```
#start
################################
########### jalase 1############
################################
x=-10:10
y=(1-x)^2
y

plot(x,y)
plot(x,y,type="l",col=2,lty=2)
abline=1
abline(v=1,col=2)
abline(h=0,col=4,lty=2)
################################
#rasm nemodar
curve((1-x)^2,-10,10)
curve(x*log(x),0,1)
################################
########## jalase 2 ############
################################
x=-10:10
y=(1-x)^2
plot(x,y,typ="b")
curve((1-x)^2,-10,10)
curve(x*log(x),0,0.6,type="b")
abline(h=-0.367,lty=2,col=2)
abline(v=0.35,lty=2,col=4)
x<-c(0.34,0.35,0.36,0.37,0.38,0.39)
x*log(x)
cbind(x,x*log(x))
rbind(x,x*log(x))
abline(h=-0.36787,v=0.37,lty=2,col=3)

curve(x*sin(2*pi*x),0,0.5)
curve(x*exp(-x),0,10)
curve(log10(x)/(1+x),0,10)
abline(h=0.1209356,v=3.59,lty=2,col=3)
x<-seq(3,4,by=0.1)
x
cbind(x,log10(x)/(1+x))
#####2 ragham ashar
x<-seq(3.59,3.7,by=0.01)
x
cbind(x,log10(x)/(1+x))
################################
########### jalase 3############
################################
curve((1+x*0.4)/2,-1,1)

y<-
c(0.41,0.91,-.61,.38,.37,.36,.01,-.28,-.33,.99,-.35,.1,.31,.75,-.34)
y
##### function####
La<-function(a){
```

```
y<-
c(0.41,0.91,-.61,.38,.37,.36,.01,-.28,-.33,.99,-.35,.1,.31,.75,-.34)
m=matrix(0,ncol=length(a),nrow=length(y))
for(i in 1:length(a)){
m[,i]=(1+a[i]*y)/2
}
z<-apply(m,2,prod)
return(z)
}
La(0.5)
La(c(0.5,0.6))
curve(La(x),-1,1)
##########################
#######jalase 4###########
##########################
FG<-function(alpha,y){
m=y^(alpha-1)*exp(-y)/gamma(alpha)
return(m)
}

y<-rgamma(10,2)
FG(2,y)

FG1<-function(alpha){
y<-rgamma(300,2)
m1=matrix(0,ncol=length(alpha),nrow=length(y))
for(i in 1:length(alpha)){
m1[,i]=FG(alpha[i],y)
}
z<-apply(m1,2,prod)
return(z)
}

FG1(c(1,2,3,4,5))

curve(FG1(x),1,3)

logFG<-function(alpha,y){
m=((alpha-1)*log(y)-y)-log(gamma(alpha))
return(m)
}

FG2<-function(alpha){
y<-rgamma(1000,2)
m1=matrix(0,ncol=length(alpha),nrow=length(y))
for(i in 1:length(alpha)){
m1[,i]=logFG(alpha[i],y)
}
z<-apply(m1,2,sum)
return(z)
}
curve(FG2(x),1,3)

###################################
```

```
########### Practice ##############
####################################
#f(x)=x^p-1*(1-x)^q-1/B(p,q)
#0=x=1;p,q>0
#f(x) = x^(alpha-1)*(1+x)^(-alpha -beta)/B(alpha,beta)
####################################
#rasm chegali tozi beta####
curve(dbeta(x,1,1),0,1)
####################################
?rbeta

FB<-function(alpha,Beta,y){
m= y^(alpha-1)*(1-y)^(Beta - 1)/beta(alpha,Beta)
return(m)
}

y<-rbeta(n=5,2,3)
y
FB(2,3,y)

FB1<-function(alpha){
y<-rbeta(10,0.5,3)
Beta<-3
m1=matrix(0,ncol=length(alpha),nrow=length(y))
for(i in 1:length(alpha)){
m1[,i]=FB(alpha[i],Beta,y)
}
z<-apply(m1,2,prod)
return(z)
}


FB1(c(1,2,3,4,5))

curve(FB1(x),0,1)
abline(v=0.5,col=2,lty=2)

####### log-likelihood######
logFB<-function(alpha,Beta,y){
m=(alpha-1)*log(y)+(Beta-1)*log(1-y)-log(beta(alpha,Beta))
return(m)
}

FB2<-function(alpha){
y<-rbeta(10,0.5,3)
Beta<-3
m1=matrix(0,ncol=length(alpha),nrow=length(y))
for(i in 1:length(alpha)){
m1[,i]=logFB(alpha[i],Beta,y)
}
z<-apply(m1,2,sum)
return(z)
}
```

```
curve(FB2(x),0,2)
abline(v=0.5,col=4,lty=2)

par(mfrow=c(1,2))
curve(FB1(x),0,2)
title("lik")
abline(v=0.5,col=2,lty=2)
curve(FB2(x),0,2)
title("log-lik")
abline(v=0.5,col=4,lty=2)

########################################
#f(x)=lamda^x*exp(-lamda)/factorial(x)
#lamda>0,x=0,1,2,...
########################################
?dpois

PO<-function(lambda,y){
m1=lambda^y*exp(-lambda)/factorial(y)
m2=(-lambda+y*log(lambda)-log(factorial(y)))
m=list(L=m1,logL=m2)
return(m)
}

y<-rpois(10,3)
PO(3,y)

PO1<-function(lambda){
y<-rpois(n=1000,3)
m1=matrix(0,ncol=length(lambda),nrow=length(y))
m2=matrix(0,ncol=length(lambda),nrow=length(y))
for(i in 1:length(lambda)){
m<-PO(lambda[i],y)
m1[,i]=m$L
m2[,i]=m$logL
}
z1<-apply(m1,2,prod)
z2<-apply(m2,2,sum)
z<-list(L=z1,logL=z2)
return(z)
}

PO1(1:10)

par(mfrow=c(1,2))
curve(PO1(x)$L,2,5)
abline(v=3,col=2,lty=2)
curve(PO1(x)$logL,2,4)
abline(v=3,col=2,lty=2)
########################
######## jalase 6########
########################
?optim
?uniroot
```

```
?optimize
#minimom (1-x)^2 by optimize#
f<-function(x) (1-x)^2
f(2)
optimize(f,lower=0,upper=5)
############################
#f(x)=(1+alpha*x)/2
#  -1<alpha<1
#max f(x)
# x1...xn
y<-
c(0.41040018,0.91061564,-0.61106896,0.39736684,0.37997637,0.34565436
,0.01906680,-0.28765977,-0.33169289,0.99989810
,-0.35203164,0.10360470,0.30573300,0.75283842,-0.33736278,-0.6145510
1,-0.76222116,0.27150040,-0.01257456
,0.68492778,-0.72343908,0.45530570,0.86249107,0.52578673,0.14145264,
0.76645754,-0.65536275,
0.12497668,0.74971197,0.53839119)
y

La<-function(a,y){
m=matrix(0,ncol=length(a),nrow=length(y))
for(i in 1:length(a)){
m[,i]=(1+a[i]*y)/2
}
z<-apply(m,2,prod)
return(z)
}
La(0.5,y)
La(c(0.5,0.6),y)
curve(La(x,y),-1,1)
################################################
f<-function(alpha){
y<-
c(0.41040018,0.91061564,-0.61106896,0.39736684,0.37997637,0.34565436
,0.01906680,-0.28765977,-0.33169289,0.99989810
,-0.35203164,0.10360470,0.30573300,0.75283842,-0.33736278,-0.6145510
1,-0.76222116,0.27150040,-0.01257456
,0.68492778,-0.72343908,0.45530570,0.86249107,0.52578673,0.14145264,
0.76645754,-0.65536275,
0.12497668,0.74971197,0.53839119)
-La(alpha,y)
}

P<-optimize(f,lower=-1,upper=1)
P
P$minimum
names(P)
curve(La(x,y),-1,1)
abline(v=P$minimum,col=2,lty=2)
################################################
############### practise for 7 meeting############
################################################
FG<-function(alpha,y){
```

```
m1=y^(alpha-1)*exp(-y)/gamma(alpha)
m2=((alpha-1)*log(y)-y)-log(gamma(alpha))
m=list(L=m1,logL=m2)
return(m)
}

y<-rgamma(n=10,3)
FG(3,y)

FG2<-function(alpha,y){
m1=matrix(0,ncol=length(alpha),nrow=length(y))
m2=matrix(0,ncol=length(alpha),nrow=length(y))
for(i in 1:length(alpha)){
m<-FG(alpha[i],y)
m1[,i]=m$L
m2[,i]=m$logL
}
z1<-apply(m1,2,prod)
z2<-apply(m2,2,sum)
z<-list(L=z1,logL=z2)
return(z)
}
FG2(1,y)

f1<-function(alpha){
set.seed(10)
y<-rgamma(n=10,3)
-FG2(alpha,y)$logL
}
f2<-function(alpha){
set.seed(10)
y<-rgamma(n=10,3)
-FG2(alpha,y)$L
}
f1(2)
f2(3)
op1<-optimize(f1,lower=0.001,upper=100)
op1
op1$minimum
op2<-optimize(f2,lower=0.001,upper=100)
op2
op2$minimum
set.seed(10)
y<-rgamma(n=10,3)
par(mfrow=c(1,2))
curve(FG2(x,y)$logL,1,10)
abline(v=op1$minimum,col=2,lty=2)
curve(FG2(x,y)$L,1,10)
abline(v=op2$minimum,col=2,lty=2)
?optimize
####################################
PO<-function(lambda,y){
m1=lambda^y*exp(-lambda)/factorial(y)
m2=(-lambda+y*log(lambda)-log(factorial(y)))
```

```
m=list(L=m1,logL=m2)
return(m)
}

y<-rpois(10,3)
PO(3,y)

PO2<-function(lambda,y){
m1=matrix(0,ncol=length(lambda),nrow=length(y))
m2=matrix(0,ncol=length(lambda),nrow=length(y))
for(i in 1:length(lambda)){
m<-PO(lambda[i],y)
m1[,i]=m$L
m2[,i]=m$logL
}
z1<-apply(m1,2,prod)
z2<-apply(m2,2,sum)
z<-list(L=z1,logL=z2)
return(z)
}
f1<-function(lambda){
set.seed(10)
y<-rpois(n=10,3)
-PO2(lambda,y)$logL
}
f2<-function(lambda){
set.seed(10)
y<-rpois(n=10,3)
-PO2(lambda,y)$L
}
f1(1:10)
f2(1:10)
op1<-optimize(f1,lower=0.001,upper=100)
op1
op1$minimum
op2<-optimize(f2,lower=0.001,upper=100)
op2
op2$minimum
set.seed(10)
y<-rpois(n=10,3)
par(mfrow=c(1,2))
curve(PO2(x,y)$logL,1,10)
abline(v=op1$minimum,col=2,lty=2)
curve(PO2(x,y)$L,1,10)
abline(v=op2$minimum,col=2,lty=2)
#####################################
############ jalase 7################
#####################################
###########simulation################
#####################################
n<-100000000
x<-sample(1:6,n,replace = TRUE)
#x
sum(x==6)
```

```
p=sum(x==6)/n
p
1/6
#####################################
rep=100000
z<-0
for(i in 1:rep){
y<-sample(1:100,5)
z[i]<-sum(y>90)
}
#z
#z==0
p<-mean(z==0)
p
#to make function
f.p<-function(rep){
z<-0
for(i in 1:rep){
y<-sample(1:100,5)
z[i]<-sum(y>90)
}
p<-mean(z==0)
return(p)
}
############
f.p(100)
rep1<-100
m=0
for(j in 1:rep1){
m[j]=f.p(100000)
}
m
plot(m,type="l",ylim=c(0.5,0.7))
abline(h=mean(m),lty=2,col=2)
#############################
#example 1.3,page6
x<-sample(365,25,replace=T)
x
y<-unique(x)
y
25-length(y)
rep=100000
m=0
for(i in 1:rep){
x<-sample(365,25,replace=T)
y<-unique(x)
m[i]=25-length(y)
}
#sum(m==0)
p=mean(m==0)
1-p
hist(m)
#################################
#########jalase 9################
```

```
################################
n=10
p=0.5
x<-rbinom(n,1,p)
x
sum(x)
phat<-mean(x)
phat
alpha=0.05
U<-phat+qnorm(1-alpha/2)*sqrt(phat*(1-phat)/n)
L<-phat-qnorm(1-alpha/2)*sqrt(phat*(1-phat)/n)
cbind(L,U)
IN<-(L<=p & p<=U)
IN
cbind(L,U,IN)
################################################
#to make function above
Conf<-function(n=10,p=0.5,alpha=0.05){
x<-rbinom(n,1,p)
phat<-mean(x)
U<-phat+qnorm(1-alpha/2)*sqrt(phat*(1-phat)/n)
L<-phat-qnorm(1-alpha/2)*sqrt(phat*(1-phat)/n)
IN<-(L<=p & p<=U)
return(cbind(L,U,IN))
}
U
L
#Conf()
#Conf(n=30,p=0.5)
rep=100
m<-matrix(NA,ncol=3,nrow=rep)
for(i in 1:rep){
m[i,]<-Conf(n=30,p=0.9)
}
m
#m[95,2]
p.co<-mean(m[,3])
p.co
y<-rbinom(500,10,0.9)
hist(y)
##############################
#########jalase 10############
##############################
#generating random numbers
#exampel 2.1 ,page 25
d<-53
b<-0
a<-20
n<-100
r<-numeric(n)
#r
seed=12
r[1]<-seed
for(i in 1:n){
```

```
r[i+1]=(a*r[i]+b)%%d
}
r
u<-(r+0.5)/d
u
hist(u)
#example 2.2,page28
d<-86436
b<-18257
a<-1093
n<-1000
r<-numeric(n)
seed=7
r[1]<-seed
for(i in 1:n){
r[i+1]=(a*r[i]+b)%%d
}
u<-(r+0.5)/d
hist(u)
u1<-u[1:(n-1)]
u2<-u[2:n]
plot(u1,u2,pch=20)
plot(u1,u2,xlim=c(0,0.1),ylim=c(0,0.1),pch=20)
#example2.3

d<-2^31
b<-0
a<-65539
n<-1000000
r<-numeric(n)
seed=7
r[1]<-seed
for(i in 1:n){
r[i+1]=(a*r[i]+b)%%d
}
u<-(r+0.5)/d
hist(u)
u1<-u[1:(n-1)]
u2<-u[2:n]
plot(u1,u2,pch=20)
plot(u1,u2,xlim=c(0,0.02),ylim=c(0,0.02),pch=20)
#######################################
##############jalase11################
#######################################
#x~exp(lambda)
#generating random numbers of x _mesal 1
lambda=2
n=100000
u<-runif(n)
u
x<-(-lambda*log(1-u))
x
mean(x)
hist(x,freq=FALSE)
```

```
abline(v=mean(x),col=2,lty=3)
abline(v=lambda,col=5,lty=3)
#text(10,0.3,"mean of x",col=25,cex=2)
legend("topright",c("hist of x","mean of
x","lambda"),col=c(1,2,5),lty=c(1,3,3))
lines(density(x),col=2,lty=2)
lambda=3
curve(1/lambda*exp(-x/lambda),0,20,col=6,cex=2,add=TRUE)
############################
#use Q-Q plot
x<-rnorm(1000)
y<-rt(1000,2)
#H0: X=Y in distribution
qqplot(x,y)
abline(0,1,col=2,lty=2)
curve(dnorm(x),-100,100,xlim=c(-10,-5),ylim=c(0,0.1))
curve(dt(x,2),-100,100,col=2,add=TRUE)
############################
#edame mesal 1
ppoints(100)
n=100000
u<-runif(n)
x<-(-lambda*log(1-u))
x
qqplot(qexp(ppoints(100),2),x)
qqline(x,distribution=function(p) qexp(p,2))

qqplot(qt(ppoints(100),4),x)
qqline(x,distribution=function(p) qt(p,4))
########################################
##x~GAMMA(alpha,gamma)
#y^(alpha-1)*exp(-y)/gamma(alpha)

alpha=2
n=10
u<-runif(n)
x<-gamma(alpha/2,x/2)/gamma(alpha/2)
x
mean(x)
hist(x,freq=FALSE)
abline(v=mean(x),col=2,lty=2)
lines(density(x),col=2,lty=2)
alpha=2
curve(x^(alpha-1)*exp(-x)/gamma(alpha),0,10,col=2,lty=2)

ppoint(100)
qqplot(qgama(ppoint(100),2),x)
qqline(x,distribution=function(p) qgama(2,gama))
#####################################
#############jalase 12###############
#####################################
#azmon adadi k.s test baraye 2 tozi
lambda=2
n=100
```

```
u<-runif(n)
x<-(-lambda*log(1-u))
#x
#?ks.test
y<-rnorm(n)
#y
ks.test(x,y)
z<-rexp(lambda)
ks.test(x,z)
ks.test(x,"pexp",1/2)
#?pexp
ks.test(x,"pnorm")
##################################
alpha=2
n=100
u<-runif(n)
x<-u^(1/alpha)
mean(x)
hist(x,freq=FALSE)
abline(v=mean(x),col=2,lty=2)
lines(density(x),col=3,lty=5)
curve(dbeta(x,2,1),0,2,col=2,lty=5,add=TRUE)

y<-rbeta(100,2,1)
qqplot(x,y)
abline(0,1,col=2,lty=2)
alpha=2
n=1000
u<-runif(n)
x<-u^(1/alpha)
alpha=2
ppoints(100)
qqplot(qbeta(ppoints(100),alpha,1),x)
qqline(x,distribution=function(p) qbeta(p,alpha,1))
ks.test(x,"pbeta",alpha,1)
########################################
#############jalase 13#################
########################################
n=1000
u<-runif(n)
x<-u^2
par(mfrow=c(1,2))
hist(x,freq=FALSE)
lines(density(x),col=2,lty=2)
curve(dbeta(x,0.5,1),0,1,col=3,lty=3,add=TRUE)
qqplot(qbeta(ppoints(100),0.5,1),x)
qqline(x,distribution=function(p) qbeta(p,0.5,1))
text(0.2,0.8,"p-value = 0.6366",cex=2,col=2)
ks.test(x,"pbeta",0.5,1)
########################################
n=1000
u<-runif(n)
x<--2*log(u)
par(mfrow=c(1,2))
```

```
hist(x,freq=FALSE)
lines(density(x),col=2,lty=2)
curve(dchisq(x,2),0,20,col=3,lty=3,add=TRUE)
qqplot(qchisq(ppoints(100),2),x)
qqline(x,distribution=function(p) qchisq(p,2))
#??pchisq
ks.test(x,"pchisq",2)
?distribution
########################################
#box_muller transformation
########################################
#to generat random sample from normal distribution
########################################
n=1000
u1<-runif(n)
u2<-runif(n)
z1<-sqrt(-2*log(u1))*cos(2*pi*u2)
z1
z2<-sqrt(-2*log(u1))*sin(2*pi*u2)
z2
par(mfrow=c(2,2))
hist(z1,freq=FALSE)
lines(density(z1),col=2,lty=2)
curve(dnorm(x),-4,4,col=3,lty=3,add=TRUE)
qqnorm(z1)
qqline(z1)
hist(z2,freq=FALSE)
lines(density(z2),col=2,lty=2)
curve(dnorm(x),-4,4,col=3,lty=3,add=TRUE)
qqnorm(z2)
qqline(z2)
shapiro.test(z1)
shapiro.test(z2)
########################################
###########jalase14####################
########################################
#chapter 3
#remann  approximation
#integrate e^-x in [0,1]
a=0
b=1
n=10000
w<-(b-a)/n
ksi<-seq(a+w/2,b-w/2,length=n)
h<-exp(-ksi)
s<-sum(h*w)
s
real<-1-exp(-1)
real
error<-real-s
error
#####integrate e^(-x^2) in [0,1]
a=0
b=1
```

```
n=10
w<-(b-a)/n
ksi<-seq(a+w/2,b-w/2,length=n)
h<-exp(-ksi^2)
s<-sum(h*w)
s
real<-integrate(function(x) exp(-x^2),a,b)
real
#names(real)
error<-real$value-s
error
#?integrate
################################
#Monte Carlo approximation
#random sampleing
n=1000000
z<-rnorm(n)
p<-mean((0<=z) & (z<=1))
p
##############################################
#intrgerate cos(2*pi*x)*exp(-x*sin(2*pi*x))
a=-1
b=1
n=1000000
w=(b-a)/n
ksi<-seq(a+w/2,b-w/2,length=n)
h<-exp(-ksi)
s<-sum(sin(2*x)+cos(x^2))*exp(-ksi))
s
real<-integrate(function(x) cos(2*pi*x)*exp(-x+sin(2*pi*x)),a,b)
real
error<-real$value-s
error
##############################################
########### jalase 15#########################
##############################################
#generating sample random from f(x)=1/2*exp(-|x|+x)
n=1000000
u<-runif(n)
#?ifelse
x<-ifelse(u<=0.5,log(2*u),-log(2*(1-u)))
hist(x,freq=FALSE,ylim=c(0,0.5))
lines(density(x),lty=2,col=2)
curve(1/2*exp(-abs(x)),-6,6,col=4,add=TRUE)
##############################################
#end
```