Contents

1 LOOPS

A loop statement allows us to execute a ststement or group of statement multiple times.A loop when refered to in a sequence of statements that are repeated over and over again and perfom some operations as long as the condition is reached (True).
Items inside a loop are typicall contained within curly brackets

1.1 THE FOR () LOOP

It repeats a chunk of code multiple times for each element within an object. This statement allows us to specify that a certain opration should be repeated a fixed number of times.
Syntax for(name in vector)commands
This statement sets a variable called name equal to each element of vector in sequence .
For each value whatever commands are listed within will be perfomed . The serve to group a commands so that the are treated by r as a single command
.

Examples

1. for(r in seq(2,4))print(r*r)
   Printing a square roots of integers one to ten
   for(xin 1;10)print (sqrt(x))
   for(x in 1:10)print(round(sqrt(x)1,2))

2. for(j in 1:5)print (j^2).
   We can capture the results of our loop in a list.first we create a vector

and then we fill in its values e.g

```
n=10
x<-rep*(0,n)
for(j in 1:n){x(j)=j^2}
x
```

1.2 THE IF () STATEMENT

This statement allows to control which statement are executed .

Syntax

(a) if (condition)commands when true

    if (test-expression)statement. If the test -expression is true ,the statements get executed .but vif its FALSE ;nothing happens here test-expression can be a logical or numerical vector but e.g only the first element is taken into consideration.

(b) if (condition)command when return TRUE else command when FALSE(The else part is optional false and is only evaluated if first is false)

    This statement causes a set of command tobe invoked if condition x evaluate to TRUE ,the else part is optional and provide an alternative set of commands which are to be invoked incase the logical variable is FALSE.

```
x<-5 or x<-4
If(x>0){print("the Number")}
e.g else {print("out of range)}
```

```
m<- 3

if (m>2){y<-2*m} else {y<- 3*m}

print(y)

M=7

if (M>2) {y <-5*M} else {else y<10}

print (y)

A<-3

B<-5

if (A>B){print(A+B)} else {print(B-A)}

#counting even numbers in a vector

Z<-c(3,5,3,9,10,11,13,14,16)

count<-0

for (val in z){if (val %% z==0)}{

count=count +1}

# and if }# ends for loop

Print(count)

output
```

1.3 If...else statement

Else statement tells the interpreter to run specific lines of codes if our comparison operatoe evaluates to FALSE.

syntax

if(test-expression)statements else (statement 2).

The else part is evaluated if test-expression is false.

It is important to note that else must be in the same line as the closing braces of the if statement.

1.3.1 Nested if else

syntax

if(condition 1)command 1 else if (condition 2)command 2 else if (condition 3)command 3 else (command 4)

```
x<- -5
y<- if(x>0) 5 else 6
y
```

example

```
p<--10
if(p>0){print("positive Number")}else if (P<0)
{print("negative Number")} else {print("Zero")}
```

Output:"negative Number"

1.4 Ifelse statement

This is the vector equivalent form of the If..else statement .

syntax

ifelse(command,x,y)

where

x corresponds to value TRUE

y corresponds to value FALSE

```
aa=c(5,7,2,a)****
```

1.4.1 if-else ladder

also if-else ...if.

Allows one to execute a block of code among more than alternatives **.

syntax

```
if(test-expression){statement 1}
else (test-expression 2){statement 2} else if
(test-expression 3){statement 3} else {statement 4}
```

1.5 The while () loop

They are used to loop until a specified condition is met .

Checks a logical condition and keeps running as long as the condition is true.

Syntax

while (condition)statement

The condition is evaluated and if it evaluates to FALSE nothing more is done .If it evaluates to TRUE the statements are executed .Condition is evaluated again and the process is repeated .

```
j<-1
while (j<5){print(j);j=j+1}
```

output:1,2,3,4,5

```
a<-0
b<-1
while (b<50){ print(b) temp<-a+b  a<-b  b<- temp}
```

Output :1 1 2 3 5 8 13 21 34 **Tasks to evaluate**

(a) $\sum_{i=10}^{100}(i^3+4i^2)$

```
#print individual values

x<-(10:100)

for(i in x){res=sum(i^3 +4*i^2);

print(res)}


#prints the sum of all values.

x<-10:100

res=0

for(i in x){res=res+(i^3+4*i^2)}

print(res)
```

(b) $\sum_{i=1}^{30}\sum_{j=1}^{5}\left(\frac{i^4}{3+ij}\right)$

```
p<-1:30

q<-1:5

res<-0

for(i in p){for(j in q){res=res+[(i^4)/(10+i*j)]}}

print(res)
```

output:392251.2

```
x<-sample(1:10***)

while (x>5){print(x) break}

x<-1

while(x<5){x=x+1;

if(x==3)break;

print(x);}
```

EXAMPLE

Sample data

A_1	1	3	5	7	9	11	13	15	17	19
B_2	129	178	140	186	191	104	150	183	151	142
C_3	A	B	C	D	E	F	G	H	I	J

syntax to generate the above tablein R

```
set.seed(1,2,3)
mydata=dataframe(x1=seq(1,20,by=2),x2=sample(100,200,10,FALSE)x3=LETTER[1;10])
mydata
ifelse(condition,value if condition is true ,value if condition is false)
```

EXAMPLE

Suppose you are asked to create a binary varieble -1 or 0 based on the variable x_2 if value of a variable x_2 is greator tha 150 assign 1 or else0

```
mydata$x4=ifelse(mydata$x2>150,1,0)
mydata
```

1.5.1 Apply if else() on character variable.

If variable $X3$ contains character value $"A", "D"$,the variable $X1$ should be multiplied by 2 otherwise it should be multiplied by 3

```
mydata$y=ifelse(mydata$X3%in %c("A","D")
mydata$X1*2,mydata$X1*3)
```

EXAMPLE

```
k=100
if(k>100){print("greator than 100")} elseif (k<100){print("less tha 100")}else {
```

1.6 Break statement in while loop

Break statement will end loop abruptly once the break statement is read, the loop will be terminated.

```
t<-1
while(t<=8){if(t==4)break;print(t*t)
t=t+1}
```

Once the value of t reaches 4 the break statement terminates the loop.

1.7 Next statement in while loop

It will skip one step of the loop once the next statement is read ,the loop will skip while once.

```
x<-5
while(x<20)
{
if(x==5)
{x=x+1;next}
print(x^3)
x=x+1}
```

2 FUNCTIONS

They are used to logially break codes into simpler parts which becomes as to understand .
functions are called by another live code that sends a request to the function

to do something or return a variable.

There are two types of functions in R :

1. Built in functions:functions which are defined by say R e.g

   ```
   sum
   x<-c()
   sum(x)
   mean(x)
   ```

2. User defined Function:functions which are defined by the user to perfom specific functions

2.1 User-Defined Functions

Writing function in R is quite easy and straight forward .

syntax

```
fun.name<-function(arg 1,arg 2,...){do this}
```

It uses the reserved word function to declare a function in R .The statement within the form the body of the function .

Func-name is a variable that holds th function object.

Example

A function to raise x to power of y.

```
pow<-function(x,y){
result<-x^y
print(result)}
```

```
p<-function(x,y){

t<-x^y

print(t)

}
```

*****

You can call the following function as follows:

```
#function call

pow(5,10)
```

Example

```
pf<-function(a,b,c){

return(list(sum=a+b+c)

product=(a*b*c))

pf(6,7,8)}
```

Assignment

Write a well commented R function that obtains the real roots of a quadratic equation using the quadratic formula.
$x_1 = \frac{-b+\sqrt{b^2-4ac}}{2a}$ $x_2 = \frac{-b-\sqrt{b^2-4ac}}{2a}$ then test the function by getting thr roots of

1. $x^2 + 5x + 6 = 0$
2. $x^2 + 6x - 16 = 0$
3. $x^2 - 9x + 8 = 0$

```
roots-function(a,b,c){

x1<-((-b+sqrt(b^2-4*a*c))/2*a)
```

```
x2<-((-b+sqrt(b^2-4*a*c))/2*a)
return(list****)
roots(1,5,6)
```

```
er<-function(a,b,c){
num1<--b+sqrt(b^2-4*a*c)
num1<--b-sqrt(b^2-4*a*c)
deno<-2*a
x1<-num1/deno
x2<-num2/deno
return(list(x1=x,x2=x))
er(1,5,6)
}
```

Many are the times that you will require your function to do some processing and return back the results .This is accomplished by using the *return*() code in *R*

Syntax

```
return(expression)
```

Example Write a function which takes a numerical vector and calculate the mean of the vector.

```
#define a function
meanvec<-function(vec){
total<-sum(vec)
n<-leng(vec)
avg<-total/n
return(avg)
}
#function call
meanvec(1:10)

#define a function
mv<-function(vec){
t<-sum(vec)
n<-leng(vec)
avg<-t/n
return(avg)
}
#function call
x<-c(89,78,56)
mv(x)
mv(1:10)
```

Write a function that takes the radius of a sphere and computes the volume and surface area of the sphere

```
sphere<-function(radius){
volume<-(4/3*pi*radius^3)
surface<-(4*pi*radius^2)
return(list(volume=volume,surface=surface))}
sphere(100)
```

Task

Consider a continous function $f(x) = \begin{cases} x^2+2x+3, & \text{if } x<0; \\ \text{x+3}, & \text{if } 0 \le x < 2; \\ x^2+4x-7, & \text{if } 2 \le x \end{cases}$

write a well-commented $R\!-\!code$ that takes a vector of 100 values between -3 and 3 and returns the value of the above function.

```
f(x)<-function(x){
ifelse(x<0,x^2+2*x+3)
ifelse(x<2,x+3,
(x^2+4*x-7))}
input<-seq(-3,3,100)
f(x)(input)
```

3 ERROR ANALYSIS

There exist many approximations to error analysis but most common is linear regression analysis.Linear regression is a name of a procedure that fit a straight line in a data.
The idea is that the x value is something that the experimentor controls and the y value is the one that the experimentor measures.
The line is used to predict the value of y for a known value of x is the predictor and y is the response.
Suppose you write the equation of the line as

$$y_i = b_o + b_i x_i$$

Then for each x_i the predicted value would be

$$\bar{y} = b_0 + b_i x_i$$

But the measured value is y_i the difference is called the residual/error and is simply

$$e_i = y_i - \bar{y}$$

The assumption on the error is that they are independent identically distributed (*iid*) with mean μ and variance δ^2 i.e $e_i \sim N(\mu, \delta^2)$.The method of least square is used to choose the value of b_0 and b_1 that minimises the squares of the model errors (SSE).Mathematically this is

$$SSE = \sum_{i=1}^{n} (y_i - \bar{y}_i)^2$$

mean squared error(MSE)

$$\begin{aligned} MSE &= \frac{SSE}{n} \\ &= \frac{\sum_{i=1}^{n} (y_i - \bar{y}_i)^2}{n} \end{aligned}$$

Root mean squared error (R.M.S.E)

$$RMSE = \sqrt{MSE}$$

Fitting a simple linear regression model in R use function $lm()$

syntax

```
lm(y ~ x,data)
```

Example Write R code to perfom a regression model on th data below use the function $lm()$

x	39	65	62	90	82	75	20	98	36	78
y	47	53	58	86	62	68	65	91	51	84

```
#reading data in R
x<-c(39,...,78)
y<-c(47,...,84)
#regress y on x
```

```
model<-lm(y~x)

#extract values of b0 and b1

b0<-coef(model)[1]

b1<-coef(model)[2]

#compute the predicted values of y‘

yhat<-b0+b1x

#display the actual predicted value of y

cbind(y,yhat)

#plot x and y fitted with regression line plot(x,y)

abline(model)

#compute the value of SSE

SSE<-0

for(i in 1:length(y)){

SSE-SSE+(y[i]-y-hat(i)^2)}

print(SSE)

#mean square error(MSE)

MSE<-SSE/n,length(y)

print(MSE)

#Root mean square error(RMSE)

RMSE<_sqrt(MSE)

print(RMSE)
```

CAT

```
Hn=n(2n-1)

#seq upto 50th term

n<-1:50

hn<-(2*h^2-n)

hn
```

```
hn[hn%%2==1]
```

```
all odd nos
```

```
hn[hn%%3==0]
```

```
sum(hn[hn%%3==0])
```

exercise

Height	100	200	300	450	600	800	1000
Distance	253	337	395	451	495	534	574

From the data obtain the:

(i) coefficient

(ii) predited values

(iii) SSE

(iv) MSE

(v) Estimate distance given height=1500

4 CALCULUS

When R ia used as statistical language for numerical concept calculations ,its often useful to be able to manipulate symbols so as to avoid approximation errors.R uses the function $D()$to differentiate univariate function.

syntax

```
d(expression,"x")
```

where expression is the univariate function to be differentiated.x is what is differentiated with respect to.

example

Evaluate $\frac{d}{dx}$ of

1. $y = x^2e^2 + 7x + 10$

```
f<-expression(x^2+exp(x)+7*x+10)
D(F,"x")
```

2. $y = lnx + 10$

```
g<-expression(log(x)+10)
D(g,"x")
```

3. $y = \frac{(x^2+5x-1)}{\sqrt{x^2+2}}$

```
h<-expression((x^2+5*x-1)/(sqrt(x^2+2)))
D(h,"x")
```

4. $y = sin3x$

```
y<-expression(sin (3*x))
D(y,"x")
```

4.1 second order or higher order derivatives

In R we use the $D()$ function for higher order derivatives .To obtain a second order derivative we use the syntax $(D(Dexpression, "x"), "x")$

Example

Evaluate the $\frac{d^2y}{dx^2}$ and $\frac{d^3y}{dx^3}$ of the function $y = \frac{cosx}{x}$

```
f<-expression(cos x/x)
#2nd order
D(D(f,"x"),"x")
#3rd order
D(D(D( f,"x"),"x"),"x")
```

4.2 Differentiation

The derivative operator.

Takes the function as input ,produces a function as output.

The output function gives the slope of the input function at any point.

Algebraic algorithm for transforms e.g $x^n - nx^{n-1}$

The theory of the infinitesimal.

A sample differential operator.

```
D=function(f,dlta=0.000001){function(x),{f(x+delta)-(f(x-delta))/2*delta}}
#using D
f=function(x){x^2+2*x}
plot(f,0,10)
plot(D(f),0,10)
```

4.3 Integration

To integrate a one dimensional integral over a finite or infinite .Use R function $integrate()$ **Example** $\int_0^\infty \frac{1}{(x+1)\sqrt{x}}dx$

```
#define the integrated function
integral<-function(x){1/(x+1)*sqrt(x)}
#integrate from 0-infinity
R<-integrate(integral,lower=0,upper=inf)
R& value#output 3.141593 with absolute error 2.7*10^-3
R&abs.error #approximation error(output 2.6515e)
```

Evaluate $\int_{-1.96}^{1.96} \frac{1}{\sqrt{2\pi}} e^{\frac{-x^2}{2}} dx$

```
inte<-function(x){\/sqrt(2*pi)*exp(-x^2/2)}
```

```
Re<-integrate(inte,lower=-1.96,upper=1.96)

Re&value

Re&abs.error
```

To integrate a scalar function over a multidimensional rectangle ,use R function $adaptIntegrate$.
You need to install the R package $cubature1^{st}$,

1. Download the package file "$cubature{-}1.0zip$ via $http://cran.r.project.org/web/packages/$
2. install the package cubature in R.
   - (i) click R main package
   - (ii) Install pa** from local zip files
   - (iii) Open the download package file "cubature-1.0.zip"

   For more R packages: $http://cran.r.project.org/web/package/$.

We can use the R function $adaptIntegrate$ to compute n-fold integrals .
Example
Find the integral of $\int_0^1 \int_0^{\frac{1}{2}} \int_0^{\frac{1}{4}} \frac{3}{4}(2x_1 + x_2 + 4x_3)dx_1dx_2dx_3$

```
library(cubature)#load the package "cubature"

f<-function(x){3/4*(2*x[1]+x[2]+4*x[3])}

adaptIntegrate(f, lowerlimit=c(0,0,0)upperlimit=c(1,0.4,0.25))***
```

5 OPTIMIZATION

5.1 Roots

The most convinient function to find roots in R is polyroots.
Syntax

Polyroot(x) where x is a vector of equation coefficients starting with the coefficient of the variables with smallest degree

Examples

(a) $x^2 + 3x + 10$

(b) $3x^5 + x + 4x^3 - 7x^2 + 100$

a `x<-c(10,3,1) polyroot(x)`

b `y<-c(100,1,-7,4,0,3)polyroot(y)`

Description:Find zeros of a real or complex polynomial.
$Polyroot(z)$ Z is the vector of polynomial coeffients in decreasing order . Polyroot returns $n-1$ complex zeros of $p(x)$ using the zenkins-traub algorithm.

5.2 Linear Programming

Often you need to maximise or minimise a function subject to constrain when the function is linear and the constrains can be expressed as linear equations the problem is linear programming.
the standard form of the minimum/maximum problem in linear programming is

$$min/maxC(x) = c_1x_1 + c_2x_2 + ... + c_nx_n$$

subject to the constrain

$$\begin{aligned} d_1x_1 + ... + d_1x_k &\geq b_1 \\ d_2x_1 + ... + d_2x_1 &\leq b_2 \\ &\vdots \\ d_\mu x_1 + ... + d_\mu x_k &\geq b_\mu \end{aligned}$$

non-negative constrains

$$x_1 \geq 0, ..., x_k \geq 0$$

The idea is to find value of the decision variable$x_1, x_2, ..., x_n$ which minimises or maximises the objective function (x) subject to the constrain and non-negative conditions .

5.3 Solving Linear Programming in R

Use the package *lpsolve* which is based on the simplex method .The $lp()$ function has th nu** of parameter.

1. Objective.in (vector of coefficient of the objective function)
2. const.mat-matrix containing the coeffient of the decision variable in the left-hand side of the constrain each row corresponding to the constrain.
3. const.dir-constain inequality indicating the direction of the constraint example $" >= "," <= "," > "," < "$
4. const.rhsb -containing the constants given the right hand side of the constrains .
5. Direction- which is either "max" or "min".by default is "min"

Example

Solve the linear Programming equation

$$\begin{aligned} minc &= 5x + 8y \\ sub : x + y &\geq 2 \\ x + 2y &\geq 3 \\ x, y &\geq 0 \end{aligned}$$

```
library(ipsolve)
min.lp<-lp(objective.in=c(5,8),const.mat=matrix(c(1,1,1,2)
nrow=2,byrow=T),const.rhs=c(2,3),const.dir=c(">=","<="),direction="min")
min.lp&solution
[1]1,1
```

Example 2

$$\begin{aligned} maxc &= 10x+3y \\ sub: x+y &\geq 2 \\ x+2y &\geq 3 \\ x,y &\geq 0 \end{aligned}$$

```
library(ipsolve)
min.lp<-lp(objective.in=c(10,3),const.mat=matrix(c(1,1,1,2)
nrow=2,byrow=T),const.rhs=c(10,3),const.dir=c("<=","<="),direction="max")
max.lp&solution
[1]2,0
```

6 Pseudo Random Numbers Generator

Almost all numbers randomly generated by a computer are not purely random since they are generated using a certain function which are predetermined hence the term pseudo.
The function $sample()$ in R is an example of a pseudo random number generator therefore since it re-sample from a data vector with replacement or without replacement and has number of different forms.
Syntax

sample(n,x,replace=TRUE,prob=Pr)

n=possible outcome

x=sample size

replace=logical expression by default is FALSE (without replacement)

prob=attached prob for each outcome

e.g $sample(c(2,3,4,5,6,7,8,9)4)$

$sample(c(4,5,6,7,8,9),5,replace = T, prob = c(0,1,0,3,0**))$

1. generate a sample from tossing a coin 10 times.
2. Write a function that generates a sample of size n from rolling a fair dice.
3. The awards of grades in a mathematics test are known to be A, B, C, D, E distributed in $1:3:4:2:1$ simulate a sample of 20 students who sat for the test.
4. Generate a sample for success and fail with 90% success for 100 students $(c("success", "fail"), 100, replacement = T, ****)$

Tossing a fair coin

```
sample (c("H","T"),10,replace=T)
```

Rolling a dice

```
RD<-function(n){sample(1:6,n,replace=T)}
RD(20);table (RD(100))
```

Grading****

```
grade<-function(n){sample (c("A","B","C","D","E"),
n,replace=T,prob=c($\frac{1}{11},\frac{3}{11}
,\frac{4}{11},\frac{2}{11},\frac{1}{11}$))}
grade(20)
```

6.1 Generation of Random Variates

Random numbers are described by a distribution that's some function which specifies the probility that a random number is in some range

example

$pr(a < x < b)$ PDF is often given a probability density or by a function $Pr(x = k)$ in the discrete case R will give number drawn from different distribution using the "r" function .

in order to us them ,you only need to farmiliarilize yourself with parameters given to the function such as the mean or rate

6.1.1 Discretion Variable

1. Binomial distribution

$$f(x) = p\Big(X = x\Big) = \begin{cases} \begin{bmatrix} n \\ x \end{bmatrix} p^x(1-p)^{n-x}, & x = 0, 1, ..., n; \\ 0, & \text{otherwise} \end{cases}$$

   $x \sim Bin(n, p), n$ =sample size,p=probability attached to success.

   The function $rh.nom(x, n, p)$ generates x random number,n numbers of trials , with probability $pr$of success.

   Example

```
rbinom(2,3,0.3)
n=10,p=0.5rbinom(20,5,0.5)
rbinom(1,n,p)
rbinom(20,n,p)
```

2. Poisson distribution

$f(x) = \begin{cases} \frac{e^{-\lambda}\lambda^x}{x!}, & x = 0, 1, ..., n; \\ 0, & \text{otherwise} \end{cases}$ To generate random numbers following $x \sim p(\lambda)$ use $rpois(n, lambda) erpois(n, \lambda)$ where n is a number generated ,λ is the rate parameter e.g$n = 100, \lambda = 5$

e.g $rpois(100, \lambda)$

Continous Variable

distribution	R-Name	Aurgument
Beta	Beta	slope1,slope2
Chi-square	Chisq	df(d.of freedom ***)
uniform	unit	max/min
exponential	exp	rate
gamma	gamma	slope,rate
normal	norm	mean,sd
t-distribution	t	df

To generate random number you prefix each name with "r"

Example

Generate 100 standard norm random variables

$$rnorm(100)$$
$$rnorm(100, mean = 10, sd = 6)$$
$$rchisq(10, df = 4)$$
$$rt(100, df = 20)$$
$$rexp(100, rate = 0.5)$$
$$runif(100)$$
$$runif(100, min = 10, max = 20)$$
$$rbeta(10, 5, 2)$$
$$rgamma(10, 3, 2)$$

Exercise

Consider the model $y_1 = \beta_0 + \beta_1 x + e_i$ where $x_i \sim N(4, 0.1), \beta_0 = 1.4, \beta_1 = 3.8, e_i \sim N(0, 0.5)$ write a well commented $R - code$ that genarates 1000 values of y.

6.2 Probability Distribution

In R use the prefix $'d'$ for the density of mass function and $'p'$ for the CDF or $'q'$ for the percentile function.

Syntax

$d.name(x...)$ -pdf or pmf at x

$p.name(q...)$-CDF at q.

$q.name(p_1...)$-p^{th}percentile/quantile

Examples

$dbinom(3, size = 10, prob = 0.25)$-$p(x = 3)$

$dpois(0:2; lambda = 4)$-$p(x = 0); p(x = 1)p(x = 2)$ for x ~function

$qnorm(0.75 mean = 10, sd = 2)$- 3^{rd} quantile $x \sim N(10, 2)$

$qt(0.95, df = 20)$-95^{th}percentile of $t(20)$

$pois(0:2, 4) - CDF$

1. Example 1

   10 jobs arrive every 15 seconds on *** less than or equal to 30 seconds $(o.5min)$ $p(T \leq 0.5)$

   solution

   ```
   p(T <=0.5)
   >p exp(0.5,4)
   [1]0.8646647
   ```

2. Example 2

   Accidents in a factory occur with a poisson distribution at an average of 4 per week calculate the probability of more than 5 accident in any one week

   ```
   p(x>5)=1-p(x<=5)
   >1-ppois(5,lambda=4)
   ```

3. Example 3

   A biased coin is tossed 6 times .the probability of heads on any toss is 0.3.Let x denote the number of heads that come up.Compute

   (i)

$$\begin{aligned} p(1 < x < 5) &= p(x < 5) - P(x < 1) \\ &= p(x = 2) + p(x = 3) + p(x = 4) \end{aligned}$$

```
sum dbinom(2,size=6,prob=0.3)
           (3,size=6,prob=0.3)
           (4,size=6,prob=0.3)
```

or

```
>pbinom(4,6,0.3)-pbinom(1,6,0.3)
>pbinom(4,6,0.3)-pbinom(2,6,0.3)
```

(ii) $p(x = 2)$

```
>dbinom(2,size=6,prob=0.3)
```

7 Simmulation Application

7.1 Monte Carlo Integration

Suppose you wish to calculate $I = \int_a^b g(x)dx$ but the antiderivative of $g(x)$ cannot be found in closed form monte carlo integration can be used to approximation the value of I. Suppose you generate is independent uniform random variable $x_1, x_2, ..., x_n$ the interval $[a, b]$ to compute

$$I = (b-a)\frac{1}{n}\sum_{i=1}^{n} g(x_i)$$

By the raw of large number as n increases without bound

$$I_n - > (b-a)E[g(x)]$$

By mathematical expectation

$$\begin{aligned} E[g(x)] &= \int_a^b g(x)\frac{1}{b-a}dx \\ &= \frac{1}{b-a}I \end{aligned}$$

Example

Computethe definite integral $\int_0^{\frac{\pi}{2}} 4sin(2x)exp(-x^2)dx$ .A monte carlo estimate would be given by $(using 1000,000 *** observe)$ Monte Carlo Integration

```
    >u<-runif(1000,000,min=0,max= pi/2)
>I=pi/2*mean(4*sin(2*u)exp(-u^2))
>I***
```

Ordinary Integration

```
fnx<-function(x){4*sin(2*x)*exp(-x^2)}
res=integrate(fnx,lower=0,upper=pi/2)
res&value
```

7.2 Central Limit Theorem

The theorem states that if $x_1, ..., x_n$ is random sample from a population with mean μ,standard deviation δ and π is the sample average mean then the distribution of π is approximately normal with mean μ and standard deviation $\frac{\delta}{n} = \frac{\delta}{\sqrt{n}}$.For a large $n \longrightarrow \infty$ the central limit tendency is approximate standard deviation random variable z i.e

$$Z \sim \frac{\bar{x}-\mu}{\frac{\delta}{\sqrt{n}}}$$

$\bar{x}$-sample mean

μ population mean

δ population standard deviation

n-sample size For a binomial distribution the central limit theorem transtates into saying that if S_n has a binomial distribution with parameters n and p then

$$\frac{\delta_n - np}{\sqrt{nPq}} \sim Z$$

$Z \sim N(0,1)$

Example

Given the normal data with mean 100 and standard deviation 16 compute the central limit theorem values of 200 sample each of size 100 from the normal data.Hence generate a histogram

Vector to hold central limit theorem values.

```
>r<-c()
>mu=100,sigma=16,n=100
>for(i in 1:200){
x=rnorm(n,mean=\mu,sd=sigma)
r[i]=(mean(x)-mu)/sigma/sqrt(n)}
>hist(r,prob=T)
>lines(density(r))
```

Binomial distribution example

Consider a binomial data of size 10 and probability of success 0.25.compute the central limit theorem values of 100 samples and plot a histogram.

```
>results=c()
>n=10,p=0.25
>for(k in 1:100){
```

```
s=rbinom(1,n,p)
results[k]=(s-n*p)/sqrt(n+p*(1-p))}
>hist(results,prob=T)
>lines(density,(results))
```

7.3 Confidence Interval

Statistical theory is based on known sample distribution of some statistics such as a sample mean.This allows to make probability state about the value of the parameters such as 95% certain that the parameters is in some range of values .

This range is reffered to as confidence interval for the parameter and can be specified under different confidence levels such as 95%, 99.9% etc.

The α is given by $\alpha = 1 - 0.95$ or 95% c.level.The confidence interval for the population $\mu$is given by $C.I = \bar{x} \pm z_{\frac{\alpha}{2}} SE$ where $\bar{x}$ is sample mean.

SEstandard error the mean

$$SE = \frac{\delta}{\sqrt{n}}$$

n =sample size

δ =population standard deviation

z =critical value at $1 - \frac{\alpha}{2}$

Example

Assume the population standard deviation δ of the students height is 9.48.Find the C.I of the population mean at 95% C.I using the data below $1.54, 1.78, 1.43, 1.30, 1.56, 1.89, 2.0$

$$C.I = \bar{x} \pm z_{\frac{\alpha}{2}} SE$$

Read data in R

```
x<-c(1.54,-1.52)//sample size
n<-length(x);n  //sample mean
x_bar<-mean(x)l;x_bar//standard error of the mean
>sigma=9.48;sigma
SE=sigma /sqrt (n);SE //z-value
>alpha=1-0.95;alpha
>z-value=qnorm(1-alpha/2)//confidence interval
>C.I=x-bar+c(-z_value*SE,Z_value*SE)
>C.I=4.17-4.179657,7.571657
```

Note:

For the caseof know standard deviation use z distribution while if this standard deviation is not known use the t-distribution

$$\begin{aligned} C.I &= \bar{x} \pm t_{1-\frac{\alpha}{2}} SE \\ SE &= \frac{s}{\sqrt{n}} \\ S &= \sqrt{\frac{E(x-\bar{x})^2}{n-1}} \end{aligned}$$

Example with unknown standard deviation

Suppose a persons height after measuring himself 10 times on a regular basis .Finds his heights to be

$$175, 176, 173, 175, 174, 173, 173, 176, 173, 179$$

Find the central level for the height at 99.9% confidence level.

```
Y<-c(175,...,179);4//sample size
n<-length(y);n //sample standard deviation
s<-sd(y):s //standard error of mean
```

```
SE<-S/sqrt(n);SE //+.value

alpha<-1.0.999:alpha

t-value=qt(1-alpha/2,df=n-1)// sample mean

y-bar z-mean(y);y_bar  //confidence interval

C.I<-Y-bar+c(-t-value*SE,t-value*SE)****
```

8 Hyppothesis Testing

Definition of terminologies

(i) Hypothesis is a claim or conjucture or guess about the distribution of population parameter.

(ii) Null hypothesis is the hypothesis that is iniatially assumed to be true .It is denoted using H_0

(iii) Alternative hypothesis is the hypo.that negates/opposes the H_0 Hyp it is denoted as H_A or H_1.

(iv) Rejection/critical region .This is the range of values of the test statistic at which null-hypothesis is rejected.

(v) Level of significance -This is the probability of rejecting a null hypothesis by the test when it is really true denoted by α.This value is predetermined by the test of hypothesis.

(vi) Test statistic
T.S=(sample statistic -null hypothesisof parameter)/standard deviation of the sample statistic

(vii) The power of a statistic test -This is the probability of rejecting H_0 when its false

8.1 Types of test

The alternative hypothesis determine the type of test to use .There are two types of hypothesis test .

1. One-sided /one tailed test .

   This test represents one-side of the parameter space around some value say θ_0.Thus the null and alternative hypothesis is tested as $H_0 : \theta = \theta_0, H_1 : \theta > \theta_0$

   DIAGRAM.

2. Two sided tailed test

   It represent the two sides of the parameter space θ_0. Thus the null and alternative hypothesis are defined as ****

9 P-Value

This is the lowest level of sig** which the observed value of the test statistics is sig i.e one rejects null hypothesis

9.1 Reject H_0 if P-Value is less than α else do not reject H_0

9.1.1 Steps of testing hypothesis

1. Formulate the null and the alternative hypothesis.

2. Determine the sign level α

3. Find the P-Value /test statistic

4. Make a decision to reject/not to reject the hypothesis

5. State the final conclusion

9.2 Test for Population Mean

Testing a claim on population mean when δ is known.
To test this claim use the $Z - distribution$.
This hypothesis tested is $H_0 : \mu = \mu_0$versers $H_1 : \mu \neq \mu_0$
This is the test statistic $Z = \frac{\bar{x}-\mu_0}{\frac{\delta}{\sqrt{n}}}$
The critical values of $Z - test$ are

$$0.10 - 1.282, 0.05 - 1.645, 0.025 - 1.960, 0.01 - 2.326$$

Example

A company that produces snack foods(biscuits) uses a machine to package $454g$ packets of biscuits .Assume that the weight are normally distribution with mean of $454g$ and standard deviation of $7.8g$ test whether the machine is working properly ,given the weights of 10 packets to be $465, 454, 442, 468, 456, 447, 449, 438, 449, 446$.At 5% level of sig** does the sample provide sufficient evidence to conclude that the packaging machine is not working properly.

1. Formulate $H_0 : \mu = 454$ vs $H_1 : \mu \neq 454g$.
2. Determine level of sign .Reject H_0 if $Z^* < Z_{\frac{\alpha}{2}}$(critical value).
3. Calculate the test statistic

$$\begin{aligned} Z^* &= \frac{\bar{x}-\mu_0}{\frac{\delta}{\sqrt{n}}} \\ Z^* &= \frac{451.4-454}{\frac{7.8}{\sqrt{10}}} \\ &= -1.054 \end{aligned}$$

4. Decision making

$$\begin{aligned}|Z^*| &= 1.054\\ Z_{\frac{\alpha}{2}} &= Z_{0.025}\\ &= 1.960\end{aligned}$$

   reject H_0 since $|Z^*| < Z_{\frac{\alpha}{2}}$.Fail to reject since $1.960 > 1.054$.

5. Conclusion :The mean weight of packet of biscuits packaged by***

```
x<-c(465,...,446)
x-bar<-mean(x);x-bar
mu<-454;mu
sigma=7.8;sigma
n=length(x);n
se<-Sigma/sqrt(n):se
z-value=(x-bar-mu)/se;z-value
Pval=2*Pnorm(z-value),Pval
Pval=2*Pnorm(abs(zvalue)lowertail=FALSE)
```

9.3 Testing a claim which δ is unknown

You use one-sample t-test to test a claim about a population mean which $\delta$is unknown .The hypothesis to test $H_0 : \mu = \mu_0$ verses $H_1 : \mu \neq \mu_0$ or $\mu > \mu_0$ or $\mu < \mu_0$.The test statistic is $t_0 = (\bar{x} - \mu_0)/s/\sqrt{n}$ $df = n - 1$ where s is a sample standard deviation $s = \sqrt{\frac{\sum(x-\bar{x})^2}{n-1}}$.
Reject H_0 when $|t^*| > t_{\alpha_{(n-1)}}$ or $|t^*| > t_{\frac{\alpha}{2}_{(n-1)}}$ when $t_{\alpha_{(n-1)}}$ or $t_{\frac{\alpha}{2}_{(n-1)}}$ are critical value from the standard deviation table and $n - 1$ d.f**

Examples

The figures below shows the amount of coffee in grams filled by a machine in six randomly packed jars $15.0, 15.9, 16.3, 15.7, 15.7, 15.9$.At 5% levelof sign is true mean amount to coffee filled in a jar less than 16 grams .

1. formulate $H_0 v H_1 : H_0 \mu = 16 H_1 : \mu < 16$
2. Decision ratio eject H_0 at $\alpha = 0.05$ as $t* < t = -2$.
3. test statistics $t = (\bar{x} - \mu_0)/$s)un**$x = 15.76, n = 6, s = 0.4279, n = lenth(x); n, s = sqrt(sum(x - mean(x))^2)/n - 1); s$
4. fail to reject H_0 since $t* = -1.4535 > t^{(s)}_{0.05} = 2.015$
5. Conclusion the true amount to coffee fills a jar is less than 16 grams

```
p-val=pb(t,n-1);pval#Reject Ho if value <0.2
fail to reject Ho
```

9.4 sample /independent sample test

Thistest determines whether there is a ststistically sign differentiate between two means in two unrelated gps the hypothesis to test is

$$H_0 : \mu_1 = \mu_2 vs H_1 : \mu_1 \neq \mu_2$$

where $\mu_1$is pop mean of gp1 and ****
$S_{\pi_1 - \pi_2}$ where $S_{\pi_1 - \pi_2} = \sqrt{\left[\frac{S_1^2 + S_2^2}{n_1 + n_2 - 2}\right]\left(\frac{1}{n_1} + \frac{1}{n_2}\right)}$
S_i if $i = 1, 2$ sample variance $n : j \; j = 1, 2$ sample size π_n;sample mean. Assumption of a test is that both gps have equal pop.var.

example :consider the data behaviour for 2 gps recovery time 4 paties*** after taking a placebo and drug.

drug	15	10	13	7	9	8	21	9	10	8
placebo	15	15	12	8	14	7	16	10	15	12

Test whether the patis recovery time under the placebo and drug is equal at $\alpha = 0.05$

solution

$H_0 : \mu_1 = \mu_2 vs H_1 : \mu_1 \neq \mu_2$

```
drug<-c(15,...,12):drug.placebo <-c(15..12):t.test(drug,placebo,var.equal=TRUE)
decision fail to reject $H_{0}$ since P-value=0.6005>0.05
```

conclusion:there is no difference between a placebo and a drug.

9.5 Paired method /repeated sample t-test

This test is used to same sample of subjects measured on two different occasions that pre/before or past/after.

The hypothesis for the test $H_0 : \mu_0 = 0 vs H_1 : \mu_0 \neq 0$

The test is $t = \frac{s}{s_B}$ where $S_B = \frac{s_B}{\sqrt{n}} \bar{D} = \frac{\sum D}{n}, D = \mu_a - \mu_b S_D = \frac{s^2}{n-1}$ **Example**

The following grades were awarded to applications in two differents occassions by the grades.

grade 1	3	0	3	2	5	5	5	4	4	5
grade 2	2	1	4	1	4	3	3	2	3	5

is there a different between the grades ?Test at $\alpha = 0.05$

```
g1<-c(3,...,5)
g2<-c(2,...,5)
```

An investor assumes that the mean of daily returns of a stock since inception is greater than 10 .The average percentage 30 days daily re*** sample is 9.9 Assume the pop.sd is $00H$. can we reject the null hypothesis at 0.05 sign level?

```
xbar=9.9      z.alpha=qnorm(1-alpha)
mu=10         z.alpha
sig=1.1
n=30          pnorm(z)-lower tailtest
z-(xbar-mu)/(sig(***)) pnorm(z,lowertail=false)
*****
```

10 references