COMP 312 Assignment 3

Daniel Braithwaite

March 21, 2016

1 Python

1.1 Ellipse Problem

```
import math

class Ellipse(object):
        """The Ellipse class"""

        def __init__(self, a, b):
                self.a = a;
                self.b = b;

        def area(self):
                return math.pi * self.a * self.b

        def eccentricity(self):
                return math.sqrt(1 - math.pow(self.b/self.a, 2))


if __name__ == "__main__":
        ellipse = Ellipse(10, 5)

        print "Area: " + str(ellipse.area())
        print "Eccentricity: " + str(ellipse.eccentricity())
```

This program outputs

$$Area:\ 157.079632679 \tag{1}$$
$$Eccentricity:\ 1.0 \tag{2}$$

1.2 Chi Square Problem

1.2.1 Part B Solution

```
import random
import math
import numpy

def chisquarevariate(k):
        """Generates a variate from chi square distrubuton with paramater k"""

        s = 0
        for i in range(k):
                s += math.pow(random.normalvariate(0, 1), 2)
```

```
        return s


def estimate(k, n):
        """Estimate expected value and variance over n runs"""
        variates = []
        for i in range(n):
                variates.append(chisquarevariate(k))

        return (numpy.mean(variates), numpy.var(variates))

if __name__ == "__main__":
        k = 9
        n = 10000

        random.seed(123)

        for i in range(1, k):
                res = estimate(i, n)
                print "K = ", i, ": ", res
```

The output for this section is the following table

k	$E(X)$	$var(X)$
1	1.0310970666411978	2.1809895852678212
2	2.0196390351470401	4.0906849280367874
3	3.0104658433374825	6.0929914117158352
4	3.9938676109902742	8.0085366007510803
5	4.9926872174841925	10.172262735835909
6	5.9559869106853442	11.465470122520625
7	6.9672714050261986	13.834253416945165
8	8.0347194125058135	16.044484561872114

1.2.2 Part C Solution

```
import random
import math
import numpy

def chisquarevariate(k):
        """Generates a variate from chi square distrubuton with paramater k"""

        s = 0
        for i in range(k):
                s += math.pow(random.normalvariate(0, 1), 2)
```

```
        return s


def estimate(k, n):
        """Estimate expected value and variance over n runs"""
        variates = []
        for i in range(n):
                variates.append(chisquarevariate(k))

        return (numpy.mean(variates), numpy.var(variates))


def conf(L):
        """Compute a 95 percent confidence interval"""
        lower = numpy.mean(L) - 1.96 * numpy.std(L)/math.sqrt(len(L))
        upper = numpy.mean(L) + 1.96 * numpy.std(L)/math.sqrt(len(L))

        return (lower, upper)

if __name__ == "__main__":
        k = 9
        n = 10000
        m = 50

        random.seed(123)

        for i in range(1, k):

                Lvar = []
                Lmean = []
                for j in range(m):
                        res = estimate(i, n)

                        Lmean.append(res[0])
                        Lvar.append(res[1])

                confMean = conf(Lmean)
                confVar = conf(Lvar)

                print "K = ", i
                print "  > Conf Mean: ", confMean
                print "  > Conf Var: ", confVar
                print "\n"
```

The output from this section for k = 8 is
K = 8
Conf Mean: (7.998448633149601, 8.0155757558836989)
Conf Var: (15.966449581811148, 16.135066786590585)

1.2.3 Part D Solution

The outputs that are given from the program make sense as we know for the chi square distribution

$$E(X) \leftarrow k \tag{3}$$
$$var(X) \leftarrow 2k \tag{4}$$

And we can see that the output is approximately equal to this.