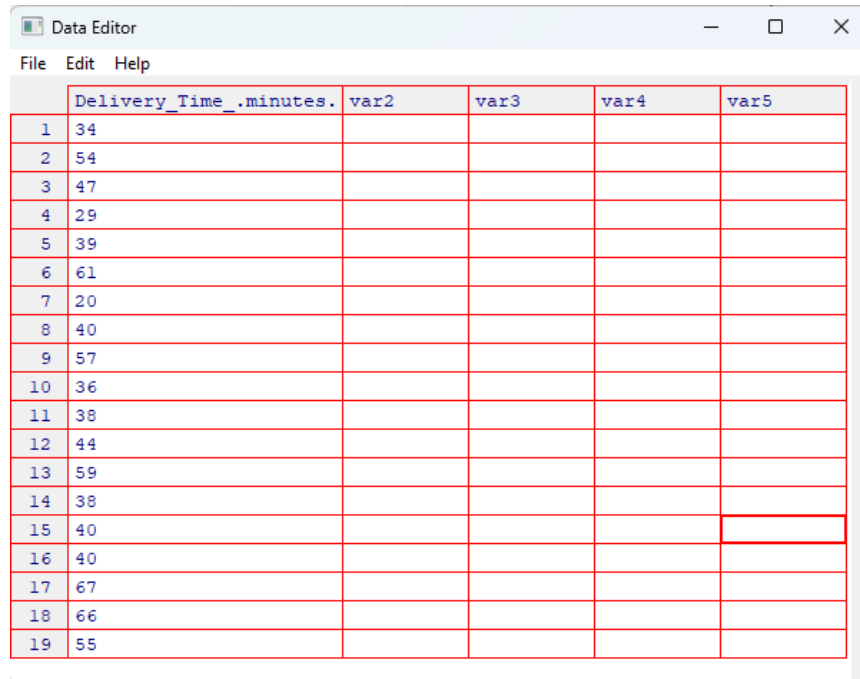


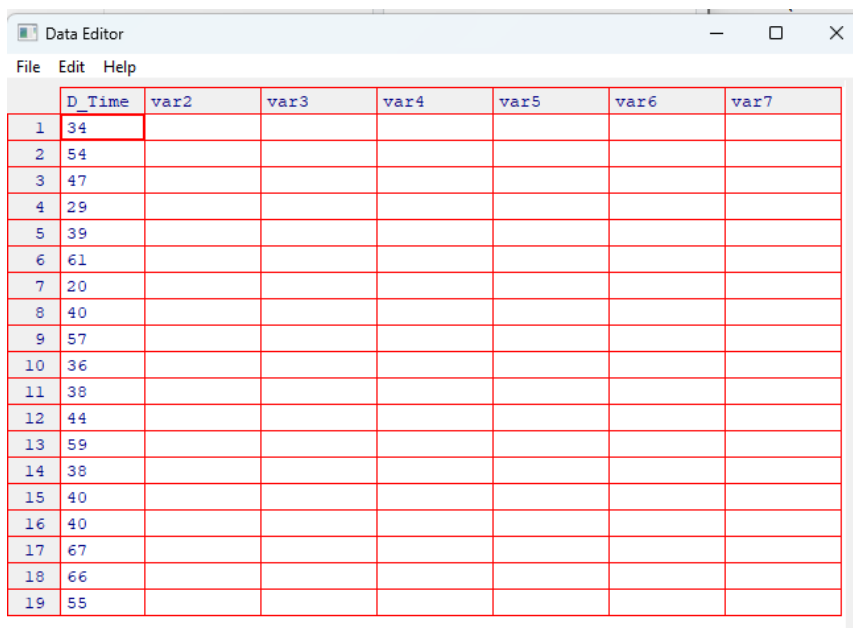
Q1).



The screenshot shows the SPSS Data Editor window. The title bar is 'Data Editor'. The menu bar contains 'File', 'Edit', and 'Help'. The data grid has 6 columns: 'Delivery_Time_.minutes.', 'var2', 'var3', 'var4', and 'var5'. There are 19 rows of data. The first column contains row numbers 1 through 19. The second column contains delivery times in minutes. The other columns are empty.

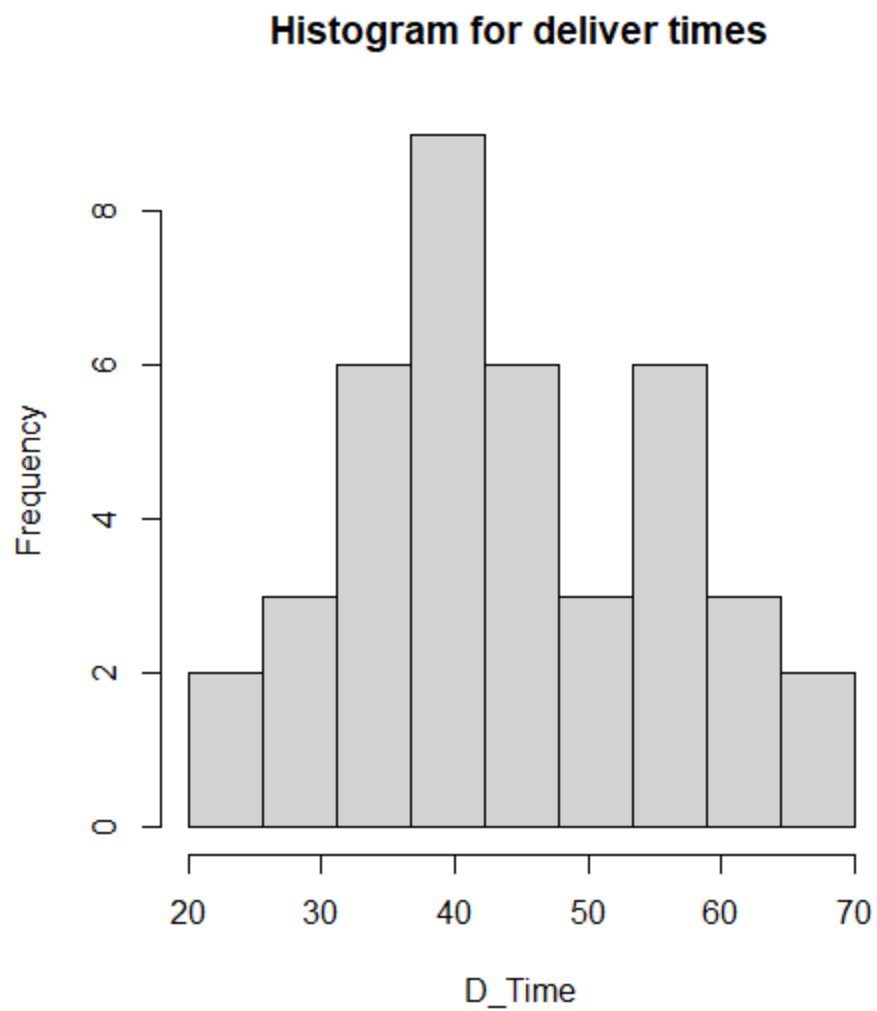
	Delivery_Time_.minutes.	var2	var3	var4	var5
1	34				
2	54				
3	47				
4	29				
5	39				
6	61				
7	20				
8	40				
9	57				
10	36				
11	38				
12	44				
13	59				
14	38				
15	40				
16	40				
17	67				
18	66				
19	55				

Q2).



The screenshot shows the SPSS Data Editor window. The title bar is 'Data Editor'. The menu bar contains 'File', 'Edit', and 'Help'. The data grid has 8 columns: 'D_Time', 'var2', 'var3', 'var4', 'var5', 'var6', and 'var7'. There are 19 rows of data. The first column contains row numbers 1 through 19. The second column contains delivery times in minutes. The other columns are empty.

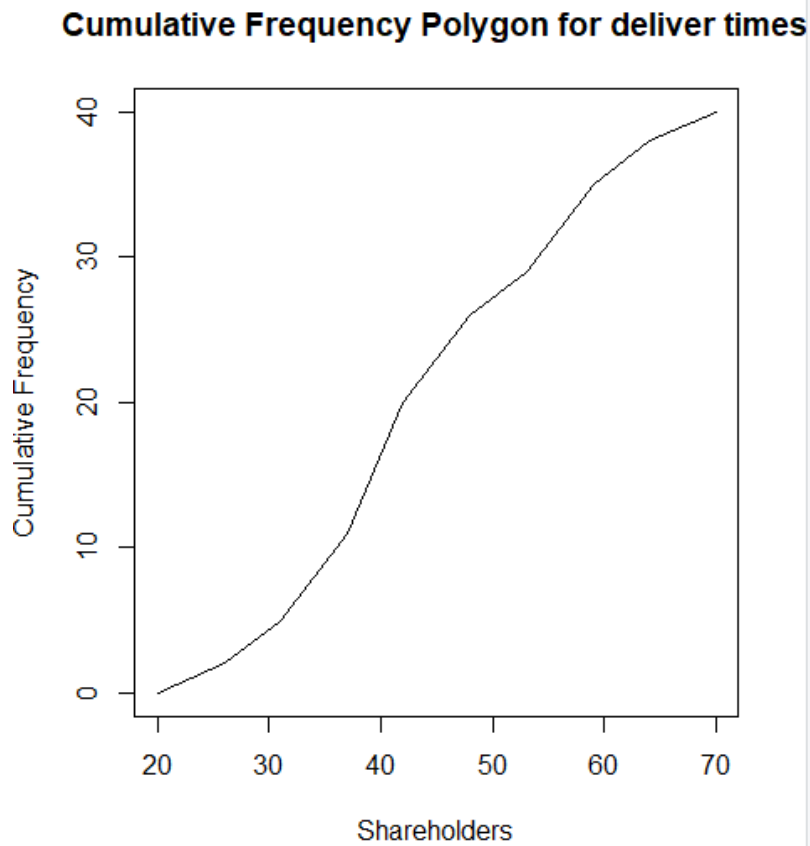
	D_Time	var2	var3	var4	var5	var6	var7
1	34						
2	54						
3	47						
4	29						
5	39						
6	61						
7	20						
8	40						
9	57						
10	36						
11	38						
12	44						
13	59						
14	38						
15	40						
16	40						
17	67						
18	66						
19	55						



Q3).

Most of the data is at the low end, forming a peak on the left, and a few high values stretch the graph out to the right.

Q4).



```
1 setwd("C:\\Users\\it24100680\\Desktop\\IT24100680")
2 Delivery_Times<-read.table("Exercise - Lab 05.txt",header=TRUE,sep = ",")
3 fix(Delivery_Times)
4 attach(Delivery_Times)
5
6 #Q2
7 names(Delivery_Times)<-c("D_Time")
8 attach(Delivery_Times)
9 fix(Delivery_Times)
10 histogram<-hist(D_Time,main="Histogram for deliver times",breaks = seq(20, 70,length = 10),right = FALSE)
11
12 #Q3
13 #Most of the data is at the low end, forming a peak on the left, and a few high values stretch the graph out to the right.
14
15 #Q4
16 breaks <- round(histogram$breaks)
17 freq <- histogram$counts
18 mids <- histogram$mids
19
20 classes <- c()
21
22 ##Creating a "for" loop to assign classes of the frequency distribution into "Classes" variable
23
24 for(i in 1:length(breaks)-1){
25   classes[i] <- paste0("[", breaks[i], ", ", breaks[i+1], ")")
26 }
27
28 #get cumulative frequencies
29 cum.freq <- cumsum(freq)
30
31 new <- c()
32
33 ## Using "for" loop to store cumulative frequencies in order to get the ogive
34 for(i in 1:length(breaks)){
35   if(i==1){
36     new[i] = 0
37   } else {
38     new[i] = cum.freq[i-1]
39   }
40 }
41
42 plot(breaks, new, type = 'l', main = "Cumulative Frequency Polygon for deliver times", xlab = "Shareholders", ylab = "Cumulative Frequency", ylim = c(0,max(cum.freq)))
43
```

```

> #q4
> breaks <- round(histogram$breaks)
> freq <- histogram$counts
> mids <- histogram$mids
> classes <- c()
>
> for(i in 1:length(breaks)-1){
+   classes[i] <- paste0("[", breaks[i], ",", breaks[i+1], ")")
+ }
>
> #get cumulative frequencies
> cum.freq <- cumsum(freq)
>
> new <- c()
>
> ## Using "for" loop to store cumulative frequencies in order to get the ogive
> for(i in 1:length(breaks)){
+   if(i==1){
+     new[i] = 0
+   } else {
+     new[i] = cum.freq[i-1]
+   }
+ }
> plot(breaks, new, type = 'l', main = "Cumulative Frequency Polygon for deliver times", xlab = "Shareholders", ylab = "Cumulative Frequency", ylim = c(0,max(cum.freq)))
> |

```