CSE3505_FDA_Lab_Exp_1

20BCE1025_Abhishek_N_N

2022-08-16

Contents

Lab 1 Exercise Understanding Basic Data Types in R

```
rm(list=ls())
```

1) initial decimal money let it be A

```
A<-10.5
A
```

```
## [1] 10.5
```

2) class name of A

```
class(A)
```

```
## [1] "numeric"
```

3) checking A is numeric?

```
is.numeric(A)
```

```
## [1] TRUE
```

4) integer value to B and Displaying it

```
B<-5
B
```

```
## [1] 5
```

5) check B is Integer ?

```
is.integer(B)
```

```
## [1] FALSE
```

6) variable c to store rupee part of A

```
C<-as.integer(A)
C
```

```
## [1] 10
```

7) cost of one chocolate

```
cost=C/B
cost
```

```
## [1] 2
```

8) representing money as char string

```
as.character(A)
```

```
## [1] "10.5"
```

9) storing first and last name of kid

```
first_name<-"Jai"
last_name<-"Ganesh"
```

10) displaying message

```
message<-paste(first_name,last_name,"bought" ,as.character(B) ,"chocolates")
message
```

```
## [1] "Jai Ganesh bought 5 chocolates"
```

11) extracting “Little” from “Twinkle Twinkle Little Star”

```
rhymes<-"Twinkle Twinkle Little Star"
rhymes
```

```
## [1] "Twinkle Twinkle Little Star"
```

```
extracted_text=substr(rhymes, 17, 22)
extracted_text
```

```
## [1] "Little"
```

12) replacing little as big

```
library("stringr")
rhymes<-sub( "Little", "Big",rhymes)
rhymes
```

```
## [1] "Twinkle Twinkle Big Star"
```

13) complex no to x

```
x<-2+3i
x
```

```
## [1] 2+3i
```

14) real part of x

```
Re(x)
```

```
## [1] 2
```

15) imaginary part of x

```
Im(x)
```

```
## [1] 3
```

16) computing square root of negative number

```
sqrt(as.complex(-15))
```

```
## [1] 0+3.872983i
```

Lab 1 Sample Practice

assign and print a variable

```
x<-1
print(x)
```

```
## [1] 1
```

```
x
```

```
## [1] 1
```

```
msg="hello"
msg
```

```
## [1] "hello"
```

```
typeof(x)
```

```
## [1] "double"
```

```
typeof(msg)
```

```
## [1] "character"
```

explicit and implicit printing

```
x<-1:20
print(x)
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
```

```
x
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
```

```
rollNo<-1:70
rollNo
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
## [26] 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
## [51] 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70
```

```
y<-20:30
y
```

```
##  [1] 20 21 22 23 24 25 26 27 28 29 30
```

```
x<-c(0.5,0.6)
x
```

```
## [1] 0.5 0.6
```

```
x<-c(TRUE,FALSE)
x
```

```
## [1]  TRUE FALSE
```

```r
x<-c(T,F)
x
```

```
## [1]  TRUE FALSE
```

```r
x<-c("a","b","c")
x
```

```
## [1] "a" "b" "c"
```

```r
x=9:29
x
```

```
##  [1]  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29
```

```r
x=c(1+0i,2+4i)
x
```

```
## [1] 1+0i 2+4i
```

```r
x<-vector("numeric",length=10)
x
```

```
##  [1] 0 0 0 0 0 0 0 0 0 0
```

to get help

```r
?type
```

```r
help(type)
```

```r
y1<-c(1.7,"a")
y1
```

```
## [1] "1.7" "a"
```

```r
y2=c(TRUE,2)
y2
```

```
## [1] 1 2
```

```r
y3<-c("a",TRUE);y3
```

```
## [1] "a"    "TRUE"
```

```r
class(y1)
```

```
## [1] "character"
```

```r
class(y2)
```

```
## [1] "numeric"
```

```r
class(y3)
```

```
## [1] "character"
```

```r
y11<-c(1.7,"3.2")
y11
```

```
## [1] "1.7" "3.2"
```

```r
class(y11)
```

```
## [1] "character"
```

```r
y22<-c(FALSE,TRUE)
y22
```

```
## [1] FALSE  TRUE
```

```r
class(y22)
```

```
## [1] "logical"
```

Explicit Coercion

```r
x<-0:6
x
```

```
## [1] 0 1 2 3 4 5 6
```

```r
class(x)
```

```
## [1] "integer"
```

```r
y=as.numeric(x)
y
```

```
## [1] 0 1 2 3 4 5 6
```

```
class(y)
```

```
## [1] "numeric"
```

```
z=as.character(x)
z
```

```
## [1] "0" "1" "2" "3" "4" "5" "6"
```

```
class(z)
```

```
## [1] "character"
```

```
l=as.logical(x)
l
```

```
## [1] FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
```

```
class(l)
```

```
## [1] "logical"
```

```
com=as.complex(x)
com
```

```
## [1] 0+0i 1+0i 2+0i 3+0i 4+0i 5+0i 6+0i
```

```
class(com)
```

```
## [1] "complex"
```

CSE3505_FDA_Lab_Exp_2

20BCE1025_Abhishek_N_N

2022-08-17

Working with Vectors in R

1. Create vector 'class' to store the class names 'class1','class2',...,'class5'

```
class <- c("class1","class2","class3","class4","class5")
class
```

```
## [1] "class1" "class2" "class3" "class4" "class5"
```

2. Use assign() function to create a vector 'avg' to store the average marks.

```
avg <- c(0,0,0,0,0)
assign("avg", c(80, 90, 95, 50, 70))
avg
```

```
## [1] 80 90 95 50 70
```

3. Display the average mark of class2.

```
avg[2]
```

```
## [1] 90
```

4. Display all the average marks except class 2.

```
avg[c(TRUE,FALSE,TRUE,TRUE,TRUE)]
```

```
## [1] 80 95 50 70
```

5. Access the average mark of class4 by its name.

```
avg[match("class4",class)]
```

```
## [1] 50
```

6. Find the minimum average mark and print the class which scored it.

```
min(avg)
```

```
## [1] 50
```

```
class[which.min(avg)]
```

```
## [1] "class4"
```

7. Find the maximum average mark and print the class which scored it.

```
max(avg)
```

```
## [1] 95
```

```
class[which.max(avg)]
```

```
## [1] "class3"
```

8. Find the total of average marks scored by all classes.

```
sum(avg)
```

```
## [1] 385
```

9. Find the mean of the average marks scored by all classes.

```
mean(avg)
```

```
## [1] 77
```

10. Find the standard deviation of the average marks scored by all classes.

```
sd(avg)
```

```
## [1] 17.88854
```

11. Arrange the average marks in ascending order.

```
avg<-sort(avg)
avg
```

```
## [1] 50 70 80 90 95
```

12. Create a vector classes by repeat the vector class twice.

```
classes<-rep(class, times=2)
classes
```

```
##  [1] "class1" "class2" "class3" "class4" "class5" "class1" "class2" "class3"
##  [9] "class4" "class5"
```

13. Create a vector marks by repeating each average mark twice.

```
marks<-rep(avg, times=2)
marks
```

```
##  [1] 50 70 80 90 95 50 70 80 90 95
```

14. Create a vector 'report' by adding the vector 'avg' with a sequence of 10 to 1 and find its length.

```
report <- c(avg,10:1)
report
```

```
##  [1] 50 70 80 90 95 10  9  8  7  6  5  4  3  2  1
```

```
length(report)
```

```
## [1] 15
```

15. Identify the classes for which average marks>70.

```
for(a in avg){
  if (a>70) print(a)
}
```

```
## [1] 80
## [1] 90
## [1] 95
```

20BCE1025_Abhishek_N_N_Lab-Ex-3-Working with matrices in R

20BCE1025_Abhishek_N_N

2022-09-09

1. Represent the height in cm information of a team of 12 basketball players as a matrix of dimension 4x3 in row major form.

```
heights <- c(150, 165, 133, 155, 177, 187, 165, 190, 173, 149, 134, 157)
m <- matrix(heights, nrow = 4, ncol = 3, byrow = TRUE)
m
```

```
##      [,1] [,2] [,3]
## [1,]  150  165  133
## [2,]  155  177  187
## [3,]  165  190  173
## [4,]  149  134  157
```

2. Access the height at row 3 and column 2.

```
m[3,2]
```

```
## [1] 190
```

3. Display all the heights in row 2.

```
m[2,]
```

```
## [1] 155 177 187
```

4. Display all the heights in column 3.

```
m[,3]
```

```
## [1] 133 187 173 157
```

5. Extract the heights in all rows but only in column 1 and 3.

```
m[,c(1,3)]
```

```
##      [,1] [,2]
## [1,]  150  133
## [2,]  155  187
## [3,]  165  173
## [4,]  149  157
```

6. Find the transpose of the matrix.

```
t(m)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]  150  155  165  149
## [2,]  165  177  190  134
## [3,]  133  187  173  157
```

7. Four more players got added to the team. Update the matrix to reflect the heights of the players.

```
m<-cbind(m,c(150, 151, 152, 153))
m
```

```
##      [,1] [,2] [,3] [,4]
## [1,]  150  165  133  150
## [2,]  155  177  187  151
## [3,]  165  190  173  152
## [4,]  149  134  157  153
```

8. Append three more players' height in the matrix.

```
m<-rbind(m,c(160, 161, 162))
```

```
## Warning in rbind(m, c(160, 161, 162)): number of columns of result is not a
## multiple of vector length (arg 2)
```

```
m
```

```
##      [,1] [,2] [,3] [,4]
## [1,]  150  165  133  150
## [2,]  155  177  187  151
## [3,]  165  190  173  152
## [4,]  149  134  157  153
## [5,]  160  161  162  160
```

20BCE1025_Abhishek_N_N_FDA Lab Experiment-4-a

20BCE1025_Abhishek_N_N

2022-09-09

1. Create a list to maintain the details of a student such as registration number, name, no. of courses registered and marks in each subject.

```
studentDetails <-
  list(
    regNo = c('20BCE1025', '19BAI1223', '21EEE7899'),
    name = c('Abhishek', 'Ram', 'Bheem'),
    courses = c('Physics', 'Chemistry', 'Mathematics'),
    marks = matrix(
      c(98, 99, 95, 43, 28, 70, 23, 78, 76),
      nrow = 3,
      ncol = 3,
      byrow = TRUE
    )
  )
studentDetails
```

```
## $regNo
## [1] "20BCE1025" "19BAI1223" "21EEE7899"
##
## $name
## [1] "Abhishek" "Ram"      "Bheem"
##
## $courses
## [1] "Physics"     "Chemistry"   "Mathematics"
##
## $marks
##      [,1] [,2] [,3]
## [1,]   98   99   95
## [2,]   43   28   70
## [3,]   23   78   76
```

2. Retrieve the name of the students.

```
studentDetails$name
```

```
## [1] "Abhishek" "Ram"      "Bheem"
```

3. Extract only the registration number and the marks of the students.

```
studentDetails[c(1,4)]
```

```
## $regNo
## [1] "20BCE1025" "19BAI1223" "21EEE7899"
##
## $marks
##      [,1] [,2] [,3]
## [1,]   98   99   95
## [2,]   43   28   70
## [3,]   23   78   76
```

4. Access the mark in the first course registered.

```
studentDetails$marks[,1]
```

```
## [1] 98 43 23
```

5. Modify the mark entry in the last course as 5 more than the existing mark.

```
studentDetails$marks[, 3] <- (studentDetails$marks[, 3] + 5)
studentDetails$marks
```

```
##      [,1] [,2] [,3]
## [1,]   98   99  100
## [2,]   43   28   75
## [3,]   23   78   81
```

Q. A college has conducted technical events for the students. It maintains the name of the participant and the score obtained in different events.

1. Create a data frame by considering 5 students and 4 events. Each event has a maximum score of 10. If a student participates in an event, its entry contains the score value and 0 otherwise.

```
eventScore <- data.frame(
  id = c(1:5),
  name = c('Ram', 'Bheem', 'Soam', 'Raheem', 'Sham'),
  coading_event = c(10, 2, 0, 7, 5),
  design_event = c(0, 10, 4, 7, 0),
  hackathon_event = c(2, 4, 10, 0, 0),
  cyber_event = c(7, 0, 8, 3, 2)
)
```

2. View the contents of the data frame.

```
eventScore
```

```
##   id   name coading_event design_event hackathon_event cyber_event
## 1  1    Ram            10            0               2           7
## 2  2  Bheem             2           10               4           0
## 3  3   Soam             0            4              10           8
## 4  4 Raheem             7            7               0           3
## 5  5   Sham             5            0               0           2
```

3. Find the total score of each participant.

```
total<-apply(eventScore[,3:6], 1, sum)
total
```

```
## [1] 19 16 22 17  7
```

4. Append a column to include the total score of the participants and view the data frame.

```
eventScore$total=total
eventScore
```

```
##   id   name coading_event design_event hackathon_event cyber_event total
## 1  1    Ram            10            0               2           7    19
## 2  2  Bheem             2           10               4           0    16
## 3  3   Soam             0            4              10           8    22
## 4  4 Raheem             7            7               0           3    17
## 5  5   Sham             5            0               0           2     7
```

5. Find the maximum score and display the name of the participant who scored it.

```
max_score<-apply(eventScore[,3:6],2,max)
max_score
```

```
##   coading_event    design_event hackathon_event     cyber_event
##              10              10              10               8
```

```
max_score_index<-apply(eventScore[,3:6] , 2, which.max)
eventScore[max_score_index,2]
```

```
## [1] "Ram"   "Bheem" "Soam"  "Soam"
```

6. Compute the average score of each events and append it as a new row in the data frame.

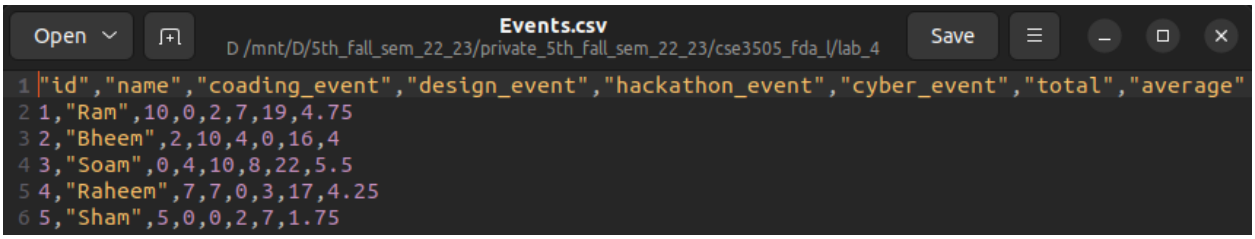
```
eventScore$average<-apply(eventScore[,3:6], 1, mean)
eventScore
```

```
##   id   name coading_event design_event hackathon_event cyber_event total
## 1  1    Ram            10            0               2           7    19
## 2  2  Bheem             2           10               4           0    16
## 3  3   Soam             0            4              10           8    22
## 4  4 Raheem             7            7               0           3    17
## 5  5   Sham             5            0               0           2     7
```

```
##   average
## 1    4.75
## 2    4.00
## 3    5.50
## 4    4.25
## 5    1.75
```

7. Store the details in a comma separated values (csv) file. Also suppress the row numbers.

```
# I am using linux so file path formats are different
write.csv(eventScore,"./Events.csv", row.names = FALSE)
```



8. Read the content of 'Events.csv' in a data frame and view it.

```
eventScore2<-read.csv("./Events.csv")
eventScore2
```

```
##   id   name coading_event design_event hackathon_event cyber_event total
## 1  1    Ram            10            0               2           7    19
## 2  2  Bheem             2           10               4           0    16
## 3  3   Soam             0            4              10           8    22
## 4  4 Raheem             7            7               0           3    17
## 5  5   Sham             5            0               0           2     7
##   average
## 1    4.75
## 2    4.00
## 3    5.50
## 4    4.25
## 5    1.75
```

9. Access the scores of participants in event2 using the column name.

```
eventScore2["design_event"]
```

```
##   design_event
## 1            0
## 2           10
## 3            4
## 4            7
## 5            0
```

10. Use index number to retrieve the same data.

```
eventScore2[4]
```

```
##   design_event
## 1            0
## 2           10
## 3            4
## 4            7
## 5            0
```

11. Extract the score of third participant in event3.

```
eventScore2[3,5]
```

```
## [1] 10
```

12. Extract the scores of the first and second participant in all the events.

```
eventScore[1:2,1:6]
```

```
##   id  name coading_event design_event hackathon_event cyber_event
## 1  1   Ram            10            0               2           7
## 2  2 Bheem             2           10               4           0
```

13. Display the names and total scores of all participants.

```
eventScore[,c(2,7)]
```

```
##     name total
## 1    Ram    19
## 2  Bheem    16
## 3   Soam    22
## 4 Raheem    17
## 5   Sham     7
```

14. Make the column "name" as the row index of the data frame.

```
rownames(eventScore2)<-eventScore$name
eventScore2
```

```
##        id   name coading_event design_event hackathon_event cyber_event total
## Ram     1    Ram            10            0               2           7    19
## Bheem   2  Bheem             2           10               4           0    16
## Soam    3   Soam             0            4              10           8    22
## Raheem  4 Raheem             7            7               0           3    17
## Sham    5   Sham             5            0               0           2     7
##        average
## Ram       4.75
## Bheem     4.00
## Soam      5.50
## Raheem    4.25
## Sham      1.75
```

15. Display the names of the students participated in event3.

```
subset(eventScore2, hackathon_event>0, select = name)
```

```
##        name
## Ram     Ram
## Bheem Bheem
## Soam   Soam
```

16. Obtain the names whose total score is above its average.

```
total_avg=mean(eventScore2$total)
total_avg
```

```
## [1] 16.2
```

```
subset(eventScore2, total>total_avg, select = name)
```

```
##          name
## Ram       Ram
## Soam     Soam
## Raheem Raheem
```

20BCE1025_Abhishek_N_N_Ex-4(b) Basic Operations in data frame

20BCE1025_Abhishek_N_N

2022-09-12

1. Install the package MASS.

```
install.packages("MASS")
```

```
## Installing package into '/home/abhishek_n_n_20bce1025/R/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

2. Import the package MASS.

```
library("MASS")
```

3. Display the structure of the data survey.

```
str(survey)
```

```
## 'data.frame':    237 obs. of  12 variables:
##  $ Sex   : Factor w/ 2 levels "Female","Male": 1 2 2 2 2 1 2 1 2 2 ...
##  $ Wr.Hnd: num  18.5 19.5 18 18.8 20 18 17.7 17 20 18.5 ...
##  $ NW.Hnd: num  18 20.5 13.3 18.9 20 17.7 17.7 17.3 19.5 18.5 ...
##  $ W.Hnd : Factor w/ 2 levels "Left","Right": 2 1 2 2 2 2 2 2 2 2 ...
##  $ Fold  : Factor w/ 3 levels "L on R","Neither",..: 3 3 1 3 2 1 1 3 3 3 ...
##  $ Pulse : int  92 104 87 NA 35 64 83 74 72 90 ...
##  $ Clap  : Factor w/ 3 levels "Left","Neither",..: 1 1 2 2 3 3 3 3 3 3 ...
##  $ Exer  : Factor w/ 3 levels "Freq","None",..: 3 2 2 2 3 3 1 1 3 3 ...
##  $ Smoke : Factor w/ 4 levels "Heavy","Never",..: 2 4 3 2 2 2 2 2 2 2 ...
##  $ Height: num  173 178 NA 160 165 ...
##  $ M.I   : Factor w/ 2 levels "Imperial","Metric": 2 1 NA 2 2 1 1 2 2 2 ...
##  $ Age   : num  18.2 17.6 16.9 20.3 23.7 ...
```

4. Check the class and type of the data set survey in MASS.

```
class(survey)
```

```
## [1] "data.frame"
```

```
typeof(survey)
```

```
## [1] "list"
```

5. Get the number of rows and columns of the survey data frame.

```
nrow(survey)
```

```
## [1] 237
```

```
ncol(survey)
```

```
## [1] 12
```

6. Get the dimension of the survey data frame.

```
dim(survey)
```

```
## [1] 237  12
```

7. Provide the statistical summary of the data frame.

```
summary(survey)
```

```
##      Sex          Wr.Hnd          NW.Hnd        W.Hnd          Fold
##  Female:118   Min.   :13.00   Min.   :12.50   Left : 18   L on R : 99
##  Male  :118   1st Qu.:17.50   1st Qu.:17.50   Right:218   Neither: 18
##  NA's  :  1   Median :18.50   Median :18.50   NA's :  1   R on L :120
##               Mean   :18.67   Mean   :18.58
##               3rd Qu.:19.80   3rd Qu.:19.73
##               Max.   :23.20   Max.   :23.50
##               NA's   :1       NA's   :1
##      Pulse             Clap       Exer       Smoke         Height
##  Min.   : 35.00   Left   : 39   Freq:115   Heavy: 11   Min.   :150.0
##  1st Qu.: 66.00   Neither: 50   None: 24   Never:189   1st Qu.:165.0
##  Median : 72.50   Right  :147   Some: 98   Occas: 19   Median :171.0
##  Mean   : 74.15   NA's   :  1              Regul: 17   Mean   :172.4
##  3rd Qu.: 80.00                            NA's :  1   3rd Qu.:180.0
##  Max.   :104.00                                        Max.   :200.0
##  NA's   :45                                            NA's   :28
##        M.I           Age
##  Imperial: 68   Min.   :16.75
##  Metric  :141   1st Qu.:17.67
##  NA's    : 28   Median :18.58
##                 Mean   :20.37
##                 3rd Qu.:20.17
##                 Max.   :73.00
##
```

8. Display the column names of the survey data frame

```
names(survey)
```

```
##  [1] "Sex"    "Wr.Hnd" "NW.Hnd" "W.Hnd"  "Fold"   "Pulse"  "Clap"   "Exer"
##  [9] "Smoke"  "Height" "M.I"    "Age"
```

9. Retrieve the top 3 rows from the data frame.

```
head(survey,3)
```

```
##      Sex Wr.Hnd NW.Hnd W.Hnd   Fold Pulse    Clap Exer Smoke Height      M.I
## 1 Female   18.5   18.0 Right R on L    92    Left Some Never  173.0   Metric
## 2   Male   19.5   20.5  Left R on L   104    Left None Regul  177.8 Imperial
## 3   Male   18.0   13.3 Right L on R    87 Neither None Occas     NA     <NA>
##      Age
## 1 18.250
## 2 17.583
## 3 16.917
```

10. Extract the bottom 2 rows from the data frame.

```
tail(survey,2)
```

```
##        Sex Wr.Hnd NW.Hnd W.Hnd   Fold Pulse  Clap Exer Smoke Height    M.I
## 236   Male   21.0   21.5 Right R on L    90 Right Some Never  183.0 Metric
## 237 Female   17.6   17.3 Right R on L    85 Right Freq Never  168.5 Metric
##        Age
## 236 17.167
## 237 17.750
```

20BCE1025_Abhishek_N_N_Lab_5

20BCE1025_Abhishek_N_N

13/09/2022

Q1. Create a data frame for the sports day events as shown in Table 1. It records the points scored by three teams namely – Orange, Yellow and Blue Teams in various sports day events. (1Mark)

```
sportsScoreBoard <- data.frame(
  Sports = c(
    "Football",
    "Cricket",
    "Throw Ball",
    "Badminton",
    "Track and Field"
  ),
  Orange = c(50, 25, 19, 30, 23),
  Yellow = c(0, 20, 25, 43, 21),
  Blue = c(20, 45, 26, 21, 0)
)
sportsScoreBoard
```

```
##            Sports Orange Yellow Blue
## 1        Football     50      0   20
## 2         Cricket     25     20   45
## 3      Throw Ball     19     25   26
## 4       Badminton     30     43   21
## 5 Track and Field     23     21    0
```

Write appropriate R code to implement the following:

a) Include a new team "Green" with the score (10,10,30,30,40) (1 Mark)

```
sportsScoreBoard$Green<-c(10,10,30,30,40)
sportsScoreBoard
```

```
##            Sports Orange Yellow Blue Green
## 1        Football     50      0   20    10
## 2         Cricket     25     20   45    10
## 3      Throw Ball     19     25   26    30
## 4       Badminton     30     43   21    30
## 5 Track and Field     23     21    0    40
```

b) Display the scores of Football and Throw ball events. (1 Mark)

```
sportsScoreBoard[c(1,3),]
```

```
##       Sports Orange Yellow Blue Green
## 1   Football     50      0   20    10
## 3 Throw Ball     19     25   26    30
```

c) Find the total points scored by each team and bind the total score with the data set. (2Marks)

```
sportsScoreBoard <- rbind(
  sportsScoreBoard,
  data.frame(
    Sports = "Total",
    Orange = sum(sportsScoreBoard$Orange),
    Yellow = sum(sportsScoreBoard$Yellow),
    Blue = sum(sportsScoreBoard$Blue),
    Green = sum(sportsScoreBoard$Green)
  )
)
sportsScoreBoard
```

```
##            Sports Orange Yellow Blue Green
## 1        Football     50      0   20    10
## 2         Cricket     25     20   45    10
## 3      Throw Ball     19     25   26    30
## 4       Badminton     30     43   21    30
## 5 Track and Field     23     21    0    40
## 6           Total    147    109  112   120
```

d) Display the name of the winning team. (1 Mark)

```
names(sportsScoreBoard)[which.max(sportsScoreBoard[6,2:5])+1]
```

```
## [1] "Orange"
```

e) Find the least points scored by Orange Team and Blue Team. (1 Mark)

```
apply(sportsScoreBoard[1:5,c(2,4)],2,min)
```

```
## Orange   Blue
##     19      0
```

f) Display the favorable game of Yellow Team and Green Team. (1.5 Marks)

```
# assuming favorable game is in which they scored max
sportsScoreBoard[c(apply(sportsScoreBoard[1:5,c(3,5)],2,which.max)),1]
```

```
## [1] Badminton       Track and Field
## Levels: Badminton Cricket Football Throw Ball Track and Field Total
```

g) Display the average score of each event with the corresponding sports name. (1.5 Marks)

```
sportsScoreBoard$average<-apply(sportsScoreBoard[,2:5],1,mean)
sportsScoreBoard
```

```
##            Sports Orange Yellow Blue Green average
## 1        Football     50      0   20    10      20
## 2         Cricket     25     20   45    10      25
## 3      Throw Ball     19     25   26    30      25
## 4       Badminton     30     43   21    30      31
## 5 Track and Field     23     21    0    40      21
## 6           Total    147    109  112   120     122
```

Q2. Assume that a Fall semester registration report contains various fields such as registration no.,name of the student, course code, course name, credits, faculty name and slot.

A) Identify a suitable data structure to store the names of the courses offered in the Fall semester and justify your choice. [1 mark]. ans: data.frame its a report containing various attributes so with data.frame we can easily represent, visualize and compute required things

B) Write an R code to do the following [3 marks]:

i) Create the identified data structure with a sample set of courses.

```
report<-data.frame(
  regno=c(1,2,3,4,5,6,7,8,9,10),
  name=c('a','b','c','d','e','f','g','h','j','k'),
  code=c(10,20,30,40,50,60,70,80,90,100),
  cname=c("c1","c2","c3","c4","c5","c6","c7","c8","c9","c10"),
  credits=c(1,2,3,4,1,2,3,4,1,2),
  fname=c("f1","f2","f3","f4","f5","f6","f7","f8","f9","f10"),
  slot=c("s1","s2","s3","s4","s5","s6","s7","s8","s9","s10")
)
report
```

```
##    regno name code cname credits fname slot
## 1      1    a   10    c1       1    f1   s1
## 2      2    b   20    c2       2    f2   s2
## 3      3    c   30    c3       3    f3   s3
## 4      4    d   40    c4       4    f4   s4
## 5      5    e   50    c5       1    f5   s5
## 6      6    f   60    c6       2    f6   s6
## 7      7    g   70    c7       3    f7   s7
## 8      8    h   80    c8       4    f8   s8
## 9      9    j   90    c9       1    f9   s9
## 10    10    k  100   c10       2   f10  s10
```

ii) Extract the course names in the data structure stored at locations 2, 5 and 7.

```
report[c(2,4,7),4]
```

```
## [1] c2 c4 c7
## Levels: c1 c10 c2 c3 c4 c5 c6 c7 c8 c9
```

iii) Extract all the course names except at locations 1, 2 and 3.

```
report[c(-1,-2,-3),4]
```

```
## [1] c4  c5  c6  c7  c8  c9  c10
## Levels: c1 c10 c2 c3 c4 c5 c6 c7 c8 c9
```

C) Suggest a suitable data structure to store the values of all fields for a student and justify your choice. [2 marks]. ans: data.frame its a record containing various attributes so with data.frame we can easily represent, visualize and compute required things

D) Write an R code to do the following: [4 marks]

E) Create the identified data structure with a sample values.

```
studentRecord<-data.frame(
  regno=c(1,2,3,4,5,6,7,8,9,10),
  name=c('a','b','c','d','e','f','g','h','j','k'),
  code=c(10,20,30,40,50,60,70,80,90,100),
  cname=c("c1","c2","c3","c4","c5","c6","c7","c8","c9","c10"),
  credits=c(1,2,3,4,1,2,3,4,1,2),
  fname=c("f1","f2","f3","f4","f5","f6","f7","f8","f9","f10"),
  slot=c("s1","s2","s3","s4","s5","s6","s7","s8","s9","s10")
)
studentRecord
```

```
##    regno name code cname credits fname slot
## 1      1    a   10    c1       1    f1   s1
## 2      2    b   20    c2       2    f2   s2
## 3      3    c   30    c3       3    f3   s3
## 4      4    d   40    c4       4    f4   s4
## 5      5    e   50    c5       1    f5   s5
## 6      6    f   60    c6       2    f6   s6
## 7      7    g   70    c7       3    f7   s7
## 8      8    h   80    c8       4    f8   s8
## 9      9    j   90    c9       1    f9   s9
## 10    10    k  100   c10       2   f10  s10
```

II) Assign names to all the values stored in the data structure

```
rownames(studentRecord)<-c("record1","record2","record3","record4","record5","record6","record7","recor
studentRecord
```

```
##          regno name code cname credits fname slot
## record1      1    a   10    c1       1    f1   s1
## record2      2    b   20    c2       2    f2   s2
```

```
## record3      3    c   30    c3       3    f3   s3
## record4      4    d   40    c4       4    f4   s4
## record5      5    e   50    c5       1    f5   s5
## record6      6    f   60    c6       2    f6   s6
## record7      7    g   70    c7       3    f7   s7
## record8      8    h   80    c8       4    f8   s8
## record9      9    j   90    c9       1    f9   s9
## record10    10    k  100   c10       2   f10  s10
```

III) Extract the name of the student, course name, credits and slot in any 2 ways.

```
studentRecord[,c(2,4,5)]
```

```
##          name cname credits
## record1     a    c1       1
## record2     b    c2       2
## record3     c    c3       3
## record4     d    c4       4
## record5     e    c5       1
## record6     f    c6       2
## record7     g    c7       3
## record8     h    c8       4
## record9     j    c9       1
## record10    k   c10       2
```

Q3. As a data analyst you have been provided with the following matrix

Write a R code snippet to do the following (5x1=5 Marks)

i) Create the Matrix and display the matrix

```
peopleMatrix <-
  matrix(
    c(
      "Satchin",
      "Virat",
      "Rohit",
      "Dhoni",
      "Amitab",
      "Amir",
      "Akhya",
      "Salman",
      "Modi",
      "Amit",
      "Rahul",
      "Neharu",
      "Delhi",
      "Chennai",
      "Kolkata",
      "Mumbai"
```

```
    ),
    nrow = 4,
    ncol = 4
  )
peopleMatrix
```

```
##      [,1]      [,2]     [,3]     [,4]
## [1,] "Satchin" "Amitab" "Modi"   "Delhi"
## [2,] "Virat"   "Amir"   "Amit"   "Chennai"
## [3,] "Rohit"   "Akhya"  "Rahul"  "Kolkata"
## [4,] "Dhoni"   "Salman" "Neharu" "Mumbai"
```

ii) Define the column names as Players, Actors, Politicians, Metro city

```
colnames(peopleMatrix)<-c("Players","Actors","Politicians","Metro city")
peopleMatrix
```

```
##      Players   Actors   Politicians Metro city
## [1,] "Satchin" "Amitab" "Modi"      "Delhi"
## [2,] "Virat"   "Amir"   "Amit"      "Chennai"
## [3,] "Rohit"   "Akhya"  "Rahul"     "Kolkata"
## [4,] "Dhoni"   "Salman" "Neharu"    "Mumbai"
```

iii) Define the row names as Record1, Record2, Record3, Record4

```
rownames(peopleMatrix)<-c("Record1","Record2","Record3","Record4")
peopleMatrix
```

```
##         Players   Actors   Politicians Metro city
## Record1 "Satchin" "Amitab" "Modi"      "Delhi"
## Record2 "Virat"   "Amir"   "Amit"      "Chennai"
## Record3 "Rohit"   "Akhya"  "Rahul"     "Kolkata"
## Record4 "Dhoni"   "Salman" "Neharu"    "Mumbai"
```

iv) Display only the names of cricket players and Politicians

```
peopleMatrix[,c(1,3)]
```

```
##         Players   Politicians
## Record1 "Satchin" "Modi"
## Record2 "Virat"   "Amit"
## Record3 "Rohit"   "Rahul"
## Record4 "Dhoni"   "Neharu"
```

v) Display the records that contain Kolkata and Mumbai

```
peopleMatrix[c(3,4),]
```

```
##         Players Actors   Politicians Metro city
## Record3 "Rohit" "Akhya"  "Rahul"     "Kolkata"
## Record4 "Dhoni" "Salman" "Neharu"    "Mumbai"
```

Q4. As a data analyst, you have been asked to retrieve the following information from the given string "8/08/2022". Write an R code snippet to do the following (5x1=5 Marks)

i) Display the given string in date format

```
d1<-as.Date("8/08/2022",format="%d/%m/%Y")
d1
```

```
## [1] "2022-08-08"
```

ii) As a data analyst extract from the date object, the day number in that year, for example, 8/25/2022 is 237th day in this year.

```
format(d1,format="%j")
```

```
## [1] "220"
```

iii) As a data analyst extract from the date object, the month number in that year. For example, 08/25/2022 has the month number 08.

```
format(d1,format="%m")
```

```
## [1] "08"
```

iv) Convert the following dates which are in string format to date format "12/11/2010", "13/12/1990","30/1/2001" and "15/08/2022" and display the dates on the console.

```
ds1<-c("12/11/2010","13/12/1990","30/1/2001","15/08/2022")
d2<-as.Date(ds1,format="%d/%m/%Y")
d2
```

```
## [1] "2010-11-12" "1990-12-13" "2001-01-30" "2022-08-15"
```

```
class(d2)
```

```
## [1] "Date"
```

v) Extract two dates "30/01/2001" and "12/11/2010" from the above created object and calculate the difference between the given dates in terms of days and months and display the results.

```r
difftime(d2[3],d2[1], units = "days")
```

```
## Time difference of -3573 days
```

```r
# direct months argument in units is not accepted so
print(paste("Time difference of",as.double(difftime(d2[3],d2[1], units = "days"))*12/365,"months",sep="
```

```
## [1] "Time difference of -117.468493150685 months"
```

20BCE1025_Abhishek_N_N_Lab_6_Exp-6 Statistical Analysis of qualitative data

20BCE1025 Abhishek N N

20/09/2022

Use the survey data in MASS package to do the following:

1. Import the package MASS

```
library("MASS")
```

2. List the rows of data that has missing values.

```
survey[rowSums(is.na(survey)) > 0,]
```

```
##        Sex Wr.Hnd NW.Hnd W.Hnd    Fold Pulse    Clap Exer Smoke Height      M.I
## 3     Male   18.0   13.3 Right  L on R    87 Neither None Occas     NA     <NA>
## 4     Male   18.8   18.9 Right  R on L    NA Neither None Never 160.00   Metric
## 12    Male   21.0   21.0 Right  R on L    68    Left Freq Never     NA     <NA>
## 13  Female   16.0   16.0 Right  L on R    NA   Right Some Never 155.00   Metric
## 15    Male   16.0   15.5 Right  R on L    60   Right Some Never     NA     <NA>
## 16  Female   17.5   17.0 Right  R on L    NA   Right Freq Never 156.00   Metric
## 19    Male   20.5   20.5 Right  L on R    NA    Left Some Never 190.50 Imperial
## 25  Female   17.0   17.5 Right  R on L    64    Left Some Never     NA     <NA>
## 26    Male   18.5   18.5 Right Neither    90 Neither Some Never     NA     <NA>
## 29    Male   17.8   17.8 Right  L on R    76 Neither Freq Never     NA     <NA>
## 31  Female   18.5   18.0 Right  R on L    76   Right None Occas     NA     <NA>
## 35    Male   18.0   19.0 Right  L on R    54 Neither Some Regul     NA     <NA>
## 37  Female   16.0   16.5 Right  L on R    NA   Right Some Never 168.00   Metric
## 40    Male   19.0   19.0 Right  R on L    NA Neither Freq Occas 171.00   Metric
## 41  Female   17.5   16.0 Right  L on R    NA   Right Some Never 169.00   Metric
## 43    Male     NA     NA Right  R on L    60    <NA> Some Never 172.00   Metric
## 45  Female   13.0   13.0  <NA>  L on R    70    Left Freq Never 180.34 Imperial
## 46    Male   17.0   17.5 Right  R on L    NA Neither Freq Never 180.34 Imperial
## 56    Male   18.5   18.5 Right  L on R    NA Neither Freq Never 171.00   Metric
## 58    Male   19.5   19.7 Right  R on L    72   Right Freq Never     NA     <NA>
## 60    Male   20.6   21.0  Left  L on R    NA    Left Freq Occas 175.26 Imperial
## 64  Female   18.7   18.0  Left  L on R    NA    Left None Never 170.00   Metric
## 66    Male   19.5   19.8 Right Neither    NA   Right Freq Never 183.00   Metric
## 67  Female   19.0   19.1 Right  L on R    NA Neither Freq Never 172.00   Metric
## 68  Female   18.5   18.0 Right  R on L    64   Right Freq Never     NA     <NA>
## 69    Male   19.0   19.0 Right  L on R    NA   Right Some Never 180.00   Metric
## 70    Male   21.0   19.5 Right  L on R    80    Left None  <NA>     NA     <NA>
```

```
## 72    Male   19.4   19.5 Right  R on L    NA   Right Freq Heavy 176.00   Metric
## 78  Female   18.6   18.0 Right  L on R    NA Neither Freq Heavy 165.10 Imperial
## 80    Male   20.0   20.5 Right  L on R    NA   Right Freq Never 185.42 Imperial
## 81    Male   19.5   19.5  Left  R on L    66    Left Some Never     NA     <NA>
## 83  Female   17.5   17.5 Right  R on L    98    Left Freq Never     NA     <NA>
## 84  Female   17.0   17.4 Right  R on L    NA Neither Some Never     NA     <NA>
## 90  Female   18.0   17.7  Left  R on L    92    Left Some Never     NA     <NA>
## 92  Female   17.5   18.0 Right Neither    NA   Right Some Never     NA     <NA>
## 94  Female   18.2   18.5 Right  R on L    NA   Right Some Never 168.00   Metric
## 96  Female   19.0   18.8 Right  L on R    NA   Right Some Never     NA     <NA>
## 99    Male   19.5   19.4 Right Neither    NA   Right Freq Never 165.00   Metric
## 101   Male   21.9   22.2 Right  R on L    NA   Right Some Never 187.00   Metric
## 103 Female   16.0   16.0 Right Neither    NA   Right Some Never 159.00   Metric
## 107 Female   16.2   16.4 Right  R on L    NA   Right Freq Occas 172.00   Metric
## 108 Female   17.0   15.9 Right  R on L    85   Right Freq Never     NA     <NA>
## 121   Male   20.0   20.0 Right  R on L    80 Neither Freq Occas     NA     <NA>
## 126   Male   19.3   19.4 Right  R on L    NA   Right Freq Never 180.34 Imperial
## 133 Female   18.9   20.0 Right  R on L    86   Right Some Never     NA     <NA>
## 137   <NA>   19.8   19.0  Left  L on R    73 Neither Freq Never 172.00   Metric
## 139   Male   20.0   19.5 Right  L on R    NA   Right Freq Never 170.00   Metric
## 142 Female   18.3   19.0 Right  R on L    NA   Right None Never 165.00   Metric
## 157   Male   14.0   15.5 Right  L on R    NA Neither Freq Heavy     NA     <NA>
## 159   Male   20.0   20.5 Right  R on L    NA   Right None Never 187.96 Imperial
## 162   Male   18.1   18.2  Left Neither    NA   Right Some Never 168.00   Metric
## 165   Male   19.1   19.1 Right Neither    NA   Right Some Never 177.00   Metric
## 169   Male   19.0   18.5 Right  L on R    NA Neither Freq Never 189.00   Metric
## 171 Female   16.5   17.0 Right  L on R    NA   Right Some Never 168.00   Metric
## 173 Female   15.5   15.5 Right Neither    50   Right Some Regul     NA     <NA>
## 179 Female   20.5   20.5 Right  R on L    NA    Left Freq Regul     NA     <NA>
## 195 Female   16.7   15.1 Right Neither    NA   Right None Never 157.48 Imperial
## 203 Female   18.8   17.8 Right  R on L    76   Right Some Never     NA     <NA>
## 210 Female   20.8   20.7 Right  R on L    NA Neither Freq Never 171.50   Metric
## 213   Male   18.0   18.5 Right  R on L    78   Right Freq Never     NA     <NA>
## 216   Male   19.5   20.0 Right Neither    NA   Right Some Never 170.00   Metric
## 217 Female   16.3   16.2 Right  L on R    NA   Right None Never     NA     <NA>
## 219 Female   17.0   17.3 Right  L on R    NA Neither Freq Never 173.00   Metric
## 221   Male   23.2   23.3 Right  L on R    NA   Right None Heavy 171.00   Metric
## 224 Female   17.5   17.6 Right  L on R    NA   Right Freq Never 150.00   Metric
## 225 Female   17.6   17.2 Right  L on R    NA   Right Some Never     NA     <NA>
## 226 Female   17.5   17.8 Right  R on L    96   Right Some Never     NA     <NA>
## 232   Male   18.0   16.0 Right  R on L    NA   Right Some Never 180.34 Imperial
## 235 Female   17.5   16.5 Right  R on L    NA   Right Some Never 170.00   Metric
##        Age
## 3   16.917
## 4   20.333
## 12  18.250
## 13  18.750
## 15  17.167
## 16  17.167
## 19  19.750
## 25  19.167
## 26  17.583
## 29  21.917
## 31  41.583
```

```
## 35  17.750
## 37  19.000
## 40  19.917
## 41  17.500
## 43  28.583
## 45  17.417
## 46  18.500
## 56  18.333
## 58  17.417
## 60  18.417
## 64  19.833
## 66  18.000
## 67  30.667
## 68  16.917
## 69  19.917
## 70  18.333
## 72  17.833
## 78  17.167
## 80  18.750
## 81  16.750
## 83  17.667
## 84  17.167
## 90  17.583
## 92  18.000
## 94  17.083
## 96  17.083
## 99  18.083
## 101 18.917
## 103 20.833
## 107 17.000
## 108 18.500
## 121 17.500
## 126 19.833
## 133 19.083
## 137 21.500
## 139 21.417
## 142 21.083
## 157 21.083
## 159 19.667
## 162 21.167
## 165 19.917
## 169 17.417
## 171 73.000
## 173 18.500
## 179 19.250
## 195 18.167
## 203 18.583
## 210 18.500
## 213 17.500
## 216 21.250
## 217 19.250
## 219 19.167
## 221 20.917
## 224 20.750
```

```
## 225 19.917
## 226 18.667
## 232 20.750
## 235 18.583
```

```
#or
survey[!complete.cases(survey),]
```

```
##        Sex Wr.Hnd NW.Hnd W.Hnd    Fold Pulse    Clap Exer Smoke Height      M.I
## 3     Male   18.0   13.3 Right  L on R    87 Neither None Occas     NA     <NA>
## 4     Male   18.8   18.9 Right  R on L    NA Neither None Never 160.00   Metric
## 12    Male   21.0   21.0 Right  R on L    68    Left Freq Never     NA     <NA>
## 13  Female   16.0   16.0 Right  L on R    NA   Right Some Never 155.00   Metric
## 15    Male   16.0   15.5 Right  R on L    60   Right Some Never     NA     <NA>
## 16  Female   17.5   17.0 Right  R on L    NA   Right Freq Never 156.00   Metric
## 19    Male   20.5   20.5 Right  L on R    NA    Left Some Never 190.50 Imperial
## 25  Female   17.0   17.5 Right  R on L    64    Left Some Never     NA     <NA>
## 26    Male   18.5   18.5 Right Neither    90 Neither Some Never     NA     <NA>
## 29    Male   17.8   17.8 Right  L on R    76 Neither Freq Never     NA     <NA>
## 31  Female   18.5   18.0 Right  R on L    76   Right None Occas     NA     <NA>
## 35    Male   18.0   19.0 Right  L on R    54 Neither Some Regul     NA     <NA>
## 37  Female   16.0   16.5 Right  L on R    NA   Right Some Never 168.00   Metric
## 40    Male   19.0   19.0 Right  R on L    NA Neither Freq Occas 171.00   Metric
## 41  Female   17.5   16.0 Right  L on R    NA   Right Some Never 169.00   Metric
## 43    Male     NA     NA Right  R on L    60    <NA> Some Never 172.00   Metric
## 45  Female   13.0   13.0  <NA>  L on R    70    Left Freq Never 180.34 Imperial
## 46    Male   17.0   17.5 Right  R on L    NA Neither Freq Never 180.34 Imperial
## 56    Male   18.5   18.5 Right  L on R    NA Neither Freq Never 171.00   Metric
## 58    Male   19.5   19.7 Right  R on L    72   Right Freq Never     NA     <NA>
## 60    Male   20.6   21.0  Left  L on R    NA    Left Freq Occas 175.26 Imperial
## 64  Female   18.7   18.0  Left  L on R    NA    Left None Never 170.00   Metric
## 66    Male   19.5   19.8 Right Neither    NA   Right Freq Never 183.00   Metric
## 67  Female   19.0   19.1 Right  L on R    NA Neither Freq Never 172.00   Metric
## 68  Female   18.5   18.0 Right  R on L    64   Right Freq Never     NA     <NA>
## 69    Male   19.0   19.0 Right  L on R    NA   Right Some Never 180.00   Metric
## 70    Male   21.0   19.5 Right  L on R    80    Left None  <NA>     NA     <NA>
## 72    Male   19.4   19.5 Right  R on L    NA   Right Freq Heavy 176.00   Metric
## 78  Female   18.6   18.0 Right  L on R    NA Neither Freq Heavy 165.10 Imperial
## 80    Male   20.0   20.5 Right  L on R    NA   Right Freq Never 185.42 Imperial
## 81    Male   19.5   19.5  Left  R on L    66    Left Some Never     NA     <NA>
## 83  Female   17.5   17.5 Right  R on L    98    Left Freq Never     NA     <NA>
## 84  Female   17.0   17.4 Right  R on L    NA Neither Some Never     NA     <NA>
## 90  Female   18.0   17.7  Left  R on L    92    Left Some Never     NA     <NA>
## 92  Female   17.5   18.0 Right Neither    NA   Right Some Never     NA     <NA>
## 94  Female   18.2   18.5 Right  R on L    NA   Right Some Never 168.00   Metric
## 96  Female   19.0   18.8 Right  L on R    NA   Right Some Never     NA     <NA>
## 99    Male   19.5   19.4 Right Neither    NA   Right Freq Never 165.00   Metric
## 101   Male   21.9   22.2 Right  R on L    NA   Right Some Never 187.00   Metric
## 103 Female   16.0   16.0 Right Neither    NA   Right Some Never 159.00   Metric
## 107 Female   16.2   16.4 Right  R on L    NA   Right Freq Occas 172.00   Metric
## 108 Female   17.0   15.9 Right  R on L    85   Right Freq Never     NA     <NA>
## 121   Male   20.0   20.0 Right  R on L    80 Neither Freq Occas     NA     <NA>
## 126   Male   19.3   19.4 Right  R on L    NA   Right Freq Never 180.34 Imperial
## 133 Female   18.9   20.0 Right  R on L    86   Right Some Never     NA     <NA>
```

```
## 137   <NA>   19.8   19.0  Left  L on R    73 Neither Freq Never 172.00   Metric
## 139   Male   20.0   19.5 Right  L on R    NA   Right Freq Never 170.00   Metric
## 142 Female   18.3   19.0 Right  R on L    NA   Right None Never 165.00   Metric
## 157   Male   14.0   15.5 Right  L on R    NA Neither Freq Heavy     NA     <NA>
## 159   Male   20.0   20.5 Right  R on L    NA   Right None Never 187.96 Imperial
## 162   Male   18.1   18.2  Left Neither    NA   Right Some Never 168.00   Metric
## 165   Male   19.1   19.1 Right Neither    NA   Right Some Never 177.00   Metric
## 169   Male   19.0   18.5 Right  L on R    NA Neither Freq Never 189.00   Metric
## 171 Female   16.5   17.0 Right  L on R    NA   Right Some Never 168.00   Metric
## 173 Female   15.5   15.5 Right Neither    50   Right Some Regul     NA     <NA>
## 179 Female   20.5   20.5 Right  R on L    NA    Left Freq Regul     NA     <NA>
## 195 Female   16.7   15.1 Right Neither    NA   Right None Never 157.48 Imperial
## 203 Female   18.8   17.8 Right  R on L    76   Right Some Never     NA     <NA>
## 210 Female   20.8   20.7 Right  R on L    NA Neither Freq Never 171.50   Metric
## 213   Male   18.0   18.5 Right  R on L    78   Right Freq Never     NA     <NA>
## 216   Male   19.5   20.0 Right Neither    NA   Right Some Never 170.00   Metric
## 217 Female   16.3   16.2 Right  L on R    NA   Right None Never     NA     <NA>
## 219 Female   17.0   17.3 Right  L on R    NA Neither Freq Never 173.00   Metric
## 221   Male   23.2   23.3 Right  L on R    NA   Right None Heavy 171.00   Metric
## 224 Female   17.5   17.6 Right  L on R    NA   Right Freq Never 150.00   Metric
## 225 Female   17.6   17.2 Right  L on R    NA   Right Some Never     NA     <NA>
## 226 Female   17.5   17.8 Right  R on L    96   Right Some Never     NA     <NA>
## 232   Male   18.0   16.0 Right  R on L    NA   Right Some Never 180.34 Imperial
## 235 Female   17.5   16.5 Right  R on L    NA   Right Some Never 170.00   Metric
##        Age
## 3   16.917
## 4   20.333
## 12  18.250
## 13  18.750
## 15  17.167
## 16  17.167
## 19  19.750
## 25  19.167
## 26  17.583
## 29  21.917
## 31  41.583
## 35  17.750
## 37  19.000
## 40  19.917
## 41  17.500
## 43  28.583
## 45  17.417
## 46  18.500
## 56  18.333
## 58  17.417
## 60  18.417
## 64  19.833
## 66  18.000
## 67  30.667
## 68  16.917
## 69  19.917
## 70  18.333
## 72  17.833
## 78  17.167
```

```
## 80  18.750
## 81  16.750
## 83  17.667
## 84  17.167
## 90  17.583
## 92  18.000
## 94  17.083
## 96  17.083
## 99  18.083
## 101 18.917
## 103 20.833
## 107 17.000
## 108 18.500
## 121 17.500
## 126 19.833
## 133 19.083
## 137 21.500
## 139 21.417
## 142 21.083
## 157 21.083
## 159 19.667
## 162 21.167
## 165 19.917
## 169 17.417
## 171 73.000
## 173 18.500
## 179 19.250
## 195 18.167
## 203 18.583
## 210 18.500
## 213 17.500
## 216 21.250
## 217 19.250
## 219 19.167
## 221 20.917
## 224 20.750
## 225 19.917
## 226 18.667
## 232 20.750
## 235 18.583
```

3. Create a data frame 'newsurvey' that contains the survey data after removing the na values. Use it for answering further queries

```
newsurvey<-na.omit(survey)
```

4. How many male and female students participated in the survey?

```
table(newsurvey["Sex"])
```

```
##
## Female   Male
##     84     84
```

5. How many the left and right handers are there?

```
table(newsurvey["W.Hnd"])
```

```
##
##  Left Right
##    12   156
```

6. Find the relative frequency distribution of left and right handers and display them with the precision of two decimal places.

```
round(table(newsurvey["W.Hnd"])/length(newsurvey$W.Hnd),2)
```

```
##
##  Left Right
##  0.07  0.93
```

7. Display the male left hander and female left hander in the column format.

```
t <- tapply(newsurvey$W.Hnd, newsurvey$Sex, table)
t<-unname(unlist(t))
d <-
  data.frame(
    male_left_hander = c(t[3]),
    female_left_hander = c(t[1])
  )
d
```

```
##   male_left_hander female_left_hander
## 1                7                  5
```

8. What percentage of male left handers never smokes?

```
install.packages("dplyr")
```

```
## Installing package into '/home/abhishek_n_n_20bce1025/R/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```
library("dplyr")
```

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:MASS':
##
##     select

## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
male_left_never=length(filter(newsurvey, Sex=="Male" & W.Hnd=="Left" & Smoke=="Never"))
male_left=length(filter(newsurvey, Sex=="Male" & W.Hnd=="Left"))
male_left_never/male_left*100
```

```
## [1] 100
```

20BCE1025_Abhishek_N_N_Exp-7 Statistical Analysis of quantitative data

20BCE1025_Abhishek_N_N

2022-10-03

Use the newsurvey data obtained by cleaning 'na' values in survey data of MASS package to do the following:

```
#install.packages("MASS")
library(MASS)
newSurvey = na.omit(survey)
```

1. Find the range of students' age participated in the survey.

```
age=newSurvey$Age
range(age)
```

```
## [1] 16.917 70.417
```

2. Break the age range into non-overlapping sub-intervals by defining a sequence of equal distance break points of 10 by rounding the range to nearest integer.

```
breaks = seq(16.917, 70.417, by=10)
breaks
```

```
## [1] 16.917 26.917 36.917 46.917 56.917 66.917
```

3. Find the distribution of the age range according to the sub-intervals with cut with its right boundary opened. Display it in column form.

```
age.cut = cut(age, breaks, right=FALSE)
age.freq = table(age.cut)
age.freq=cbind(age.freq)
```

4. Which age range of students has mostly participated in the survey.

```
age.freq[which.max(age.freq)]
```

```
## [1] 155
```

5. Similarly, find the frequency distribution of Wr.Hnd span and display it in column format.

```r
wrHnd=newSurvey$Wr.Hnd
range(wrHnd)
```

```
## [1] 13.0 23.2
```

```r
breaks = seq(13.0, 23.2, by=2)
breaks
```

```
## [1] 13 15 17 19 21 23
```

```r
wrHnd.cut = cut(wrHnd, breaks, right=FALSE)
wrHnd.freq = table(wrHnd.cut)
wrHnd.freq=cbind(wrHnd.freq)
wrHnd.freq
```

```
##         wrHnd.freq
## [13,15)          2
## [15,17)         16
## [17,19)         82
## [19,21)         41
## [21,23)         22
```

6. Find the relative frequency of Wr.Hnd and display it by correcting to 3 decimal places.

```r
wrHnd.relfreq = wrHnd.freq / nrow(newSurvey)
round(wrHnd.relfreq, 3)
```

```
##         wrHnd.freq
## [13,15)      0.012
## [15,17)      0.095
## [17,19)      0.488
## [19,21)      0.244
## [21,23)      0.131
```

20BCE1025_Abhishek_N_N_Experiment_8_Visualization using basic graphics

20BCE1025_Abhishek_N_N

2022-11-03

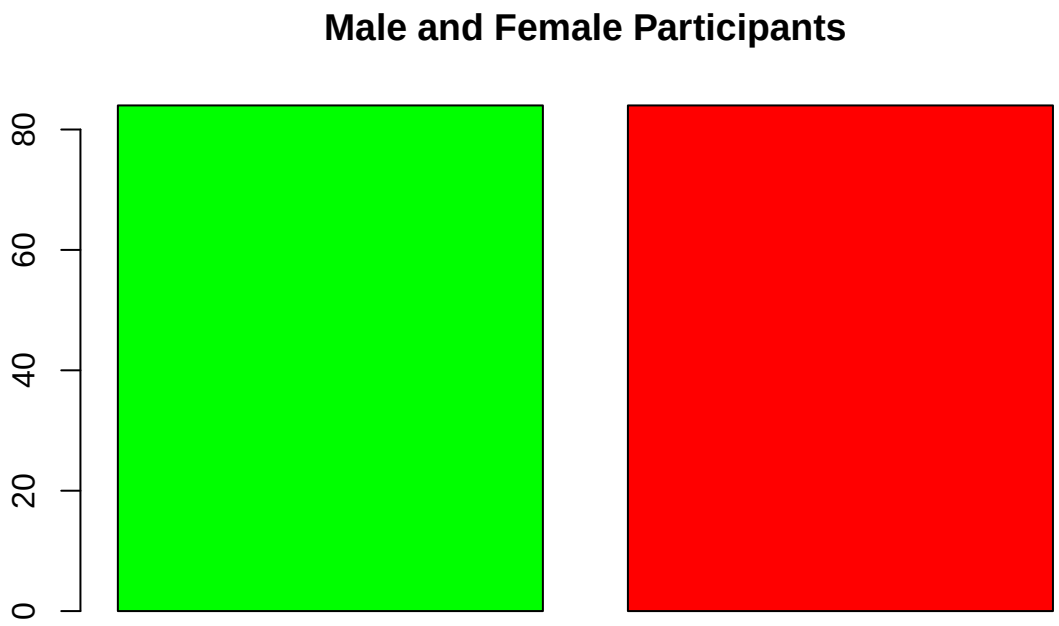
Use the newsurvey data obtained by cleaning 'na' values in survey data of MASS package to do the following:

```
library("MASS")
cleansurvey=survey[complete.cases(survey),]
head(cleansurvey)
```

```
##      Sex Wr.Hnd NW.Hnd W.Hnd    Fold Pulse  Clap Exer Smoke Height      M.I
## 1 Female   18.5   18.0 Right  R on L    92  Left Some Never 173.00   Metric
## 2   Male   19.5   20.5  Left  R on L   104  Left None Regul 177.80 Imperial
## 5   Male   20.0   20.0 Right Neither    35 Right Some Never 165.00   Metric
## 6 Female   18.0   17.7 Right  L on R    64 Right Some Never 172.72 Imperial
## 7   Male   17.7   17.7 Right  L on R    83 Right Freq Never 182.88 Imperial
## 8 Female   17.0   17.3 Right  R on L    74 Right Freq Never 157.00   Metric
##      Age
## 1 18.250
## 2 17.583
## 5 23.667
## 6 21.000
## 7 18.833
## 8 35.833
```

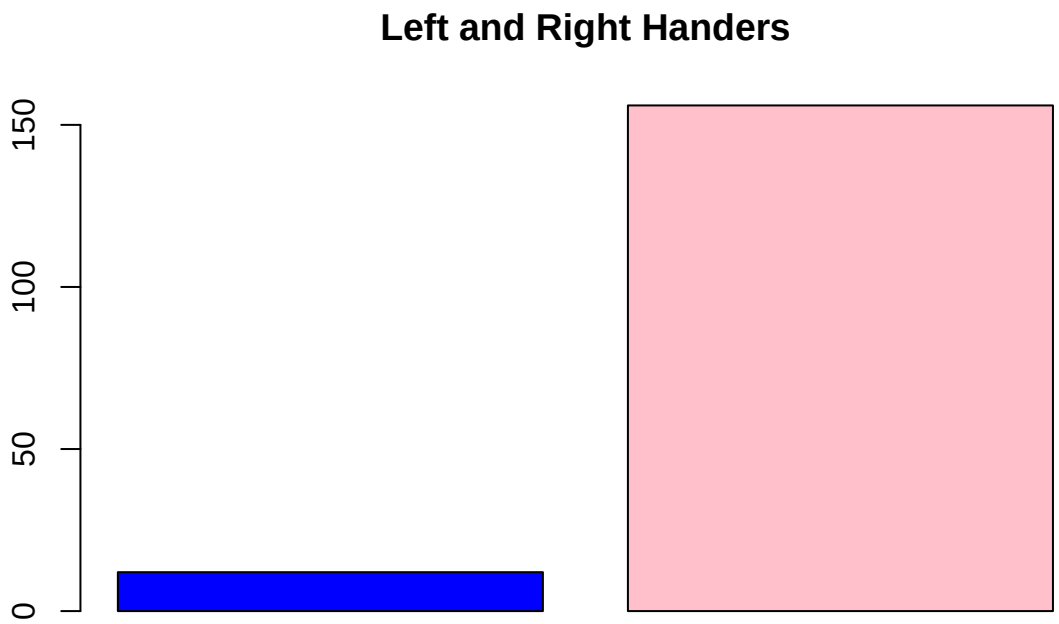
1. Plot a bar graph for the number of male and female participants in the survey. Provide the titleas "Male and Female participants" and specify the colours for the bars.

```
x=c(sum(cleansurvey$Sex=="Male"),sum(cleansurvey$Sex=="Female"))
barplot(x,main="Male and Female Participants",col=c("green","red"))
```



2. Plot a bar graph for the number of left handers and right handers in the survey. Provide the title as "Left Handers and Right Hnaders" and specify the colours for the bars.

```
x=c(sum(cleansurvey$W.Hnd=="Left"),sum(cleansurvey$W.Hnd=="Right"))
barplot(x,main="Left and Right Handers",col=c("blue","pink"))
```

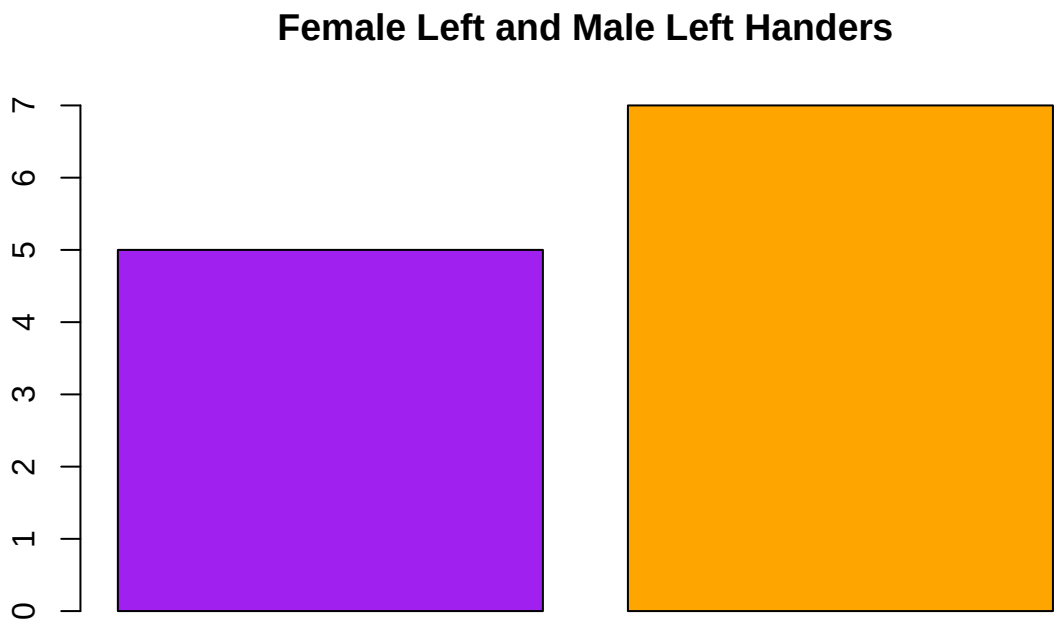


3. Plot the distribution between male left handers and female left handers using bar chart. Provide the title as “Female Left Handers and Male Left Handers” and specify the colours for the bars.

```
library(dplyr)
```

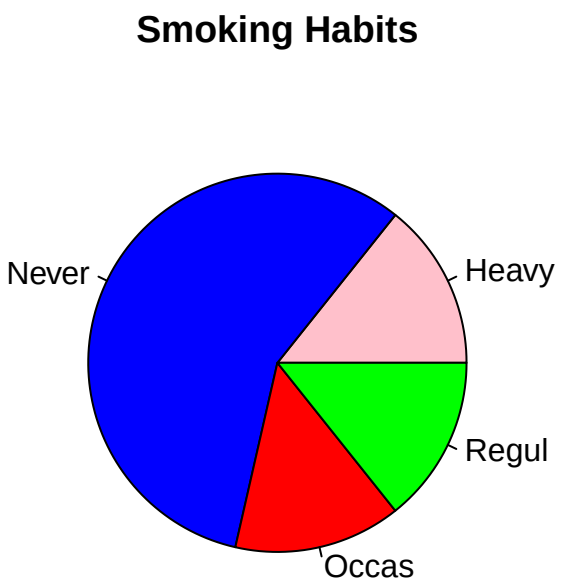
```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:MASS':
##
##     select

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
x=filter(cleansurvey,Sex=="Female")
y=filter(cleansurvey,Sex=="Male")
x=c(sum(x$W.Hnd=="Left"),sum(y$W.Hnd=="Left"))
barplot(x,main="Female Left and Male Left Handers",col=c("purple","orange"))
```



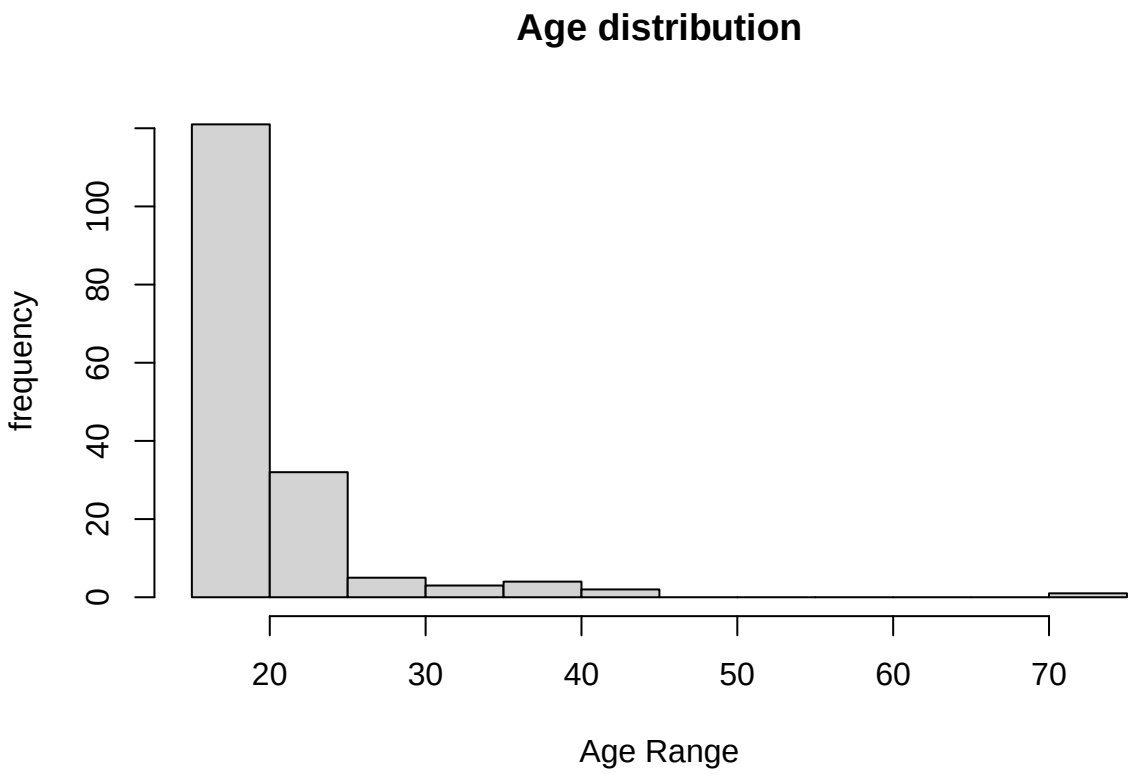
4. Draw the distribution of smoking habits of male left handers using pie chart.

```
y=filter(cleansurvey,Sex=="Male")
y=filter(y,W.Hnd=="Left")
pie(table(y$Smoke),col = c("pink","blue","red","green"),main="Smoking Habits")
```



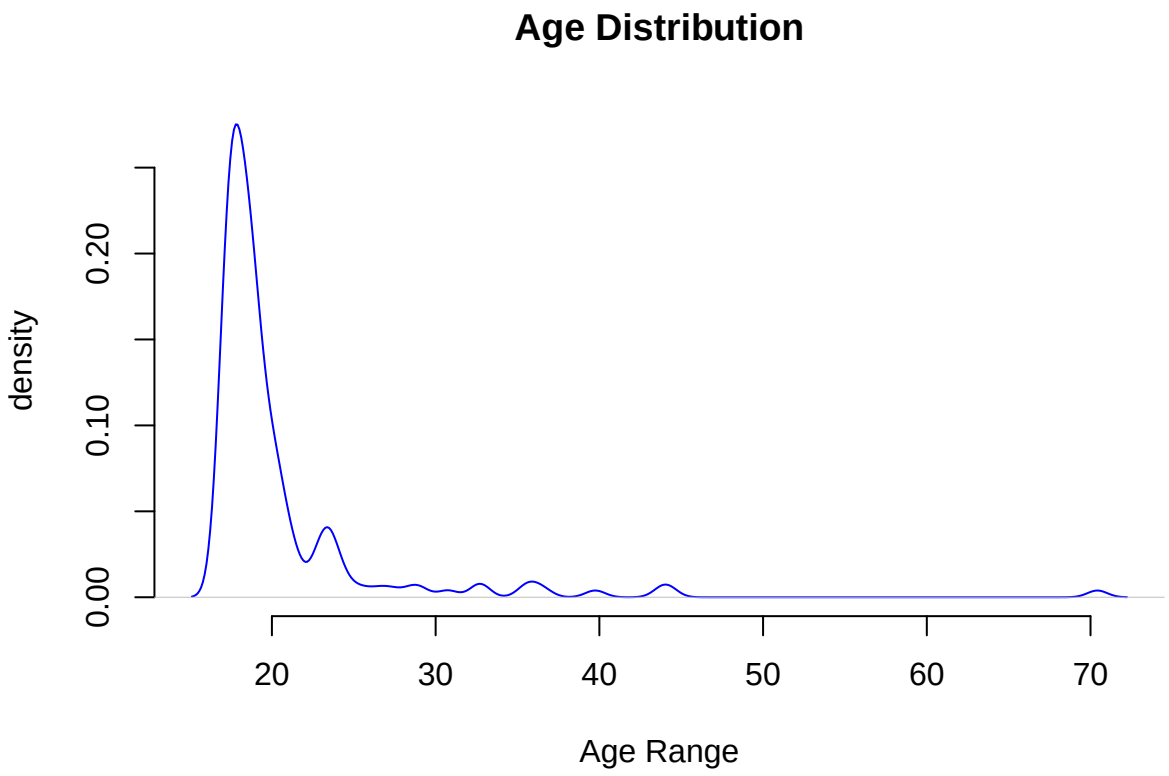
5.Draw the histogram of age distribution with the title as 'Age distribution' and xlabel as 'Age range' and ylabel as 'frequency'.

```
hist(cleansurvey$Age,main="Age distribution",xlab="Age Range",ylab="frequency")
```



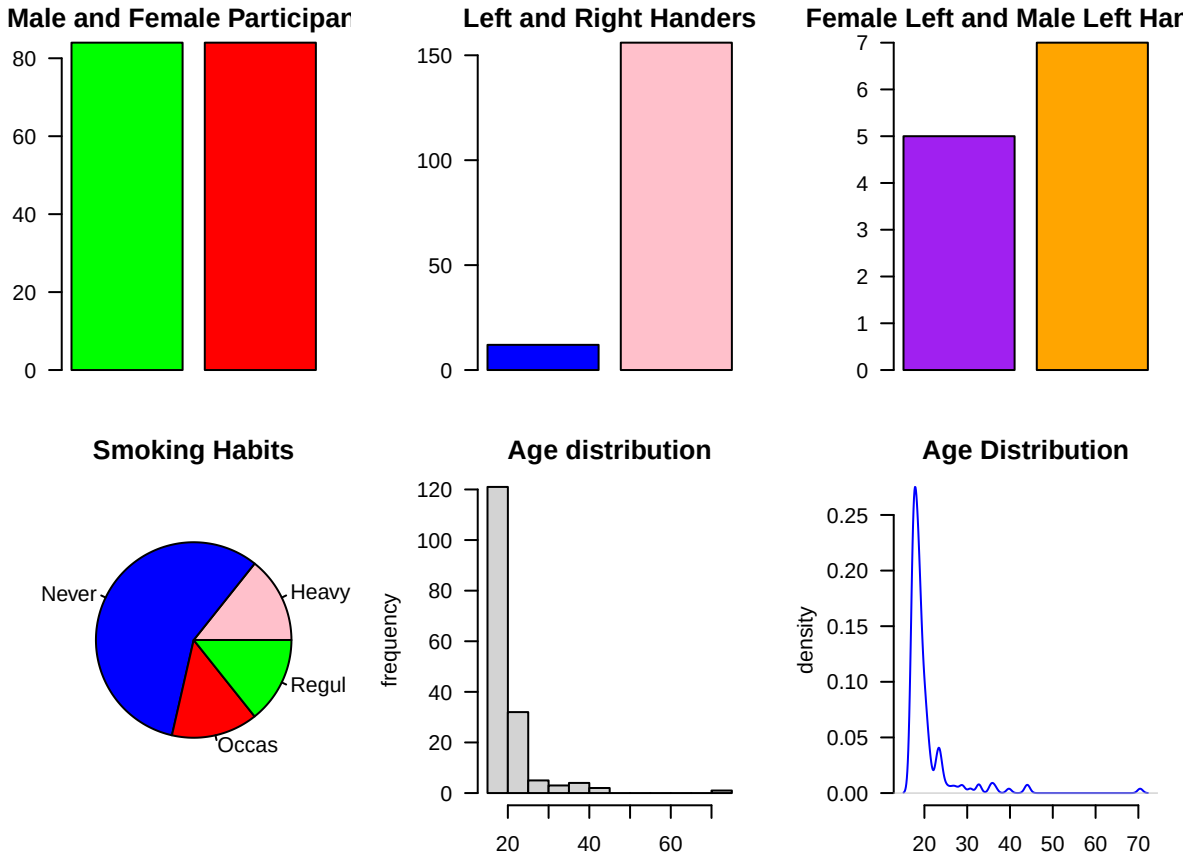
6. Plot the density distribution of age distribution with title as 'Age distribution' and xlabel as 'Age range' and ylabel as 'density'.

```
den <- density(cleansurvey$Age)
plot(den, frame = FALSE, col = "blue",main='Age Distribution',xlab="Age Range",ylab="density")
```



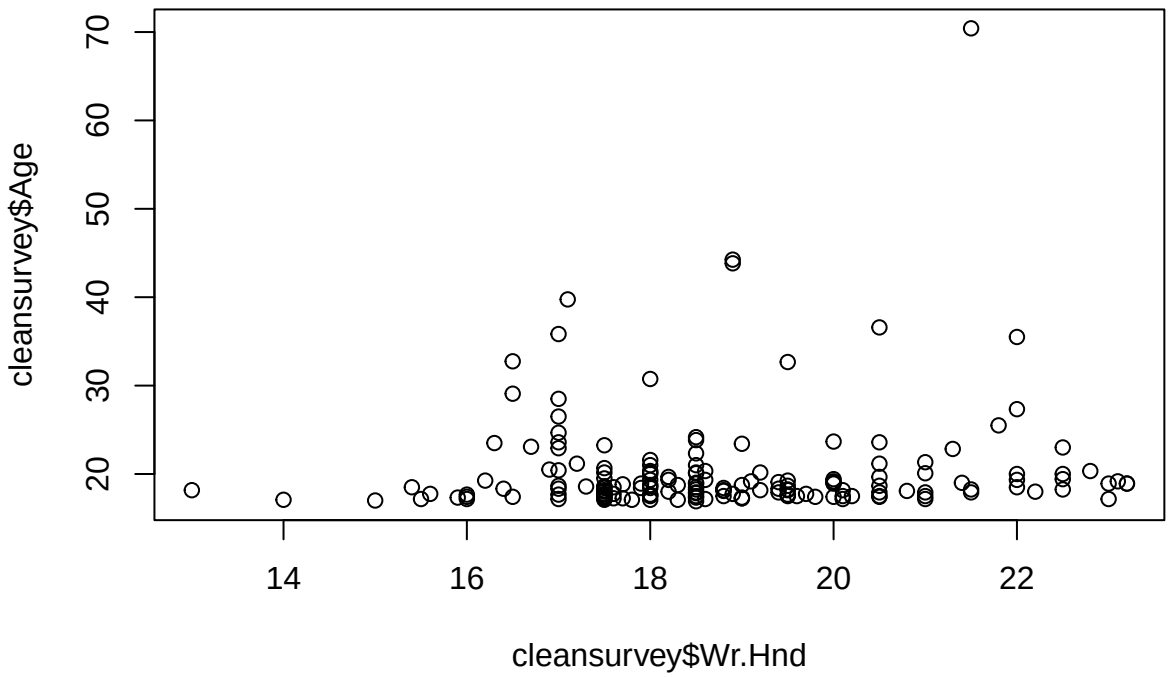
7. Create a suitable grid for projecting the multiple charts obtained earlier

```
par(mfrow=c(2,3), mar=c(2,5,2,1),las=1,bty='n')

x=c(sum(cleansurvey$Sex=="Male"),sum(cleansurvey$Sex=="Female"))
barplot(x,main="Male and Female Participants",col=c("green","red"))

x=c(sum(cleansurvey$W.Hnd=="Left"),sum(cleansurvey$W.Hnd=="Right"))
barplot(x,main="Left and Right Handers",col=c("blue","pink"))

library(dplyr)
x=filter(cleansurvey,Sex=="Female")
y=filter(cleansurvey,Sex=="Male")
x=c(sum(x$W.Hnd=="Left"),sum(y$W.Hnd=="Left"))
barplot(x,main="Female Left and Male Left Handers",col=c("purple","orange"))

y=filter(cleansurvey,Sex=="Male")
y=filter(y,W.Hnd=="Left")
pie(table(y$Smoke),col = c("pink","blue","red","green"),main="Smoking Habits")

hist(cleansurvey$Age,main="Age distribution",xlab="Age Range",ylab="frequency")

den <- density(cleansurvey$Age)
plot(den, frame = FALSE, col = "blue",main='Age Distribution',xlab="Age Range",ylab="density")
```



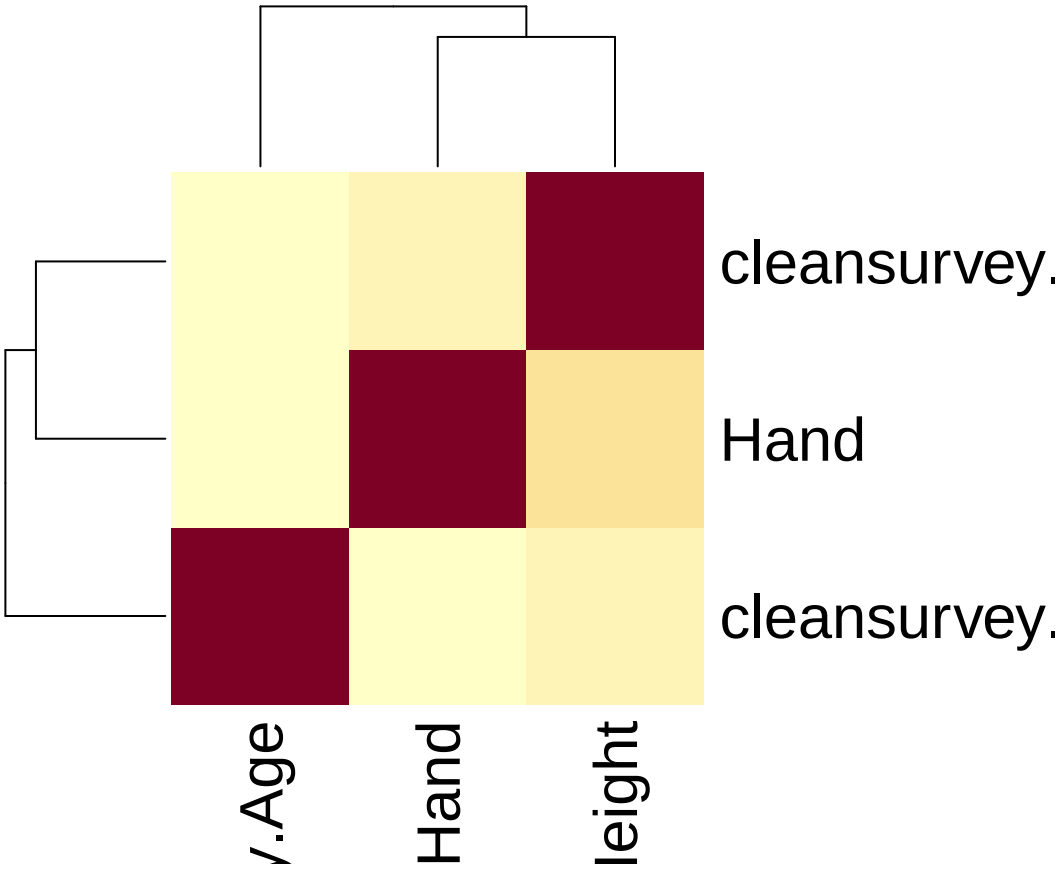
8. Reveal the relationship between the age and writing hand span using scatter plot

```
plot(cleansurvey$Wr.Hnd,cleansurvey$Age)
```

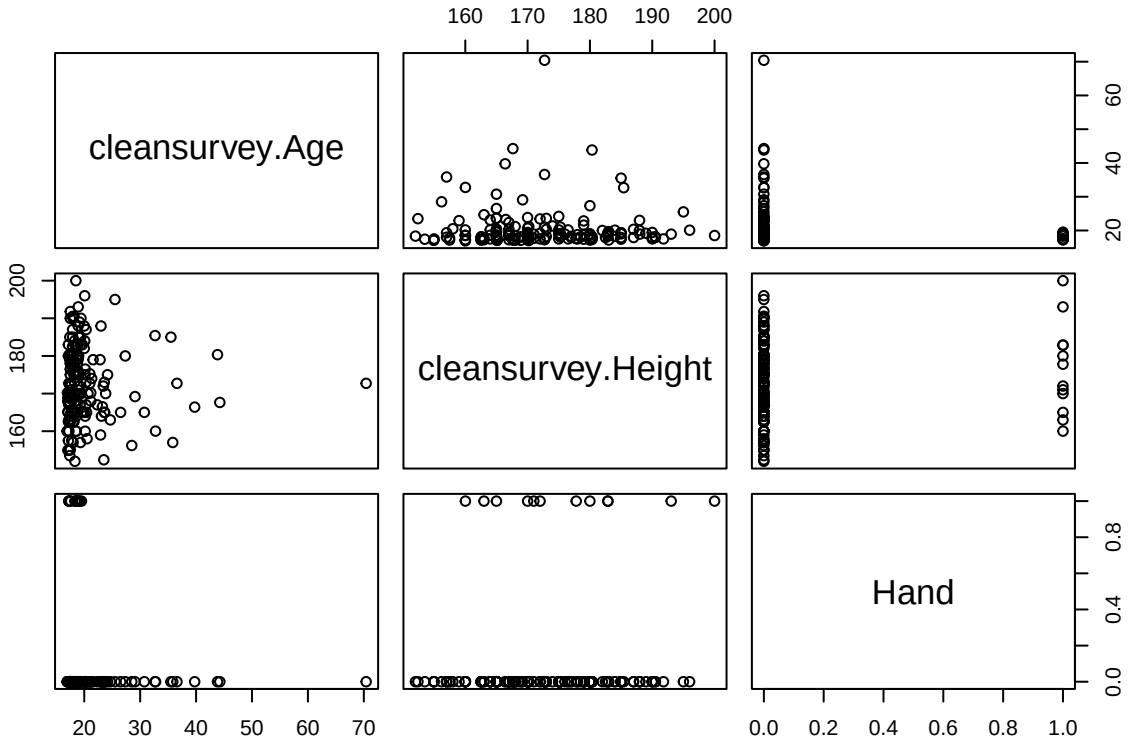


9. Plot the relationship between age, height and writing hand span in a single chart.

```
Hand = cleansurvey$W.Hnd=='Left'
df=data.frame(cleansurvey$Age,cleansurvey$Height,Hand)
corr= cor(df)
heatmap(corr)
```

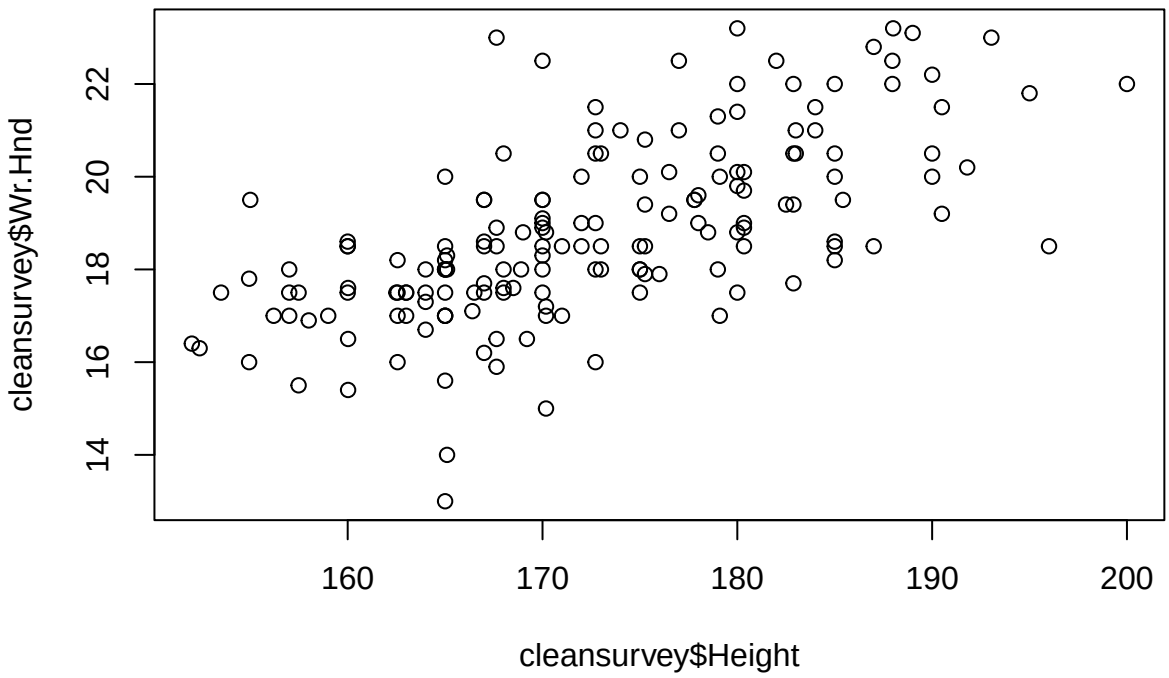



```
pairs(df)
```



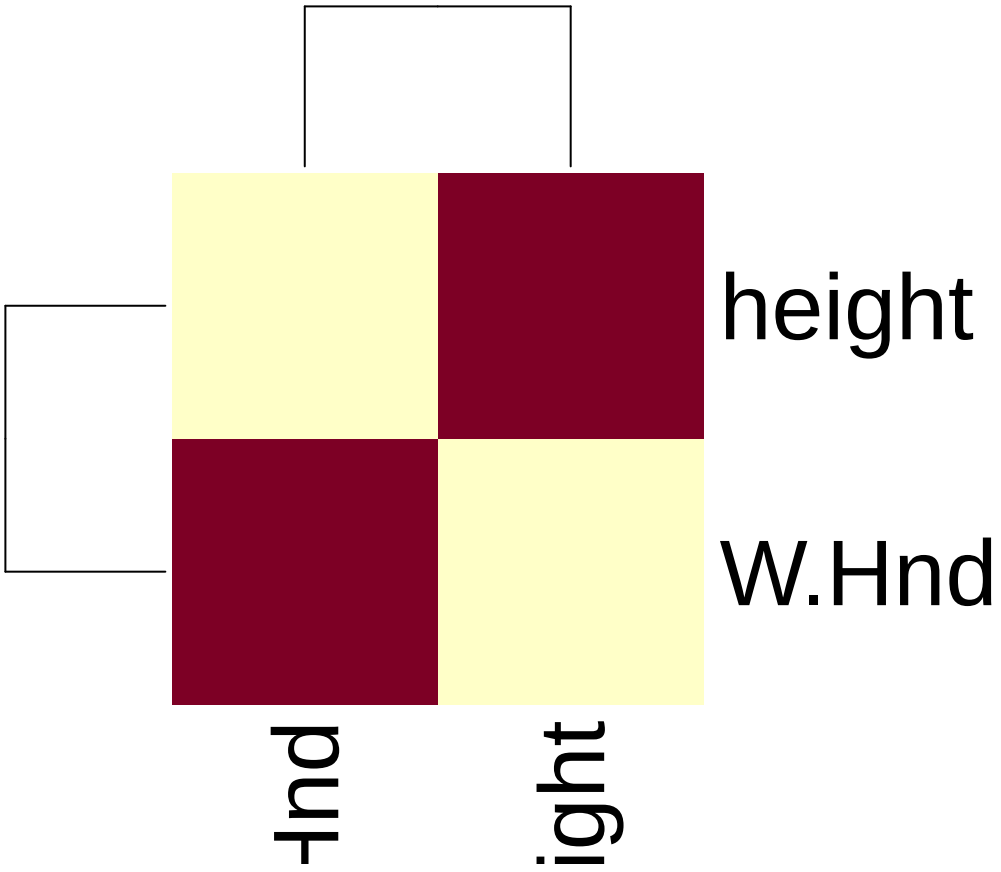
10. Plot the relationship between height and writing hand span

```
plot(cleansurvey$Height,cleansurvey$Wr.Hnd)
```

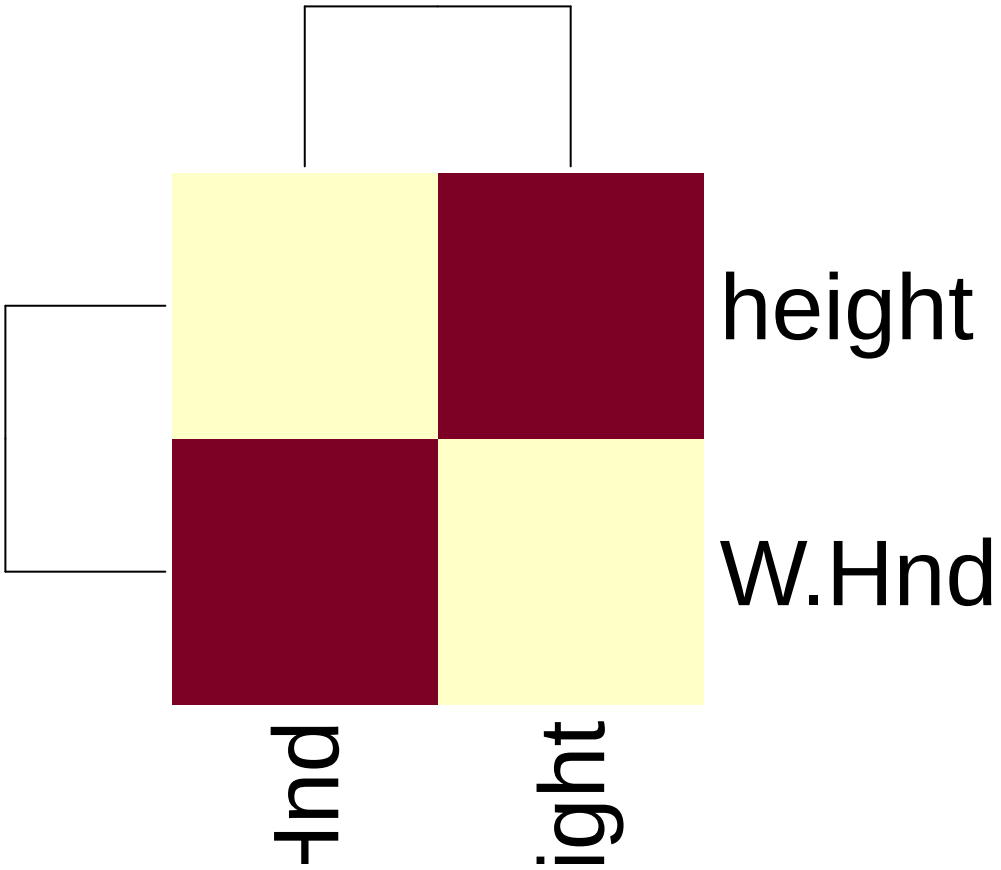


11. Plot the relationship between height and writing hand span based on gender and left and right handers.

```
maleData = cleansurvey%>%filter(Sex=='Male')
femaleData = cleansurvey%>%filter(Sex=='Female')
maleHnd = cleansurvey$W.Hnd=="Left"
femaleHnd = cleansurvey$W.Hnd=="Left"
corrMale= cor(data.frame(W.Hnd=maleHnd, height=maleData$Height))
corrFemale= cor(data.frame(W.Hnd=femaleHnd, height=femaleData$Height))
heatmap(corrMale)
```

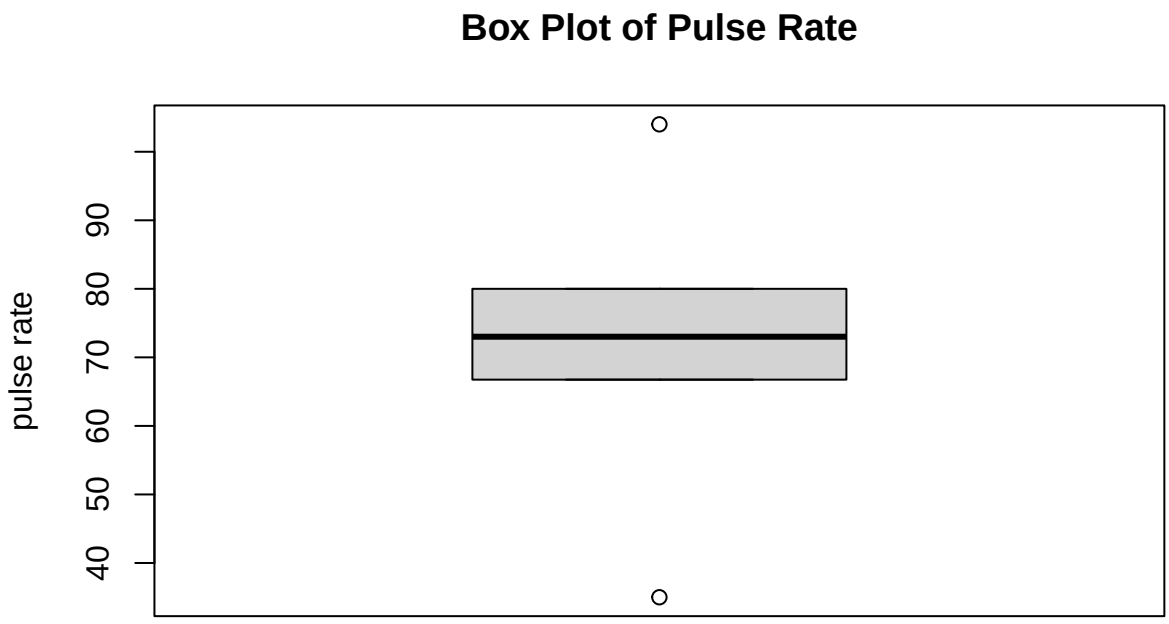



```
heatmap(corrFemale)
```



12. Draw the boxplot for pulse rate to analyse the five summary statistics. Provide appropriate title and label

```
n<-summary(cleansurvey$Pulse)
boxplot(n,main="Box Plot of Pulse Rate", ylab="pulse rate")
```


Box Plot of Pulse Rate
pulse rate
40
50
60
70
80
90

20BCE1025_Abhishek_N_N_Expeiment-10 Visualization using grammar of graphics

20BCE1025_Abhishek_N_N

2022-11-03

Use the newsurvey data obtained by cleaning 'na' values in survey data of MASS package and ggplot2 package to do the following:

```
library(MASS)
newSurvey=na.omit(survey)
head(newSurvey)
```

```
##      Sex Wr.Hnd NW.Hnd W.Hnd    Fold Pulse  Clap Exer Smoke Height      M.I
## 1 Female   18.5   18.0 Right  R on L    92  Left Some Never 173.00   Metric
## 2   Male   19.5   20.5  Left  R on L   104  Left None Regul 177.80 Imperial
## 5   Male   20.0   20.0 Right Neither    35 Right Some Never 165.00   Metric
## 6 Female   18.0   17.7 Right  L on R    64 Right Some Never 172.72 Imperial
## 7   Male   17.7   17.7 Right  L on R    83 Right Freq Never 182.88 Imperial
## 8 Female   17.0   17.3 Right  R on L    74 Right Freq Never 157.00   Metric
##      Age
## 1 18.250
## 2 17.583
## 5 23.667
## 6 21.000
## 7 18.833
## 8 35.833
```

1. Install the package ggplot2 and import it.

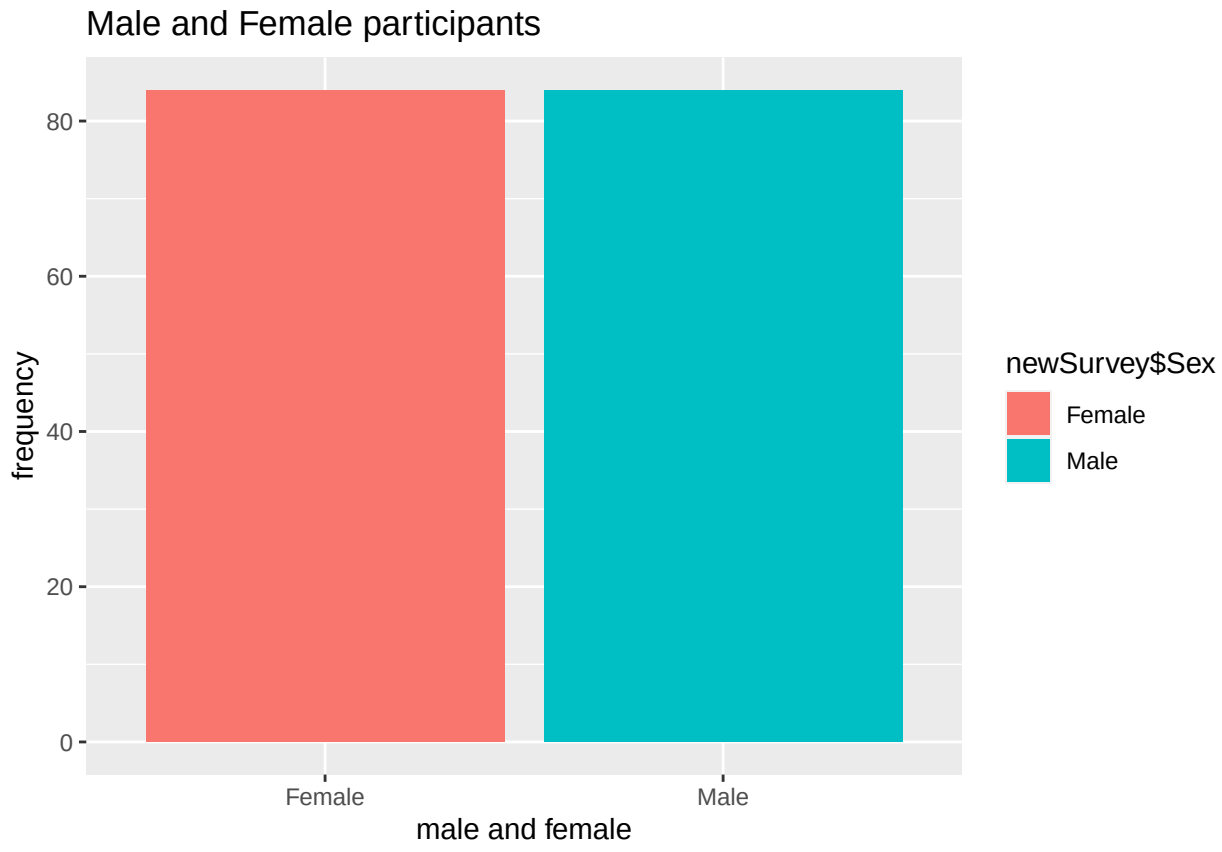
```
#install.packages("ggplot2")
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:MASS':
##
##     select

## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

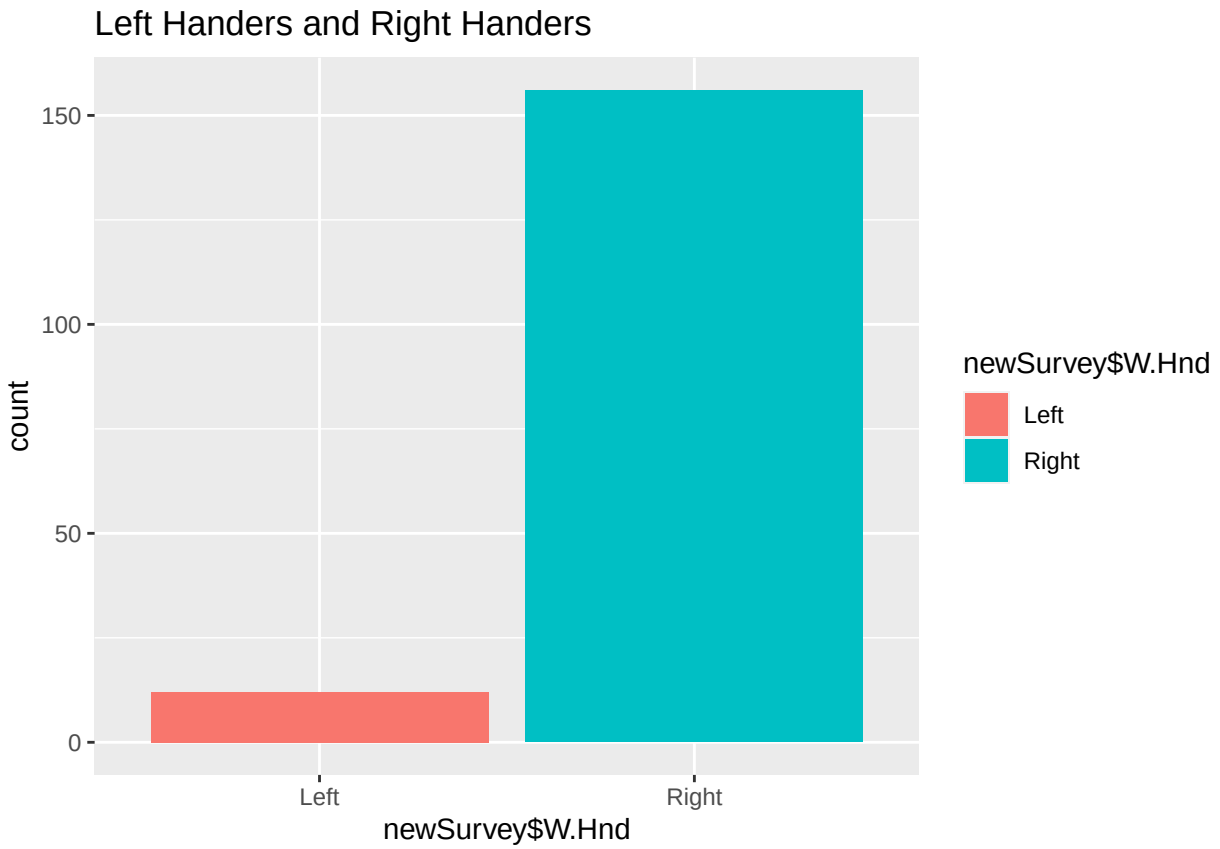
2. Plot a bar graph for the number of male and female participants in the survey. Provide the title as "Male and Female participants", y-axis label as "frequency" and specify the colours for the bars.

```
library(ggplot2)
ggplot(newSurvey)+
  geom_bar(aes(newSurvey$Sex,fill=newSurvey$Sex)) +
  ggtitle("Male and Female participants") +
  xlab("male and female") +
  ylab("frequency")
```



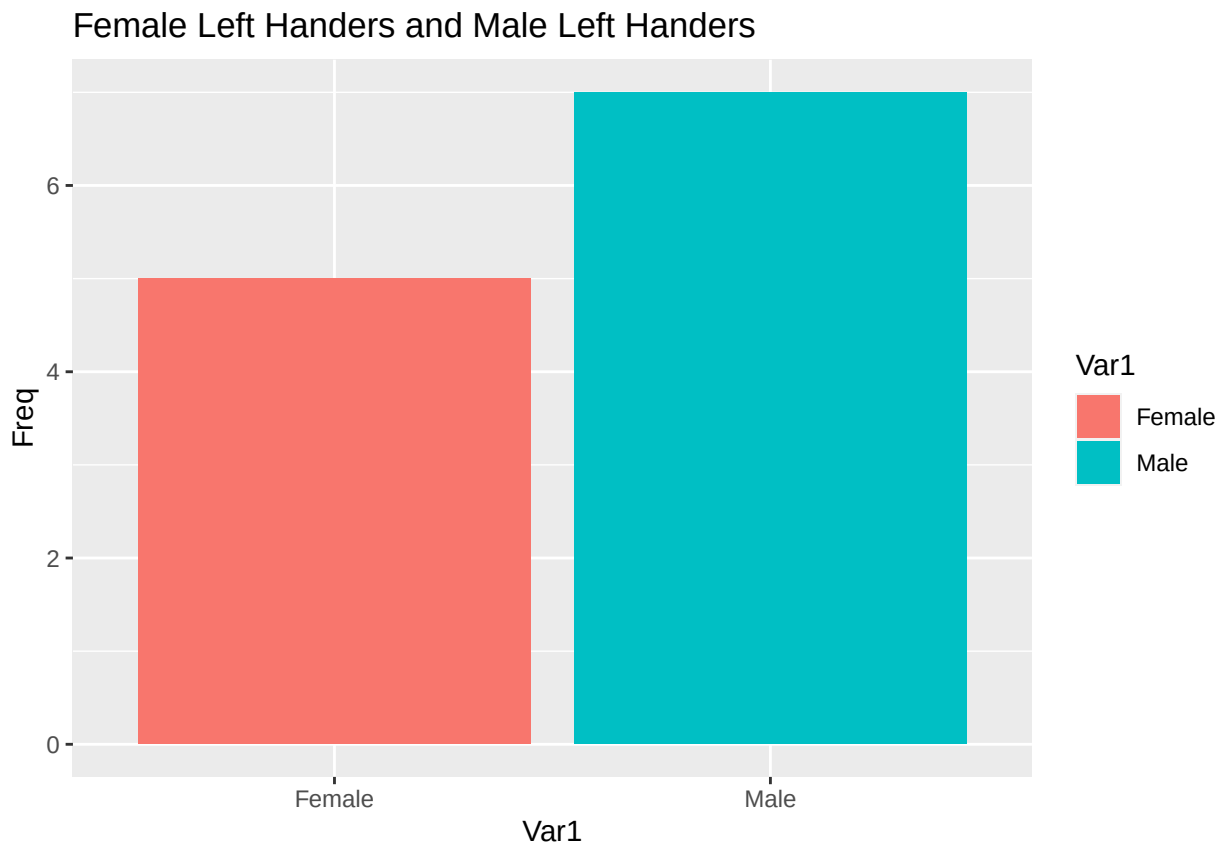
3. Plot a bar graph for the number of left handers and right handers in the survey. Provide the title as "Left Handers and Right Handers", y-axis label as "count" and specify the colours for the bars.

```
ggplot(newSurvey)+
  geom_bar(aes(newSurvey$W.Hnd,fill=newSurvey$W.Hnd)) +
  ggtitle("Left Handers and Right Handers") +
  ylab("count")
```



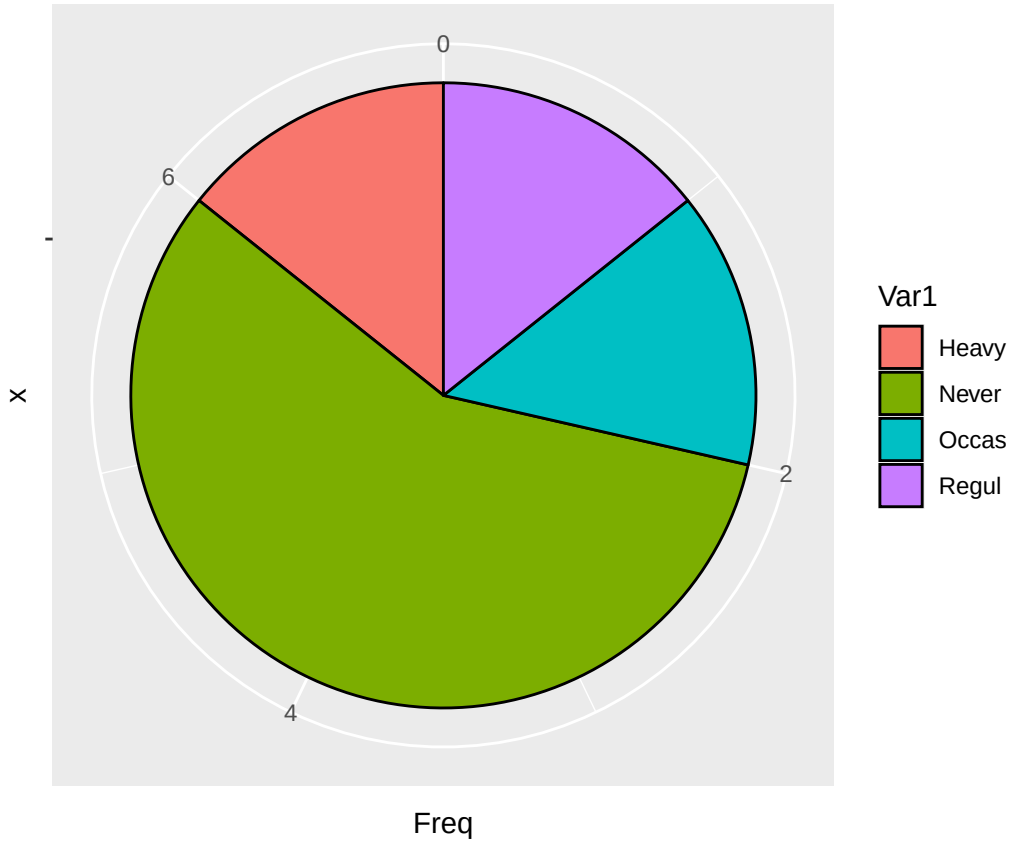
4. Plot the distribution between male left handers and female left handers using bar chart. Provide the title as "Female Left Handers and Male Left Handers , y-axis label as "count" and specify the colours for the bars.

```
male_left_handers= newSurvey[newSurvey$W.Hnd=="Left" & newSurvey$Sex=="Male",]
female_left_handers= newSurvey[newSurvey$W.Hnd=="Left" & newSurvey$Sex=="Female",]
custom_data = rbind(male_left_handers, female_left_handers)
ggplot(as.data.frame(table(custom_data$Sex)), aes(x=Var1, y = Freq,
fill=Var1)) +
geom_bar(stat="identity")+
labs(title="Female Left Handers and Male Left Handers")
```



5. Draw the distribution of smoking habits of male left handers using pie chart.

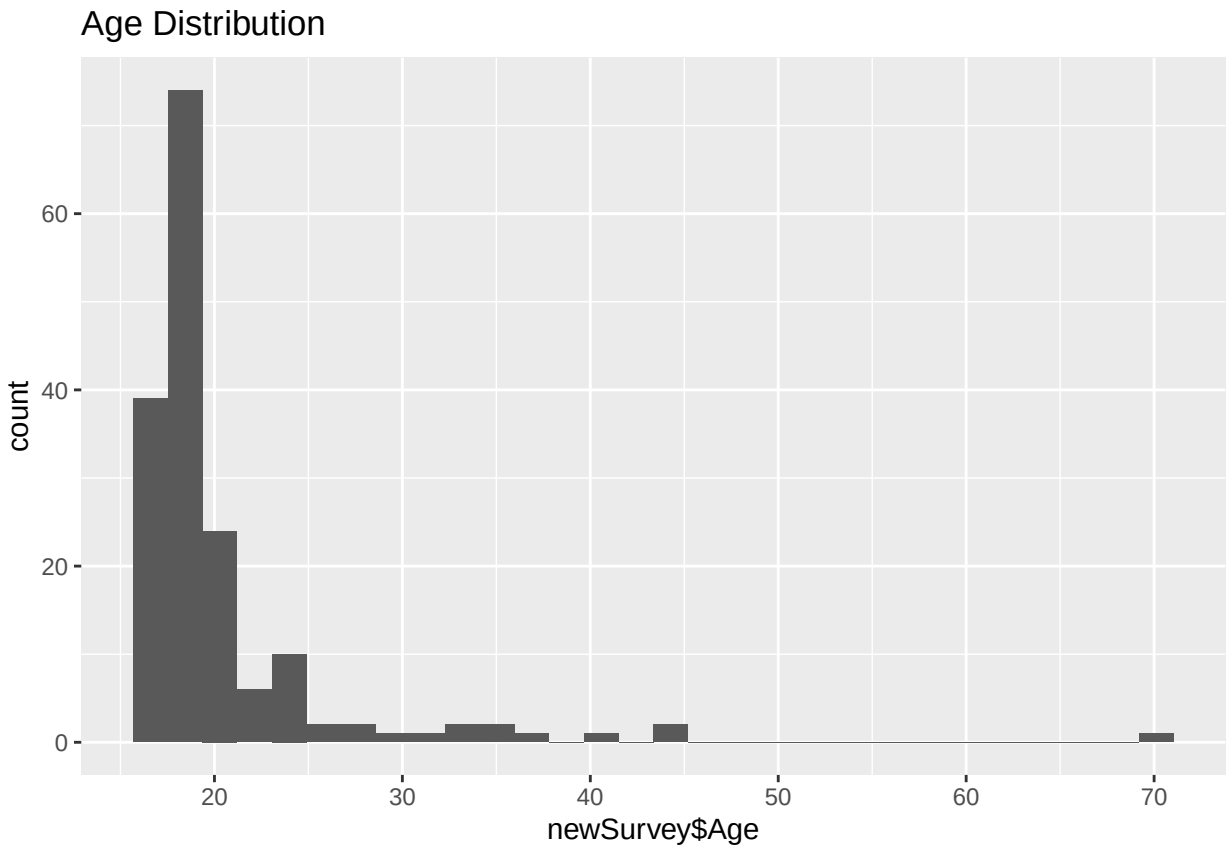
```
smoking_habits = ggplot(as.data.frame(table(male_left_handers$Smoke)),
aes(x = "", y =Freq, fill=Var1)) +
geom_bar(stat="identity")+
geom_col(color = "black") +
coord_polar("y", start=0)
smoking_habits
```



6. Draw the histogram of age distribution with the title as 'Age distribution' and xlabel as 'Age range' and ylabel as 'frequency'.

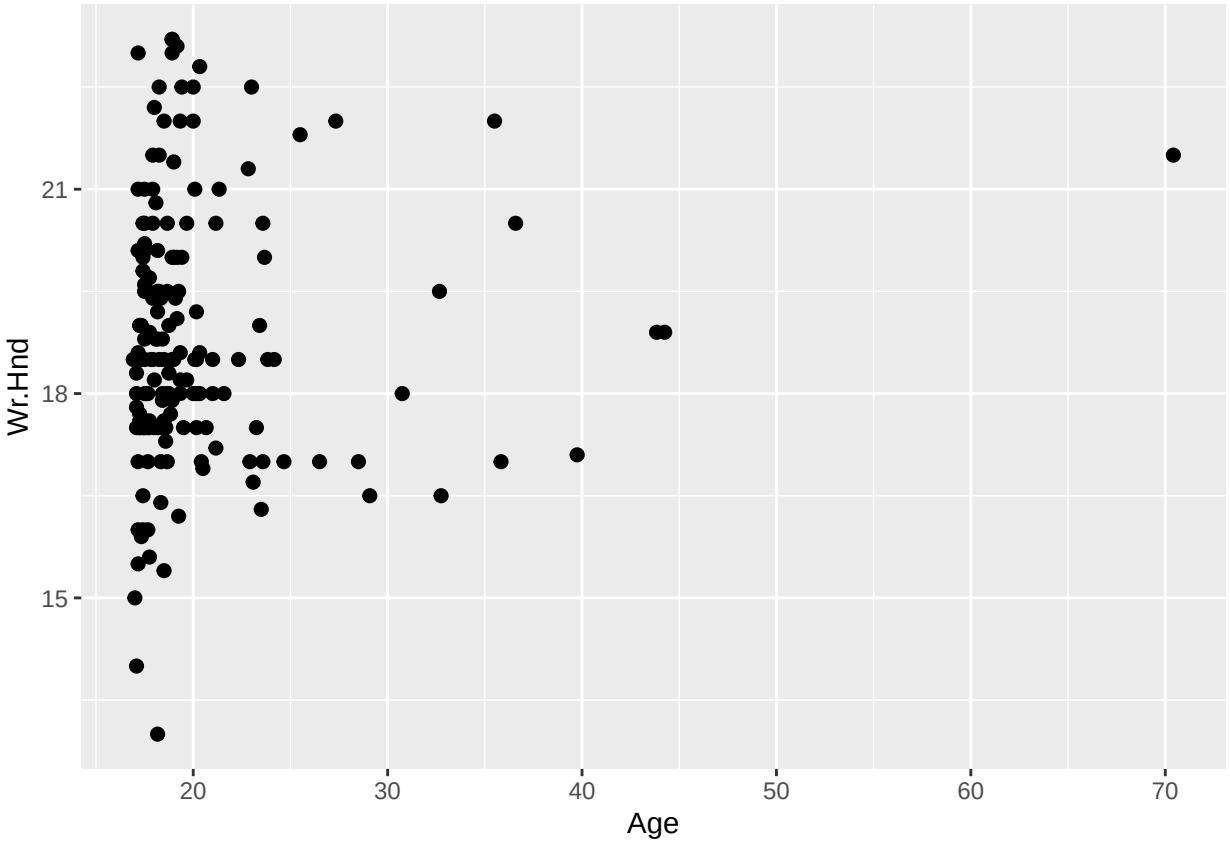
```
ggplot(newSurvey, aes(x=newSurvey$Age)) +
  geom_histogram() +
  stat_bin(bins = 30) +
  labs(title="Age Distribution", xlabel="Age Range", ylabel="Frequency")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



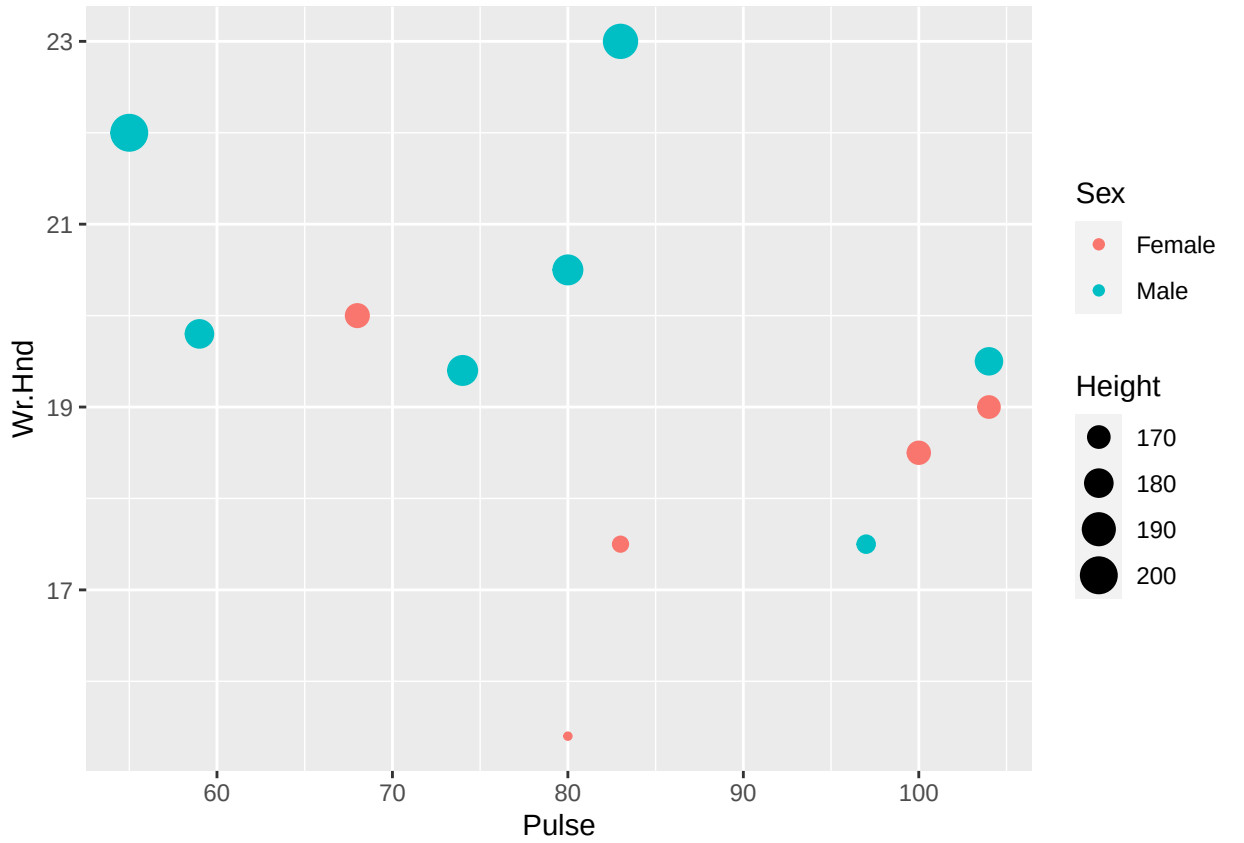
7. Reveal the relationship between the age and writing hand span using scatter plot.

```
ggplot(newSurvey, aes(x=Age, y=Wr.Hnd)) + geom_point(size=2)
```



8. Plot the distribution of writing hand span vs. pulse rate of left handers. Provide colour based on gender and vary the size of the point based on height of the student.

```
ggplot(custom_data, aes(x=Pulse, y=Wr.Hnd, color=Sex, size=Height)) + geom_point()
```



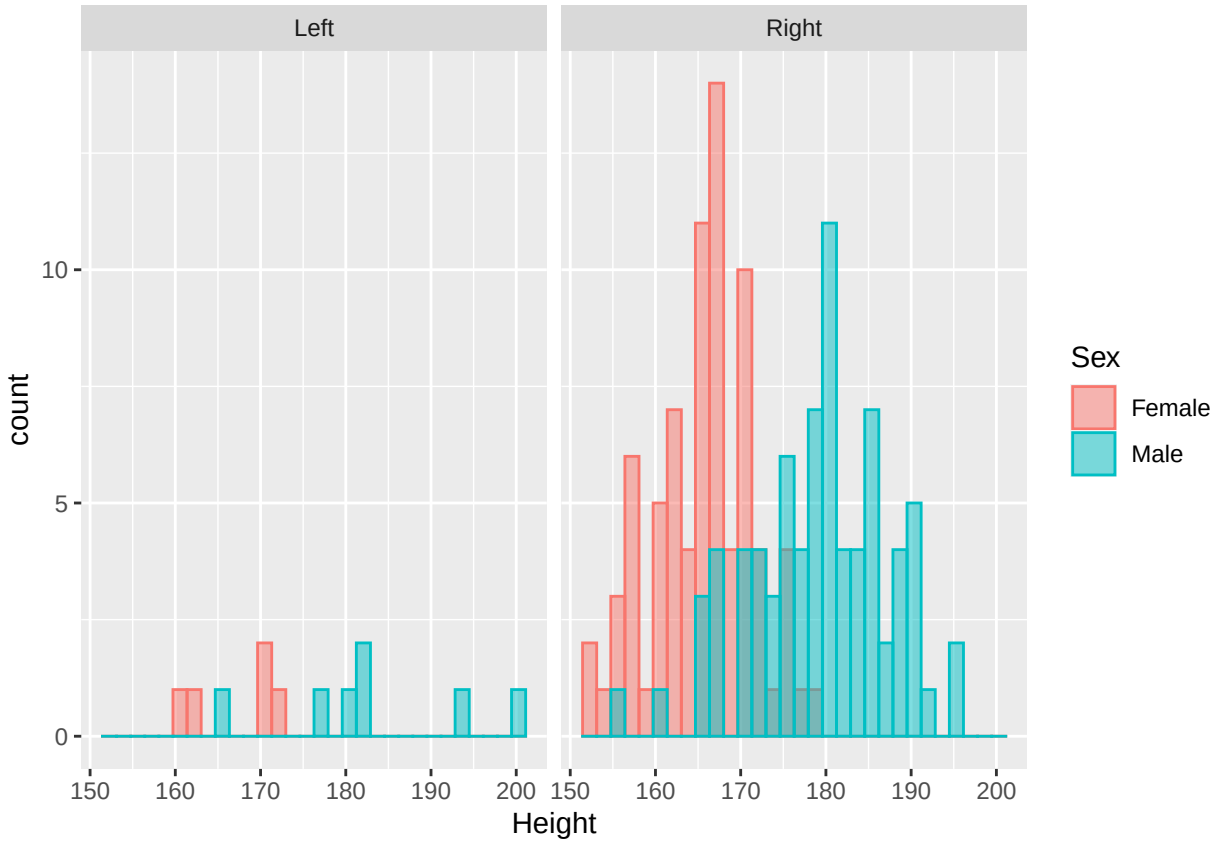
9. Plot the distribution of height of the students with filled colour based on gender with facet based on left and write handers.

```
head(newSurvey)
```

```
##      Sex Wr.Hnd NW.Hnd W.Hnd    Fold Pulse  Clap Exer Smoke Height      M.I
## 1 Female   18.5   18.0 Right  R on L    92  Left Some Never 173.00   Metric
## 2   Male   19.5   20.5  Left  R on L   104  Left None Regul 177.80 Imperial
## 5   Male   20.0   20.0 Right Neither    35 Right Some Never 165.00   Metric
## 6 Female   18.0   17.7 Right  L on R    64 Right Some Never 172.72 Imperial
## 7   Male   17.7   17.7 Right  L on R    83 Right Freq Never 182.88 Imperial
## 8 Female   17.0   17.3 Right  R on L    74 Right Freq Never 157.00   Metric
##      Age
## 1 18.250
## 2 17.583
## 5 23.667
## 6 21.000
## 7 18.833
## 8 35.833
```

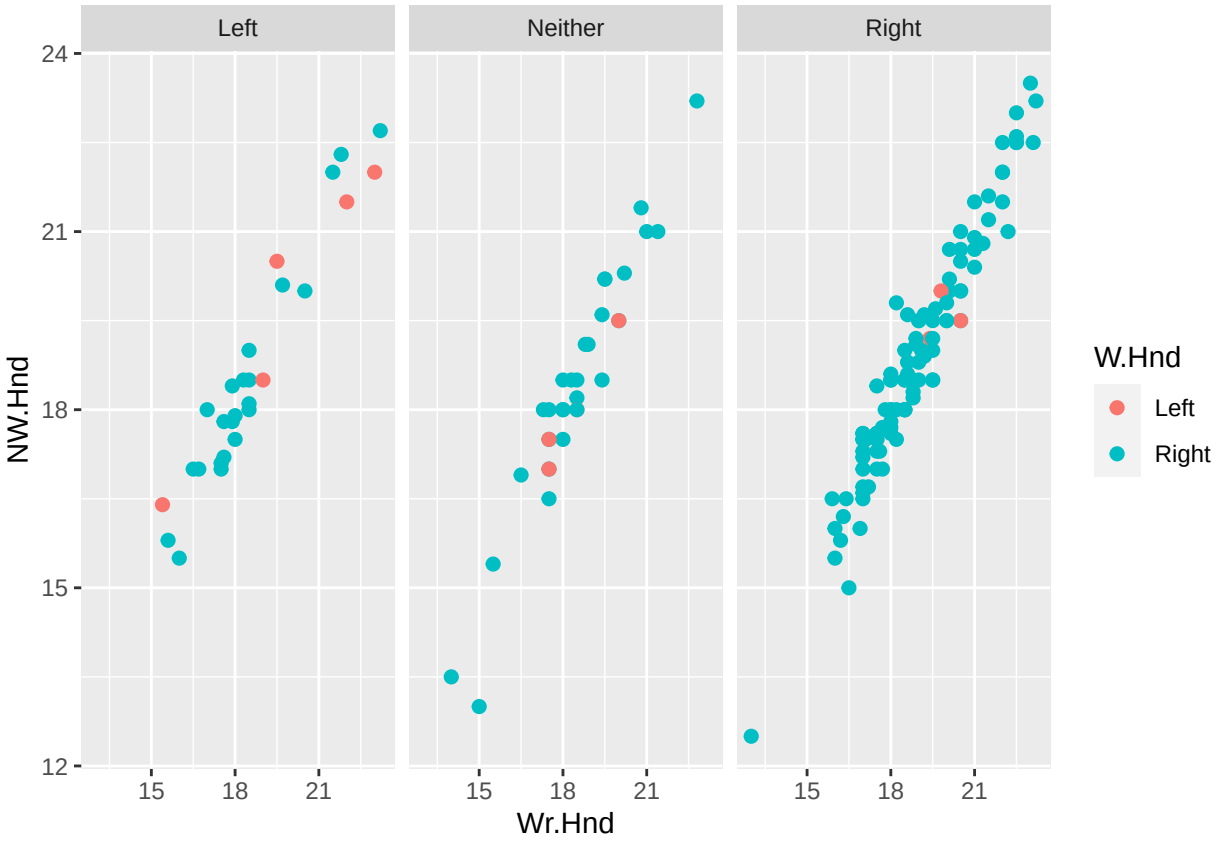
```
ggplot(newSurvey, aes(x=Height, color=Sex,fill=Sex)) +
  geom_histogram( alpha=0.5, position="identity") +
  facet_wrap(~W.Hnd)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



10. Plot the trend of span of writing hand vs. non-writing hand coloured and grouped based on left and right handers with facet label based on clap.

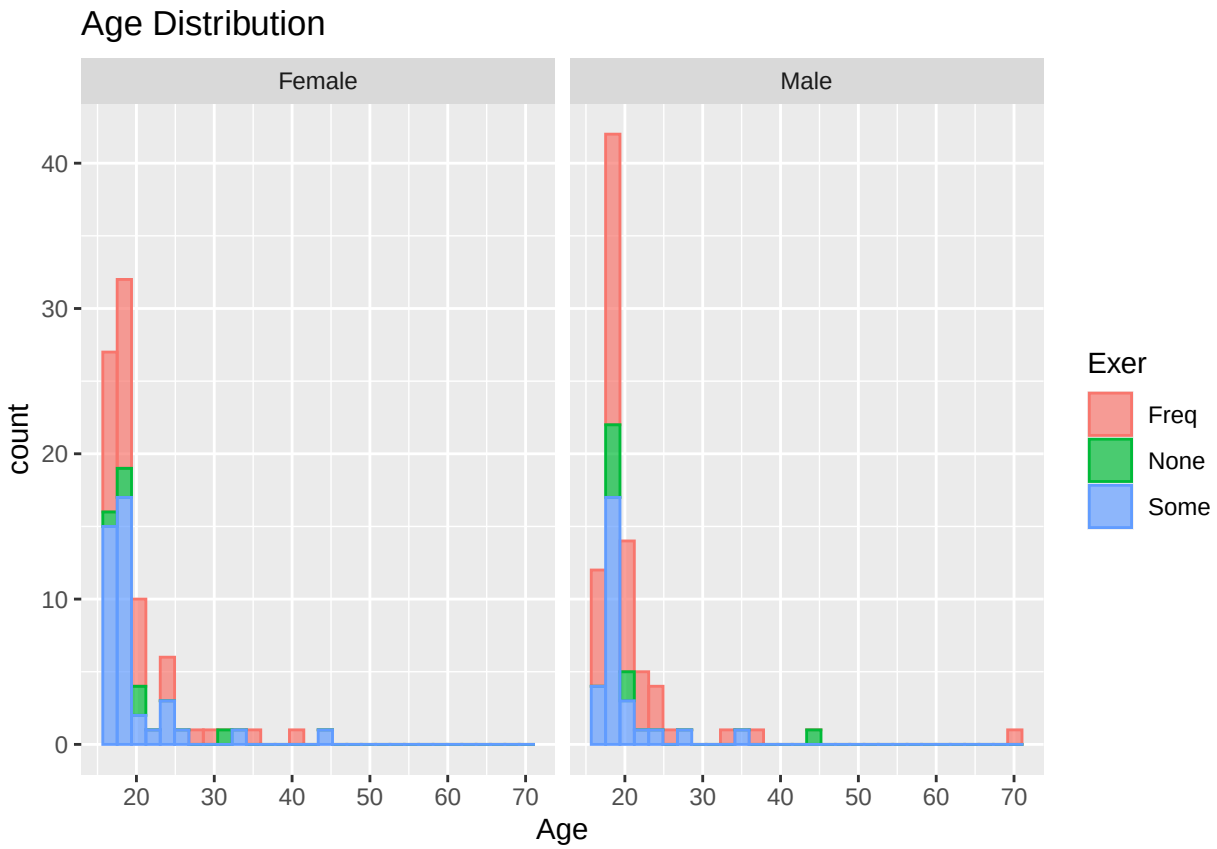
```
ggplot(newSurvey, aes(x=Wr.Hnd, y=NW.Hnd, color=W.Hnd, )) +
facet_grid(.~Clap) +
geom_point(size=2)
```



11. Plot the distribution of age of students based on categories under exercise with facet wrap based on gender.

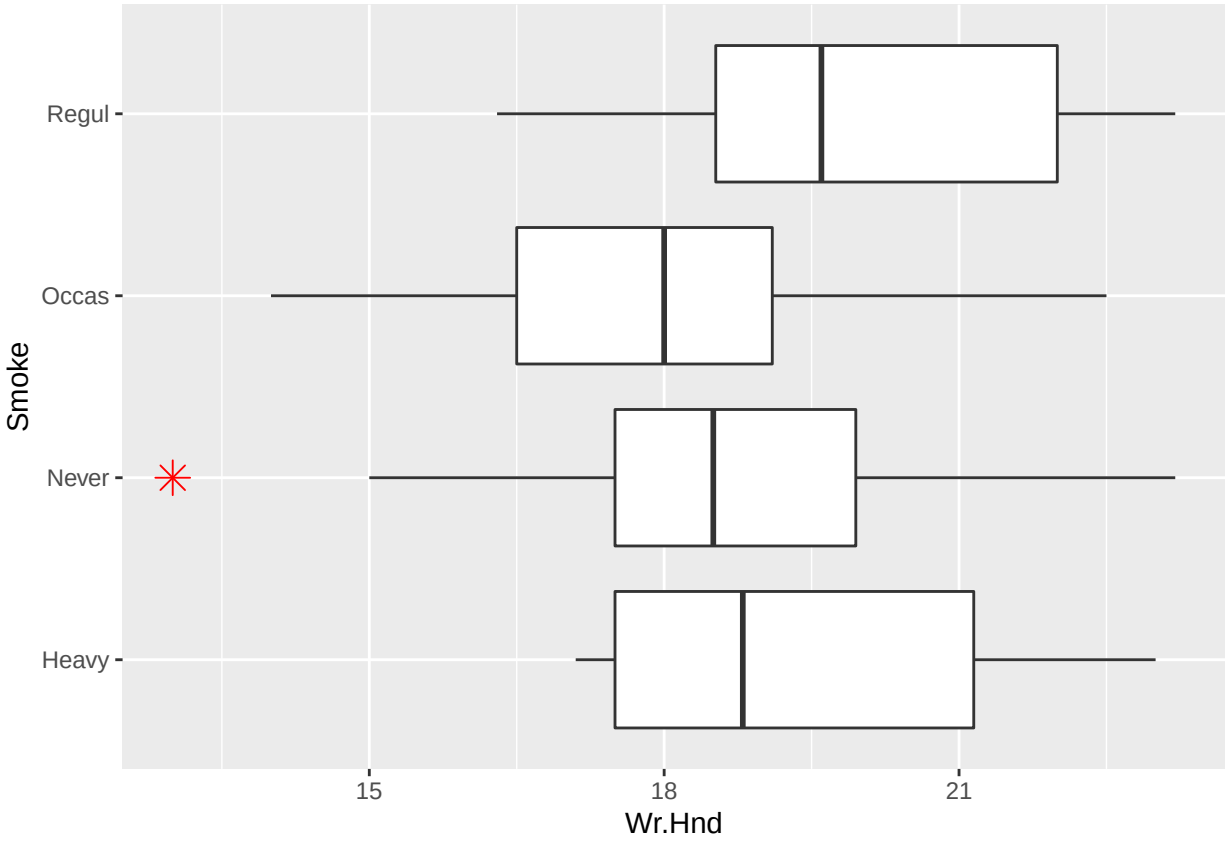
```
ggplot(newSurvey, aes(x=Age,color=Exer,fill=Exer)) +
facet_grid(.~Sex)+
geom_histogram(alpha=0.7)+
labs(title="Age Distribution", xlabel="Age Range", ylabel="Frequency")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



12. Plot the box plot of writing hand span with respect to smoking habits of students.

```
ggplot(newSurvey, aes(x=Wr.Hnd, y=Smoke))+
geom_boxplot(outlier.colour="red", outlier.shape=8,outlier.size=4)
```


Regul
Occas
Never
Heavy
Smoke
15
18
21
Wr.Hnd