-1, 6, 12, 18, 24), labels = c("Night", "Morning", "Afternoon",
"Evening"), include.lowest = TRUE),
      HourNum = as.numeric(Hour), # Temp column for AM/PM conversion
      HourAMPM = ifelse(HourNum == 0, "12 AM", ifelse(HourNum < 12, paste(HourNum, "AM"),
ifelse(HourNum == 12, "12 PM", paste(HourNum - 12, "PM")))),
      HourAMPM = factor(HourAMPM, levels = c("12 AM", paste(1:11, "AM"), "12 PM", paste(1:11, "PM")))
    ) %>%
    # Ensure 'TimeOfDay' is a factor with all four periods as levels
    mutate(TimeOfDay = factor(TimeOfDay, levels = c("Night", "Morning", "Afternoon", "Evening")))
}
data_prepared <- prepare_time_data(data, "Time.stamp")
```

Creating Our Graphs

With our data ready, it's time to reveal the secrets it holds. We're going to create three graphs:

1. **Visits by Hour**: This shows us how many birds were spotted during each hour of the day in 24-hour time.

2. **Visits by Time of Day**: This divides the day into four parts - Night, Morning, Afternoon, and Evening - to see when birds prefer to visit.

3. **Visits by Hour with AM/PM**: A closer look at the hours using AM and PM, which we are more familiar with!

Graph #1: Bird Visits By Hour

This graph shows us how many bird visits were recorded during each hour of the day. It helps us see which hours are the busiest and which are the quietest. Replace the title (where it says "Enter Descriptive Title Here!") with a descriptive title of what this graph is showing!

```{r}
ggplot(data_prepared, aes(x = Hour)) +
  geom_bar(fill = "skyblue", color = "black") +
  scale_x_discrete(drop = FALSE) +
  labs(title = "Enter Descriptive Title Here!", x = "Hour", y = "Frequency") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        plot.title = element_text(hjust = 0.5))
```

Graph #2: **Bird Visits by Time of Day**

This graph divides the day into four parts—Night, Morning, Afternoon, and Evening—to see when birds like to visit most.

```{r}
```

```
ggplot(data_prepared, aes(x = TimeOfDay)) +
  geom_bar(fill = "lightgreen", color = "black") +
  scale_x_discrete(drop = FALSE) +
  labs(title = "Bird Visits by Time of Day", x = "Time of Day", y = "Frequency") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        plot.title = element_text(hjust = 0.5))
```

Graph #3: **Bird Visits by Hour with AM/PM**

This graph offers a closer look at the day's hours, using the familiar AM/PM format to see when our feathered friends prefer to make their appearances.

```{r}
ggplot(data_prepared, aes(x = HourAMPM)) +
  geom_bar(fill = "coral", color = "black") +
  scale_x_discrete(drop = FALSE) +
  labs(title = "Bird Visits by Hour (AM/PM)", x = "Hour (AM/PM)", y = "Frequency") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        plot.title = element_text(hjust = 0.5))
```

Pick Your Favorite Graph

Think back to the three graphs we created together:

1. **Bird Visits by Hour**

2. **Bird Visits by Time of Day**

3. **Bird Visits by Hour (AM/PM)**

Which one did you like the most? Was it the detailed hourly breakdown, the broad view of the day's times, or the familiar AM/PM format?

Paste in Your Code

Now, take the code for your favorite graph and paste it here:

```{r}
# Paste your favorite graph's code here!


```

Bonus Activity: Customize Your Graph

It's time to add your personal touch. We can do this by changing the colors. Remember the part in the code that looks like **`fill = "skyblue"`** or **`fill = "lightgreen"`**? That's where you decide what color your bars will be!

Run the following code to see the LONG list of colors that R understands:

```{r}
colors()
```

More Bonus Customization Options

Feeling adventurous? Here are a few more ways you can customize your graph:

- **Outline Color**: Change the **`color = "black"`** part to another color like **`"grey"`** or **`"darkblue"`** to modify the outline color of your bars.

- **Title Style**: Customize your title further by adding **`face = "bold"`** or changing its color with **`color = "darkgreen"`** inside the **`plot.title = element_text()`** part.

- **Background Color**: Give your plot a background color with **`theme(plot.background = element_rect(fill = "lightgrey"))`**, replacing **`"lightgrey"`** with your desired background color.

Example of Full Customization

Here's how your code might look after some customization:

```{r}
ggplot(data_prepared, aes(x = Hour)) +
  geom_bar(fill = "plum", color = "darkblue") +
  scale_x_discrete(drop = FALSE) +
  labs(title = "Example Graph: Bird Visits by Hour", x = "Hour", y = "Frequency") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, color = "navy"),
        plot.title = element_text(hjust = 0.5, face = "bold", color = "darkgreen"),
        plot.background = element_rect(fill = "lightgrey"))
```

Evaluation Activity:

- Once students have completed their graphs and styled them to their liking, they will be asked to save their final project on posit.cloud.
- Teachers will log on to posit.cloud, access each student's project, and save the image they created for their final graph.
- Teachers will print out each student's work and bring it to their final class where they will incorporate their graph in a poster.