Assignment #3 – Working With PDFs

To begin, open Jupyter Notebook through your preferred environment such as Google Colab or Anaconda. Once opened, create a new notebook. In Google Colab this is done by selecting **File** then **New Notebook**. Next, carefully write the code provided into each cell of the notebook as shown, including any comments.

Execute each cell by clicking the **Run or Play** button located on the left side of each cell. Should you have an error during execution, review the code within the cell. Pay close attention to potential typos, incorrect spacing, or misspellings, verifying each line against the provided code. Correct and rerun the cell.

Continue this process for all cells. Then save as a pdf by selecting **File** then **Print**.

Submit PDF to Canvas. Code and output must be clearly identified for full credit.

50 points

dataWrangling_hw3_workingWithPDFs

March 27, 2024

[ ]:
```
# install tabula python package
!pip install tabula.py
```

[ ]:
```
!pip install tabulate
```

[ ]:
```
# import the necessary libraries
from tabula import read_pdf
from tabulate import tabulate
```

[ ]:
```
import warnings

# ignore all warnings
warnings.filterwarnings("ignore")
```

[ ]:
```
# filename variable of the pdf file which needs to be uploaded into the folder/
 ↪environment
pdf_file = 'FoodList.pdf'

# extract data from page 1 of the pd file
page_number = 1

# returns the extracted tables as pandas dataframes
tables_df = read_pdf(pdf_file, pages=page_number)

# print the tables from page 1 of the pdf
print(tables_df)


# ignore any warnings
```

[ ]:
```
# use list comprehension to create a new list, loop through each dataframe,␣
 ↪drops any columns that contain NaN (missing) values
cleaned_tables = [table.dropna(axis='columns') for table in tables_df]

# loop through the table and print everything, should not have any NaN values
for idx, table in enumerate(cleaned_tables):
    print(f"Table {idx+1} after dropping NaN values:")
```

```
    print(table)
```

[ ]:
```
# extract data from page 1 of the pdf file
page_number = 3

# returns the extracted tables as pandas dataframes
tables_df = read_pdf(pdf_file, pages=page_number)

# print the tables from page 1 of the pdf
print(tables_df)
```

[ ]:
```
# use list comprehension to convert the dataframe into a JSON string
tables_json = [table.to_json() for table in tables_df]

# loop over each JSON string to print data from the table
for idx, table_json in enumerate(tables_json):
    print(f"Table {idx + 1}:")
    print(table_json)
    # add a space/newline between tables
    print()
```

[ ]:
```
# extract tables from all pages
tables = read_pdf(pdf_file, pages='all', multiple_tables=True)

# print the tables extracted from each page
print(tables)
```

[ ]:
```
# set flag to process information page by page, performance optimizer
stream_option = True

# extract contents from page 4
page_number = 4

# extract tables in a rectangular area defined by coordinates (top, left,␣
 ↪bottom, right)
area = (270, 13, 790, 900)

# extract from the specified area using the stream option
tables_df = read_pdf(pdf_file, pages=page_number, stream=stream_option,␣
 ↪area=area)

# loop over the table, print the information
for idx, table in enumerate(tables_df):
    print(f"Table {idx + 1}:")
    print(table)
```