Homework: Case Study 2

In the six exercises of this case study, we will find and plot the distribution of word frequencies for different translations of Hamlet. Perhaps the distribution of word frequencies of Hamlet depends on the translation -- let's find out!

For this case study, the functions `count_words_fast` and `word_stats` are defined as in the Case 2 Videos (Videos 3.2.1 through 3.2.6). The code for these functions, which you will need for the following exercises, is given here:

```
import os
import pandas as pd
import numpy as np
from collections import Counter
```

```
def count_words_fast(text):
    text = text.lower()
    skips = [".", ",", ";", ":", "'", '"',
"\n", "!", "?", "(", ")"]
    for ch in skips:
        text = text.replace(ch, "")
    word_counts = Counter(text.split(" "))
    return word_counts
```

```
def word_stats(word_counts):
    num_unique = len(word_counts)
    counts = word_counts.values()
    return (num_unique, counts)
```

Case Study 2 Homework: Exercises 1-4

Exercise 1

1/1 point (graded)

Note that `book_titles` is a nested dictionary, containing book titles within authors within languages, all of which are strings. These books are all stored online, and are accessed throughout this case study. In Exercise 1, we will first read in and store each translation of Hamlet.

Instructions

Read in the data as a pandas dataframe using `pd.read_csv`. Use the `index_col` argument to set the first column in the csv file as the index for the dataframe. The data can be found at this link within the courseware

External link

, and at this link when coming from outside the courseware.

Complete the following line of code to read in the data:

```
hamlets = ## Complete this line of code! ##
```

How many Hamlet translations are there?

Answer = [3]

Code = [

```
hamlets = pd.read_csv("https://courses.edx.org/asset-
v1:HarvardX+PH526x+2T2019+type@asset+block@hamlets.csv",
                      index_col=0)

hamlets
language    text
1     English     The Tragedie of Hamlet\n ...
2     GermanHamlet, Prinz von Dännemark.\n ...
3     Portuguese  HAMLET\n DRAMA EM ...
]
```

Exercise 2

0/1 point (graded)

In Exercise 2, we will summarize the text for a single translation of Hamlet in a `pandas` dataframe.

Instructions

Find the dictionary of word frequency in `text` by calling `count_words_fast()`. Store this as `counted_text`.

Create a `pandas` dataframe named `data`.

Using `counted_text`, define two columns in `data`:

- `word`, consisting of each unique word in `text`.
- `count`, consisting of the number of times each word in `word` is included in the text.

Here's the code to get you started:

```
language, text = hamlets.iloc[0]

# Enter your code here.
```

How many times does the word Hamlet appear in the text?

Answer = [97]

Code = [

```
counted_text = count_words_fast(text)

data = pd.DataFrame({
    "word": list(counted_text.keys()),
    "count": list(counted_text.values())
})

data.head(10)
```

]

Exercise 3

1/1 point (graded)

In Exercise 3, we will continue to define summary statistics for a single translation of Hamlet.

Instructions

Add a column to `data` named `length`, defined as the length of each word.

Add another column named `frequency`, which is defined as follows for each word in `data`:

- If `count > 10`, `frequency` is "frequent".
- If `1 < count <= 10`, `frequency` is "infrequent".
- If `count == 1`, `frequency` is "unique".

How many unique words appear in the text?

Answer = [3348]

Code = [

```
    data["length"] = data["word"].apply(len)

data.loc[data["count"] > 10,  "frequency"] = "frequent"
data.loc[data["count"] <= 10, "frequency"] = "infrequent"
data.loc[data["count"] == 1,  "frequency"] = "unique"

data.groupby('frequency').count()
```

]

Exercise 4

1/1 point (graded)

In Exercise 4, we will summarize the statistics in `data` into a smaller `pandas` dataframe.

Instructions

Create a `pandas` dataframe named `sub_data` including the following columns:

- `language`, which is the language of the text (defined in Exercise 2).
- `frequency`, which is a list containing the strings "frequent", "infrequent", and "unique".
- `mean_word_length`, which is the mean word length of each value in `frequency`.
- `num_words`, which is the total number of words in each frequency category.

What is the average word length of the infrequent words?

Answer = [5.825243]

Code = [

```
 sub_data = pd.DataFrame({
    "language": language,
    "frequency": ["frequent","infrequent","unique"],
    "mean_word_length": data.groupby(by = "frequency")["length"].mean(),
    "num_words": data.groupby(by = "frequency").size()
})

sub_data
      language     frequency    mean_word_length   num_words
frequency
frequent     English      frequent     4.371517     323
infrequent   English      infrequent   5.825243     1442
unique       English      unique7.005675     3348
```

]

Case Study 2 Homework: Exercises 5-6

Exercise 5

2/2 points (graded)

In Exercise 5, we will join all the data summaries for text Hamlet translation.

Instructions

The previous code for summarizing a particular translation of Hamlet is consolidated into a single function called `summarize_text`. Create a `pandas` dataframe `grouped_data` consisting of the results of `summarize_text` for each translation of Hamlet in `hamlets`.

- Use a `for` loop across the row indices of `hamlets` to assign each translation to a new row.
- Obtain the `ith` row of `hamlets` to variables using the `.iloc` method, and assign the output to variables `language` and `text`.
- Call `summarize_text` using `language` and `text`, and assign the output to `sub_data`.
- Use the pandas `.append()` function to append pandas dataframes row-wise to `grouped_data`.

The code below defines `summarize_text`:

```
def summarize_text(language, text):
    counted_text = count_words_fast(text)

    data = pd.DataFrame({
        "word": list(counted_text.keys()),
        "count": list(counted_text.values())
    })

    data.loc[data["count"] > 10,  "frequency"] =
"frequent"
    data.loc[data["count"] <= 10, "frequency"] =
"infrequent"
    data.loc[data["count"] == 1,  "frequency"] =
"unique"

    data["length"] = data["word"].apply(len)

    sub_data = pd.DataFrame({
```

```
        "language": language,
        "frequency":
["frequent","infrequent","unique"],
        "mean_word_length": data.groupby(by =
"frequency")["length"].mean(),
        "num_words": data.groupby(by =
"frequency").size()
    })

    return(sub_data)

# write your code here!
```

What is the average word length of the frequent words in the German translation?

Answer = [4.528053]

How many frequent words are there in the Portugese translation?

Answer = [261]

Code = [

```
grouped_data = pd.DataFrame(columns = ["language", "frequency",
"mean_word_length", "num_words"])

for i in range(hamlets.shape[0]):
    language, text = hamlets.iloc[i]
    sub_data = summarize_text(language, text)
    grouped_data = grouped_data.append(sub_data)

grouped_data
      language    frequency   mean_word_length   num_words
frequent    English     frequent    4.371517    323
infrequent  English     infrequent  5.825243    1442
unique      English     unique7.005675    3348
frequent    Germanfrequent    4.528053    303
infrequent  Germaninfrequent  6.481830    1596
unique      Germanunique9.006987    5582
```

```
frequent     Portuguese   frequent     4.417625     261
infrequent   Portuguese   infrequent   6.497870     1643
unique       Portuguese   unique8.669778     5357
```

]

Exercise 6

1/1 point (graded)

In Exercise 6, we will plot our results and look for differences across each translation.

Instructions

Plot the word statistics of each translation on a single plot. Note that we have already done most of the work for you. Consider whether the word statistics differ by translation.

This code will do most of the plotting work:

```
colors = {"Portuguese": "green", "English": "blue",
"German": "red"}
markers = {"frequent": "o","infrequent": "s", "unique":
"^"}
import matplotlib.pyplot as plt
for i in range(grouped_data.shape[0]):
    row = grouped_data.iloc[i]
    plt.plot(row.mean_word_length, row.num_words,
        marker=markers[row.frequency],
        color = colors[row.language],
        markersize = 10
    )

color_legend = []
```

```
marker_legend = []
for color in colors:
    color_legend.append(
        plt.plot([], [],
        color=colors[color],
        marker="o",
        label = color, markersize = 10,
linestyle="None")
    )
for marker in markers:
    marker_legend.append(
        plt.plot([], [],
        color="k",
        marker=markers[marker],
        label = marker, markersize = 10,
linestyle="None")
    )
plt.legend(numpoints=1, loc = "upper left")

plt.xlabel("Mean Word Length")
plt.ylabel("Number of Words")
# write your code to display the plot here!
```

For which word category do the statistics differ most by translation?

frequent

infrequent

unique
correct
Explanation

You just need to add this line of code to display the plot: `plt.show();`. Looking at the plot, unique words have the largest difference in statistics by translation - unique English are shorter than either unique Portuguese or unique German words, and there are also fewer unique English words than either unique German or unique Portuguese words.