Pratical 1: A car travels 25 miles at 25 miles per hour (mi/h), 25 miles at 50 mph, and 25 miles at 75 mph. Write a program to find the arithmetic mean of the three velocities and the harmonic mean of the three velocities. Which is correct?

Code:

```
dist <- c(25,25,25)

speed <- c(25,50,75)


my_data <- rep(speed,dist)


# To find Arithmatic mean

arithmetic_mean_speed <- mean(my_data)

arithmetic_mean_speed


# call library

library(psych)


# to find harmonic mean

harmonic_mean_speed <- harmonic.mean(my_data)

harmonic_mean_speed


print("Since speed have fraction unit measure therefore ,Harmonic mean is correct mean")
```

Output:

arithmetic_mean_speed

[1] 50

harmonic_mean_speed

[1] 40.90909

[1] "Since speed have fraction unit measure therefore ,Harmonic mean is correct mean"

Pratical 2:

Enter the following details of wages of 65 employees at the ABC Ltd. In Excel: Wages
Number of Employees

25000-25999	8
26000-26999	10
27000-27999	16
28000-28999	14
29000-29999	10
30000-30999	5
31000-31999	2
Total	65

Import the data in R and find the mean, standard deviation and variance of wage and mode wage of the 65 employees.

Code:

```
library(data.table)

salarydata<-data.frame(

  salaries_low=c(25000,26000,27000,28000,29000,30000,31000),

  salaries_high=c(25999,26999,27999,28999,29999,30999,32999),

  Numbers=c(8,10,16,14,10,5,2)

)

View(salarydata)

cal_data <- salarydata

# center value of group

cal_data$center_value = (cal_data$salaries_high+cal_data$salaries_low)/2

attach(cal_data)

calculation_data <- rep(center_value,Numbers)

calculation_data

# To find mean of salary

mean_salary <- mean(calculation_data)

mean_salary

# To find Standard deviation of salary

sd_salary <- sd(calculation_data)

sd_salary
```

To find Variance of Salary

var_salary <- var(calculation_data)

var_salary

Output:

	salaries_low	salaries_high	Numbers
1	25000	25999	8
2	26000	26999	10
3	27000	27999	16
4	28000	28999	14
5	29000	29999	10
6	30000	30999	5
7	31000	32999	2

mean_salary

[1] 27991.81

sd_salary

[1] 1609.28

var_salary

[1] 2589784

Pratical 3:

Enter the following details of wages of 65 employees at the ABC Ltd. In Excel:

Wages	Number of Employees
25000-25999	8
26000-26999	10
27000-27999	16
28000-28999	14
29000-29999	10
30000-30999	5
31000-31999	2
Total	65

Import the data in R and find the median wage and mode wage of the 65 employees.

Code:

```
library(data.table)
salarydata<-data.frame(
  salaries_low=c(25000,26000,27000,28000,29000,30000,31000),
  salaries_high=c(25999,26999,27999,28999,29999,30999,32999),
  Numbers=c(8,10,16,14,10,5,2)
)
print(salarydata)
salarydata <- as.data.table(salarydata)
salarydata <-salarydata[,cumNumbers:=cumsum(Numbers)]
salarydata

# Compute total count (N)
total_count <- sum(salarydata$Numbers)

# Find the median group using data.table syntax
median_group <- salarydata[
  (cumNumbers - Numbers) <= (total_count / 2) &
    cumNumbers >= (total_count / 2)
]
# Print or use these variables as needed
print(median_group)
# Extract relevant values
l1 <- median_group$salaries_low
l2 <- median_group$salaries_high
f  <- median_group$Numbers
pcf <- median_group$cumNumbers - median_group$Numbers
n  <- sum(salarydata$Numbers)
```

```
median<-l1+(((n/2)-pcf)/f*(l2-l1))

median

#For Mode

print(salarydata)

salarydata <- as.data.table(salarydata)

salarydata[ , prevNumbers := shift(Numbers,1)]

salarydata[ , nextNumbers := shift(Numbers,-1)]

salarydata

#identifying mode group

modegroup <- salarydata[Numbers == max(Numbers)]

modegroup

#creating the variables needed to calculate mode

l1 = modegroup[,salaries_low]

l2 = modegroup[,salaries_high]

f1 = modegroup[,Numbers]

f0 = modegroup[,prevNumbers]

f2 = modegroup[,nextNumbers]

#calculating mode

groupmode <- l1 + ((f1-f0)/(f1-f0+f1-f2)*(l2-l1))

groupmode
```

Output:

```
> print(salarydata)
  salaries_low salaries_high Numbers
1        25000         25999       8
2        26000         26999      10
3        27000         27999      16
4        28000         28999      14
5        29000         29999      10
6        30000         30999       5
7        31000         32999       2
> print(median_group)
   salaries_low salaries_high Numbers cumNumbers
          <num>         <num>   <num>      <num>
1:        27000         27999      16         34
> median
[1] 27905.34
> salarydata
   salaries_low salaries_high Numbers cumNumbers prevNumbers nextNumbers
          <num>         <num>   <num>      <num>       <num>       <num>
1:        25000         25999       8          8          NA          10
2:        26000         26999      10         18           8          16
3:        27000         27999      16         34          10          14
4:        28000         28999      14         48          16          10
5:        29000         29999      10         58          14           5
6:        30000         30999       5         63          10           2
7:        31000         32999       2         65           5          NA
> #identifying mode group
> modegroup <- salarydata[Numbers == max(Numbers)]
```

```
> modegroup
   salaries_low salaries_high Numbers cumNumbers prevNumbers nextNumbers
          <num>         <num>   <num>      <num>       <num>       <num>
1:        27000         27999      16         34          10          14
> groupmode <- l1 + ((f1-f0)/(f1-f0+f1-f2)*(l2-l1))
> groupmode
[1] 27749.25
```

Pratical 4:

Enter the following data sets in Excel: a) 12, 6, 7, 3, 15, 10, 18, 5 b) 9, 3, 8, 8, 9, 8, 9, 18. Import the data in R and find standard deviation and variance of the data sets using R.

Code:

```
input<-data.frame(
  X1=c(12, 6, 7, 3, 15, 10, 18, 5),
  X2=c(9, 3, 8, 8, 9, 8, 9, 18)
)
print("Standard Deviaton of X1 and X2 are : ")
sd(input$X1)
sd(input$X2)
print("Variance of X1 and X2 are : ")
var(input$X1)
var(input$X2)
```

Output:

```
[1] "Standard Deviaton of X1 and X2 are : "
> sd(input$X1)
[1] 5.209881
> sd(input$X2)
[1] 4.140393
```

```
> print("Variance of X1 and X2 are : ")
[1] "Variance of X1 and X2 are : "
> var(input$X1)
[1] 27.14286
> var(input$X2)
[1] 17.14286
```

Pratical 5:

Enter the following table of three distributions f1, f2 and f3 for the variable X in EXCEL.

X	f1	f2	f3
0	10	1	1
1	5	2	2
2	2	14	2
3	2	2	5
4	1	1	10

Import the data in R and write and program to find Pearson's first and second coefficients of skewness.

Code:

```
my_data<- data.frame(read.csv("C:\\Users\\prath\\Desktop\\data.csv"))
my_data
getmode <- function(v) {
  uniqv <- unique(v)
  uniqv[which.max(tabulate(match(v, uniqv)))]
}
#First Pearson
mean1<-mean(rep(my_data$x,my_data$F1))
median1<-median(rep(my_data$x,my_data$F1))
mode1<-getmode(rep(my_data$x,my_data$F1))
sd1<-sd(rep(my_data$x,my_data$F1))
```

```
Pearsonfirst1<-((mean1-mode1)/sd1)

Pearsonsecond1<-(3*(mean1-median1)/sd1)

print("Pearson first coefficient of skewness with x and f1 is:")

print(Pearsonfirst1)

print("Pearson Second coefficient of skewness with x and f1 is:")

print(Pearsonsecond1)

#Second Pearson

mean2<-mean(rep(my_data$x,my_data$F2))

median2<-median(rep(my_data$x,my_data$F2))

mode2<-getmode(rep(my_data$x,my_data$F2))

sd2<-sd(rep(my_data$x,my_data$F2))

Pearsonfirst2<-((mean2-mode2)/sd1)

Pearsonsecond2<-(3*(mean2-median2)/sd2)

print("Pearson first coefficient of skewness with x and f2 is:")

print(Pearsonfirst2)

print("Pearson Second coefficient of skewness with x and f2 is:")

print(Pearsonsecond2)

#Third Pearson

mean3<-mean(rep(my_data$x,my_data$F3))

median3<-median(rep(my_data$x,my_data$F3))

mode3<-getmode(rep(my_data$x,my_data$F3))

sd3<-sd(rep(my_data$x,my_data$F3))

Pearsonfirst3<-((mean3-mode3)/sd3)

Pearsonsecond3<-(3*(mean3-median3)/sd3)

print("Pearson first coefficient of skewness with x and f3 is:")

print(Pearsonfirst3)

print("Pearson Second coefficient of skewness with x and f3 is:")

print(Pearsonsecond3)
```

Output:

```
> my_data
  x F1 F2 F3
1 0 10  1  1
2 1  5  2  2
3 2  2 14  5
4 3  2  2  5
5 4  1  1 10
```

[1] "Pearson first coefficient of skewness with x and f1 is:"

[1] 0.7696196

[1] "Pearson Second coefficient of skewness with x and f1 is:"

[1] 1.09367

[1] "Pearson first coefficient of skewness with x and f2 is:"

[1] 0

[1] "Pearson Second coefficient of skewness with x and f2 is:"

[1] 0

[1] "Pearson first coefficient of skewness with x and f3 is:"

[1] -0.9037158

[1] "Pearson Second coefficient of skewness with x and f3 is:"

[1] -0.2168918

Pratical 6:

Many casinos use card-dealing machines to deal cards at random. Occasionally, the machine is tested to ensure an equal likelihood of dealing for each suit. To conduct the test, 1,500 cards are dealt from the machine, while the number of cards in each suit is counted. Theoretically, 375 cards should be dealt from each suit. But this is not the case as shown in the following table:

Spades	Diamonds	Clubs	Hearts	
Observed	402	358	273	467
Expected	375	375	375	375

Enter the data in Excel. Import the date in R and write a program using chi-square test to determine if the discrepancies are significant. If the discrepancies are significant, then the game would not be fair.

Code:

```
data <- data.frame(
  spade = 402,
  diamond = 358,
  club = 273,
  heart = 467
)
result <- chisq.test(data)
result
if (result$p.value > 0.05) {
  print("Null hypothesis is Accepted")
} else {
  print("Null hypothesis is Rejected")
}
```

Output:

```
> result

	Chi-squared test for given probabilities

data:  data
X-squared = 53.029, df = 3, p-value = 1.807e-11
[1] "Null hypothesis is Rejected"
```

Pratical 7:

A business owner had been working to improve employee relations in his company. He predicted that he met his goal of increasing employee satisfaction from 65% to 80%. Employees from four departments were asked if they were satisfied with the working conditions of the company. The results are shown in the following table:

Finance	Sales	HR	Technology	
Satisfied	12	38	5	8
Dissatisfied	7	19	3	1
Total	19	57	8	9

Enter the data in Excel. Import the date from Excel to R and write a program suing chi-square test to determine whether the results support or reject the business owner's prediction.

Code:

```
data <- data.frame(
  finance = c(12,7),
  sales = c(38,19),
  hr = c(5,3),
  technology = c(8,1)
)
result <- chisq.test(data)
result
if (result$p.value > 0.05) {
  print("Null hypothesis is Accepted")
} else {
  print("Null hypothesis is Rejected")
}
```

Output:

```
> result

	Pearson's Chi-squared test

data:  data
X-squared = 2.1553, df = 3, p-value = 0.5408
[1] "Null hypothesis is Accepted"
```

Pratical 8:

Suppose the number of games in which major league baseball players play during their careers is normally distributed with mean equal to 1500 games and standard deviation equal to 350 games. Use R to solve the following problems. (a) What percentage play in fewer than 750 games? (b) What percentage play in more than 2000 games? (c) Find the 90th percentile for the number of games played during a career.

Code:

```
print("What percentage play in fewer than 750 games")

pa<-pnorm(750, mean = 1500, sd = 350)

Percenta <- pa*100

print(Percenta)

print("What percentage play in more than 2000 games")

pb<-pnorm(2000, mean = 1500, sd = 350, lower.tail = FALSE)

Percentb <- pb*100

print(Percentb)

print("the 90th percentile for the number of games played during a career")

p05<-round(qnorm(0.05,mean = 1500, sd = 350),0)

p95<-round(qnorm(0.95,mean = 1500, sd = 350),0)

cat("Range for 90 Percentile is : ",p05,"-",p95)
```

Output:

```
[1] "What percentage play in fewer than 750 games"

[1] 1.606229

[1] "What percentage play in more than 2000 games"

[1] 7.656373

[1] "the 90th percentile for the number of games played during a career"

Range for 90 Percentile is :  924 - 2076
```

Pratical 9:

Enter the following table which shows the heights(H) to the nearest inch (in) and the weights(W) to the nearest pound (lb) of a sample of 12 male students drawn at random from the first-year students at College.

H	70	63	72	60	66	70	74	65	62	67	65	68
W	155	150	180	135	156	168	178	160	132	145	139	152

Import the data in R and write a program to fit a least squares line using a) H as the independent variable

b) H as dependent variable

Code:

```
height <- c(70,63,72,60,66,70,74,65,62,67,65,68)
weight <- c(155,150,180,135,156,168,178,160,132,145,139,152)
# A) H is independent Variable
# linear regression
line_reg1 <- lm(weight ~ height)
line_reg1
print(paste("weight = ",line_reg1$coefficients[1],"+",
        line_reg1$coefficients[2],"* Height"))
# B) H is dependent Variable
line_reg2 <- lm(height ~ weight)
line_reg2
print(paste("Height = ",line_reg2$coefficients[1],"+",
        line_reg2$coefficients[2],"* Weight"))
```

Output:

```
> line_reg1
Call:
lm(formula = weight ~ height)
Coefficients:
(Intercept)      height
   -60.746       3.216
[1] "weight =  -60.7460869565214 + 3.21565217391304 * Height"
Call:
lm(formula = height ~ weight)
Coefficients:
(Intercept)      weight
   31.1078      0.2317
[1] "Height =  31.1077829301918 + 0.23173329991227 * Weight"
```

Pratical 10:

Enter the total agricultural exports in millions of dollars in Excel:

Year	2000	2001	2002	2003	2004	2005
Total Value	51246	53659	53115	59364	61383	62958

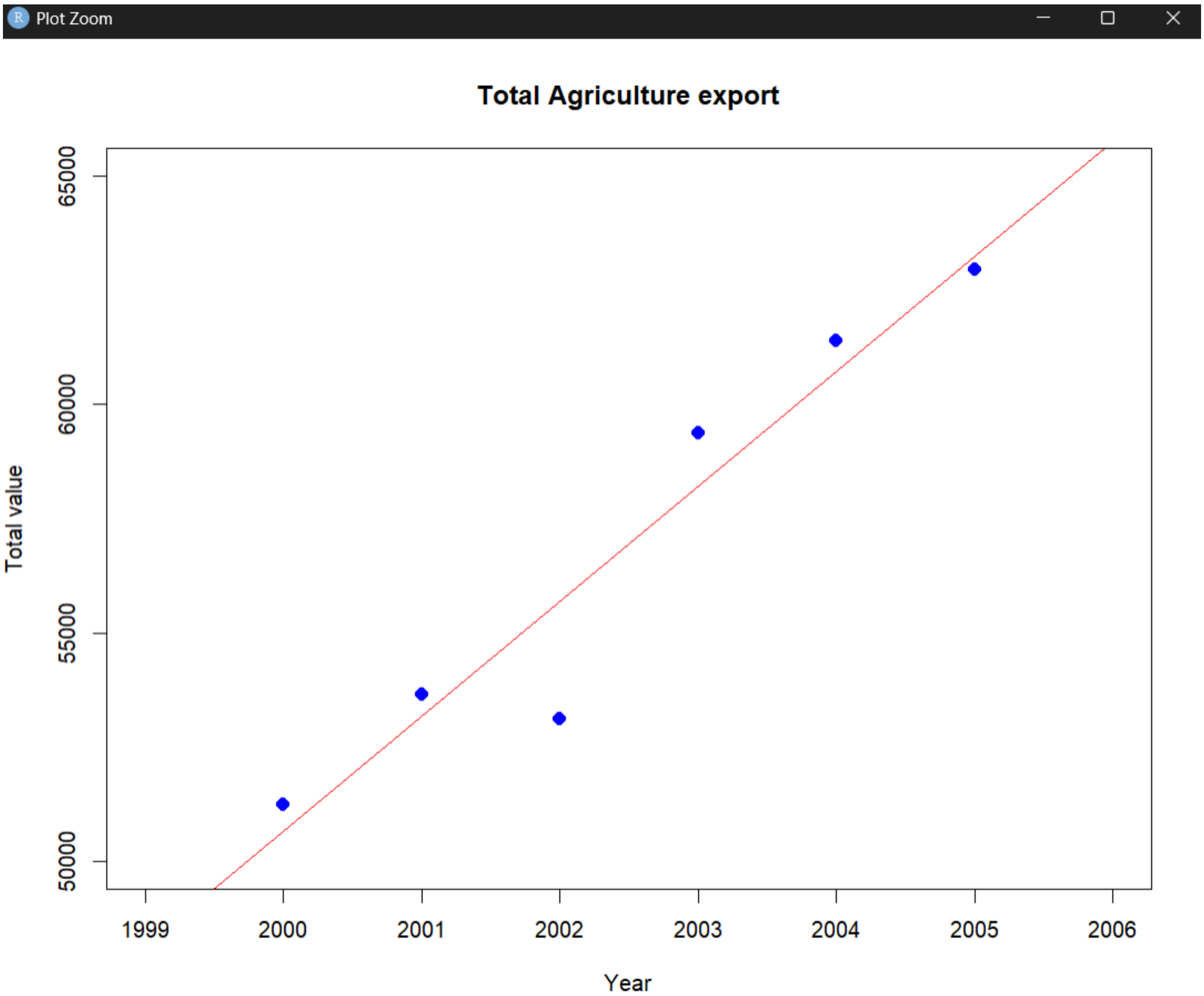
Import the data in R and perform the following

(a) Graph the data and show the least-squares regression line. (b) Find and plot the trend line for the data.

(c) Estimate the value of total agricultural exports in the year 2006.

Code:

```
year <- c(2000:2005)

Total_value <- c(51246, 53659,53115,59364, 61383, 62958)

reg_line <- lm(Total_value ~ year)

reg_line

plot(year,Total_value,

   main = "Total Agriculture export",

   xlab = "Year",

   ylab = "Total value",

   cex = 1.3,

   pch = 16,

   abline(reg_line, col = "red"),

   xlim = c(1999,2006),

   ylim = c(50000,65000),

   col = "blue")

predict(reg_line,data.frame(year = c(2006)))
```



(Leave half page to stick this graph)

> reg_line

Call:

lm(formula = Total_value ~ year)

Coefficients:

(Intercept) year

-4976816 2514

> predict(reg_line,data.frame(year = c(2006)))

1

65752.27

Pratical 11:

Enter the following table in Excel which shows the first two grades (denoted by First Quiz X and Second Quiz Y, respectively) of 10 students on two short quizzes in biology.

X	6	5	8	8	7	6	10	4	9	7
Y	8	7	7	10	5	10	8	6	8	6

Import the data in R and write programs for the following: (a) Find the least-squares regression line of Y on X.

(b) Find the least-squares regression line of X on Y.

Code:

```
x <- c(6,5,8,8,7,6,10,4,9,7)

y <- c(8,7,7,10,5,10,8,6,8,6)

# A) Y on X

# linear regression

line_reg1 <- lm(y ~ x)

line_reg1

print(paste("weight = ",line_reg1$coefficients[1],"+",

        line_reg1$coefficients[2],"* Height"))

# B) x on Y

line_reg2 <- lm(x ~ y)

line_reg2

print(paste("Height = ",line_reg2$coefficients[1],"+",

        line_reg2$coefficients[2],"* Weight"))
```

Output:

```
> line_reg1

Call:

lm(formula = y ~ x)

Coefficients:

(Intercept)        x

   5.8667      0.2333

[1] "weight =  5.86666666666666 + 0.233333333333334 * Height"
```

> line_reg2

Call:

lm(formula = x ~ y)

Coefficients:

(Intercept)   y

4.8571   0.2857

[1] "Height = 4.85714285714285 + 0.285714285714286 * Weight"

Pratical 12(Same as 10 need to ask to repeat or not)

Pratical 13 and 15 omit

Pratical 14 :

Write a program in R to create two matrices A and B of order 3 X 3 and perform the following operations:

a. Add matrices A and B
b. Multiply matrices A and B
c. Find the inverse of matrix A
d. Find the inverse of matrix B
e. Find the transpose of matrix B

Code:

```
# Create two 3x3 matrices A and B

A <- matrix(c(2, 4, 3, 1, 5, 7, 6, 8, 9), nrow=3, ncol=3)

B <- matrix(c(9, 8, 7, 6, 5, 4, 3, 2, 1), nrow=3, ncol=3)


# Display matrices

print("Matrix A:")

print(A)


print("Matrix B:")

print(B)
```

```r
# a. Add matrices A and B
sum_matrix <- A + B
print("Sum of A and B:")
print(sum_matrix)

# b. Multiply matrices A and B
product_matrix <- A %*% B
print("Product of A and B:")
print(product_matrix)

# c. Find the inverse of matrix A
if (det(A) != 0) {
  inverse_A <- solve(A)
  print("Inverse of A:")
  print(inverse_A)
} else {
  print("Matrix A is singular and does not have an inverse.")
}

# d. Find the inverse of matrix B
if (det(B) != 0) {
  inverse_B <- solve(B)
  print("Inverse of B:")
  print(inverse_B)
} else {
  print("Matrix B is singular and does not have an inverse.")
}
```

```
# e. Find the transpose of matrix B
transpose_B <- t(B)
print("Transpose of B:")
print(transpose_B)
```

Output:

```
[1] "Matrix A:"
     [,1] [,2] [,3]
[1,]    2    1    6
[2,]    4    5    8
[3,]    3    7    9
[1] "Matrix B:"
     [,1] [,2] [,3]
[1,]    9    6    3
[2,]    8    5    2
[3,]    7    4    1
[1] "Sum of A and B:"
     [,1] [,2] [,3]
[1,]   11    7    9
[2,]   12   10   10
[3,]   10   11   10
[1] "Product of A and B:"
     [,1] [,2] [,3]
[1,]   68   41   14
[2,]  132   81   30
[3,]  146   89   32
```

```
[1] "Inverse of A:"
           [,1]  [,2]       [,3]
[1,] -0.2500000  0.75 -0.5000000
[2,] -0.2727273  0.00  0.1818182
[3,]  0.2954545 -0.25  0.1363636
[1] "Matrix B is singular and does not have an inverse."
[1] "Transpose of B:"
     [,1] [,2] [,3]
[1,]    9    8    7
[2,]    6    5    4
[3,]    3    2    1
```