

BUS 210 - QUIZ 2 - Python (Loops-Functions-Conditionals), Pandas, and NumPy

Spring 2024

✓ START HERE

Run the code block below and respond in the output.

Instructions:

This is a CLOSED-BOOK individual quiz. No resources can be used for this quiz.

Any violation of this will result in an academic violation.

Read all parts of this Jupyter Notebook carefully.

1. Ensure you **save a copy** of this quiz **to your drive** and **RENAME** it as "BUS210_Quiz2_YourFullName.ipynb"
 2. Share your Jupyter Notebook with me: amyremionkhale@depauw.edu
 3. Make sure you save your completed quiz as a pdf (via print)
 4. Submit your PDF to Moodle.
-

✓ Confirmation that you read the instructions.

Do NOT edit this code block below.

Run it and fill out the boxes (Hit enter/return when done).

```
#### Do NOT edit this.
#### Just RUN it and enter your responses in the output boxes below.

name = ""
confirmed = False

while not confirmed:
    name = input("Enter your name: ")

    instructions_confirm = input("Have you read the instructions? Yes or No: ").lower().strip()

    if instructions_confirm == "yes":
        confirmed = True
        print(f"Thank you {name}, your acknowledgment of the instructions is confirmed! \nYou can now start the quiz. \nRu
    else:
        print(f"Please read instructions and confirm {name} before proceeding.")

Enter your name: Anh Bui
Have you read the instructions? Yes or No: Yes
Thank you Anh Bui, your acknowledgment of the instructions is confirmed!
You can now start the quiz.
Run the Start Time Code Block below.
```

✓ **Run this code block below to mark the time you started the quiz.**

```
## Start Time Code Block.
print("I am starting the quiz now.")
import datetime
import pytz

# Define the Eastern Time Zone
eastern = pytz.timezone('US/Eastern')

# Convert the UTC time to Eastern Time
start_time = datetime.datetime.now(eastern)

print(f"Start time: {start_time}")

print("-----")

I am starting the quiz now.
Start time: 2024-04-18 12:48:09.667669-04:00
-----
```

✓ Quiz Questions

```
# Import necessary libraries
import pandas as pd
import numpy as np
import os

# Question 1 (10 points)
print("Question 1: Explain the key differences between NumPy arrays and Pandas DataFrames. Discuss their respective advantages and disadvantages.")

# Answer placeholder
answer1 = ''
My answer:
1. NumPy arrays
- Best for homogeneous numerical data, meaning they store elements of the same data type.
- They are multi-dimensional and primarily used for numerical computations.
- Provide efficient storage and operations for numerical data.
- Use cases: scientific computing, numerical simulations, and mathematical operations.

2. Pandas DataFrames
- Good with handling heterogeneous data, which means it allows different data types in columns.
- They are 2-dimensional and resemble a table, making them suitable for handling structured data.
- Offer labeled axes (rows and columns) and support data alignment, missing data handling, data manipulation, and relationships.
- Use cases:
  + Data cleaning, exploration, and analysis in data science projects.
  + Working with tabular data such as CSV files, Excel spreadsheets, and SQL databases...
  + Generating summary statistics, and visualizing data...
...
print(answer1)
```

Question 1: Explain the key differences between NumPy arrays and Pandas DataFrames. Discuss their respective advantages and disadvantages.

My answer:

- NumPy arrays
 - Best for homogeneous numerical data, meaning they store elements of the same data type.
 - They are multi-dimensional and primarily used for numerical computations.

- Provide efficient storage and operations for numerical data.
- Use cases: scientific computing, numerical simulations, and mathematical operations.

2. Pandas DataFrames

- Good with handling heterogeneous data, which means it allows different data types in columns.
- They are 2-dimensional and resemble a table, making them suitable for handling structured data.
- Offer labeled axes (rows and columns) and support data alignment, missing data handling, data manipulation, and
- Use cases:
 - + Data cleaning, exploration, and analysis in data science projects.
 - + Working with tabular data such as CSV files, Excel spreadsheets, and SQL databases...
 - + Generating summary statistics, and visualizing data...

Question 2 (15 points)

```
print("Question 2a: Create a NumPy array named 'arr' with the following values: [10, 15, 20, 25, 30]")
print("Question 2b: Calculate the mean and standard deviation of 'arr' ")
```

Answer code

```
print()
arr = np.arange(10, 35, 5)
print('Question 2a:',arr)

print('Question 2b:')
print('Mean of arr is:',np.mean(arr))
print('Standard deviation of arr is:', np.std(arr))
```

Question 2a: Create a NumPy array named 'arr' with the following values: [10, 15, 20, 25, 30]
 Question 2b: Calculate the mean and standard deviation of 'arr'

Question 2a: [10 15 20 25 30]
 Question 2b:
 Mean of arr is: 20.0
 Standard deviation of arr is: 7.0710678118654755

Question 3 (10 points)

```
print("Question 3: Create a Pandas DataFrame using the following data and print the DataFrame.")
data = {'Product': ['A', 'B', 'C'], 'Sales': [1000, 1500, 2000], 'Profit': [100, 150, 200]}
```

Answer code

```
data_frame = pd.DataFrame(data)
print(data_frame)
```

Question 3: Create a Pandas DataFrame using the following data and print the DataFrame.

	Product	Sales	Profit
0	A	1000	100
1	B	1500	150
2	C	2000	200

Question 4 (15 points)

```
print("Question 4: Given the sales data loaded below, calculate and print the total sales for all products.")
sales_data = np.array([[100, 150, 200], [120, 180, 250], [90, 110, 190]])
```

Answer code

```
print('Total sales for all products:',np.sum(sales_data))
```

Question 4: Given the sales data loaded below, calculate and print the total sales for all products.
 Total sales for all products: 1390

Question 5 (15 points)

```
print("Question 5: Explain the concept of broadcasting in NumPy. Provide an example of when broadcasting can be useful.")
```

Answer code

```
answer2 = '''
```

```
    Broadcasting is a powerful mechanism that allows numpy to work with arrays of different shapes when performing arithmetic.
    For example, we can add a scalar to each element of an array, or add arrays of different but compatible shapes.
    ...
```

```
print(answer2)
```

Question 5: Explain the concept of broadcasting in NumPy. Provide an example of when broadcasting can be useful.

Broadcasting is a powerful mechanism that allows numpy to work with arrays of different shapes when performing arithmetic. For example, we can add a scalar to each element of an array, or add arrays of different but compatible shapes.

Question 6 (25 points)

```
print("Question 6: Perform the following data handling tasks: ")
```

```
print("a) Set your working directory to a specific folder on your Google Drive.")
```

```
print("b) Use the given URL (https://tinyurl.com/yuzyb2nr) to download a CSV file and save it to your working directory.")
```

```
print("c) Load the CSV file into a Pandas DataFrame and display the first five rows of the dataset.")
```

Answer code

a) Set the wd

Step 1: Mount the Google Drive

```
from google.colab import drive
```

```
drive.mount('/content/drive')
```

#Step 2: load the operating system package

```
import os
```

#Step 3: changing the working directory to a specific folder on my Google Drive

```
os.chdir('/content/drive/MyDrive/Colab Notebooks/Data_BUS210AClass')
```

#Step 4: load the data from my working directory

b) download the given CSV file

c) Load the CSV file

```
sales_data_df = pd.read_csv('sales_data_quiz2.csv')
```

```
print(sales_data_df.head())
```

Question 6: Perform the following data handling tasks:

a) Set your working directory to a specific folder on your Google Drive.

b) Use the given URL (<https://tinyurl.com/yuzyb2nr>) to download a CSV file and save it to your working directory.

c) Load the CSV file into a Pandas DataFrame and display the first five rows of the dataset.

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remo

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	\
0	10107	30	95.70	2	2871.00	
1	10121	34	81.35	5	2765.90	
2	10134	41	94.74	2	3884.34	
3	10145	45	83.26	6	3746.70	
4	10159	49	100.00	14	5205.27	

	ORDERDATE	STATUS	QTR_ID	MONTH_ID	YEAR_ID	...	\
0	2/24/2003 0:00	Shipped	1	2	2003	...	
1	5/7/2003 0:00	Shipped	2	5	2003	...	
2	7/1/2003 0:00	Shipped	3	7	2003	...	
3	8/25/2003 0:00	Shipped	3	8	2003	...	
4	10/10/2003 0:00	Shipped	4	10	2003	...	

	ADDRESSLINE1	ADDRESSLINE2	CITY	STATE	\
0	897 Long Airport Avenue	NaN	NYC	NY	
1	59 rue de l'Abbaye	NaN	Reims	NaN	
2	27 rue du Colonel Pierre Avia	NaN	Paris	NaN	
3	78934 Hillside Dr.	NaN	Pasadena	CA	

```

4          7734 Strong St.          NaN  San Francisco    CA

   POSTALCODE COUNTRY TERRITORY CONTACTLASTNAME CONTACTFIRSTNAME DEALSIZE
0      10022    USA      NaN          Yu          Kwai    Small
1      51100  France    EMEA    Henriot          Paul    Small
2      75508  France    EMEA    Da Cunha    Daniel    Medium
3      90003    USA      NaN    Young      Julie    Medium
4        NaN    USA      NaN    Brown      Julie    Medium

```

[5 rows x 25 columns]

```

from google.colab import drive
drive.mount('/content/drive')

```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remo

```

# Question 7 (20 points)
print("Question 7a: Load the 'sales_data.csv' file (from class - should be in your working directory) into a pandas DataFr
print("Question 7b: Using the 'df' DataFrame, calculate the total sales for each product category.")
print("Question 7c: Print the results.")

```

```

# Answer code
data1 = pd.read_csv('sales_data.csv')
df = pd.DataFrame(data1)
total_sales_per_category = df.groupby('Product')['Sales'].sum().reset_index()
print("Question 7b: Total Sales for Each Product Category")
print(total_sales_per_category)

```

```

Question 7a: Load the 'sales_data.csv' file (from class - should be in your working directory) into a pandas DataFrame
Question 7b: Using the 'df' DataFrame, calculate the total sales for each product category.
Question 7c: Print the results.
Question 7b: Total Sales for Each Product Category
   Product  Sales
0  Widget A  164.0
1  Widget B   81.5
2  Widget C   75.0
3  Widget D  183.6

```

```

# Question 8 (20 points)
print("Question 8a: Write a Python function 'calculate_discount' that takes two arguments: 'price' and 'discount_rate' (in
print("Question 8b: Call your created function and pass it some arguments to show how it works.")

```

```

# Answer code
def calculate_discount(price, discount_rate):
    return price * (1 - discount_rate / 100)

# Test
print('The shirt has the original price at $100')
print('Applying 20% discount')
print('price after discount:', calculate_discount(100, 20))

```

```

Question 8a: Write a Python function 'calculate_discount' that takes two arguments: 'price' and 'discount_rate' (in pe
Question 8b: Call your created function and pass it some arguments to show how it works.
The shirt has the original price at $100
Applying 20% discount
price after discount: 80.0

```

```

# Question 9 (15 points)
print("Question 9: Write a Python function to calculate net profit. ")
print(".          You should subtract the total expenses from the total revenues and return the result.")
print(".          Call your function using the data below for the arguments.")
# Data to test your created function
revenues = [1000, 2000, 3000]

```

```
revenues = [1000, 2000, 3000]
expenses = [500, 600, 700]
```

```
# Answer code
```

```
def calculate_net_profit(revenues, expenses):
    total_revenues = np.sum(revenues)
    total_expenses = np.sum(expenses)
    net_profit = total_revenues - total_expenses
    return net_profit
```

```
# Call the function and print the result
```

```
print('Net profit:', calculate_net_profit(revenues, expenses))
```

Question 9: Write a Python function to calculate net profit.

. You should subtract the total expenses from the total revenues and return the result.

. Call your function using the data below for the arguments.

Net profit: 4200

```
# Getting the end time of your quiz in Eastern Time
```

```
end_time = datetime.datetime.now(eastern)
```

```
print(f"End time: {end_time}")
```

```
# Calculating the total time it took you to complete this quiz.
```

```
duration = end_time - start_time
```

```
print(f"Total duration: {duration}")
```



```
End time: 2024-04-18 13:49:25.455910-04:00
```

```
Total duration: 1:01:15.788241
```

```
print(" -----The END-----")
```

```
-----The END-----
```