#1. Consider mtcars and iris (Motor Trend Car Road Tests and Iris) #data sets available in R-statistical software, and find #i. Mean, median, mode, 1st quartile, 2nd quartile, 3rd quartile, variance, #standard deviation, and covariance between any two variables using inbuilt #functions and by writing your own function with proper documentation. rm(list=ls(all=TRUE)) #checklen function used in almost all programs checklen=function(x){ i=1; while(!is.na(x[i,1])){ i=i+1; if(is.na(x[i,1])){ break; } } checklen=i-1 checklen } #——–finding mean of column mpg and Sepal.Length x=mtcars y=iris mean(x[,1]) mean(y[,1]) #ans=20.09062 for mtcars #ans=5.843333 for iris

#for finding sum of observations of mpg column and Sepal.Length #here we are passing dataset and column number as parameter findMean = function(x,n){ i=1; #to keep track of iterations sum=0; total=checklen(x) for(i in 1:total){ sum=sum+x[i,n] } #mean= sum of observations/total number of observations findMean=sum/checklen(x) findMean } #ans 20.09062 for mtcars #ans 5.843333 for iris

#——–finding median of column mpg and Sepal.Length median(x[,1]) median(y[,1]) #ans=19.2 for mtcars #ans=5.8 for iris findMedian=function(x){ #code below is to sort the array. t is a temporary variable used in swapping for(i in 1:(checklen(x)-1)){ for(j in (i+1):checklen(x)){ if(x[j,1]<x[i,1]){ t=x[i,1] x[i,1]=x[j,1] x[j,1]=t } } } #if number of elements odd then median at (n+1)/2th position # %% used to check remainder if(checklen(x)%%2==1){ med=x[((checklen(x)+1)/2),1] } else{ m1=x[((checklen(x))/2),1] m2=x[(((checklen(x))/2)+1),1] med=(m1+m2)/2 } findMedian=med findMedian } #ans=19.2

#——–finding mode of column mpg and Sepal.Length f=table(x[,1]) table(y[,1]) #frequencies of numbers shown. 10.4, 15.2, 19.2, 21, 21.4, 22,8, 30.4 occur #occur two times #function to find mode without in build function findMode=function(x){ #sorting array for(i in 1:(checklen(x)-1)){ for(j in ((i+1):(checklen(x)))){ if(x[j,1]<x[i,1]){ t=x[i,1] x[i,1]=x[j,1] x[j,1]=t } } }

h=rep(0,(checklen(x))) #checking for frequency and updating for(i in 1:(checklen(x)-1)){ for(j in (i+1):checklen(x)){ if(x[i,1]==x[(j),1]){ h[i]=h[i]+1 } }

} #finding max frequency max=h[1] for(i in 1:(checklen(x))){ if(h[i]>max){ max=h[i] } } #printing numbers with max frequency for(i in 1:(checklen(x))){ if(h[i]==max){ print(x[i,1]) } } }

#——–finding quartiles and median of column mpg and Sepal.Length quartile=quantile(x[,1],c(0.25,0.5,0.75)) #this in build function gives value of 1st quartile, median and 3rd quartile #ans = 15.425, 19.2, 22.8 for mtcars #ans = 5.1, 5.8, 6.4 #creating our own function findQuartile=function(x,n){ stopifnot(n>= 1 && n<=3) #sorting array for(i in 1:(checklen(x)-1)){ for(j in (i+1):(checklen(x))){ if(x[j,1]<x[i,1]){ t=x[i,1] x[i,1]=x[j,1] x[j,1]=t } } } #first quartile if(n==1){ findQuartile=(x[as.integer((checklen(x)+1)/4),1]) findQuartile } #third quartile else if(n==3){ findQuartile=(x[as.integer(3\*(checklen(x)+1)/4),1]) findQuartile } #median else if(n==2){ findQuartile=(x[as.integer((checklen(x)+1)/2),1]) findQuartile } }

#——–finding variance of column mpg and Sepal.Length var(x[,1]) var(y[,1]) #in build function, ans=36.3241 #in build function, ans=0.685693

findVar=function(x){ sum=0 i=1 mean=findMean(x) for(i in 1:checklen(x)){ sum=sum+((x[i,1]-mean)^2)/(checklen(x)-1) } findVar=sum findVar }

#——–finding standard deviation of column mpg and Sepal.Length sd(x[,1]) sd(y[,1]) #in build function, ans=6.026948 for mtcars #ans=0.8280661 for iris findSd=function(x){ findSd=sqrt(findVar(x)) findSd }

#——–finding covariance of column mpg and Sepal.Length cov(x[,1],x[,2]) #-9.172379 cov(y[,1],y[,2]) #-0.042434 findCov=function(x){ sum=0 for(i in 1:(checklen(x))){ sum=sum+(((x[i,1]-findMean(x,1))\*(x[i,2]-findMean(x,2)))/(checklen(x)-1)) } findCov=sum findCov }

#——–ii. Describe data using summary in R summary(x) summary(y)

#——–iii. Make use of histogram, bar chart, pie chart and boxplot to #illustrate data on different variables.

x=mtcars; #for(i in 1:11){ hist(x[,1], xlab=“mpg”, ylab=“frequency”, main=(“mpg of mtcars”), col=“lavender” ) #} hist(x[,2], xlab=“cyl”, ylab=“frequency”, main=(“cyl of mtcars”), col=“lavender” )

hist(x[,3], xlab=“disp”, ylab=“frequency”, main=(“disp of mtcars”), col=“lavender” )

hist(x[,4], xlab=“hp”, ylab=“frequency”, main=(“hp of mtcars”), col=“lavender” )

hist(x[,5], xlab=“drat”, ylab=“frequency”, main=(“drat of mtcars”), col=“lavender” )

hist(x[,6], xlab=“wt”, ylab=“frequency”, main=(“wt of mtcars”), col=“lavender” )

hist(x[,7], xlab=“qsec”, ylab=“frequency”, main=(“qsec of mtcars”), col=“lavender” )

hist(x[,8], xlab=“vs”, ylab=“frequency”, main=(“vs of mtcars”), col=“lavender” )

hist(x[,9], xlab=“am”, ylab=“frequency”, main=(“am of mtcars”), col=“lavender” )

hist(x[,10], xlab=“gear”, ylab=“frequency”, main=(“gear of mtcars”), col=“lavender” )

hist(x[,11], xlab=“carb”, ylab=“frequency”, main=(“carb of mtcars”), col=“lavender” )

#pie chart on number of cylinders pie(table(x[,2]), main=“Number of Cylinders” )

boxplot(mtcars$mpg, main=‘Distribution of mpg values’, ylab=‘mpg’, col=‘lavender’, border=‘black’)

barplot( table(x[,2]), xlab=“number of cylinders”, ylab=“number of cars”, main=“cylinders in mtcars” )

#hist, pie and bar chart for iris y=iris;

hist(iris$Sepal.Length, col=‘steelblue’, main=‘Histogram’, xlab=‘Length’, ylab=‘Frequency’)

hist(iris$Sepal.Width, col=‘steelblue’, main=‘Histogram’, xlab=‘Width’, ylab=‘Frequency’)

hist(iris$Petal.Length, col=‘steelblue’, main=‘Histogram’, xlab=‘Length’, ylab=‘Frequency’)

hist(iris$Petal.Width, col=‘steelblue’, main=‘Histogram’, xlab=‘Width’, ylab=‘Frequency’)

#hist for species throws error as it requres numeric inputs #pie chart on number of species pie(table(y[,5]), main=“Species of Irises” )

boxplot(mtcars$mpg, main=‘Distribution of mpg values’, ylab=‘mpg’, col=‘lavender’, border=‘black’)

#barplot for iris table barplot( table(y[,5]), #gives frequency of species xlab=“number of Species”, ylab=“frequency”, main=“frequency of species” )

#——–2. Make use of plot, lines and legend functions in R to plot the #graph of PMF/PDFs and CDFs of following statistical distributions #corresponding to various parameter values on the same x-axis.

#i. binomial #dbinom x = 1:80

# size = 80, prob = 0.2

plot(dbinom(x, size = 80, prob = 0.2), type = “l”, main = “Binomial probability function”, ylab = “P(X = x)”, xlab = “Number of successes”, col=“black”, lty=1, lwd = 3, )

# size = 80, prob = 0.3

lines(dbinom(x, size = 80, prob = 0.3), type = “l”, lty=2, lwd = 2, col = “blue”)

# size = 80, prob = 0.4

lines(dbinom(x, size = 80, prob = 0.4), type = “l”, lty = 3, lwd= 1, col = “red”)

# Add a legend

legend(“topright”, legend = c(“80 0.2”, “80 0.3”, “80 0.4”), title = “size prob”, lty=c(1,2,3), lwd=c(3,2,1), col=c(“black”,“blue”,“red”) ) #pbinom

# size = 80, prob = 0.2

plot(pbinom(x, size = 80, prob = 0.2), type = “l”, lty=1, lwd = 3, main = “Binomial distribution function”, xlab = “Number of successes”, ylab = “F(x)”, col=“black” )

# size = 80, prob = 0.3

lines(pbinom(x, size = 80, prob = 0.3), type = “l”, lwd = 2, lty = 2, col=“red” )

# size = 80, prob = 0.4

lines(pbinom(x, size = 80, prob = 0.4), type = “l”, lty=3, lwd = 1, col = “blue”)

# Add a legend

legend(“bottomright”, legend = c(“80 0.2”, “80 0.3”, “80 0.4”), title = “size prob”, lty=c(1,2,3), lwd=c(3,2,1), col=c(“black”,“red”,“blue”) )

#ii. poisson x=0:50 #using dpois to find pmf lambda=5 plot(x,dpois(x,lambda), type=‘l’, main=“Poisson Probability Mass Function”, ylab=“P(X=x)”, xlab=(“Number of events”), col=‘black’, lty=1, lwd=3 ) lambda=10 lines(dpois(x,lambda), type=‘l’, col=‘red’, lty=2, lwd=2 ) lambda=20 lines(x,dpois(x,lambda), type=‘l’, col=‘green’, lty=3, lwd=1 )

legend(“topright”, legend = c(“5”,“10”,“20”), title=“lambda”, col=c(“black”,“red”,“green”), lty=c(1,2,3), lwd=c(3,2,1), ) #ppois lambda=5 plot(x,ppois(x,lambda), type=‘l’, main=“Poisson CDF”, ylab=“F(x)”, xlab=(“Number of events”), col=‘black’, lty=1, lwd=3 ) lambda=10 lines(ppois(x,lambda), type=‘l’, col=‘red’, lty=2, lwd=2 ) lambda=20 lines(x,ppois(x,lambda), type=‘l’, col=‘green’, lty=3, lwd=1 )

legend(“topright”, legend = c(“5”,“10”,“20”), title=“lambda”, col=c(“black”,“red”,“green”), lty=c(1,2,3), lwd=c(3,2,1), )

#iii. Uniform x <- seq(-4, 4, length=100) plot(x, dunif(x,min=-3, max=3), type = ‘l’, lty=1, lwd = 2, ylim = c(0, .3), col=‘blue’, xlab=‘x’, ylab=‘Probability’, main=‘Uniform Distribution Plot’) lines(x,dunif(x, min=-2, max=2), type=‘l’, col=‘green’, lty=2, lwd=1 )

legend(“topright”, legend = c(“-3 to 3”,“-2 to 2”), title=“min max values”, col=c(“blue”,“green”), lty=c(1,2), lwd=c(2,1), cex=0.6 ) #punif plot(x, punif(x,min=-3, max=3), type = ‘l’, lty=1, lwd = 2, ylim = c(0, 1.5), col=‘blue’, xlab=‘x’, ylab=‘Probability’, main=‘Uniform Distribution Plot’) lines(x,punif(x, min=-2, max=2), type=‘l’, col=‘green’, lty=2, lwd=1 )

legend(“topright”, legend = c(“-3 to 3”,“-2 to 2”), title=“min max values”, col=c(“blue”,“green”), lty=c(1,2), lwd=c(2,1), cex=0.6 ) #iv. Exponential Function #dexp x=seq(0,8,0.1) lambda=0.5 plot(x,dexp(x,lambda),typ=“l”, ylab=“P(x)”, xlab=“x”, col=‘black’, lty=1, lwd=3 ) lambda=1 lines(x,dexp(x,lambda), type=‘l’, col=‘red’, lty=2, lwd=2 ) lambda=2 lines(x,dexp(x,lambda), type=‘l’, col=‘green’, lty=3, lwd=1 )

legend(“topright”, legend = c(“0.5”,“1”,“2”), title=“lambda”, col=c(“black”,“red”,“green”), lty=c(1,2,3), lwd=c(3,2,1), )

#pexp x=seq(0,8,0.1) lambda=0.5 plot(x,pexp(x,lambda),typ=“l”, ylab=“F(x)”, xlab=“x”, col=‘black’, lty=1, lwd=3 ) lambda=1 lines(x,pexp(x,lambda), type=‘l’, col=‘red’, lty=2, lwd=2 ) lambda=2 lines(x,pexp(x,lambda), type=‘l’, col=‘green’, lty=3, lwd=1 )

legend(“topright”, legend = c(“0.5”,“1”,“2”), title=“lambda”, col=c(“black”,“red”,“green”), lty=c(1,2,3), lwd=c(3,2,1), )

#v. gamma function #dgamma x=seq(0,2,0.01) curve(dgamma(x, shape=2, rate=1), from=0, to=5, ylim=c(0,1), col=‘black’)

curve(dgamma(x, shape=3, rate=2), from=0, to=7, col=‘red’, add=TRUE)

curve(dgamma(x, shape=4, rate=3),  
from=0, to=10, col=‘blue’, add=TRUE) legend(“topright”, legend=c(“shape 2, scale 1”,“shape 3, scale 2”,“shape 4, scale 3”), text.col=c(“black”,“red”,“blue”), cex=0.5 #for setting the text size ) #pgamma x=seq(0,2,0.01) curve(pgamma(x, shape=2, rate=1), from=0, to=5, ylim=c(0,1), col=‘black’)

curve(pgamma(x, shape=3, rate=2), from=0, to=7, col=‘red’, add=TRUE)

curve(pgamma(x, shape=4, rate=3),  
from=0, to=10, col=‘blue’, add=TRUE) legend(“bottomright”, legend=c(“shape 2, scale 1”,“shape 3, scale 2”,“shape 4, scale 3”), text.col=c(“black”,“red”,“blue”), cex=0.5 #for setting the text size )

#vi. normal distribution #dnorm x <- seq(-4, 8, 0.1)

#mean=0, sd=1 plot(x, dnorm(x, mean = 0, sd = 1), type = “l”, ylim = c(0, 0.6), xlab=“x”, ylab = “P(X==x)”, lwd = 2, col = “red”) #mean=3, sd=1 lines(x,dnorm(x, mean=3, sd=1), col=“blue”, lty=1, lwd=2 ) #legend legend(“topright”, legend = c(“0 1”, “3 1”), col = c(“red”, “blue”), title = expression(paste(mu,” “,sigma)), title.adj = 0.9, lty = 1, lwd = 2)

#mean same, sd different # Mean 1, sd 1 plot(x, dnorm(x, mean = 1, sd = 1), type = “l”, ylim = c(0, 1), ylab = “P(X==x)”, lwd = 2, col = “red”) # Mean 1, sd 0.5 lines(x, dnorm(x, mean = 1, sd = 0.5), col = “blue”, lty = 1, lwd = 2)

# Adding a legend

legend(“topright”, legend = c(“1 1”, “1 0.5”), col = c(“red”, “blue”), title = expression(paste(mu, ” “, sigma)), title.adj = 0.75, lty = 1, lwd = 2)

#pnorm # Same sd, different mean # Mean 0, sd 1 plot(x, pnorm(x, mean = 0, sd = 1), type = “l”, ylim = c(0, 1), ylab = “F(x)”, lty=2, lwd = 1, col = “red”) # Mean 3, sd 1 lines(x, pnorm(x, mean = 3, sd = 1), col = “blue”, lty = 1, lwd = 2)

# Legend

legend(“topleft”, legend = c(“0 1”, “3 1”), col = c(“red”, “blue”), title = expression(paste(mu, ” “, sigma)), lty = c(2,1), lwd = c(1,2),)

# Same mean, different sd

# Mean 1, sd 1

plot(x, pnorm(x, mean = 1, sd = 1), type = “l”, ylim = c(0, 1), ylab = “F(x)”, lty=2, lwd = 1, col = “red”) # Mean 1, sd 0.5 lines(x, pnorm(x, mean = 1, sd = 0.5), col = “blue”, lty = 1, lwd = 2)

# Legend

legend(“topleft”, legend = c(“1 1”, “1 0.5”), col = c(“red”, “blue”), title = expression(paste(mu, ” “, sigma)), lty=c(2,1), lwd=c(1,2) )

#vi. normal distribution x = seq(-4, 8, 0.1)

# Same sd, different mean

# Mean 0, sd 1

plot(x, dnorm(x, mean = 0, sd = 1), type = “l”, ylim = c(0, 0.6), ylab = “P(X==x)”, lty=1, lwd = 2, col = “red”) # Mean 3, sd 1 lines(x, dnorm(x, mean = 3, sd = 1), col = “blue”, lty = 2, lwd = 1)

# Adding a legend

legend(“topright”, legend = c(“0 1”, “3 1”), col = c(“red”, “blue”), title = expression(paste(mu, ” “, sigma)), lty=c(1,2), lwd=c(2,1))

# Same mean, different standard deviation

# Mean 1, sd 1

plot(x, dnorm(x, mean = 1, sd = 1), type = “l”, ylim = c(0, 1), ylab = “P(X==x)”, lty=1, lwd = 2, col = “red”) # Mean 1, sd 0.5 lines(x, dnorm(x, mean = 1, sd = 0.5), col = “blue”, lty = 2, lwd = 1)

# Adding a legend

legend(“topright”, legend = c(“1 1”, “1 0.5”), col = c(“red”, “blue”), title = expression(paste(mu, ” “, sigma)), lty = c(1,2), lwd = c(2,1))

#pnorm # Same sd, different mean # Mean 0, sd 1 plot(x, pnorm(x, mean = 0, sd = 1), type = “l”, ylim = c(0, 1), ylab = “F(x)”, lty=1, lwd = 2, col = “red”) # Mean 3, sd 1 lines(x, pnorm(x, mean = 3, sd = 1), col = “blue”, lty = 2, lwd = 1)

# Legend

legend(“bottomright”, legend = c(“0 1”, “3 1”), col = c(“red”, “blue”), title = expression(paste(mu, ” “, sigma)), lty = c(1,2), lwd = c(2,1))

# Same mean, different sd

# Mean 1, sd 1

plot(x, pnorm(x, mean = 1, sd = 1), type = “l”, ylim = c(0, 1), ylab = “F(x)”, lty=1, lwd = 2, col = “red”) # Mean 1, sd 0.5 lines(x, pnorm(x, mean = 1, sd = 0.5), col = “blue”, lty = 2, lwd = 1)

# Legend

legend(“bottomright”, legend = c(“1 1”, “1 0.5”), col = c(“red”, “blue”), title = expression(paste(mu, ” “, sigma)), lty=c(1,2), lwd=c(2,1))

#vii. Log-normal distribution #dlnorm curve(dlnorm(x, meanlog=0, sdlog=.3), from=0, to=10, col=‘blue’) curve(dlnorm(x, meanlog=0, sdlog=.5), from=0, to=10, col=‘red’, add=TRUE) curve(dlnorm(x, meanlog=0, sdlog=1), from=0, to=10, col=‘purple’, add=TRUE) legend(“topright”, title=“sdlog”, legend=c(“0.3”,“0.5”,“1”), text.col=c(“blue”,“red”,“purple”) )

#plnorm curve(plnorm(x, meanlog=0, sdlog=.3), from=0, to=10, col=‘blue’) curve(plnorm(x, meanlog=0, sdlog=.5), from=0, to=10, col=‘red’, add=TRUE) curve(plnorm(x, meanlog=0, sdlog=1), from=0, to=10, col=‘purple’, add=TRUE) legend(“bottomright”, title=“sdlog”, legend=c(“0.3”,“0.5”,“1”), text.col=c(“blue”,“red”,“purple”) )

#viii Weibull distribution #dweibull curve(dweibull(x, shape=2, scale=1), from=0, to=5, col=‘black’)

curve(dweibull(x, shape=3, scale=2), from=0, to=7, col=‘red’, add=TRUE)

curve(dweibull(x, shape=4, scale=3),  
from=0, to=10, col=‘blue’, add=TRUE)

legend(“topright”, legend=c(“shape 2, scale 1”,“shape 3, scale 2”,“shape 4, scale 3”), text.col=c(“black”,“red”,“blue”), cex=0.5 #for setting the text size ) #pweibull curve(pweibull(x, shape=2, scale=1), from=0, to=5, col=‘black’)

curve(pweibull(x, shape=3, scale=2), from=0, to=7, col=‘red’, add=TRUE)

curve(pweibull(x, shape=4, scale=3),  
from=0, to=10, col=‘blue’, add=TRUE) legend(“bottomright”, legend=c(“shape 2, scale 1”,“shape 3, scale 2”,“shape 4, scale 3”), text.col=c(“black”,“red”,“blue”), cex=0.5 #for setting the text size )