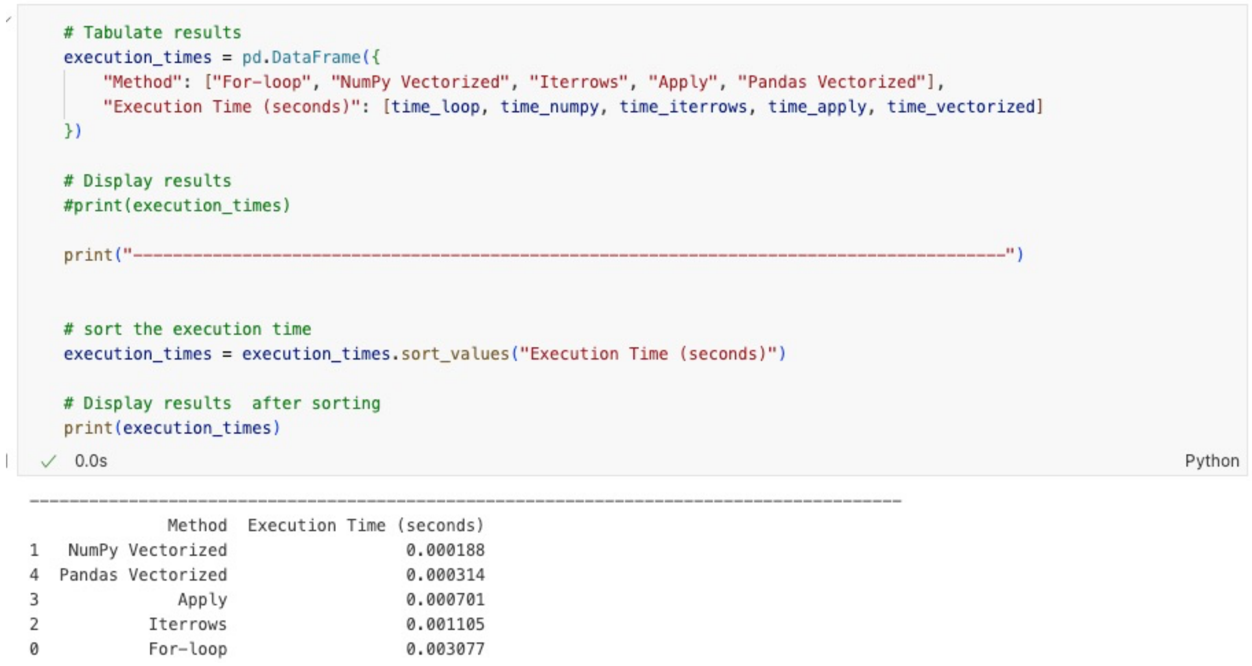
Wen-shan, Liu
Week3-Assignment

Week3_Assignment:

Source code: In GitHub: Week3-Assignment folder (Assignment.ipynb and Week3-Assignment.Rmd)

1.Execution time from the Python code.

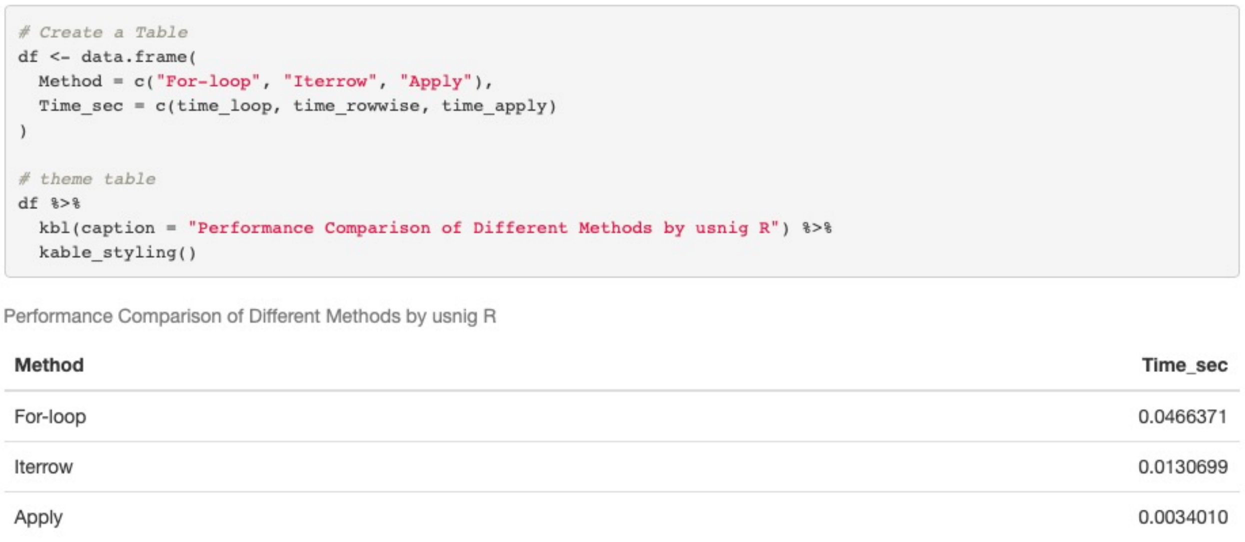


```python
# Tabulate results
execution_times = pd.DataFrame({
    "Method": ["For-loop", "NumPy Vectorized", "Iterrows", "Apply", "Pandas Vectorized"],
    "Execution Time (seconds)": [time_loop, time_numpy, time_iterrows, time_apply, time_vectorized]
})

# Display results
#print(execution_times)

print("-----------------------------------------------------------------------------------------")


# sort the execution time
execution_times = execution_times.sort_values("Execution Time (seconds)")

# Display results  after sorting
print(execution_times)
```

✓ 0.0s Python

```
------------------------------------------------------------------------------------------
              Method  Execution Time (seconds)
1   NumPy Vectorized                  0.000188
4  Pandas Vectorized                  0.000314
3              Apply                  0.000701
2           Iterrows                  0.001105
0           For-loop                  0.003077
```

2.Execution time from the R code.



```r
# Create a Table
df <- data.frame(
  Method = c("For-loop", "Iterrow", "Apply"),
  Time_sec = c(time_loop, time_rowwise, time_apply)
)

# theme table
df %>%
  kbl(caption = "Performance Comparison of Different Methods by usnig R") %>%
  kable_styling()
```

Performance Comparison of Different Methods by usnig R

Method	Time_sec
For-loop	0.0466371
Iterrow	0.0130699
Apply	0.0034010

3.Based on the computational efficiency of implementations in Python and R, which one would you prefer? Based on a consideration of implementation (i.e., designing and

implementing the code), which approach would you prefer? Taking both of these (run time and coding time), which approach would you prefer?

Ans: I prefer using Python.

In regard to computing efficiency, the three techniques ran in both Python and R, with all approaches in Python demonstrating superior speed compared to R. In the for-loop function, Python requires 0.003 seconds, but R necessitates 0.004 seconds. In the interrow approach, Python requires 0.0011 seconds, whereas R necessitates 0.0013 seconds. Finally, in the Apply function, Python requires 0.0007 seconds, but R necessitates 0.003 seconds. Furthermore, the Python code results indicate that the most efficient solution is NumPy Vectorized, which requires 0.0001 seconds. Pandas Vectorized is the next most efficient solution. Thus, based on this result, using vectorization could reduce execution times.

Furthermore, from an implementation perspective, I think Python is far simpler to develop. Python would benefit from an increased number of open-source packages. Furthermore, Python offers a greater variety of data types, which are applicable in many situations.

Based on these two reasons, I prefer using Python.

4. Identify and describe one or two other considerations, in addition to these two, in determining which of the two environments – Python or R – is preferable to you.

Beyond the two factors mentioned, Python seamlessly integrates with various tools and platforms, including databases, cloud services (e.g., Google Cloud Platform), and distributed computing frameworks like Apache Spark. This makes it an excellent choice for large-scale data processing and real-world applications.

Moreover, Python boasts a rich ecosystem of well-established machine learning libraries, such as scikit-learn, PyTorch, and TensorFlow, which are widely used in both industry and academia for predictive modeling, deep learning, and AI applications.

Another key advantage is Python's extensive community support. Its vast, active global community provides comprehensive documentation, tutorials, and open-source resources—beneficial for both beginners learning programming concepts and professionals working on big data analytics and AI-driven projects.

Therefore, for machine learning, deep learning, or large-scale data engineering, Python is the preferred choice.