590D Mini Exercise - Reservoir Sampling

Anirudha Desai, Suhas Keshavmurthy

September 12, 2017

1 Code

```
import random
import numpy as np
import time
'''function to install any packages that may be needed'''
def install_and_import(package):
    import importlib
    try:
        importlib.import_module(package)
    except ImportError:
        import pip
        pip.main(['install', package])
        importlib.invalidate_caches()
    finally:
        globals()[package] = importlib.import_module(package)


def RsSamplingOne(iterations):
    #size of input is 100
    size = 100
    #array of size 100 with default value 0
    count = [0] * size

    #run the reservoir sampling for the iterations provided as input to the function
    for j in range(1, iterations + 1):
        #the first item is selected first time as reservoir is empty
        res = 1

        for i in range(2, size+1):
            '''choose the ith item with a probability of 1\i'''
            if random.randint(1,i) == i:
                res = i
        #maintain count of the item in the reservoir at the end
        count[res - 1] += 1
    #return the counts of item selected in the reservoir and the last item in the reservoir
    return count, res


def plotData(xArray, yArray, numIterations):

    '''calculate mean, standard deviation and
    coefficient of variation of input dataset'''

    yArray_np = np.asarray(yArray)
    m = np.mean(yArray_np)
    deviation = np.std(yArray_np)
    cv = deviation/m
    cv_formatted = float("{0:.3f}".format(cv))

    #use bokeh library to obtain the plot
    plotTitle = str(numIterations) \
                + ' Iterations;\nCoefficient of Variation = ' + str(cv_formatted)
    p = figure(title=plotTitle,
               x_axis_label='item',
               y_axis_label='Number of times item is selected',
                width=500, height=300)
    p.circle(x=xArray, y=yArray, size=6, color="firebrick",
        alpha=0.8, legend='Selection Count')
    p.line(x=xArray, y=yArray, line_width=2)
    p.y_range.start = 0
    p.title.align = 'center'
    p.legend.location = "bottom_right"

    return p

def main():

    #item sampled in one run of the algorithm
    __,result = RsSamplingOne(1)
    print('Item sampled in one iteration')
    print(result)

    '''Repeat the algorithm for 1000 times and plot the
    number of times each element is selected'''
    count1000,__ = RsSamplingOne(1000)
    print('Count of items sampled in 1000 iterations')
    print(count1000)
    p1 = plotData(range(1,101), count1000, 1000)

    '''Repeat the algorithm for 10000 times and plot the
    number of times each element is selected'''
    count10000,__ = RsSamplingOne(10000)
    print('Count of items sampled in 10000 iterations')
    print(count10000)
    p2 = plotData(range(1, 101), count10000, 10000)

    '''Repeat the algorithm for 100000 times and plot the
    number of times each element is selected'''
    count100000,__ = RsSamplingOne(100000)
    print('Count of items sampled in 100000 iterations')
    print(count100000)
    p3 = plotData(range(1, 101), count100000, 100000)

    # make a grid
    grid = gridplot([[p1, p2], [p3, None]])

    # show the results
    show(grid)

if __name__ == "__main__":
    install_and_import('bokeh')

    from bokeh.plotting import figure
    from bokeh.layouts import row, gridplot
    from bokeh.io import output_notebook, show

    main()
```

2 Results

1. The item sampled in one run of the algorithm :
2. Plot with algorithm run 1000 times

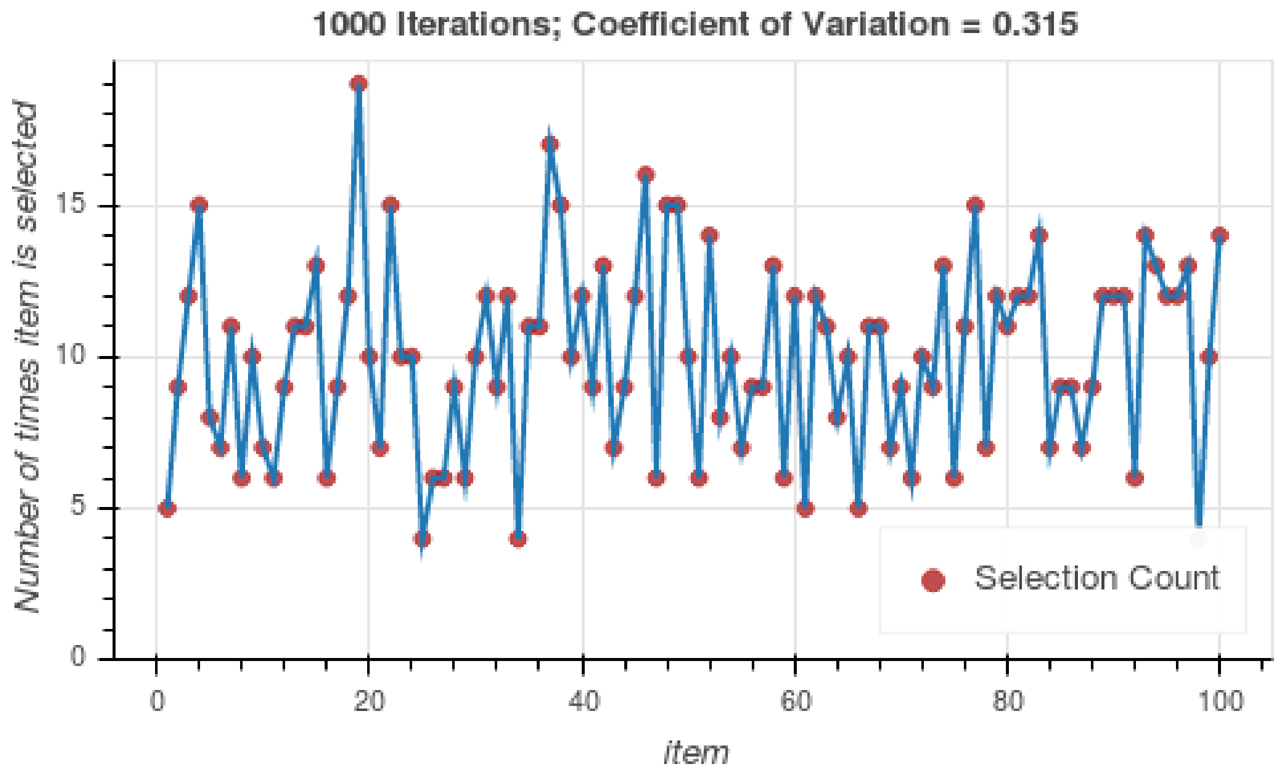

Figure 1: Plot with 1000 runs

3. Plot with algorithm run 10000 times
4. Plot with algorithm run 100000 times

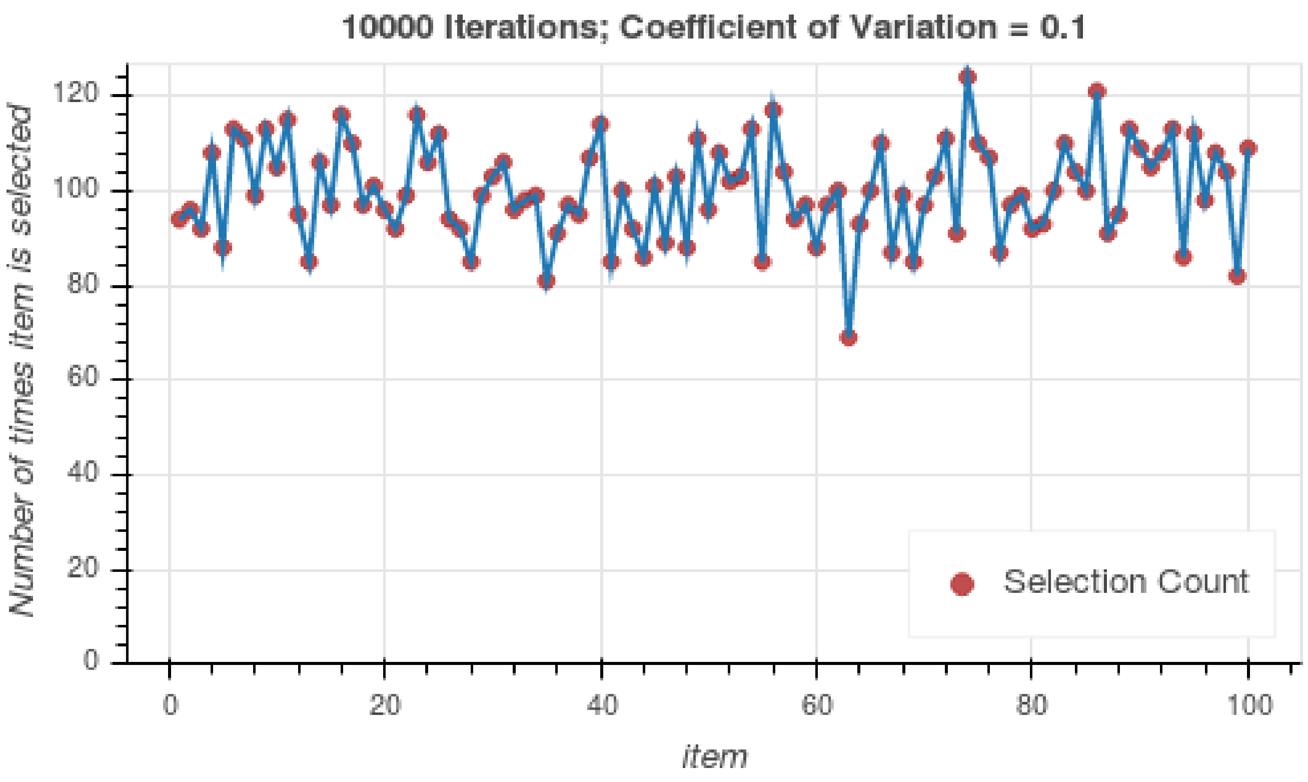

Figure 2: Plot with 10000 runs

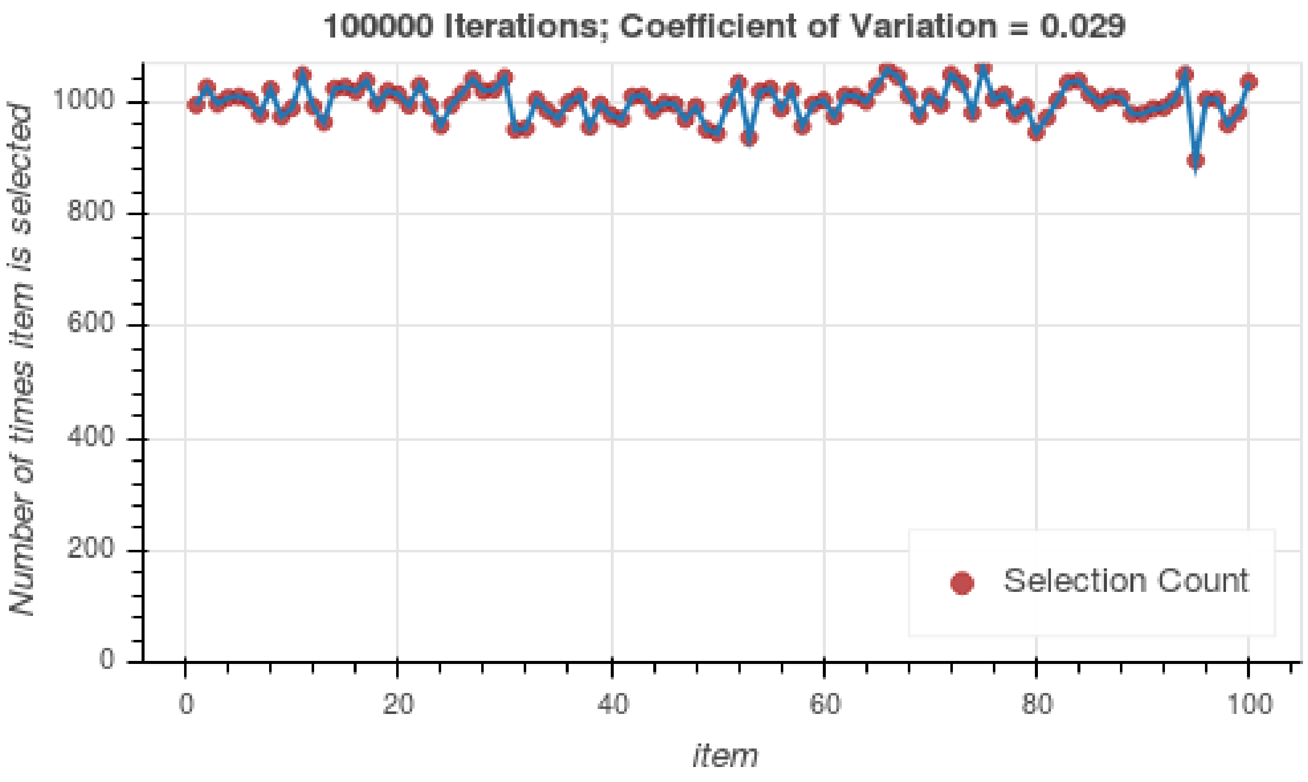

Figure 3: Plot with 100000 runs