STAT2110 Engineering Statistics

Semester 1, 2022

Python Week 4: Sampling and emperical probabilities

In [ ]:
```
!python --version
```

```
Python 3.8.8
```

In [ ]:
```
import numpy as np
import scipy as sp
import pandas as pd
import matplotlib as plt
print(np.__name__, np.__version__)
print(sp.__name__, sp.__version__)
print(pd.__name__, pd.__version__)
print(plt.__name__, plt.__version__)
```

```
numpy 1.21.4
scipy 1.7.2
pandas 1.3.4
matplotlib 3.5.0
```

Note: You may need to install **pandas** package. On **PyCharm** to install packages complete the following:

> Click on View, select "Tool Window" and click on "Python Packages"
>
> Search "pandas" in the searchbar in the window which opens below. Then click on install (Left hand top corner of Python Packages Window).
>
> See Week 1 Workshop for more information

A standard deck of cards consists of 52 cards, 13 cards for each of 4 suits: hearts, diamonds, spades, and clubs. In this exercise, we will compare probabilities when drawing 5 cards from the deck, both with replacement (i.e. after each draw, we replace the drawn card in the deck before drawing again) and without replacement (i.e. after each draw, we do not replace the drawn card before drawing again).

Let X be the number of clubs in our hand when we draw 5 cards with replacement, and let Y be the number of clubs in our hand when we draw 5 cards without replacement.

Step 1: Define a vector to contain the card suit strings.

First, let's generate a string called `cards`, from which we will simulate our card draws. Note the use of the `repeat()` function from the **NumPy** package will be required to repeat the elements in the suits vector 13 times:

Firstly, make sure to import the **NumPy** Package as above

In [ ]:
```
suits=["Hearts", "Diamonds", "Spades", "Clubs"] #4 suits
cards=np.repeat(suits, 13)  #generate card deck of 52 cards
```

We will now simulate $100,000$ different draws of 5 cards, both with and without replacement.

Step 2: Set up the simulation.

Next, let's initialize vectors X and Y to contain the counts of the number of clubs drawn for each of the simulations:

In [ ]:
```
n_sim=10000
X=[] #Variable for Number of clubs with replacement
Y=[] #Variable for Number of clubs without replacement
```

To sample cards, a useful function is the `random.choice()` function from the **NumPy** Package, which allows you to sample a specified number of elements from a supplied vector, both with and without replacement. For example, let's define a vector containing the states and territories of Australia:

In [ ]:
```
states=["ACT", "NSW", "NT", "QLD", "SA", "TAS", "VIC", "WA"]
```

Now, let's draw 4 elements from this vector, both with and without replacement, with the `np.random.choice()` function, and use the `sum()` function to count the number of times "QLD" is in the sample:

In [ ]:
```
draw4replace=np.random.choice(states,4,replace=True)  #With replacement
draw4replace
```

Out[ ]:
```
array(['NT', 'VIC', 'SA', 'SA'], dtype='<U3')
```

In [ ]:
```
draw4noreplace=np.random.choice(states,4,replace=False)  #Without replacement
draw4noreplace
```

Out[ ]:
```
array(['WA', 'TAS', 'QLD', 'ACT'], dtype='<U3')
```

In [ ]:
```
sum(draw4replace=="QLD")
```

Out[ ]:
```
0
```

In [ ]:
```
sum(draw4noreplace=="QLD")
```

Out[ ]:
```
1
```

Step 3: Use a for loop to simulate card draws and record the counts.

Now, you should write a for loop to simulate drawing 5 cards, with and without replacement, and store the resulting counts of the number of clubs drawn each simulation in the vectors X and Y, respectively. Use the `np.random.choice()` and `sum()` functions, as well as `append()` to add new values to X and Y i.e.:

`a.append(1)` add value 1 to a

Step 4: Generate frequency tables.

Having generated X and Y, we can summarise the empirical counts of the number of clubs drawn in each simulation via the code below. This will require **pandas** packages to be loaded. To install See steps at start or Week 1 Workshop.

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

Use the following code to convert X and Y into Dataframe and create pandas dataframes from *TabX* and *TabY* (Save empirical counts):

In [ ]:
```
X1=pd.DataFrame(X,columns=list("X"))    #Data frame format
Y1=pd.DataFrame(Y,columns=list("Y"))    #Data frame format
TabX=pd.DataFrame(columns=['X', 'Count', 'Percent'])    #Data frame to save empirical c
TabY=pd.DataFrame(columns=['Y', 'Count', 'Percent'])    #Data frame to save empirical c
```

Following will compute counts and percentages for frequency of counts of clubs from 5 cards for Replacement (X) and without Replacement (Y)

In [ ]:
```
TabX['Count']=X1['X'].value_counts().sort_index() #Save counts in order for replacemen
TabX['Percent']=X1['X'].value_counts().sort_index()/n_sim #Save Percent in order for r
TabX['X'] = TabX.index    #Number of clubs
TabX
```

In [ ]:
```
TabY['Count']=Y1['Y'].value_counts().sort_index()    #Save Counts for Y in order for nc
TabY['Percent']=Y1['Y'].value_counts().sort_index()/n_sim  #Save Percent in order for
TabY['Y'] = TabX.index    #Number of clubs
TabY
```

The resulting tables contain 3 columns: the first column contains the values of the X and Y variables ($0-5$). The second column contains the empirical counts, and the third column contains the corresponding percentages (out of 100).

Step 5: Graph the empirical probability mass functions.

We can graph the empirical probability mass functions using the `stem()` function from the **matplotlib** package. Instead of creating two separate graphs, we will combine them onto a single graph. We do this by off-setting the x-axis values corresponding to the number of clubs drawn by ± 0.05 so that the masses appear next to each other, for with and without replacement, at each number of clubs.

This will require **matplotlib** packages to be loaded, which is preinstalled on PyCharm.

In [ ]:
```
import matplotlib.pyplot as plt
```

The following code will produce the combined plot:

In [ ]:
```
offset=0.05
plt.stem(TabX['X']-offset,TabX['Percent'], markerfmt='bo', label='With Replacement')
plt.stem(TabY['Y']+offset,TabY['Percent'], markerfmt='go', label='Without Replacement'
plt.legend()
plt.xlabel("Number of clubs drawn")
plt.ylabel("Empirical probability")
plt.title("Empirical Probability Mass Function")
plt.show()
```

Exercise 1

Having generated the empirical probability mass functions, answer the following questions.

a) - Is it more likely that we would draw 1 club in a set of 5 cards with or without replacement?

b) - Is it more likely that we would draw 2 clubs in a set of 5 cards with or without replacement?

c) - Is it more likely that we would draw 3 clubs in a set of 5 cards with or without replacement?

d) - Using concepts from **Week 2**, verify by hand that $P(Y=1)=0.411$, $P(Y=2)=0.274$, and $P(Y=3)=0.082$. Do your empirical probabilities for variable Y resemble these theoretical probabilities?