Workshop #1

Water Quality and Summary Statistics

Appalachian A. I. Corps @ UTK

Notes

Section 1

Think about the water you drink. Where is it from? What do you know about it?

Notes

Lesson Objective

In this lesson, you will learn how to use the Python programming language to calculate summary statistics and investigate nitrates and nitrates in water samples.

Materials Needed:

- Your computer
- A web browser (Chrome, Firefox, or Safari)
- A calculator

Notes

Section 2

Workshop Structure

Notes

Workshop Structure

In the workshops, we will use:
- slides
- interactive modules
- handouts

Please have the module pulled up on your device and handout ready.

Navigate to: https://appalachianaicorps.org/ > Modules > Water Quality Monitoring > Lesson 1

Note: If you see a laptop icon in the slides, that means there is a corresponding activity in the module to complete!

Notes

Workshop Structure

There are multiple pages in each module. You navigate between them in the left-hand sidebar:

Lesson Contents

1. Overview
2. Intro to Water Quality
3. Nitrates Lab

4. Intro to Statistics
5. Python: Lists
6. Center & Spread

7. Boxplots Activity
8. Boxplots & Outliers

9. Exit Ticket

Optional Exercise

Notes

Workshop Structure

Each page contains **checkpoints**. The checkpoints are linked at the top of the page and include code blocks to run and questions to answer.
They are numbered by the page they are on (number) and the order they are within each page (letter).

Lesson Contents

Checkpoints on this Page:

- **Checkpoint 2.a:** Role of Nitrogen
- **Checkpoint 2.b:** Nitrogen in Your Water
- **Checkpoint 2.c:** Nitrogen Summary

Notes

Workshop Structure

Notes

The checkpoints will also be highlighted in the body of the module. When you come across one, that means there is something new to complete there!

Checkpoint 2.a: Role of Nitrogen

Section 3

Let's Get Started!

Notes

Notes

What is Python?

Python is a **programming language** — a way to give instructions to a computer.

Python can:

- analyze data
- do math
- make graphs

In these workshops, we will learn Python together!

Notes

Let's Try Python Right Now!

Python will print text. In programming, we call text **strings**.

Checkpoint 1.a: Print Statements

Click the Run Code button to run the block.

```
# This is a comment - Python ignores lines that start with #
# Let's make Python print a message!

print("Hello, Water Scientist!")
```

What happens?

Notes

Let's Try Python Right Now!

Python will print text. In programming, we call text **strings**.

Checkpoint 1.a: Print Statements

Click the Run Code button to run the block.

```
# This is a comment - Python ignores lines that start with #
# Let's make Python print a message!

print("Hello, Water Scientist!")
```

```
Hello, Water Scientist!
```

You just ran Python code! See how it printed "Hello, Water Scientist!" below the block?

Note that strings are always surrounded by quotation marks.

Notes

Python Does Math!

Python can do math problems way faster than we can!

Checkpoint 1.b: Mathematical Operators

Click the Run Code button to run the blocks.

Addition:

```
2 + 3
```

Subtraction:

```
3 - 2
```

Multiplication:

```
2 * 3
```

Division:

```
3 / 2
```

Notes

Python Does Math!

Checkpoint 1.b: Mathematical Operators

Click the Run Code button to run the blocks.

Addition:

```
2 + 3
```

```
5
```

Subtraction:

```
3 - 2
```

```
1
```

Multiplication:

```
2 * 3
```

```
6
```

Division:

```
3 / 2
```

```
1.5
```

Notes

Python Will Compare Values

Read these mathematical equations out loud. Let's try the first together!

Checkpoint 1.c: Comparing Values

Click the Run Code button to run the block.

```
2 + 3 >= 2 * 3
```

"The sum of 2 and 3 is greater than or equal to the product of 2 and 3."

Is this statement **TRUE** or **FALSE**?

Python can decide!

```
2 + 3 >= 2 * 3
```

```
False
```

Notes

Python Will Compare Values

You try the second one!

```
2 + 3 <= 2 * 3
```

Read the statement and make a prediction. Is this statement **TRUE** or **FALSE**?

Python says:

```
2 + 3 <= 2 * 3
```

```
True
```

Notes

Combining Math & Strings

You can combine strings and math!

Checkpoint 1.d: Print Statements with Math

Click the Run Code button to run the block.

```
print("2 + 3 =", 2 + 3)
```

Make a prediction. What will Python print?

```
print("2 + 3 =", 2 + 3)
```

```
2 + 3 = 5
```

Notes

Your Own Print Statements

Your turn! Edit the code in the block.

Checkpoint 1.e: Your own print statements with math!

Change "Student" to your first name.
Create a new line 9 to calculate 15 times 4. Use lines 5 and 6 as models.
Try other math problems.

Click the Run Code button to run the block.

```
# Change this message to your name!
print("My name is Student")

# Try different math problems
print("5 + 3 =", 5 + 3)
print("20 - 7 =", 20 - 7)

# Calculate 15 times 4 below. Print the equation and result.
```

Notes

Section 4

Intro to Water Quality

Notes

What is Water?

Question: What do you know about water?

H2O is pure water!

- But water is almost always mixed with other things:
  - minerals
  - salts
  - other chemicals

Some of these things help water be safe to drink, while others make it unsafe.

Notes

Water Quality

Notes

Water Quality

In the next several workshops, we will become water quality citizen scientists!

The video discussed several indicators of water quality. We'll focus on **nitrogen** today. It can serve as both a **nutrient** and a **pollutant**.

Dissolved Oxygen	Salinity Levels	pH
Water Temperature	Nutrient Levels	Chlorophyll Concentrations
Bacteria Levels	Concentrations of Pollutants	Pesticides Herbicies Heavy Metals

Notes

Nitrogen as Nutrient

Nitrogen is naturally occurring.
When it combines with air and water, it forms ions: nitrates (NO_3) and nitrites (NO_2).
Nitrogen within water is an important part of the nitrogen cycle, an important process necessary for life.

Notes

Nitrogen as Nutrient

Notes

Nitrogen as Pollutant

Too much nitrogen in water can be a **pollutant**.
Excess nitrogen–in the form of nitrates and nitrites–can result from:

- Agricultural operations (fertilizer runoff and livestock manure)
- Sewage and septic systems (human waste)
- Acid rain

Notes

Nitrogen as Pollutant

Notes

Water Quality: Nitrogen

Checkpoint 2.a: Role of Nitrogen

What is the role of nitrogen in our water?
Do we want nitrates in our streams?
Our drinking water?
How about water with lots of nitrogen?

Notes

Monitoring Nitrogen: Your Utility

The U.S. Environmental Protection Agency (EPA) sets limits on allowable concentrations of nitrates and nitrites in drinking water.

! Nitrate and Nitrite EPA Limits

Nitrates: 10 mg/L
Nitrites: 1 mg/L

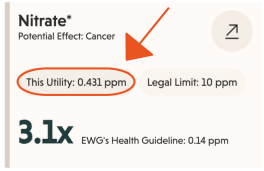
Notes

Monitoring Nitrogen: Your Utility

You can look up the reported nitrate concentration for water from your utility by using your zipcode.

Checkpoint 2.b: Nitrogen in Your Water

Using data from EWG, what is the most recently reported nitrate concentration by your water utility?



Notes

Monitoring Nitrogen: YOU!

We can also monitor our own water for nitrogen!

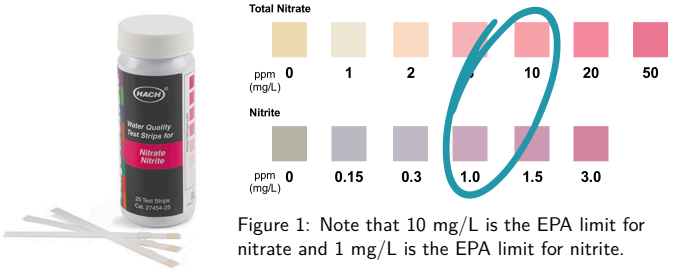


Figure 1: Note that 10 mg/L is the EPA limit for nitrate and 1 mg/L is the EPA limit for nitrite.

Notes

Section 5

Nitrate Lab

Notes

Nitrate Lab

We are going to conduct a brief water quality lab to practice gathering data about nitrates and nitrates in water samples.

Materials Needed:
- 4+ water quality samples per group
- 4+ nitrate/nitrate test strips per group
- 1 lab notesheet per person
- 1 computer with the submission form open for data entry
- link to form (provided by your teacher)
- pencils
- 4+ sets of multicolor sticky notes (one set per water sample)
- 1 Sharpie markers
- timing device (watch, clock, etc.)

Notes

Nitrate Lab: Roles

Assign roles to each person in your group:

Role 1: *Data Recorder (Lab Report)*
- 1 lab handout
- pencil

Role 2: *Data Recorder (Computer)*
- computer (everyone else can put theirs away temporarily)
- submission form pulled up in browser

Role 3: *Data Recorder (Sticky Notes)*
- sets of small sticky notes (4+ sets, one per water sample)
- Sharpie

Role 4: *Lab Technician*
- water samples
- test strips

Notes

Nitrate Lab: Procedure

Read the procedure on your lab sheet. Make sure everyone in your group understands the procedure and has the materials needed for their role.

LET'S BEGIN

When finished: With the help of the *Data Recorder (Lab Report)*, make sure everyone in your group has a copy of the data table on their own sheet.

Notes

Section 6

Intro to Statistics

Notes

What is Statistics?

How can we make sense of the data we just collected during the nitrates lab?

- We can use statistics!

Statistics is a **collection of tools** that can be used to **analyze data**.

Statistics helps us:

- Summarize lots of numbers into one useful number
- Find patterns
- Make decisions
- Spot things that don't fit

Notes

What is Statistics?

Notes

Measures of Center

We use **measures of center** to find the central value of a group of numbers. There are different types of central values. We'll use **mean** and **median** in this lesson.

Sample Dataset: 1, 2, 6, 5, 1

Mean

The *mean* is often called the "average". To find the mean, we add up all of the numbers of interest and divide by how many numbers there are.
$\frac{1+2+6+5+1}{5} = 3$
The mean value of this sample dataset is **3**.

Notes

Measures of Center

We use **measures of center** to find the central value of a group of numbers. There are different types of central values. We'll use **mean** and **median** in this lesson.

Sample Dataset: 1, 2, 6, 5, 1

Median

The *median* is a different kind of measure of center. To calculate the median, you first line up all the values in your data set, then you find the middle value by position.
$1 + 1 + 2 + 5 + 6$

Notes

Statistics: Your Turn

Materials Needed:
- Lab sheet
- Calculator

Checkpoint 4.a: Calculations—Mean & Median

Find the mean and median of your team's lab data:
- Calculate mean nitrate concentration
- Calculate mean nitrite concentration
- Calculate median nitrate concentration
- Calculate median nitrite concentration

Record your results on your lab sheet.

Notes

Section 7

Python: Lists!

Notes

Python: Lists!

Sample Dataset: 1, 2, 6, 5, 1

Lists

In Python, we store multiple numbers in something called a **list**.

```
# Our sample dataset from above
# (the square brackets [ ] make it a list)

sample_data = [1, 2, 6, 5, 1]

print("Our sample data:", sample_data)
```

```
Our sample data: [1, 2, 6, 5, 1]
```

Breaking it down:
- `sample_data` = the variable name we gave our list (you can name it anything!)
- = means "store this in the variable `sample_data`"
- `[1, 2, 6, 5, 1]` = the actual numbers, separated by commas

Notes

Python: Make Your Own Lists!

Checkpoint 5.a: Storing data as lists!

Replace the ??, ??, ??, and ?? placeholders with the nitrate and nitrite readings from your group!

Click the Run Code button to run the block.

```
# Create a list of your group's nitrate and nitrite readings!
# Change the 10, 20, 30, 40 to your group's real values

group_nitrate = [10, 20, 30, 40]
group_nitrite = [10, 20, 30, 40]

print("My group's nitrate readings:", group_nitrate)
print("My group's nitrite readings:", group_nitrite)
```

```
My group's nitrate readings: [10, 20, 30, 40]
My group's nitrite readings: [10, 20, 30, 40]
```

Notes

Python: Make Your Own Lists!

Checkpoint 5.b: Basic functions with lists.

len()

How many items are in a list?

```
print("Number of samples tested:", len(group_nitrate))
```

```
Number of samples tested: 4
```

min()

Smallest number in a list.

```
print("Lowest nitrate:", min(group_nitrate))
```

```
Lowest nitrate: 10
```

max()

Biggest number in a list.

```
print("Highest nitrate:", max(group_nitrate))
```

```
Highest nitrate: 40
```

Notes

Example: Water Safety Check!

Checkpoint 5.c: Making Comparisons—EPA Limits.

Replace the ?? placeholder with the EPA limit for nitrate!

Click the Run Code button to run the blocks.

```
# Let's check using your group's nitrate readings
max_nitrate = max(group_nitrate)

print("The water has a max nitrate concentration of", max_nitrate)

# Python can make decisions using 'if' statements!
if max_nitrate <= ??:
    print(" Nitrate concentration is in the safe range!")
else:
    print(" Nitrate concentration is outside the safe range!")
```

Notes

Example: Water Safety Check!

Checkpoint 5.c: Making Comparisons—EPA Limits.

Replace the ?? placeholder with the EPA limit for nitrate!

Click the Run Code button to run the blocks.

```
# Let's check using your group's nitrate readings
max_nitrate = max(group_nitrate)

print("The water has a max nitrate concentration of", max_nitrate)

# Python can make decisions using 'if' statements!
if max_nitrate <= 10:
    print(" Nitrate concentration is in the safe range!")
else:
    print(" Nitrate concentration is outside the safe range!")
```

```
The water has a max nitrate concentration of 40
 Nitrate concentration is outside the safe range!
```

Notes

Section 8

Center & Spread

Notes

Python: Measures of Center

We can also use Python to calculate measures of center.

- We'll import a library of functions called NumPy (Numerical Python) to help us.

Mean

```
import numpy as np

# Same sample dataset
sample_data = [1, 2, 6, 5, 1]

# Calculate mean the easy way!
mean_sample = np.mean(sample_data)

print("mean of sample data:", mean_sample)
```

```
mean of sample data: 3.0
```

Notes

Python: Measures of Center

Mean: Your Turn!

Checkpoint 6.a: Calculating group mean nitrate.
Replace the 10, 20, 30, and 40 placeholders with the nitrate readings from your group!
Click the Run Code button to run the block.

```
# Recreate the list of your group's nitrate readings below

group_nitrate = [10, 20, 30, 40]

mean_nitrate = np.mean(group_nitrate)

print("Group nitrate readings (mg/L):", group_nitrate)
print("Group mean nitrate:", mean_nitrate, "mg/L")

# Is it safe? (Remember: the EPA's limit is less than 10 mg/L)
if mean_nitrate < 10:
    print("  Average nitrate is SAFE")
else:
    print("  Average nitrate is TOO HIGH")
```

```
Group nitrate readings (mg/L): [10, 20, 30, 40]
Group mean nitrate: 25.0 mg/L
  Average nitrate is TOO HIGH
```

Notes

Class Data

Python can handle large amounts of data with ease!
We can consider the nitrate and nitrite data your whole class collected.
To do that, paste the .csv link from your teacher in the code to import the Google Form data.

! Your teacher will provide the link to the CSV file for you to use.

Notes

Class Data

Checkpoint 6.b: Import class data.

Replace the placeholder (Line 4) with CSV URL from your teacher. Be sure the keep the quotation marks! This will pull the class nitrate and nitrite data from the CSV file so you can use it later down the page.

Click the Run Code button to run the block.

```
# Replace the url with the one provided from your teacher.
# Make sure to keep the quotation marks!

csv_url = "replace_this_with_your_csv_url"

class_nitrate, class_nitrite = load_class_data(csv_url)
```

```
print("Nitrate values:", class_nitrate)
print("Nitrite values:", class_nitrite)
```

```
Nitrate values: [0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0, 5.5, 6.0, 6.5, 7.0, 7.5, 8.0, 9.0, 10.0, 10.5, 11.0, 12.0, 13.5, 15.
Nitrite values: [0.0, 0.0, 0.1, 0.1, 0.15, 0.2, 0.2, 0.25, 0.3, 0.3, 0.35, 0.4, 0.4, 0.5, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.2, 1.5, 1.8,
```

Notes

Class Data: Mean

Checkpoint 6.c: Calculate means for class data.

Replace the ??? placeholders (Lines 3 & 4) with the variables that hold the class nitrate and nitrite data. That is, `class_nitrate` and `class_nitrite`, respectively.

Click the Run Code button to run the block.

```
# Calculate mean nitrate and nitrite for class data (replace the ???)

mean_class_nitrate = np.mean(???)
mean_class_nitrite = np.mean(???)

print("Class nitrate:", mean_class_nitrate, "mg/L")
print("Class nitrite:", mean_class_nitrite, "mg/L")
```

Notes

Class Data: Mean

Checkpoint 6.c: Calculate means for class data.

Replace the ??? placeholders (Lines 3 & 4) with the variables that hold the class nitrate and nitrite data. That is, `class_nitrate` and `class_nitrite`, respectively.

Click the Run Code button to run the block.

Notes

```
# Calculate mean nitrate and nitrite for class data (replace the ???)

mean_class_nitrate = np.mean(class_nitrate)
mean_class_nitrite = np.mean(class_nitrite)

print("Class nitrate:", mean_class_nitrate, "mg/L")
print("Class nitrite:", mean_class_nitrite, "mg/L")
```

```
Class nitrate: 7.58 mg/L
Class nitrite: 0.71 mg/L
```

Your answers will differ. This is example data.

Class Data: Median

Checkpoint 6.d: Calculate medians for class data.

Similarly, replace ??? with the correct variable names to calculate the median for the class nitrate and nitrite data.

Click the Run Code button to run the block.

Notes

```
# Calculate median nitrate and nitrite for class data (replace the ???)

median_class_nitrate = np.median(???)
median_class_nitrite = np.median(???)

print("Class nitrate:", median_class_nitrate, "mg/L")
print("Class nitrite:", median_class_nitrite, "mg/L")
```

Class Data: Median

Checkpoint 6.d: Calculate medians for class data.

Similarly, replace ??? with the correct variable names to calculate the median for the class nitrate and nitrite data.

Click the Run Code button to run the block.

```
# Calculate median nitrate and nitrite for class data (replace the ???)

median_class_nitrate = np.median(class_nitrate)
median_class_nitrite = np.median(class_nitrite)

print("Class nitrate:", median_class_nitrate, "mg/L")
print("Class nitrite:", median_class_nitrite, "mg/L")
```

```
Class nitrate: 6.5 mg/L
Class nitrite: 0.4 mg/L
```

Your answers will differ. This is example data.

Notes

Python: Measures of Spread

Spread is a second type of statistical measure.
Like it sounds, it tells us how spread out our data are.
Spread, also referred to as **variability**, is *the backbone* of statistics!

Range

The **range** is a fairly simple calculation. It's the maximum minus the minimum in a dataset.
max() - min()

```
# Same sample dataset
sample_data = [1, 2, 6, 5, 1]

sample_range = max(sample_data) - min(sample_data)

print("Range of sample data:", sample_range)
```

```
Range of sample data: 5
```

Notes

Class Data: Range

Range: Your Turn!

Checkpoint 6.e: Calculate ranges for class data.
Replace ??? with the correct variable names to calculate the ranges of the data.
Click the Run Code button to run the block.

```
class_nitrate_range = max(???) - min(???)
class_nitrite_range = max(???) - min(???)

print("Range of class nitrate data:", class_nitrate_range)
print("Range of class nitrite data:", class_nitrite_range)
```

Notes

Class Data: Range

Range: Your Turn!

Checkpoint 6.e: Calculate ranges for class data.
Replace ??? with the correct variable names to calculate the ranges of the data.
Click the Run Code button to run the block.

```
class_nitrate_range = max(class_nitrate) - min(class_nitrate)
class_nitrite_range = max(class_nitrite) - min(class_nitrite)

print("Range of class nitrate data:", class_nitrate_range)
print("Range of class nitrite data:", class_nitrite_range)
```

```
Range of class nitrate data: 22.0
Range of class nitrite data: 3.0
```

Your answers will differ. This is example data.

Notes

Class Data: Standard Deviation

Standard Deviation

The standard deviation is another measure of spread.
It tell us the average distance the data in the dataset is from the mean.

```
# Same sample dataset
sample_data = [1, 2, 6, 5, 1]

sample_stdev = np.std(sample_data)

print("Standard Deviation of sample data:", sample_stdev)
```

```
Standard Deviation of sample data: 2.0976176963403033
```

Notes

Class Data: Standard Deviation

Standard Deviation: Your Turn!

Checkpoint 6.f: Calculate standard deviations for class data.
Replace ??? with the correct variable names to calculate the standard deviations of the data.
Click the Run Code button to run the block.

```
class_nitrate_stdev = np.std(???)
class_nitrite_stdev = np.std(???)

print("Standard deviation of class nitrate data:", class_nitrate_stdev)
print("Standard deviation of class nitrite data:", class_nitrite_stdev)
```

Notes

Class Data: Standard Deviation

Standard Deviation: Your Turn!

Checkpoint 6.f: Calculate standard deviations for class data.
Replace ??? with the correct variable names to calculate the standard deviations of the data.
Click the Run Code button to run the block.

```
class_nitrate_stdev = np.std(class_nitrate)
class_nitrite_stdev = np.std(class_nitrite)

print("Standard deviation of class nitrate data:", class_nitrate_stdev)
print("Standard deviation of class nitrite data:", class_nitrite_stdev)
```

```
Standard deviation of class nitrate data: 5.40865972307373
Standard deviation of class nitrite data: 0.7532595834106592
```

Your answers will differ. This is example data.

Notes

Causes of High Variability

If a creek's reading has high standard deviation (lots of variability in comparison to other creeks), it might mean:

- **Pollution events** - A factory dumps waste occasionally
- **Storm runoff** - Rain washes fertilizer from farms into the creek
- **Natural cycles** - Some creeks naturally vary
- **Sensor problems** - The equipment might need calibration
- **Different sampling times** - Morning vs afternoon can be different

Notes

Section 9

Boxplots Activity

Notes

Boxplots: Visualizing Data

A key practice in statistics is to visualize the data!
One type of plot we can use to do this is a **boxplot**.
To make a boxplot, we will order the data and then divide the data into four equal groups (by number of observations), called **quartiles**.

Attention *Data Recorder (Sticky Notes)*

Data Recorders (Sticky Notes), sort all of your sticky notes by color. Pass them out to each group, matching the sticky's color to the group's tent color. This will result in each group having a full set of stickies representing the class nitrate data.

Notes

Boxplots: Activity

Checkpoint 7.a: As a group, construct a boxplot for the class nitrate data by hand.

Materials Needed:
- 1 Post-It grid paper sheet per group
- 1 poster marker per group
- Your group's full set of stickies (including those from other groups) from the nitrate lab
- Lab sheet handout (back)

Follow the steps on the back of your lab sheet to create a boxplot for the class nitrate data.

LET'S GO

Notes

Section 10

Boxplots & Outliers

Notes

Class Data

Whew! That was a lot of work. Wouldn't it be great if Python could do it for us? Good news. It can! But first, let's re-import our data on this page.

Checkpoint 8.a: Import Class Data.

Once more, replace the placeholder (Line 4) with CSV URL from your teacher. Be sure the keep the quotation marks! This will pull the class nitrate and nitrite data from the CSV file so you can use it later down the page.

Click the Run Code button to run the block.

```
# Replace the url with the one provided from your teacher.
# Make sure to keep the quotation marks!

csv_url = "replace_this_with_your_csv_url"

class_nitrate, class_nitrite = load_class_data(csv_url)
```

```
print("Nitrate values:", class_nitrate)
print("Nitrite values:", class_nitrite)
```

```
Nitrate values: [0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0, 5.5, 6.0, 6.5, 7.0, 7.5, 8.0, 9.0, 10.0, 10.5, 11.0, 12.0, 13.5, 15.
Nitrite values: [0.0, 0.0, 0.1, 0.1, 0.15, 0.2, 0.2, 0.25, 0.3, 0.3, 0.35, 0.4, 0.4, 0.5, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.2, 1.5, 1.8,
```

Notes

Class Data: Boxplots

Checkpoint 8.b: Create a boxplot for the class nitrate data.

Click the Run Code button to run the block and create a boxplot for the class nitrate data.

```
import matplotlib.pyplot as plt

plt.figure(figsize=(6, 6))
plt.boxplot(class_nitrate, vert=True, patch_artist=True,
            boxprops=dict(facecolor='plum'))
plt.ylabel('Nitrate (mg/L)', fontsize=12)
plt.xticks([])
plt.title('Boxplot of Class Nitrate', fontsize=13)
plt.grid(True, alpha=0.3, axis='y')
plt.show()
```

Notes

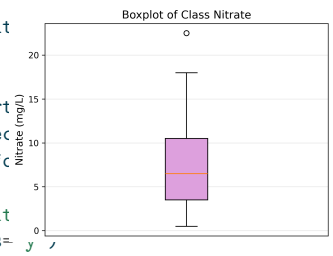
Class Data: Boxplots

Checkpoint 8.b: Create a boxplot for the class nitrate data.

Click the Run Code button to run the block and create a boxplot for the class nitrate data.

```
import matplotlib.pyplot as plt

plt.figure(figsize=(6, 6))
plt.boxplot(class_nitrate, vert
            boxprops=dict(facec
plt.ylabel('Nitrate (mg/L)', fc
plt.xticks([])
plt.title('Boxplot of Class Nit
plt.grid(True, alpha=0.3, axis=
plt.show()
```



This uses example data.

Notes

Class Data: Boxplots

Now, let's create a boxplot for the class nitrite data.

Checkpoint 8.c: Create a boxplot for the class nitrite data.

Click the Run Code button to run the block and create a boxplot for the class nitrite data.

```
plt.figure(figsize=(6, 6))
plt.boxplot(class_nitrite, vert=True, patch_artist=True,
            boxprops=dict(facecolor='lemonchiffon'))
plt.ylabel('Nitrite (mg/L)', fontsize=12)
plt.xticks([])
plt.title('Boxplot of Class Nitrite', fontsize=13)
plt.grid(True, alpha=0.3, axis='y')
plt.show()
```

Notes

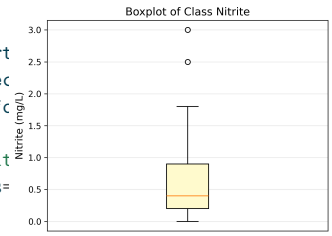
Class Data: Boxplots

Now, let's create a boxplot for the class nitrite data.

Checkpoint 8.c: Create a boxplot for the class nitrite data.

Click the Run Code button to run the block and create a boxplot for the class nitrite data.

```
plt.figure(figsize=(6, 6))
plt.boxplot(class_nitrite, vert
            boxprops=dict(facec
plt.ylabel('Nitrite (mg/L)', fc
plt.xticks([])
plt.title('Boxplot of Class Nit
plt.grid(True, alpha=0.3, axis=
plt.show()
```



This uses example data.

Notes

Outliers

Outliers are data points that are very different from the rest of your data.
- Can really change your statistics—like mean and standard deviation.
- On boxplots, outliers are represented by a circle (or sometimes a star) beyond the whiskers.

Do either the class nitrate or nitrite boxplot have outliers? How do you know?

Notes

Why Do Outliers Happen?

Outliers can happen for different reasons:

Buoy Malfunction
Real Pollution Event
- Factory dumped chemicals into the stream
- Farm fertilizer washed in after a storm
- Sewage spill

Natural Event
- Heavy rain changed water chemistry
- Algae bloom
- Seasonal variation

Your job as a scientist: Figure out which reason it is!

Notes

Section 11

Exit Ticket

Notes

Exit Ticket

Great job! You've learned so much!Share what you've learned on the **Exit Ticket**.

Notes

Section 12

Exercises

Notes

Exercises

Want to practice what we've learned?Try the **Exercises**.

Notes

Notes