Meysam Hamel Assignment2

Q1)

a.

$E[\bar{x}]$ = 5 by CLT

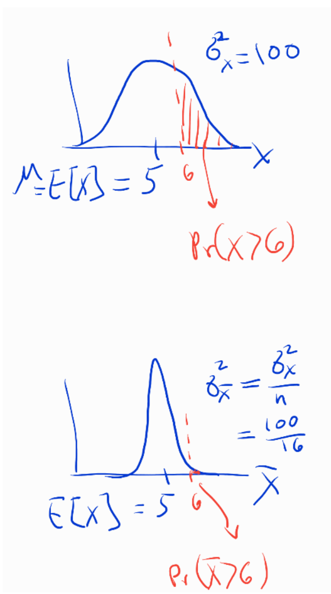
b.

var$[\bar{x}]$ = $\sigma^2/16$ = 100 /16 = 6.25 by CLT

c.

σ = sqrt 100/16 = 10 /4 = 2.5 by CLT

d.



Pr[X>6] > Pr[$\bar{x}$ > 6]

e.

Pr[X > 3] < Pr [$\bar{x}$ > 3]

Pr(x-5 / 10 > 3 – 5/10 = Pr(z > -0.2) = 1 – Pr(z0.2) = 0.5793

Pr ($\bar{x}$ -5 /2.5 > 3-5 /2.5) = 0.78

f.

upper band Pr[$\bar{x}$ > 8] ➔ Pr[$\bar{x}$ > 8] < = E[$\bar{x}$] / 8 = 5/8 = 0.625

g.

Chebyshev Pr[$\bar{x}$ > 8] ➔ ε = 8 – E[x] ➔ pr [|$\bar{x}$ – 5| >3] < = 100/16.9 = 0.694

h.

Meysam Hamel Assignment2
Chernoff-Hoeffding

Pr[$\bar{x}$ > 8]

Pr[$|\bar{x} - 5| > 3$] < = 2 exp(-2.9 – 16 /20^2) = 2exp(-18/23) = 0.9735

i.

chrrnoff → 0.022

cheby→ 0.111

markov → 0.625

markov remains unchanged since the calculation is independent from var[x]and n. σ^2/n. ε^2 for Chebyshev get smaller as n increase and band become tighter.

2/ $e^{(2\cdot \varepsilon.n/delta^2)}$ for chernoff getssmaller as n increase and bond vbecome tighter.

Q2)

```
In [5]: import numpy
        from numpy import matrix
        from numpy import linalg
        v = numpy.array([1, 2, 5, 2,-3, 1, 2, 6,2])
        u = numpy.array([-4,3,-2,2,1,-3,4,1,-2])
        w = numpy.array([3,3,-3,-1,6,-1,2,-5,-7])
        print "Dot product v,w"
        print numpy.dot(v, w)
        print numpy.dot(v, u)
        print "Dot product u,w"
        print numpy.dot(u, w)
        print "u norm 2"
        print linalg.norm(u, 2)
        print "w norm inf"
        print linalg.norm(w, numpy.inf)
```

```
Dot product v,w
-67
0
Dot product u,w
27
u norm 2
8.0
w norm inf
7.0
```

A) <v,w> = -67

B) Orthogonal vector

<v,u> = 0

C) ||u||2 = 8

D) ||w||inf = 7

Q3)

In [6]:
```
from numpy import matrix
from numpy import linalg
A = matrix ( [[2,-2],[-3,1],[5,-3]] )
B = matrix ( [[4,4,4], [-2,3,-7], [2,5,-7]] )
C = matrix ( [[4,-1,2], [-8,2,-4], [2,1,-4]] )
print "AT.B"
print A.T*B
print "C+B"
print C+B
print "A rank"
print linalg.matrix_rank(A)
print "B rank"
print linalg.matrix_rank(B)
print "C rank"
print linalg.matrix_rank(C)
print "C norm f"
print linalg.norm(C, 'fro')
print "A norm 2"
print linalg.norm(A, 2)
print linalg.inv(B)
```

```
AT.B
[[ 24  24  -6]
 [-16 -20   6]]
C+B
[[  8   3   6]
 [-10   5 -11]
 [  4   6 -11]]
A rank
2
B rank
3
C rank
2
C norm f
11.2249721603
A norm 2
7.14562300799
[[-0.11666667 -0.4         0.33333333]
 [ 0.23333333  0.3        -0.16666667]
 [ 0.13333333  0.1        -0.16666667]]
```

a. **A^TB**

```
AT.B
[[ 24  24  -6]
[-16 -20   6]]
```

b. **C + B**

```
[[  8   3   6]
 [-10   5 -11]
[  4   6 -11]]
```

c. **Which matrices are full rank? ➔ A and B**

```
A rank
  2
B rank
  3
C rank
  2
```

A and B

d. $||C||_F$

11.2249721603

e. $||A||_2$

7.14562300799

f. B^{-1}

```
[[-0.11666667 -0.4          0.33333333]
 [ 0.23333333  0.3         -0.16666667]
 [ 0.13333333  0.1         -0.16666667]]
```