**Exercise-01**

**Arithmetic operation**

CODE:

#Arithmetic operation

a<-10

b<-20

c<-a+b

print(c)

[1] 30

c<-a-b

print(c)

[1] -10

c<-a\*b

print(c)

[1] 200

c<-a/b

print(c)

[1] 0.5

c<-a%%b

print(c)

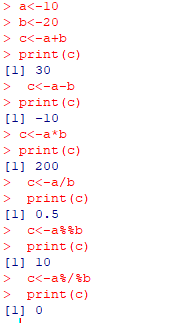
[1] 10

c<-a%/%b

print(c)

[1] 0

OUTPUT:



**Exercise-02**

**Matrix operation**

CODE:

m1<-matrix(c(1:9),nrow=3)

print(m1)

[,1] [,2] [,3]

[1,] 1 4 7

[2,] 2 5 8

[3,] 3 6 9

m2<-matrix(c(10:18),ncol=3)

print(m2)

[,1] [,2] [,3]

[1,] 10 13 16

[2,] 11 14 17

[3,] 12 15 18

m3<-m1+m2

print(m3)

[,1] [,2] [,3]

[1,] 11 17 23

[2,] 13 19 25

[3,] 15 21 27

m3<-m1-m2

print(m3)

[,1] [,2] [,3]

[1,] -9 -9 -9

[2,] -9 -9 -9

[3,] -9 -9 -9

m3<-m1\*m2

print(m3)

[,1] [,2] [,3]

[1,] 10 52 112

[2,] 22 70 136

[3,] 36 90 162

m3<-m1/m2

print(m3)

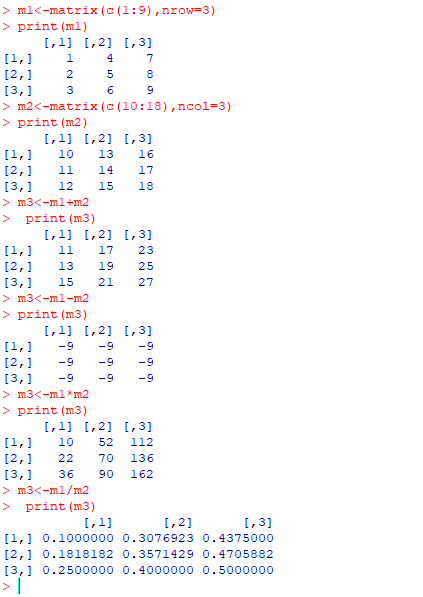
[,1] [,2] [,3]

[1,] 0.1000000 0.3076923 0.4375000

[2,] 0.1818182 0.3571429 0.4705882

[3,] 0.2500000 0.4000000 0.5000000

OUTPUT:



**Exercise – 3**

**Arithmetic operation in vector**

CODE :

a<-c(1:7)

b<-c(11,21,27,11,23,56,71)

print(a)

[1] 1 2 3 4 5 6 7

print(b)

[1] 11 21 27 11 23 56 71

result<-a+b

print(result)

[1] 12 23 30 15 28 62 78

result<-a-b

print(result)

[1] -10 -19 -24 -7 -18 -50 -64

result<-a\*b

print(result)

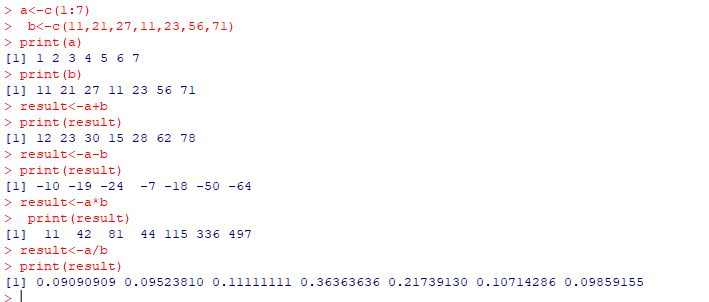
[1] 11 42 81 44 115 336 497

result<-a/b

print(result)

[1] 0.09090909 0.09523810 0.11111111 0.36363636 0.21739130 0.10714286 0.09859155

OUTPUT



**Exercise- 04**

**ODD or EVEN**

CODE:

num<-7

if(num%%2==0)

{

print("EVEN")

} else {

print("ODD")

}

[1] "ODD"

num<-20

if(num%%2==0)

{

print("EVEN")

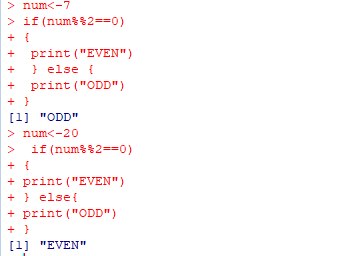
} else{

print("ODD")

}

[1] "EVEN"

OUTPUT:



**Exercise- 5**

**Iterative statement**

CODE :

#using for loop

for(i in 1:10) {

print(i)

}

[1] 1

[1] 2

[1] 3

[1] 4

[1] 5

[1] 6

[1] 7

[1] 8

[1] 9

[1] 10

#using while loop

i<-1

while(i<6){

print(i)

i<-i+1

}

[1] 1

[1] 2

[1] 3

[1] 4

[1] 5

#using repeat loop

i<-1

repeat {

print(i)

i<-i+1

if(i 7){

break

}

}

[1] 1

[1] 2

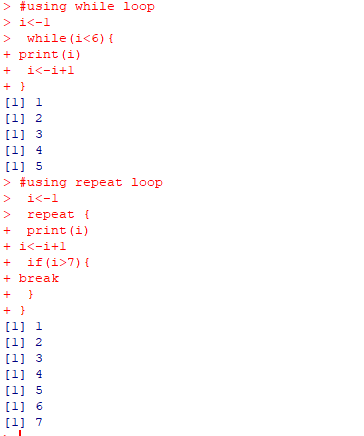
[1] 3

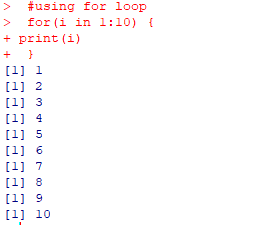
[1] 4

[1] 5

[1] 6

[1] 7

OUTPUT:



**Exercise-6**

**Factorial of a number**

CODE:

fact<-1

x<-1

while(x<=6)

{

fact=fact\*x

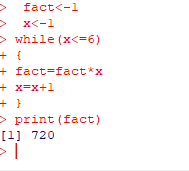
x=x+1

}

print(fact)

[1] 720

OUTPUT:



**Exercise- 7**

**Sum of the digits of a number**

CODE:

num<-456

sum<-0

while(num 0) {

y<-num%%10

sum

sum<-sum+y

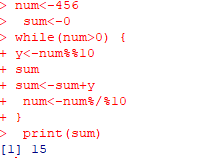
num<-num%/%10

}

print(sum)

[1] 15

OUTPUT:



**Exercise -8**

**Check for Armstrong number**

CODE :

num<-153

og<-num

sum<-0

while(num 0)

{

y<-num%%10

sum<-sum+(y^3)

num<-num%/%10

}

if(sum==og)

{

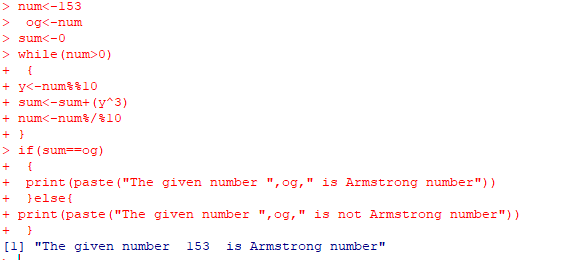
print(paste("The given number ",og," is Armstrong number"))

}else{

print(paste("The given number ",og," is not Armstrong number"))

}

OUTPUT:



**Exercise-9**

**Sum of Series**

CODE:

# Function to calculate the sum of integers from 1 to n

sum\_integers <- function(n) {

sum\_value <- sum(1:n) # Sums numbers from 1 to n

return(sum\_value)

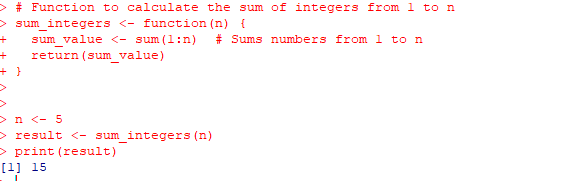
}

n <- 5

result <- sum\_integers(n)

print(result)

OUTPUT :



**Exercise - 10**

**Fibonacci series**

CODE:

a<-0

b<-1

for(i in 1:10){

print(a)

y<-a

y<-a+b

a<-b

b<-y

}

[1] 0

[1] 1

[1] 1

[1] 2

[1] 3

[1] 5

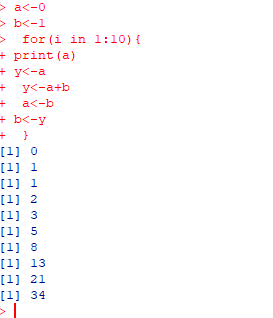
[1] 8

[1] 13

[1] 21

[1] 34

OUTPUT



**Exercise – 11**

**Statics operation**

CODE:

x<-c(23,11,45,68,92,14,65,69,27,31,99,11)

mean(x)

[1] 46.25

median(x)

[1] 38

mode(x)

[1] "numeric"

sd(x)

[1] 31.46174

sort(x)

[1] 11 11 14 23 27 31 45 65 68 69 92 99

find\_mode<-function(x){

u<-unique(x)

tab<-table(match(x,u))

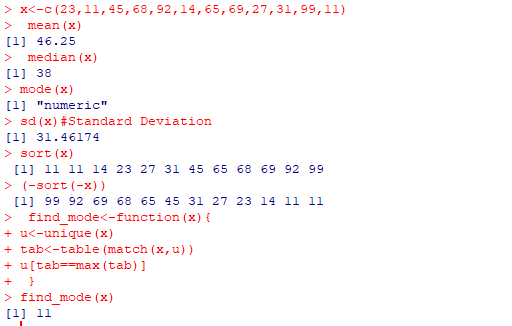
u[tab==max(tab)]

}

find\_mode(x)

[1] 11

OUTPUT :



**Exercise-12**

**Import CSV file**

CODE:

x<-read.csv("U:/cssa01/Data analysis using R/Sem1.csv")

print(x)

Subject.Code Title.of.the.paper CIA.mark Semester.mark Total.Mark STATUS

1 UCS2301 MFC 12 11 23 FAIL

2 UCS2302 WD 22 22 44 PASS

3 UCS2303 AJ 32 33 65 PASS

4 UCS2304 RR 42 42 84 PASS

5 UCS2305 MP 43 43 86 PASS

6 UCS2306 AWS 44 44 88 PASS

x<-read.csv("U:/cssa01/Data analysis using R/Sem2.csv")

print(x)

Subject.Code Title.of.the.paper CIA.mark Semester.mark Total.Mark STATUS

1 UCS2301 COA 12 11 23 FAIL

2 UCS2302 DS 22 22 44 PASS

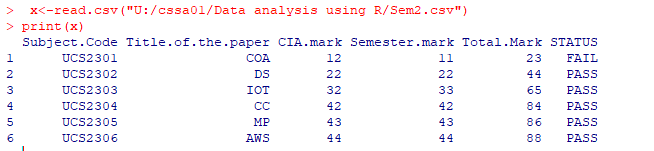
3 UCS2303 IOT 32 33 65 PASS

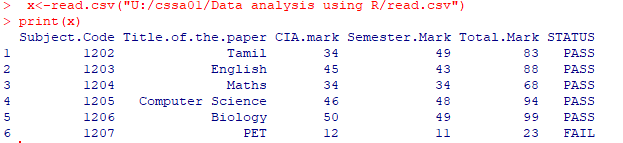
4 UCS2304 CC 42 42 84 PASS

5 UCS2305 MP 43 43 86 PASS

6 UCS2306 AWS 44 44 88 PASS

OUTPUT :





**Exercise – 13**

**Visualization in R**

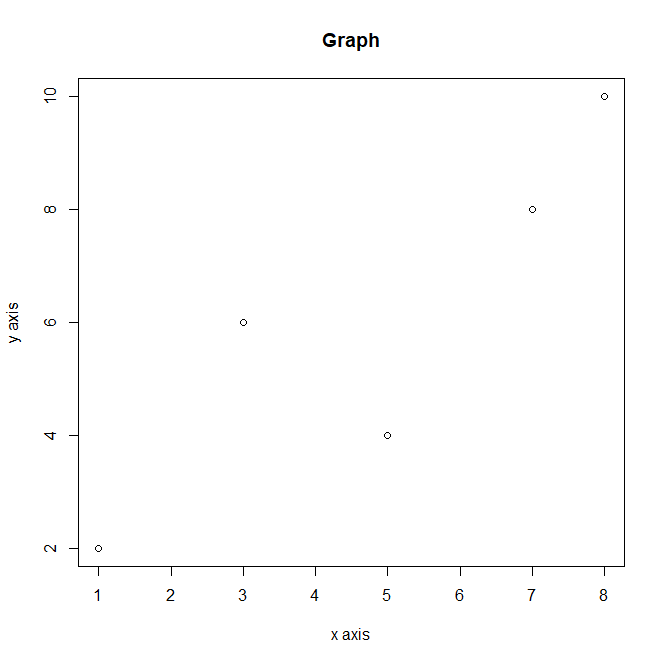
CODE:

#scatter graph

x<-c(1,5,3,7,8)

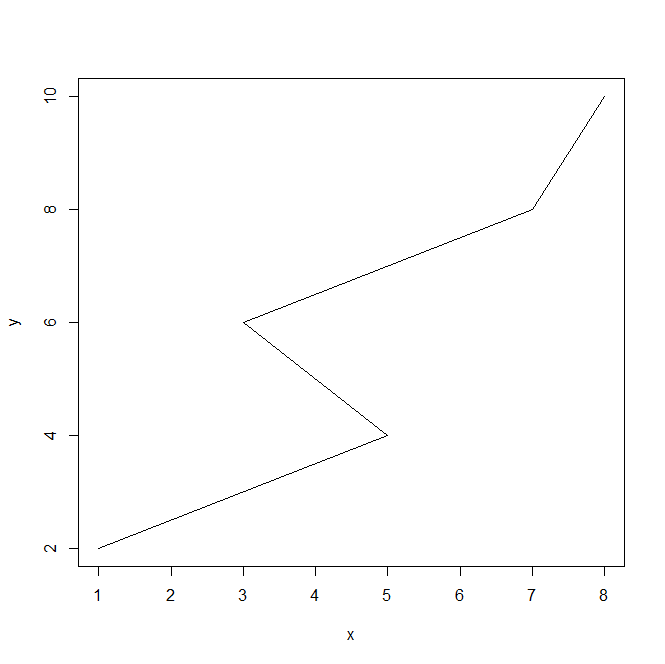
y<-c(2,4,6,8,10)

plot(x,y,main="Graph",xlab="x axis",ylab="y axis")



#line graph

plot(x,y,type="l")

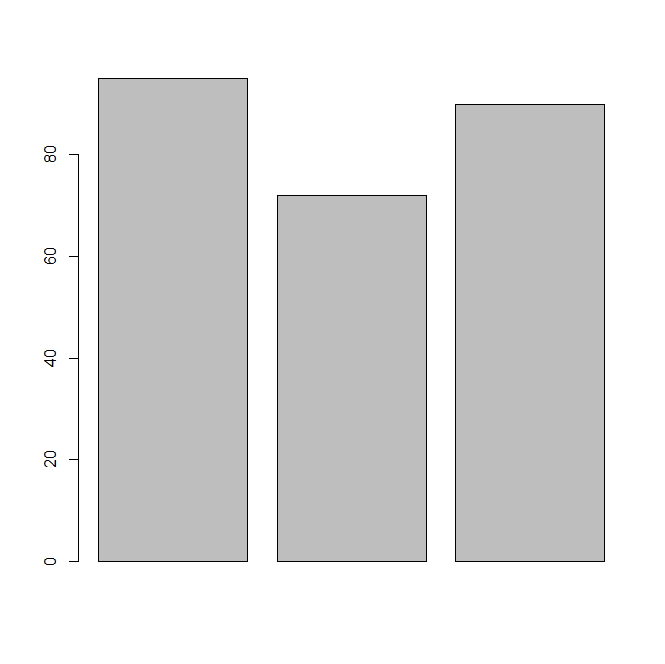


#barplot

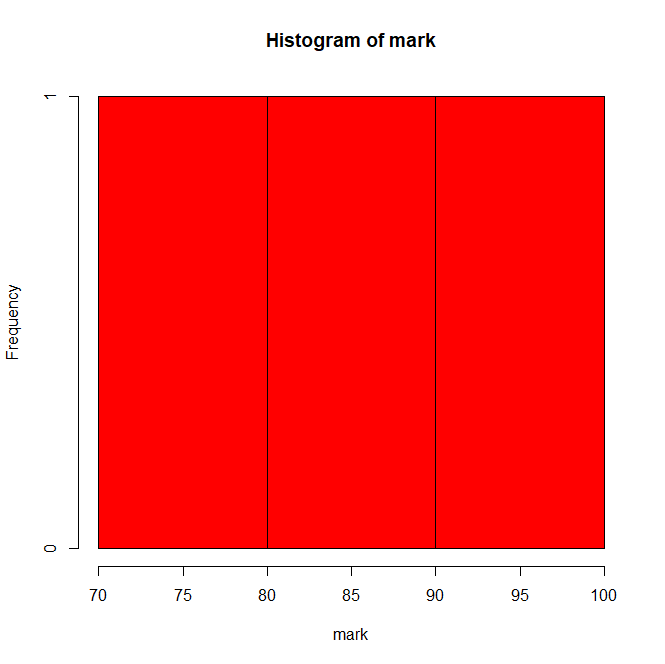
name<-c("kishore","kim","roze")

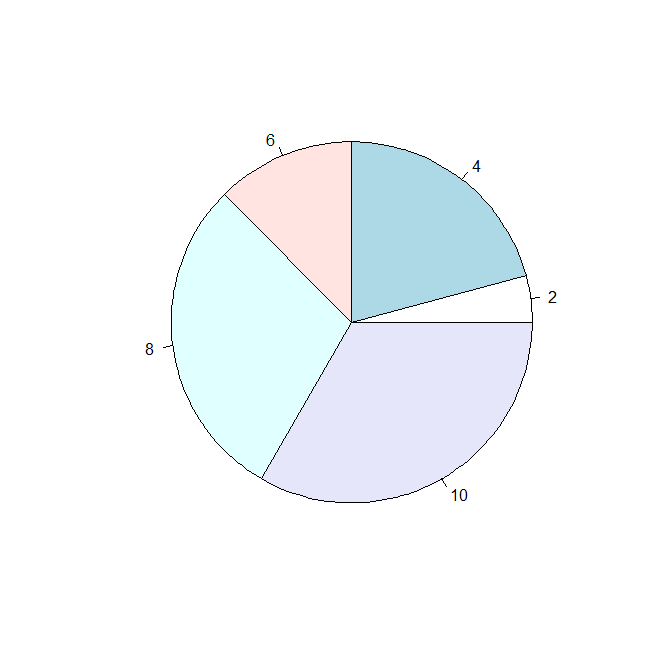
mark<-c(95,72,90)

barplot(mark,name.args=name)



hist(mark,col=”red”)



Pie(x,y)

**Exercise – 14**

**Covariance and Correlation**

**CODE**

height <- c(150, 160, 170, 180, 190)

width <- c(50, 55, 60, 65, 70)

x <- cov(height, width)

y <- cor(height, width)

print(x)

[1] 125

print(y)

[1] 1

data\_matrix <- cbind(x, y)

print("Combined Covariance and Correlation:")

[1] "Combined Covariance and Correlation:"

print(data\_matrix)

x y

[1,] 125 1

cov\_matrix <- cov(cbind(height, width))

cor\_matrix <- cor(cbind(height, width))

print("Covariance Matrix:")

[1] "Covariance Matrix:"

print(cov\_matrix)

height width

height 250 125.0

width 125 62.5

print("Correlation Matrix:")

[1] "Correlation Matrix:"

print(cor\_matrix)

height width

height 1 1

width 1 1

cov2cor\_matrix <- cov2cor(cov\_matrix)

print("Cov2cor Matrix:")

[1] "Cov2cor Matrix:"

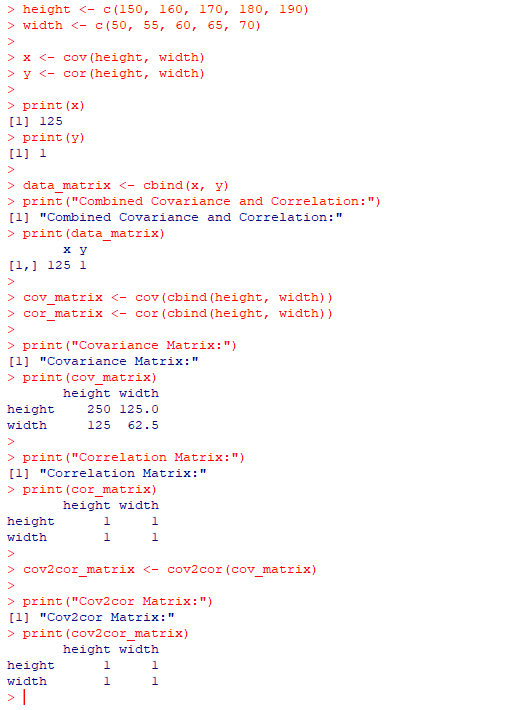
print(cov2cor\_matrix)

height width

height 1 1

width 1 1

OUTPUT



**Exercise – 15**

**Factors and Table**

**SOURCE CODE:**

height <- c(150, 160, 170, 180, 190, 165, 175, 185, 155, 200)

weight <- c(50, 55, 60, 65, 70, 58, 63, 68, 53, 75)

height\_factor <- factor(height)

weight\_factor <- factor(weight)

print("Height Factors:")

[1] "Height Factors:"

print(height\_factor)

[1] 150 160 170 180 190 165 175 185 155 200

Levels: 150 155 160 165 170 175 180 185 190 200

print("Weight Factors:")

[1] "Weight Factors:"

print(weight\_factor)

[1] 50 55 60 65 70 58 63 68 53 75

Levels: 50 53 55 58 60 63 65 68 70 75

relationship <- table(height\_factor, weight\_factor)

print("Tabulated Relationship:")

[1] "Tabulated Relationship:"

print(relationship)

weight\_factor

height\_factor 50 53 55 58 60 63 65 68 70 75

150 1 0 0 0 0 0 0 0 0 0

155 0 1 0 0 0 0 0 0 0 0

160 0 0 1 0 0 0 0 0 0 0

165 0 0 0 1 0 0 0 0 0 0

170 0 0 0 0 1 0 0 0 0 0

175 0 0 0 0 0 1 0 0 0 0

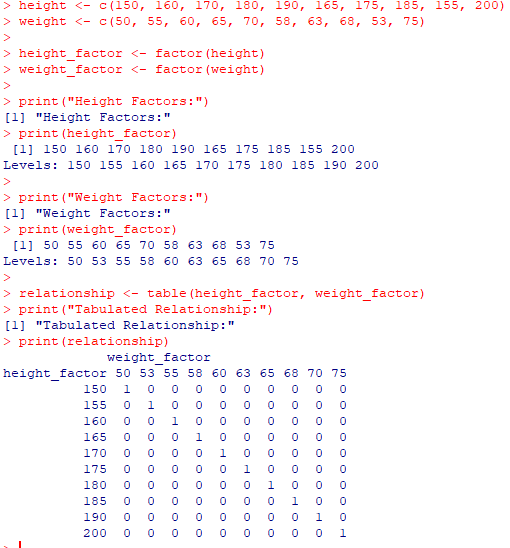
180 0 0 0 0 0 0 1 0 0 0

185 0 0 0 0 0 0 0 1 0 0

190 0 0 0 0 0 0 0 0 1 0

200 0 0 0 0 0 0 0 0 0 1

**OUTPUT**

****

**Exercise – 16**

**t.test() Function in R**

**CODE**

**men\_weight <- c(72, 85, 68, 77, 90, 66, 81, 74, 79, 69, 83, 70, 88, 76, 84)**

**women\_weight <- c(50, 65, 55, 60, 70, 52, 63, 58, 67, 54, 66, 59, 64, 53, 62)**

**result <- t.test(men\_weight, women\_weight)**

**print("Weight of Men")**

**[1] "Weight of Men"**

**print(men\_weight)**

**[1] 72 85 68 77 90 66 81 74 79 69 83 70 88 76 84**

**print("Weight of Women")**

**[1] "Weight of Women"**

**print(women\_weight)**

**[1] 50 65 55 60 70 52 63 58 67 54 66 59 64 53 62**

**print("T-test Result")**

**[1] "T-test Result"**

**print(result)**

**Welch Two Sample t-test**

**data: men\_weight and women\_weight**

**t = 7.0108, df = 26.734, p-value = 1.638e-07**

**alternative hypothesis: true difference in means is not equal to 0**

**95 percent confidence interval:**

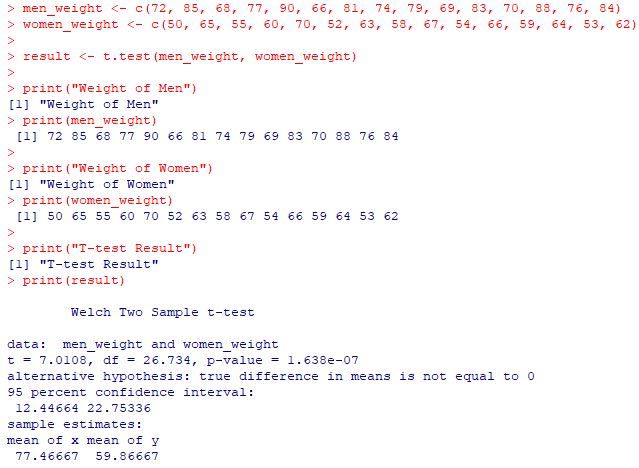
**12.44664 22.75336**

**sample estimates:**

**mean of x mean of y**

**77.46667 59.86667**

**OUTPUT**

****

**Exercise – 17**

**Linear Regression and Predict function in R**

**CODE**

height <- c(150, 160, 170, 180, 190, 155, 165, 175, 185, 195)

weight <- c(50, 55, 60, 65, 70, 53, 58, 63, 68, 73)

relation <- lm(weight ~ height)

print("Linear Regression Model Summary:")

[1] "Linear Regression Model Summary:"

print(summary(relation))

Call:

lm(formula = weight ~ height)

Residuals:

Min 1Q Median 3Q Max

-0.3030 -0.2348 0.0000 0.2348 0.3030

Coefficients:

Estimate Std. Error t value Pr( |t|)

(Intercept) -25.272727 1.049071 -24.09 9.40e-09 \*\*\*

height 0.503030 0.006061 83.00 4.95e-13 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2752 on 8 degrees of freedom

Multiple R-squared: 0.9988, Adjusted R-squared: 0.9987

F-statistic: 6889 on 1 and 8 DF, p-value: 4.952e-13

new\_height <- data.frame(height = c(158, 172, 188))

result <- predict(relation, newdata = new\_height)

print("Result for Given Heights:")

[1] "Result for Given Heights:"

print(result)

1 2 3

54.20606 61.24848 69.29697

OUTPUT

