Exercise 4 - Statistics and tests

First name: Brian

Last name: Schweigler

Matriculation number: 16-102-071

Q1: Consider the sentence “Ann plays the role with Mary and Annie”. Use RegExp to replace “Ann” by “Alice” and obtain “Alice plays the role with Mary and Annie”.

General imports and solving the question:

In [1]:
```
%load_ext autoreload
%autoreload 2
%matplotlib inline

import matplotlib.pyplot as plt
import re
import numpy as np
import lxml.etree
import os
from scipy import stats

s1 = "Ann plays the role with Mary and Annie"
replaced1 = re.sub("Ann ", "Alice ", s1)
print(replaced1)
```

```
Alice plays the role with Mary and Annie
```

Q2: Consider the sentence "$99.99to$87.80 or Fr. 75.50". Use RegExp to remove the decimal part of prices and obtain "$99to$87 or Fr. 75".

In [2]:
```
s2 = "$99.99 to $87.80 or Fr. 75.50"
replaced2 = re.sub('(\d+)\.\d*', r'\1', s2)
print(replaced2)
```

```
$99 to $87 or Fr. 75
```

Q3: For the genre “Comédie”, extract (in a list) the number of word-tokens per play. Do you obtain the same mean as the one indicated in the lecture slides (namely 9934.91 for 310 plays in this category)?

Setup and solving the task

In [3]:
```
plays = {}
counter = 0
for xml_file in os.scandir('theatre-classique'):
    tree = lxml.etree.parse(xml_file.path)
    genre = tree.find('//genre')
    lines = []
    if genre is not None and genre.text == 'Comédie':
        for line in tree.xpath('//l|//p'):
            lines.append(' '.join(line.itertext()))
            text = ' '.join(lines)
        plays[counter] = [a.lower() for a in re.split(r'\W+', text)]
        counter += 1

all_plays = [len(words) for words in plays.values()]
distinct_plays = [len(set(words)) for words in plays.values()]
print("Mean of 'Comédie' work-tokens: ", np.mean(all_plays))
```

```
Mean of 'Comédie' work-tokens:  10042.429032258064
```

Thus, we are near the result, but not quite. It could be that there are some encoding issues that led to some malformed words.

Q4: Apply the t-test to verify the hypothesis that, in mean, a French comedy contains 10,000 word-tokens.

In [4]:
```
t_test_comedie_10000 = stats.ttest_1samp(all_plays, 10000)
print("p-Value for French comedy containing 10'000 words:", t_test_comedie_10000.pvalue)
```

```
p-Value for French comedy containing 10'000 words: 0.8874058267073174
```

The p-value is high, giving a likelihood of about 88% that our data fits the hypothesis of having 10'000 words on average. To reject the null-hypothesis, we would normally go with a very low p-value of 0.05 or smaller.

Q5: Apply the t-test to verify the hypothesis that, in mean, a French tragedy contains 14,000 word-tokens.

This was done in exercise 2 (the last series), thus I assume this is a mistake as we only have one author here.

In [5]:
```
tragedies = {}
counter = 0
for xml_file in os.scandir('theatre-classique'):
    tree = lxml.etree.parse(xml_file.path)
    genre = tree.find('//genre')
    lines = []
    if genre is not None and genre.text == 'Tragédie':
        for line in tree.xpath('//l|//p'):
            lines.append(' '.join(line.itertext()))
            text = ' '.join(lines)
        tragedies[counter] = [a.lower() for a in re.split(r'\W+', text)]
        counter += 1

all_tragedies = [len(words) for words in tragedies.values()]
distinct_plays = [len(set(words)) for words in tragedies.values()]
print("Mean of 'Tragédie' work-tokens: ", np.mean(all_tragedies))

t_test_tragedies_14000 = stats.ttest_1samp(all_tragedies, 14000)
print("p-Value for French tragedies containing 14'000 words:", t_test_tragedies_14000.pvalue)
```

```
Mean of 'Tragédie' work-tokens:  14326.026666666667
p-Value for French tragedies containing 14'000 words: 0.24197018702608467
```

Here we have a lower p-value, the null-hypothesis is unlikelier than before, but normally we would only accept the alternative hypothesis at a significance level of 0.05 or lower.

Q6: Apply the t-test to verify the hypothesis that, in mean, a French tragedy contains 15,000 word-tokens.)

In [6]:
```
t_test_tragedies_15000 = stats.ttest_1samp(all_tragedies, 15000)
print("p-Value for French tragedies containing 15'000 words:", t_test_tragedies_15000.pvalue)
```

```
p-Value for French tragedies containing 15'000 words: 0.016353531252481648
```

Here we have a p-value below 0.05, thus we accept the alternative hypothesis that the mean is unlikely to be 15'000 for french tragedies, based on this data.

Q7: Describe the results you obtained from the previous tests.

This was already done at Q4, Q5, and Q6.

Namely, in a normal case, we would stick with the null-hypothesis in Q4 and Q5, but in Q6, with such a low p-value, we would accept the alternative hypothesis: "The mean words of tragedies within French plays is unlikely to be 15'000".

Q8: Apply the chi^2 test to verify the hypothesis that the distribution of the education level in the New England region is similar to the one appearing in the Pacific area.

In [7]:
```
import pandas as pd
import gzip
```

```
import numpy as np
from scipy import stats

with gzip.open('data/GSS7214_R5.DTA.gz', 'rb') as infile:
    # we restrict this (very large) dataset to the variables of interest
    columns = ['id', 'year', 'age', 'sex', 'race', 'reg16', 'degree',
               'realrinc', 'readfict']
    df = pd.read_stata(infile, columns=columns)

regions_oi = sorted(['new england', 'pacific'])
df_regions = df.loc[df['reg16'].isin(regions_oi)].copy()
df_regions['reg16'] = df_regions['reg16'].cat.remove_unused_categories()
df_regions.groupby('reg16')['degree'].value_counts(normalize=True).round(1).to_frame()
df_regions.groupby('reg16')['degree'].value_counts().to_frame()

subjects = pd.DataFrame(
    [
        # Original:
        # [134, 120, 152, 318],
        # [80, 72, 69, 112],
        # [63, 34, 20, 63],
        # [57, 19, 19, 48],
        # [32, 17, 16, 42],

        # Then removing the first and third column, as those were "foreign" and "mountain"
        [120, 318],
        [72, 112],
        [34, 63],
        [19, 48],
        [17, 42],
    ],
    index=['high school', 'bachelor', 'graduate', 'lt high school', 'junior college'],
    columns=['new england', 'pacific'])

chi, pval, dof, exp = stats.chi2_contingency(subjects)
print('p-value is: ', pval)

significance = 0.05
p = 1 - significance
critical_value = stats.chi2.ppf(p, dof)
print('chi=%.6f, critical value=%.6f\n' % (chi, critical_value))
```

```
p-value is:  0.05103567823078395
chi=9.438059, critical value=9.487729
```

If the Chi-square value is greater than or equal to the critical value: There is a significant difference between the groups, probably too great to be attributed to chance.

If the Chi-square value is less than the critical value: There is no significant difference, it is likely due to chance.

Thus, as we have a value that is just barely less than the critical value, we can assume that the difference in distribution of the education level is due to chance.