**­­­**

**Statistical Methods for Data Science (Spring 2017)**

Mini Project 1

Contributing members :

**Akhilesh Kumar Kagalvadi Chinnaswamy (axk167131)**

**Vidya Sri Mani (vxm163230)**

**Contribution of Group Members:**

We individually worked on both problems and later discussed our approaches. After reaching a consensus on our solution, we compiled the report together.

Note:

* Every question consists of problem solution and corresponding R -code if needed with output is given as screenshot as described in Instruction as section1 and section 2.

* Finally, Appendix is added at last, which includes the R code for all questions.

**1. Suppose a random variable X has the following probability density function: f(x)  
equals 4\*x^3 when x is between 0 and 1, and equals 0 otherwise.  
(a) Compute E(X), Var(X) and P(X > 0.5) analytically, i.e., using their formulas.**

**Section1: Normal.**

, else 0.

= **0.02667.**

= 1- **0.9375**

**Section 2: R- code:**

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

**(b) Explain how you would simulate a draw from the distribution of X.**

**Section 1:**

Given: Probability density function(pdf)

Cumulative density function(cdf)

Let uniform random variable, U = F(x) , , u) =>

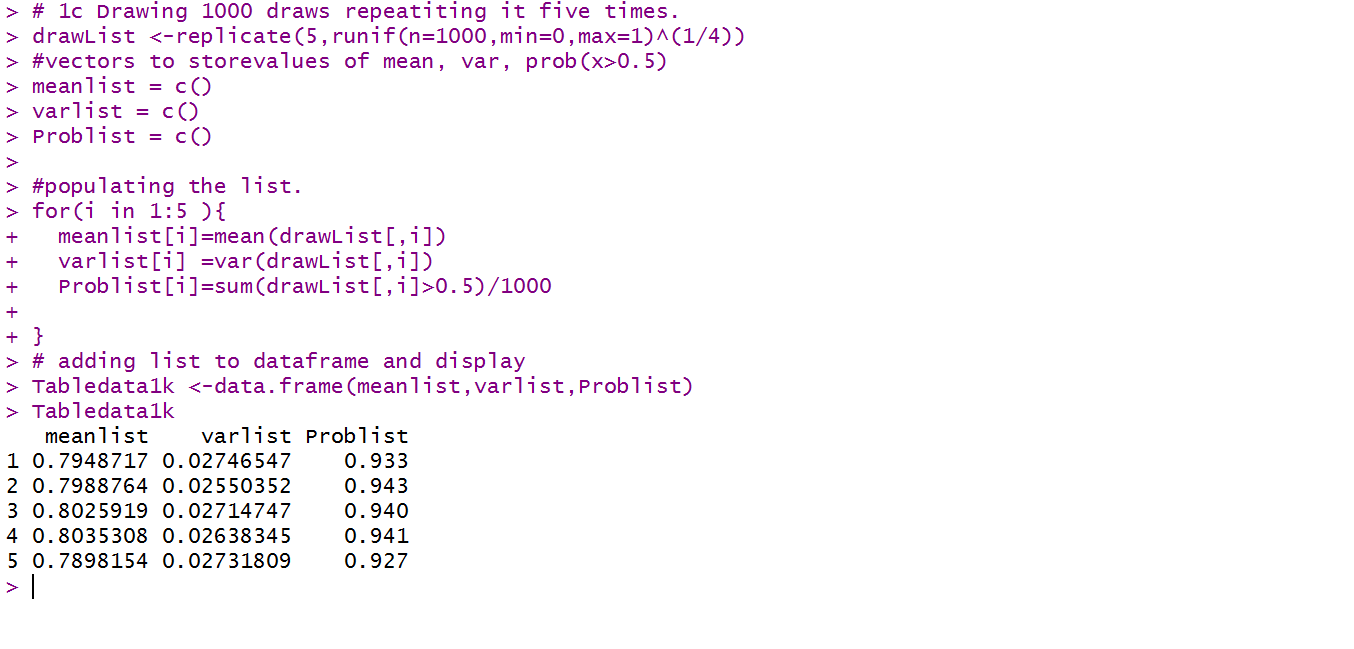
Using this function, we can simulate a draw, by using runif function in R. Example:

**Section 2:**



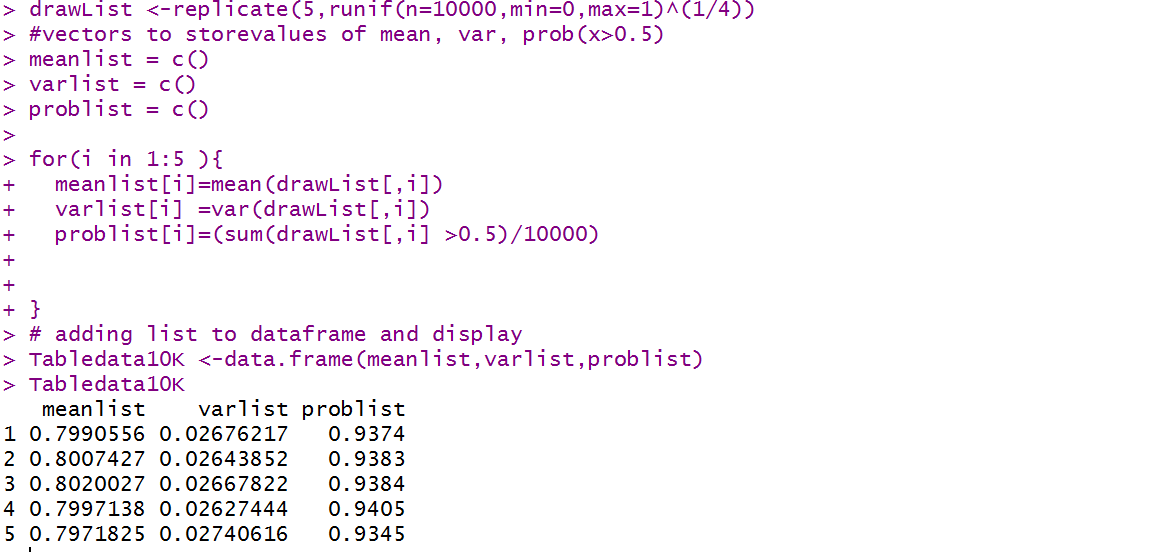
(c) Approximate E(X), Var(X) and P(X > 0.5) using Monte Carlo simulation with 1,000  
draws 5 times. Summarize the results in a table.

Section 1: Simulate MC simulation with N=1000 , use replicate function to repeat it five times. Use standard mean(),var(), sum() if P(x>0.5) /(total no of simulation) for each time and make a table display using dataframe.

Section 2:**(d) Repeat (c) with 10,000 draw**

Section 1: same as above, Change the n to 10,000.

Section 2:



**(e) Compare you results in (a), (c) and (d). Explain, with justification, what you observe.**

From A, mean = 0.8, variance = 0.0266667 , p(x>0.5) =0.9375

|  |  |
| --- | --- |
| From C with 1000 simulations | From D with 10000 simulations |

From the results, we see that the function with given pdf, the mean =0.8 and variance = 0.266667 and p(x>0.5) is 0.9375 using the standard Integration.

Considering when we increase the number of observations n, from 1000 to 10000 we notice that the values are closer to the actual mean, variance and probability(x>0.5).

**Question 2**

**2. IQ test scores have a population mean and standard deviation of 100 and 15, respectively. Assume that the scores follow a normal distribution. ­­­**

1. **Compute the 95-th percentile of this distribution the usual way.**

**Section1:**

Given:

mean = µ = 100

Standard deviation = σ = 15

To Find: The 95th percentile

Manually we do:

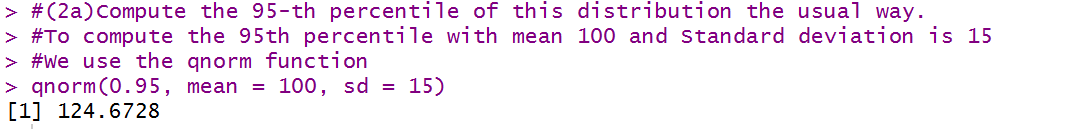
We compute Z using the table or by taking Z = qnorm(0.95) = 1.644854

X = µ + σZ

X = 100 + 15 \* (1.644854)

X = 124.67281

**Section 2:**



1. **Suppose your IQ score equals the percentile you computed in (a). What does this mean?**

**Section 1:**

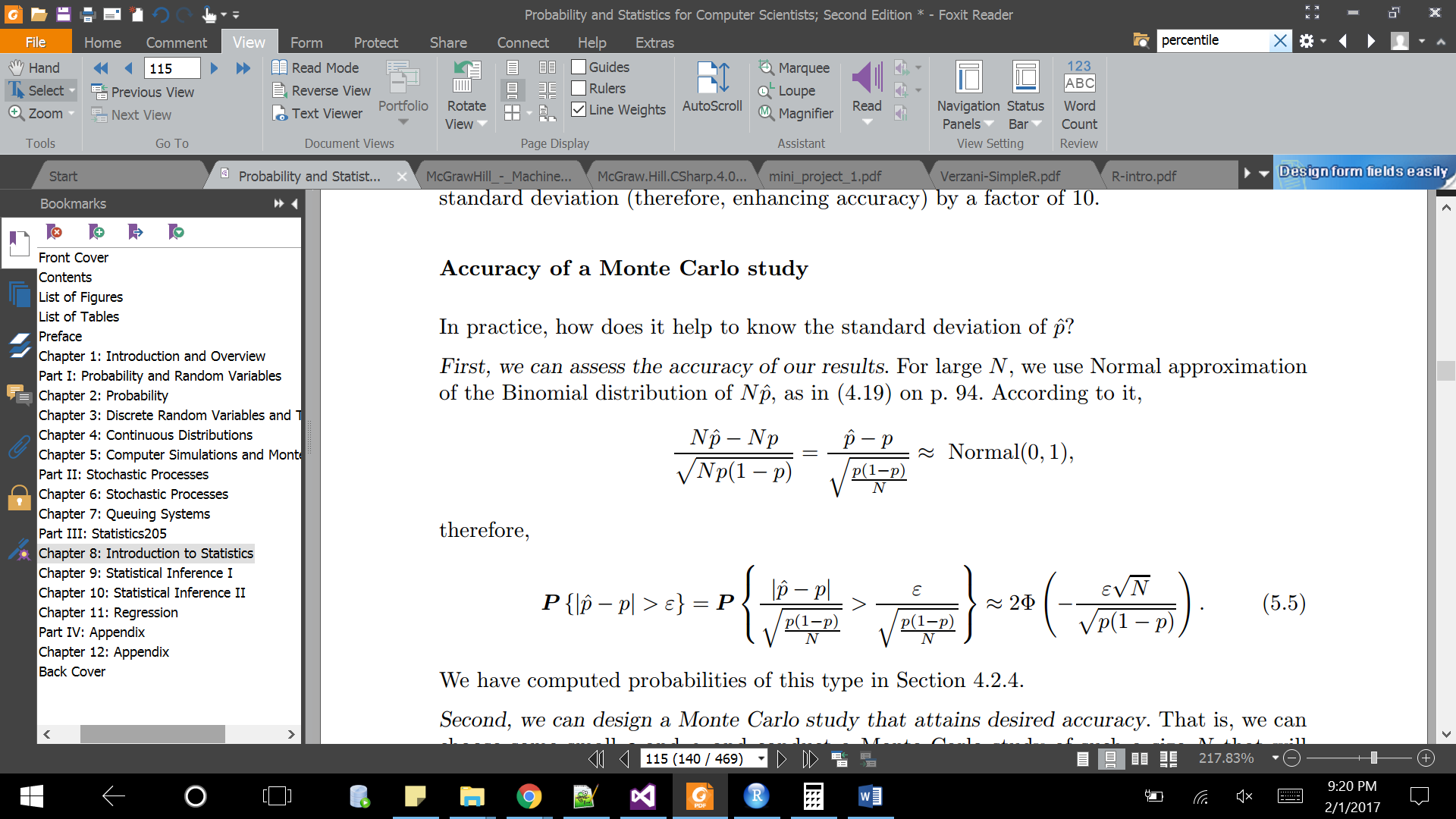
Assuming my IQ score is equal to 124.67281, this implies that 5% of the total population have an IQ greater than mine, while 95% of the population have an IQ lower than mine

1. **Explain how you would simulate a draw from the distribution of the IQ scores.**

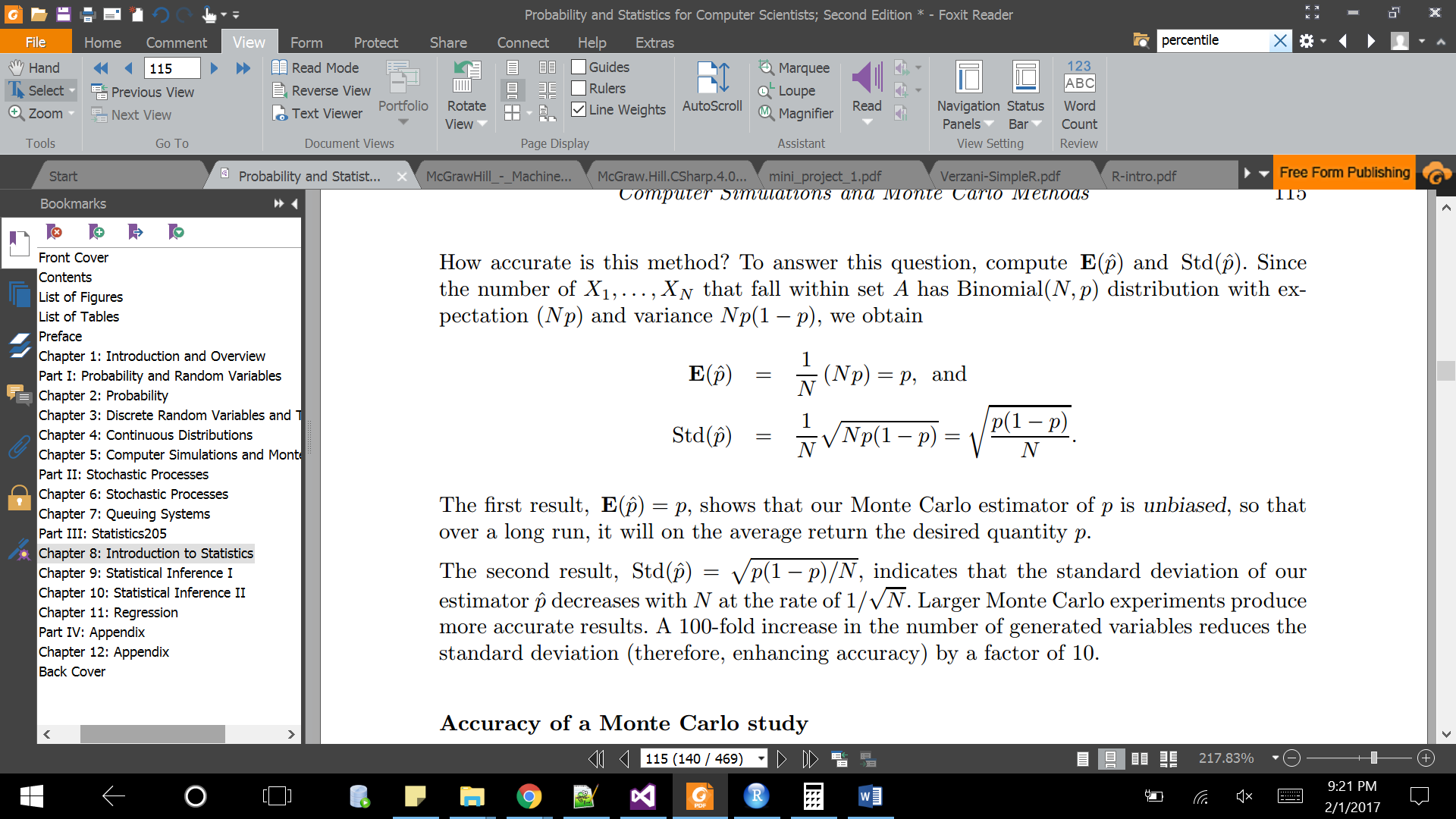
**Section 1:**

The draws can be simulated using the Monte Carlo Method.

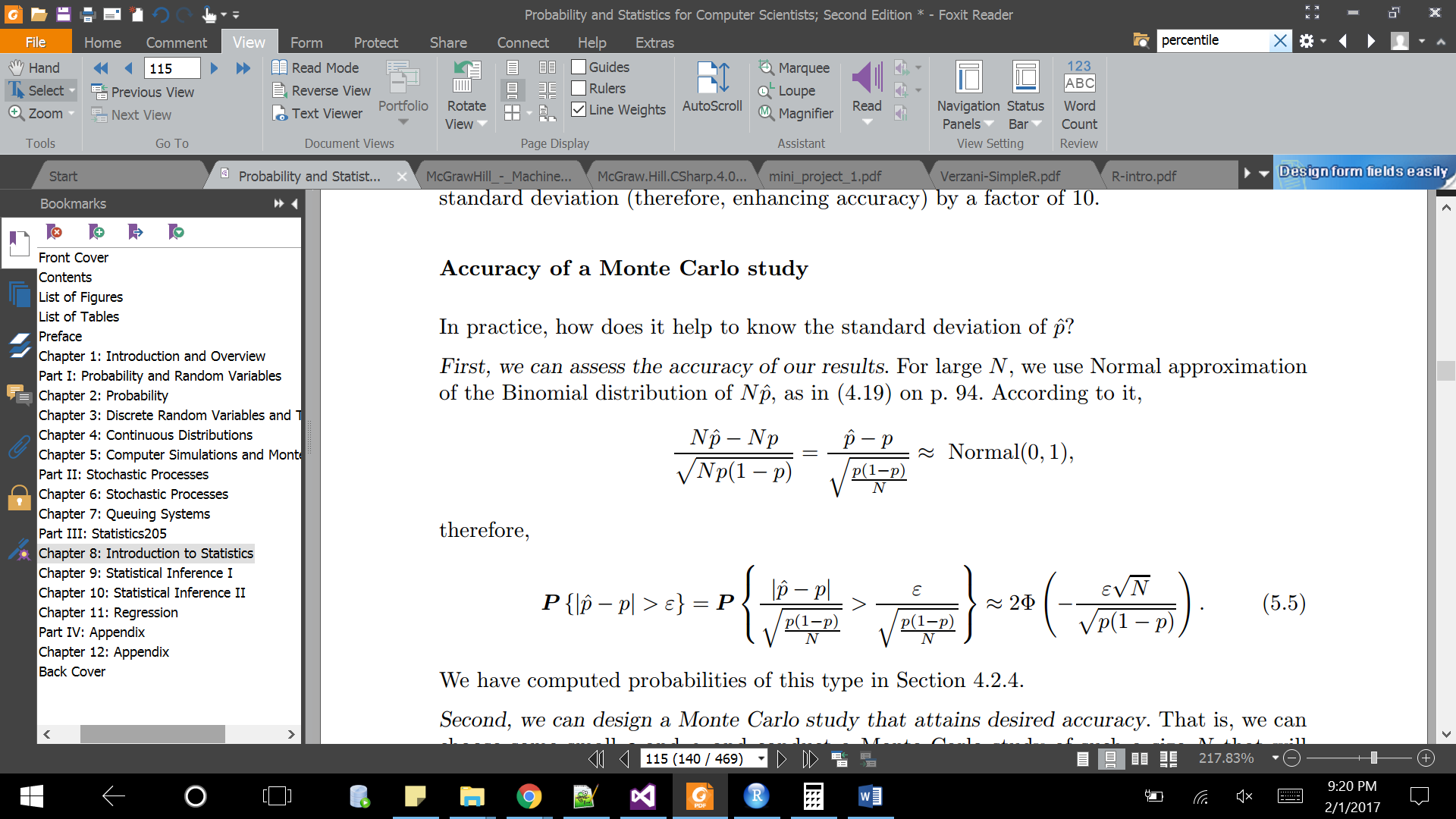
When we know the standard deviation, we can use normal approximation



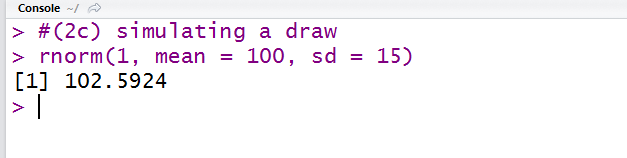
Where



Therefore,

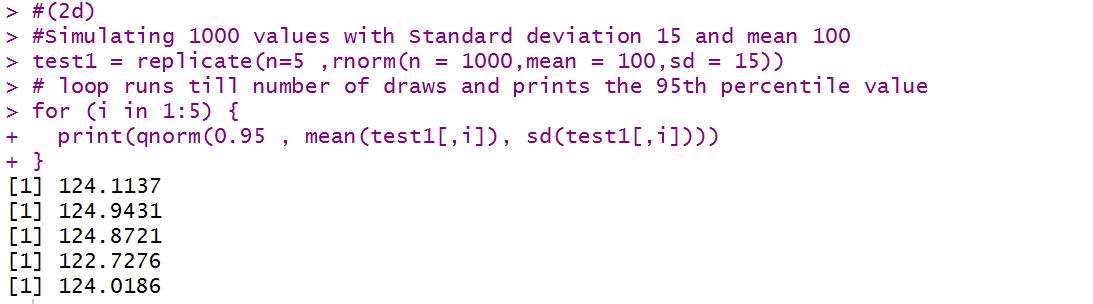


**­Section2:**



1. **Approximate the 95-th percentile of the distribution using Monte Carlo simulation with 1,000 draws 5 times**

**Section 2:**



**e) Repeat (d) with 10,000 draws.**

**Section 2:**

|  |
| --- |
|  |
| |  | | --- | |  |   **(f) Compare your results in (a), (d) and (e). Explain, with justification, what you observe**.  Results from   1. X = 124.67281   d) [1] 124.1137  [1] 124.9431  [1] 125.8721  [1] 125.7276  [1] 125.0186    e) [1] 124.218  [1] 124.7631  [1] 124.5583  [1] 124.6657  [1] 124.5721  From the results, we see that the 95th percentile for a normal distribution with Mean = 100 and SD = 15 is 124.67281  using the standard function. Considering when we increase the number of observations n, from 1000 to 10000,  we notice the values are closer to the 95th Percentile.   Even with value of n=1000, The 95% percentile is not varying very high value proving the central limit theorem, when  N is greater than thousand, the simulation gives the exact value of mean. |

**Appendix: R code for all Problems collectively:**#===================================# 1a# Integrating the function f(x)\*x from 0 to 0.5# the E(x) = f(x) \* xExpFunc <- function(x) {4\*x^3\*x}

expected <- integrate(ExpFunc, lower = 0, upper = 1)

expected$val # # print the value component of Integration result

#Var(X)

varFunc <- function(x) {((x-0.8)^2)\*4\*x^3}

var <-integrate(varFunc, lower = 0, upper = 1)

var$val # # print the value component of Integration result

#p(x>0.5) = 1-P(x<0.5)

ProbFunc <- function(x) {4\*x^3}

Prob <- integrate(ProbFunc, lower =0 , upper =0.5)#P(x<0.5)

final <- 1-Prob$val # print the value component of Integration result

final

#==================================

# 1b sample simulation of using runif with N=1 and ranging from 0 to 1.

runif(n=1,min=0,max=1)^(1/4)

#==================================

# 1c Drawing 1000 draws repeatiting it five times.

drawList <-replicate(5,runif(n=1000,min=0,max=1)^(1/4))

#vectors to storevalues of mean, var, prob(x>0.5)

meanlist = c()

varlist = c()

Problist = c()

#populating the list.

for(i in 1:5 ){

meanlist[i]=mean(drawList[,i])

varlist[i] =var(drawList[,i])

Problist[i]=sum(drawList[,i]>0.5)/1000

}

# adding list to dataframe and display

Tabledata1k <-data.frame(meanlist,varlist,Problist)

Tabledata1k

#==================================

# 1d simulating with 10,000 draws repatiting it five times

drawList <-replicate(5,runif(n=10000,min=0,max=1)^(1/4))

#vectors to storevalues of mean, var, prob(x>0.5)

meanlist = c()

varlist = c()

problist = c()

for(i in 1:5 ){

meanlist[i]=mean(drawList[,i])

varlist[i] =var(drawList[,i])

problist[i]=(sum(drawList[,i] >0.5)/10000)

}

# adding list to dataframe and display

Tabledata10K <-data.frame(meanlist,varlist,problist)

Tabledata10K

==========================

#(2a)Compute the 95-th percentile of this distribution the usual way.

#To compute the 95th percentile with mean 100 and Standard deviation is 15

#We use the qnorm function

qnorm(0.95, mean = 100, sd = 15)

#[1] 124.6728

#(2c) simulating a draw

rnorm(1, mean = 100, sd = 15)

#(2d)

#Simulating 1000 values with Standard deviation 15 and mean 100

test1 = replicate(n=5 ,rnorm(n = 1000,mean = 100,sd = 15))

# loop runs till number of draws and prints the 95th percentile value

for (i in 1:5) {

print(qnorm(0.95 , mean(test1[,i]), sd(test1[,i])))

}

#Question 2e  
#Simulating 10000 values with Standard deviation 15 and mean 100

test2 = replicate(n=5 ,rnorm(n = 10000,mean = 100,sd = 15))

# loop runs till number of draws

for (i in 1:5) {

print(qnorm(0.95 , mean(test2[,i]), sd(test2[,i])))

}