4. Write an R program to create a list containing strings, numbers, vectors and logical values and do the following manipulations over the list.

a. Access the first element in the list
b. Give the names to the elements in the list
c. Add element at some positions in the list
d. Remove the element
e. print the first and third element
f. Update the third element

```
cat("create a list containing strings, numbers, vectors and logical values\n")
list_data <- list("R programming", c(125.17, 20), c(5, 7, 9, 11), c(TRUE, FALSE, TRUE, TRUE))
print(list_data)

cat("a. Access the first element in the list\n")
print(list_data[[1]])

cat("\nb. Give the names to the elements in the list\n")
names(list_data) <- c("My_strings", "My_numbers", "My_vector", "My_logics")
print(list_data)

cat("c. Remove the element\n")
list_rem <- list_data[-2]
print(list_rem)

cat("d.Add element at some positions in the list\n")
list_data[[5]] <- matrix(c(1, 2, 3, 4), nrow = 2)
print(list_data)

cat("e. print the first and third element\n")
print(list_data[c(1, 3)])

cat("f. Update the third element\n")
list_data[[3]] <- c(4, 6, 8, 10)
print(list_data)
```

Output:

```
create a list containing strings, numbers, vectors and logical values
[[1]]
[1] "R programming"

[[2]]
[1] 125.17  20.00

[[3]]
[1]  5  7  9 11

[[4]]
[1]  TRUE FALSE  TRUE  TRUE

a. Access the first element in the list
[1] "R programming"

b. Give the names to the elements in the list
$My_strings
[1] "R programming"

$My_numbers
[1] 125.17  20.00

$My_vector
[1]  5  7  9 11

$My_logics
[1]  TRUE FALSE  TRUE  TRUE

c. Remove the element
$My_strings
[1] "R programming"

$My_vector
[1]  5  7  9 11

$My_logics
[1]  TRUE FALSE  TRUE  TRUE

d.Add element at some positions in the list
$My_strings
[1] "R programming"
```

```
$My_numbers
[1] 125.17  20.00

$My_vector
[1]  5  7  9 11

$My_logics
[1]  TRUE FALSE  TRUE  TRUE

[[5]]
     [,1] [,2]
[1,]    1    3
[2,]    2    4
```

e. print the first and third element

```
$My_strings
[1] "R programming"

$My_vector
[1]  5  7  9 11
```

f. Update the third element

```
$My_strings
[1] "R programming"

$My_numbers
[1] 125.17  20.00

$My_vector
[1]  4  6  8 10

$My_logics
[1]  TRUE FALSE  TRUE  TRUE

[[5]]
     [,1] [,2]
[1,]    1    3
[2,]    2    4
```

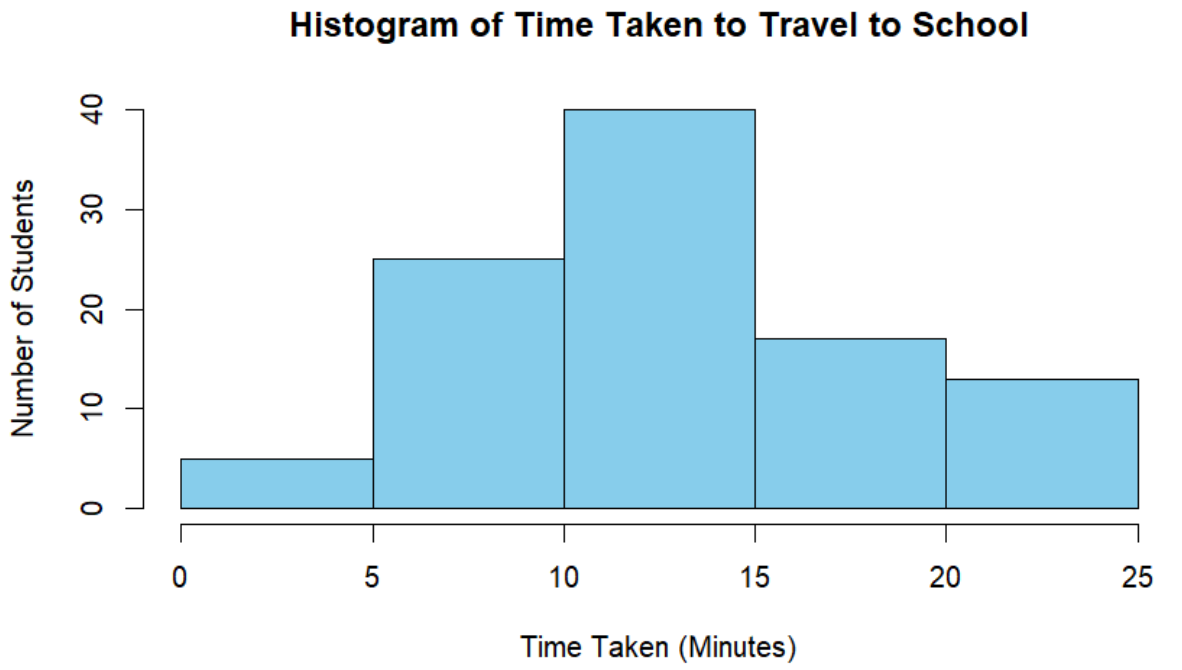
5. The following table shows the time taken (in minutes) by 100 students to travel to school on a particular day.

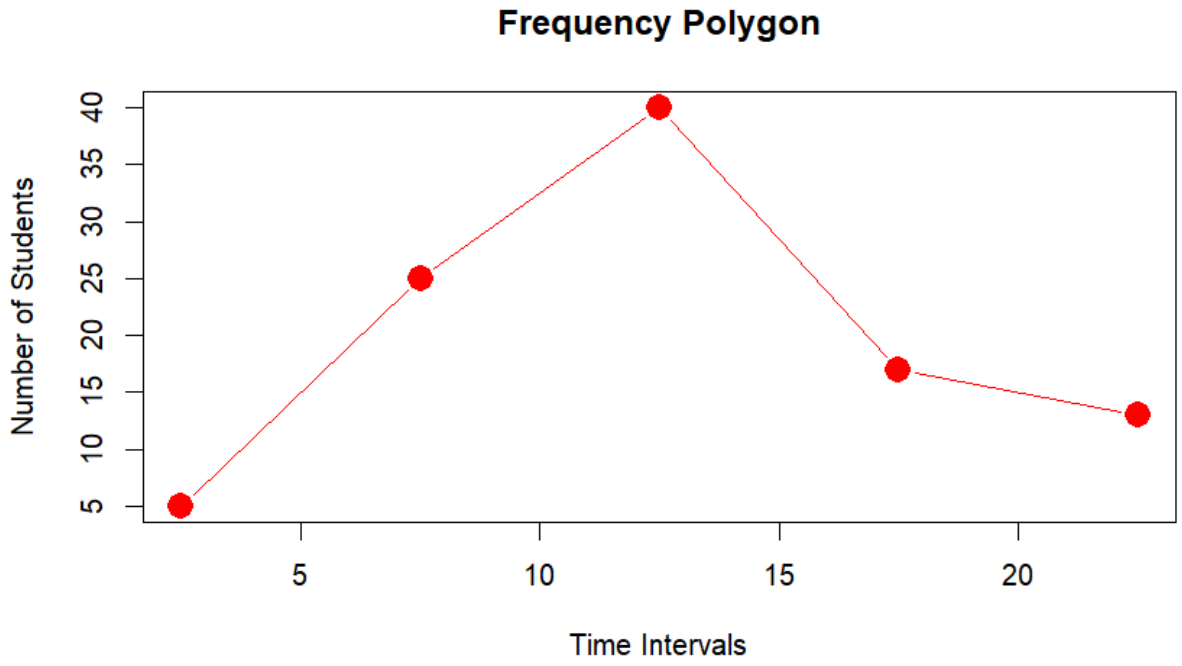
Time	0-5	5-10	10-15	15-20	20-25
Number of Students	5	25	40	17	13

a. Draw the histogram
b. Draw frequency polygon

```
lower_limit <- seq(0, 20, 5)
upper_limit <- seq(5, 25, 5)
mid_point <- (lower_limit + upper_limit) / 2
freq <- c(5, 25, 40, 17, 13)
hist_data <- rep(mid_point, freq)

hist(hist_data,
  breaks = 5,
  main = "Histogram of Time Taken to Travel to School",
  xlab = "Time Taken (Minutes)",
  ylab = "Number of Students",
  col = "skyblue"
)

polygon(mid_point, freq,  col = "red", pch = 16, xlab = "Time Intervals", ylab = "Number of Students", main = "Frequency Polygon")

plot(mid_point, freq, type = "b", xlab = "Time Intervals", ylab = "Number of Students", main = "Frequency Polygon",pch = 16, cex= 2, col = "red")
```

Output:


Histogram of Time Taken to Travel to School
Number of Students
0
10
20
30
40
0
5
10
15
20
25
Time Taken (Minutes)


Frequency Polygon
Number of Students
5
10
15
20
25
30
35
40
5
10
15
20
Time Intervals

6. Write an R program to create a Data Frame with following details and do the following operations.

ItemCode	itemCategory	ItemPrice
1001	Electronics	700
1002	DesktopSupplies	300
1003	Office Supplies	350
1004	USB	400
1005	CD Drive	800

a. Subset the Data frame and display the details of only those items whose price is greater than or equal to 350.
b. Subset the Data frame and display only the items where the category is either "Office Supplies" or "Desktop Supplies"
c. Subset the Data frame and display the items where the Itemprice between 300 and 700
d. Compute the sum of all ItemPrice
e. Create another Data Frame called “item-details” with three different fields itemCode, ItemQtyonHand and ItemReorderLvl and merge the two frames.

```
item_data <- data.frame(
  ItemCode = c("1001", "1002", "1003", "1004", "1005"),
  ItemCategory = c(
    "Electronics", "Desktop Supplies", "Office Supplies", "USB",
    "CD Drive"
  ), ItemPrice = c(700, 300, 350, 400, 800)
)
print(item_data)

cat("\na. Subset the Data frame and display the details of only those items whose price is greater than or equal to 350.\n")
subset_data_GT <- subset(item_data, ItemPrice >= 350)
print(subset_data_GT)

cat("\nb. Subset the Data frame and display only the items where the category is either Office Supplies or Desktop Supplies.\n")
subset_data_DO <- subset(item_data, ItemCategory == "DesktopSupplies" | ItemCategory == "Office Supplies")
print(subset_data_DO)
```

```
cat("\nc. Subset the Data frame and display the items where the Itemprice between 300 and 700.\n")
subset_data_Between <- subset(item_data, ItemPrice >= 300 & ItemPrice <= 700)
print(subset_data_Between)

cat("\nd. Compute the sum of all ItemPrice.\n")
total_price <- sum(item_data$ItemPrice)
print(total_price)

cat("\ne. Create another Data Frame called item-details with three different fields itemCode, ItemQtyonHand and ItemReorderLvl and merge the two frames.\n")
item_details <- data.frame(
  ItemCode = c("1001", "1002", "1003", "1004", "1005"),
  ItemQtyonHand = c(15, 10, 25, 8, 12), ItemReorderLvl = c(5, 8, 10, 6, 9)
)
merged_data <- merge(item_data, item_details, by = "ItemCode")
print(merged_data)
```

Output:

	ItemCode	ItemCategory	ItemPrice
1	1001	Electronics	700
2	1002	Desktop Supplies	300
3	1003	Office Supplies	350
4	1004	USB	400
5	1005	CD Drive	800

a. Subset the Data frame and display the details of only those items whose price is greater than or equal to 350.

	ItemCode	ItemCategory	ItemPrice
1	1001	Electronics	700
3	1003	Office Supplies	350
4	1004	USB	400
5	1005	CD Drive	800

b. Subset the Data frame and display only the items where the messageegory is either Office Supplies or Desktop Supplies.

	ItemCode	ItemCategory	ItemPrice
3	1003	Office Supplies	350

c. Subset the Data frame and display the items where the Itemprice between 300 and 700.

	ItemCode	ItemCategory	ItemPrice
1	1001	Electronics	700
2	1002	Desktop Supplies	300
3	1003	Office Supplies	350
4	1004	USB	400

d. Compute the sum of all ItemPrice.

[1] 2550

e. Create another Data Frame called item-details with three different fields itemCode, ItemQtyonHand and ItemReorderLvl and merge the two frames.

	ItemCode	ItemCategory	ItemPrice	ItemQtyonHand	ItemReorderLvl
1	1001	Electronics	700	15	5
2	1002	Desktop Supplies	300	10	8
3	1003	Office Supplies	350	25	10
4	1004	USB	400	8	6
5	1005	CD Drive	800	12	9