a)

To get the estimate standard error, runif() function is used to get a set of n numbers that is generated according to theta [0,theta]. Then calculate the maximum likelihood and the moments estimator. Replicate the step for 1000 times. Then compute the estimate standard error with the equation separately.

b)

Each time runif function is triggered, two estimators are calculated once to ensure they are based on same data set each time.

*simulate<-function(n,theta)*

*{*

*list<-runif(n,0,theta)#generate uniform number*

*moment<-2\*mean(list) # get moment*

*mle<-max(list) # get mle*

*data<-c(moment,mle) #return the vector*

*return(data)*

*}*

*c)*

*ombination of: 1,1*

*moment: 0.344258759397233 mle: 0.327080835277253*

*Combination of: 2,1*

*moment: 0.177659485071397 mle: 0.169140345419211*

*Combination of: 3,1*

*moment: 0.112387659502238 mle: 0.0999093053669077*

*Combination of: 5,1*

*moment: 0.065784856894359 mle: 0.048978116886898*

*Combination of: 10,1*

*moment: 0.0342326568422309 mle: 0.0156107756469822*

*Combination of: 30,1*

*moment: 0.0112258283146884 mle: 0.00207676940570941*

*Combination of: 1,5*

*moment: 7.96583644206151 mle: 7.99575235268213*

*Combination of: 2,5*

*moment: 4.41918344038428 mle: 4.29044529403165*

*Combination of: 3,5*

*moment: 2.91109675960776 mle: 2.55189249737367*

*Combination of: 5,5*

*moment: 1.60352364712427 mle: 1.21575642759498*

*Combination of: 10,5*

*moment: 0.854402462728854 mle: 0.390339111052917*

*Combination of: 30,5*

*moment: 0.259324432075741 mle: 0.0508835184951293*

*Combination of: 1,50*

*moment: 824.021676565861 mle: 779.282208026006*

*Combination of: 2,50*

*moment: 421.030929708223 mle: 432.95331324951*

*Combination of: 3,50*

*moment: 274.307972052006 mle: 254.992331246428*

*Combination of: 5,50*

*moment: 170.274370840227 mle: 118.052244665051*

*Combination of: 10,50*

*moment: 81.5946558043778 mle: 38.0198179871017*

*Combination of: 30,50*

*moment: 28.2963081975532 mle: 4.82360101066434*

*Combination of: 1,100*

*moment: 3254.29253784129 mle: 3429.97569447181*

*Combination of: 2,100*

*moment: 1640.11768000083 mle: 1715.17889029731*

*Combination of: 3,100*

*moment: 1147.61443743237 mle: 984.809863681098*

*Combination of: 5,100*

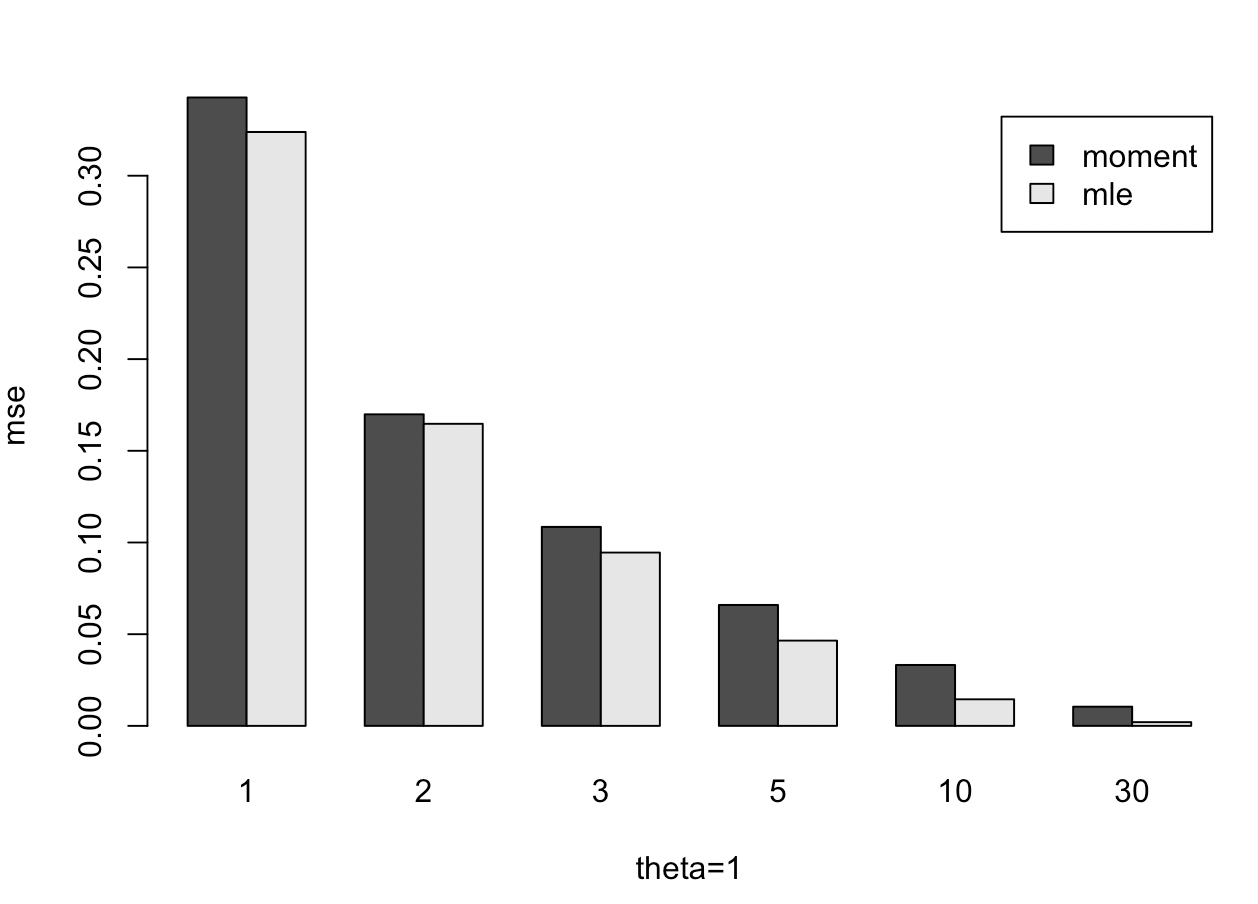
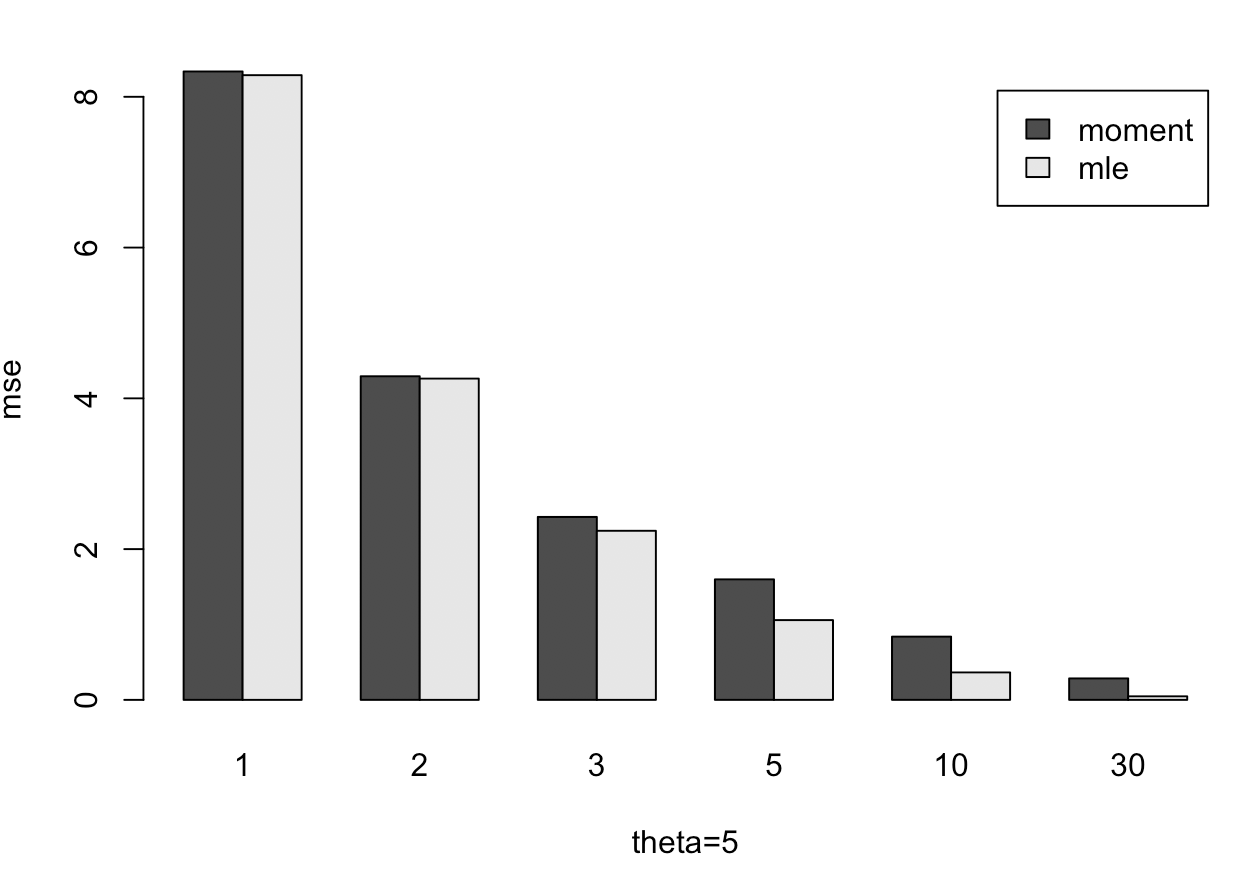
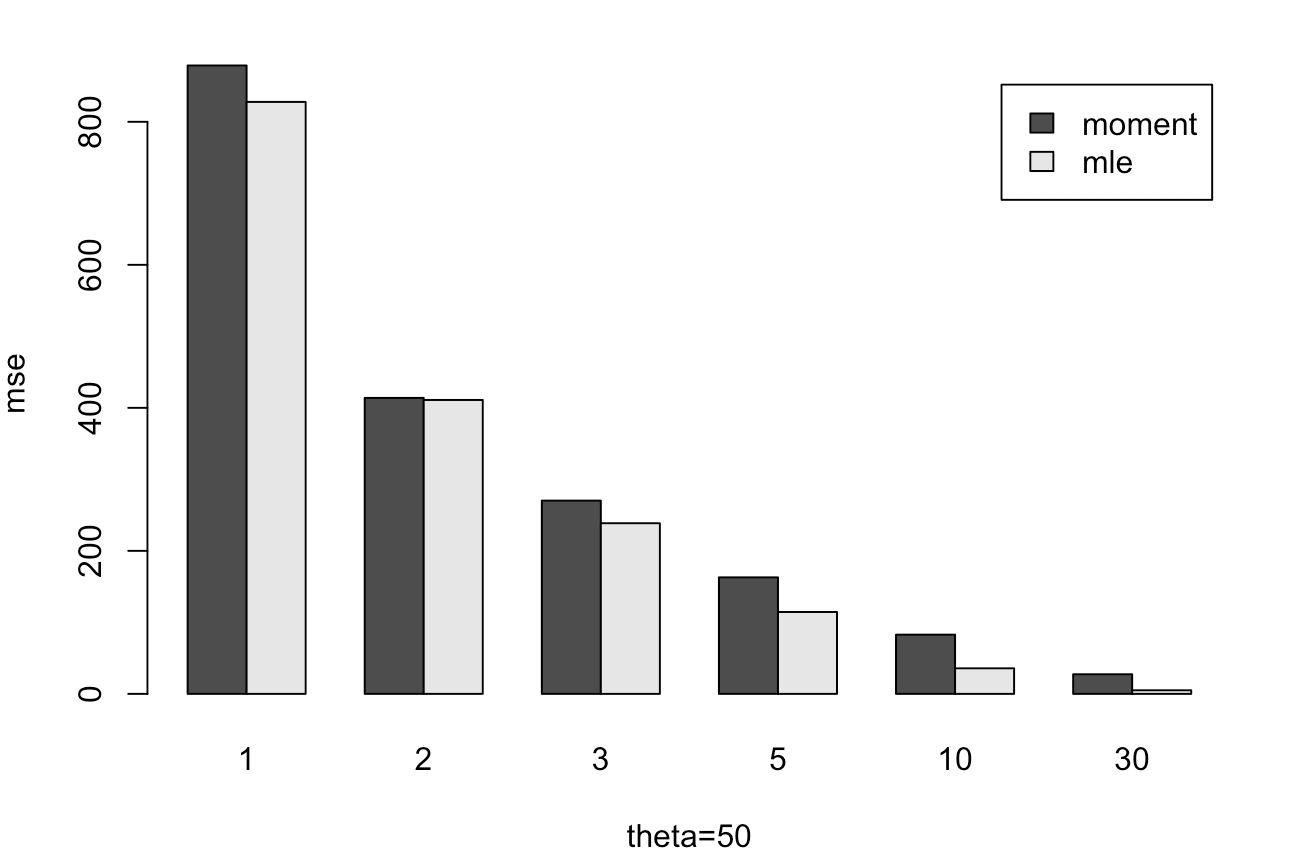
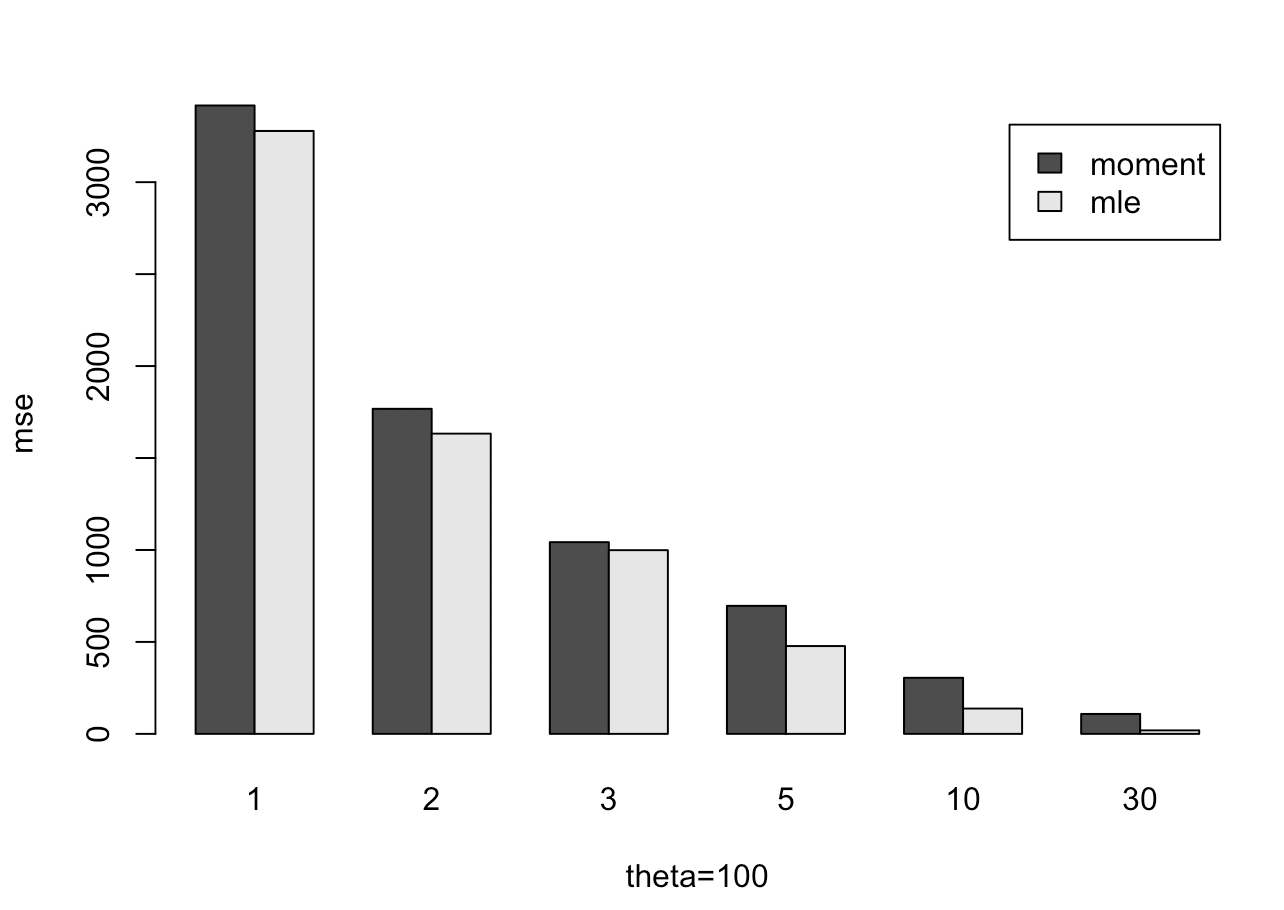
*moment: 682.872432356001 mle: 472.662704746288*

*Combination of: 10,100*

*moment: 357.175726956612 mle: 138.859461187434*

*Combination of: 30,100*

*moment: 114.757528350687 mle: 21.4376664056975*

d) Based on c, sometime the standard error of moment is lower, but most of the time, mle has lower standard error. As the numbers of data set get larger, the better standard error maximum likelihood shows. It’s clear that the answer doesn’t depend on theta. It’s also doesn’t depend on n when n is large. But when the n is small, the answer may swing (moment may have similar or even smaller standard error). The conclusion is based on output of the above function.

CODE

*simulate<-function(n,theta)*

*{*

*list<-runif(n,0,theta)#generate uniform number*

*moment<-2\*mean(list) # get moment*

*mle<-max(list) # get mle*

*data<-c(moment,mle) #return the vector*

*return(data)*

*}*

*se<-function(x,theta)*

*{*

*return ((x-theta)\*(x-theta)) #se*

*}*

*compare<-function(n,theta)*

*{*

*cat(sprintf("Combination of: %s,%s\n", n,theta))*

*data\_m<-replicate(1000,simulate(n,theta))#do it 1000 times*

*#matrix<-matrix(data\_m,nrow=2,byrow=FALSE) #set the matrix row=2 and fill the matrix by column*

*se\_temp<-apply(data\_m,1,se,theta) #calculate (head\_theta-theta)^2*

*sum<-apply(se\_temp,2,sum) #get sum of each row*

*m\_o<-sum[1]/1000;*

*m\_m<-sum[2]/1000; #get mean of se*

*cat(sprintf("moment: %s ",m\_o))*

*cat(sprintf("mle: %s\n",m\_m))*

*return(c(m\_o,m\_m))*

*}*

*n<-c(1,2,3,5,10,30)*

*theta<-c(1,5,50,100)*

*combination<-expand.grid(n,theta) # get all combination*

*v = as.vector(unlist(combination))*

*combination<-matrix(v,ncol=2,byrow=FALSE)*

*output<-apply(combination,1,function(x)do.call(compare,as.list(x)))# use each row as input to use function*

*#########get combination to draw graphic*

*n<-c(1,2,3,5,10,30)*

*theta<-c(100)*

*combination<-expand.grid(n,theta) # get combination*

*v = as.vector(unlist(combination))*

*combination<-matrix(v,ncol=2,byrow=FALSE)*

*output<-apply(combination,1,function(x)do.call(compare,as.list(x)))*

*region<-c("moment","mle")*

*barplot(output,names.arg=n,ylab="mse",xlab="theta=100",beside=T,legend.text = region)# draw barplot*

*###########################*