Exercises Frecuency Measures and Graphical Representation of Data

Abel Isaias Gutierrez-Cruz

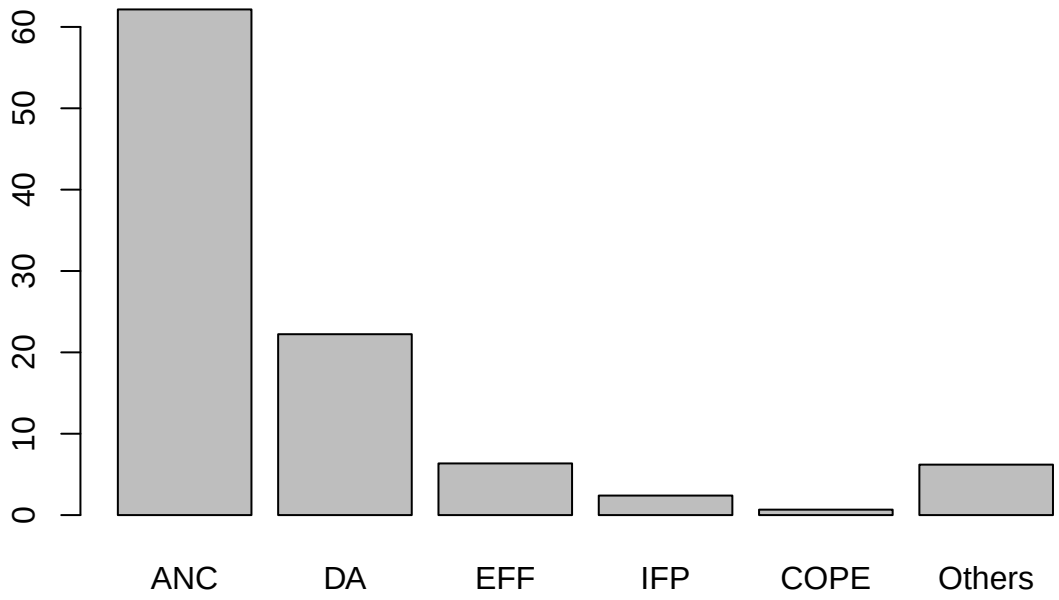
26/6/2021

Excercise 2.1 Consider the results of the national elections in South Africa in 2014 and 2009:

Party	Results 2014(%)	Results 2009(%)
ANC	36.15	65.90
DA	22.23	16.66
EFF	6.35	-
IFP	2.40	4.55
COPE	0.67	7.42
Others	6.20	5.47

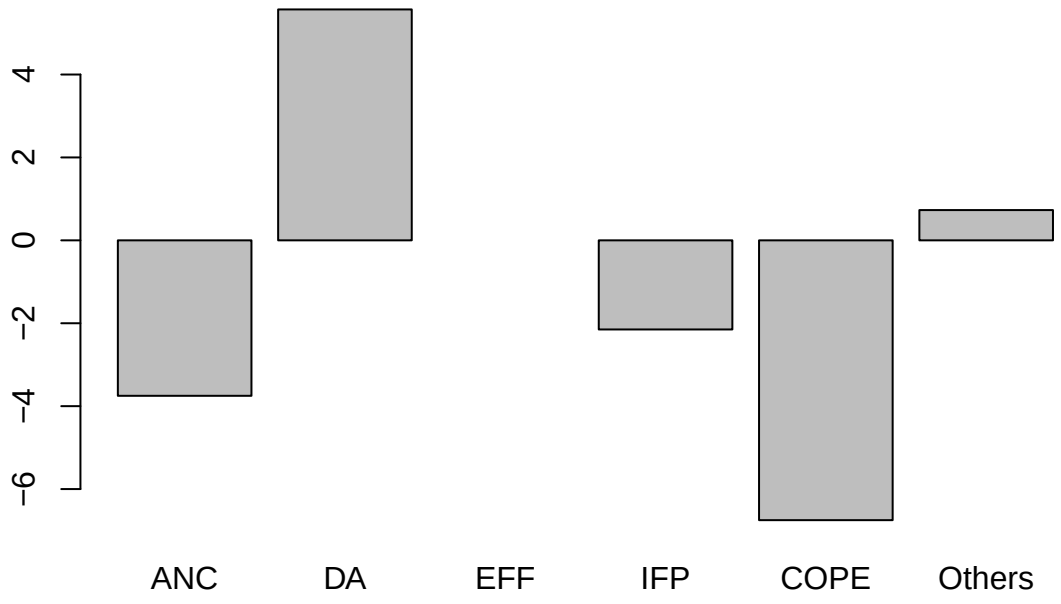
a. Summarize the results of the 2014 elections in a bar chart. Do it manually and by using R.

```
exercise2.1 <- data.frame(results2014 = c(62.15, 22.23, 6.35, 2.40, 0.67, 6.20),
                          results2009 = c(65.90, 16.66, NA, 4.55, 7.42, 5.47),
                          row.names = c("ANC", "DA", "EFF", "IFP", "COPE", "Others"))
barplot(exercise2.1$results2014, names.arg = rownames(exercise2.1))
```



b. How would you compare the results of the 2009 and 2014 elections? Offer a simple solution that can be represented in a single plot. Construct this plot in R.

```
difference <- exercise2.1$results2014 - exercise2.1$results2009
barplot(difference, names.arg = rownames(exercise2.1))
```



Exercise 2.2 Consider a variable X describing the time until the first goal was scored in the matches of the 2006 football World Cup competition. Only matches with at least one goal are considered, and goals during the xth minute of extra time are denoted as 90 + x:

a. What is the scale of X ?

Continuous scale

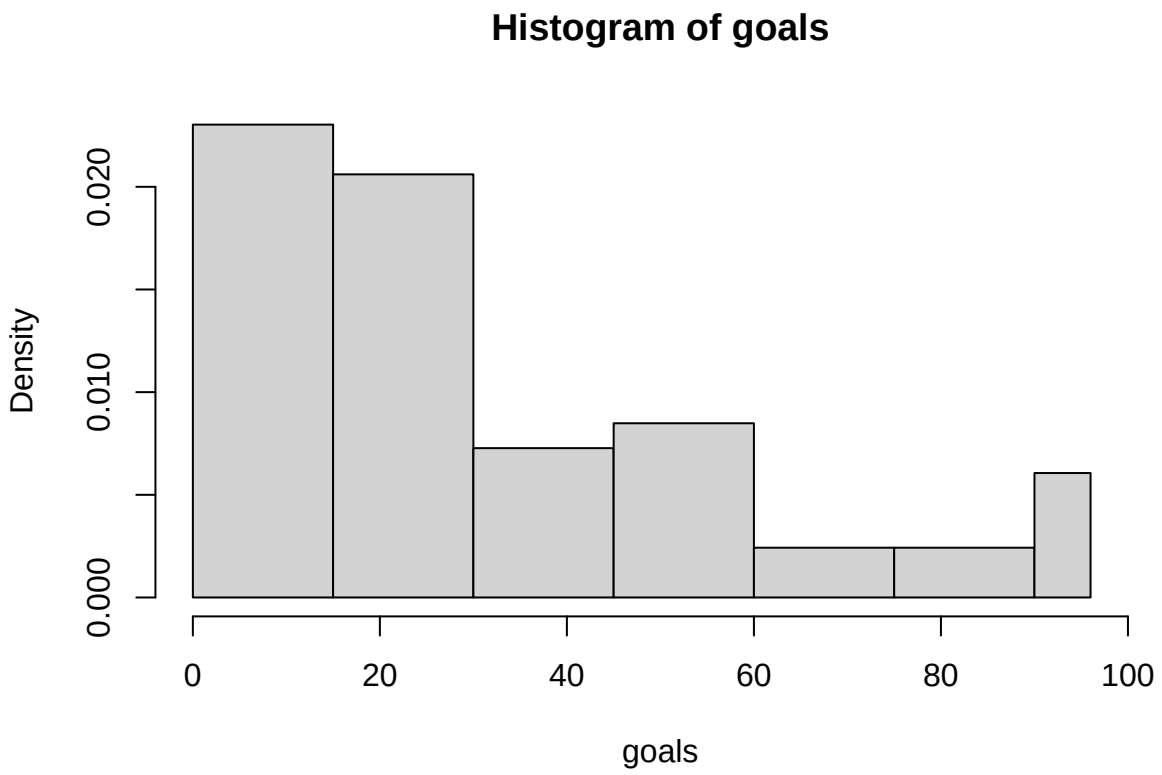
b. Write down the frequency table of X based on the following categories: [0, 15), [15, 30), [30, 45), [45, 60), [60, 75), [75, 90), [90, 96).

```
goals <- c(6,24,91,8,4,25,3,83,89,34,25,24,18,6,23,10,28,4,63,6,60,5,40,2,22,26,23,26,
44,49,34,2,33,9,16,55,23,13,23,4,8,26,70,4,6,60,23,95,28,49,6,57,33,56,7)
groups <- c(0, 15, 30, 45, 60, 75, 90, 96)
labels_goals <- c(15, 30, 45, 60, 75, 90, 96)
grouped_goals <- cut(goals, breaks = groups, labels = labels_goals, right = FALSE)
nj <- as.vector(table(grouped_goals))
fj <- as.vector(round(table(grouped_goals)/length(goals), 2))
table_frecuency_2.2 <- data.frame(j = c(1:7), "ej-1" = groups[1:7], "ej" = groups[2:8], "nj" = nj,
           "fj" = fj, "dj" = c(15, 15, 15, 15, 15, 15, 6))
kable(table_frecuency_2.2)
```

j	ej.1	ej	nj	fj	dj
1	0	15	19	0.35	15
2	15	30	17	0.31	15
3	30	45	6	0.11	15
4	45	60	5	0.09	15
5	60	75	4	0.07	15
6	75	90	2	0.04	15
7	90	96	2	0.04	6

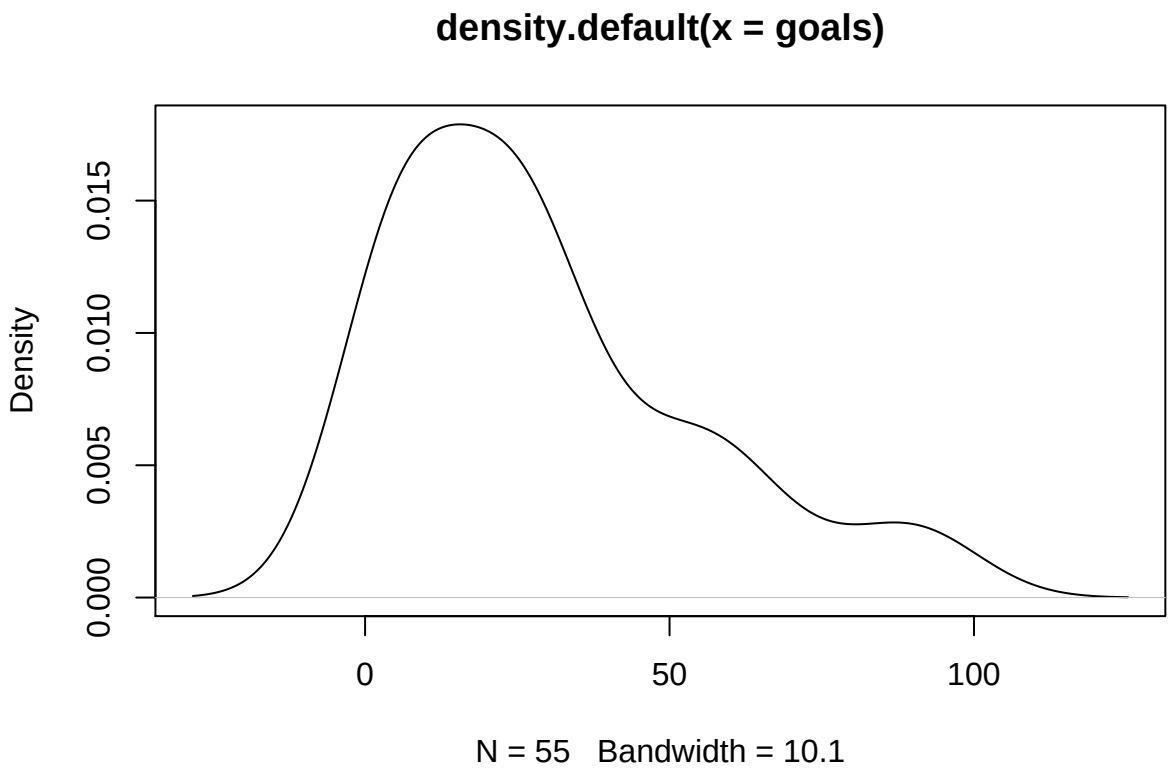
c. Draw the histogram for X with intervals relating to the groups from the frequency table.

```
hist(goals, breaks = groups, xlim = c(0, 100))
```



d.Now use R to reproduce the histogram. Compare the histogram to a kernel density plot of your choice.

```
plot(density(goals))
```



e. Calculate the empirical cumulative distribution function for the grouped data.

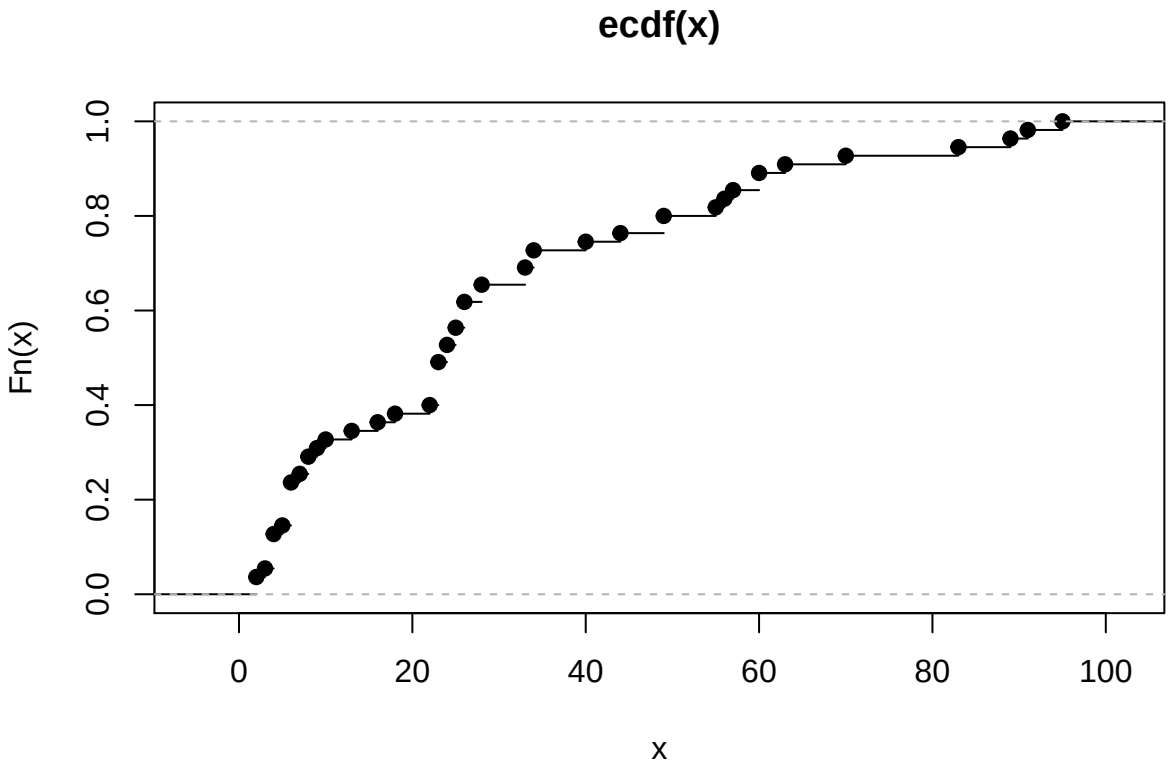
```
Fx <- cumsum(table_frecuency_2.2$fj)
table_frecuency_2.2 <- mutate(table_frecuency_2.2, "F(x)" = Fx)
kable(table_frecuency_2.2)
```

j	ej.1	ej	nj	fj	dj	F(x)
1	0	15	19	0.35	15	0.35
2	15	30	17	0.31	15	0.66
3	30	45	6	0.11	15	0.77
4	45	60	5	0.09	15	0.86
5	60	75	4	0.07	15	0.93
6	75	90	2	0.04	15	0.97
7	90	96	2	0.04	6	1.01

f. Use R to plot the ECDF (via a step function) for

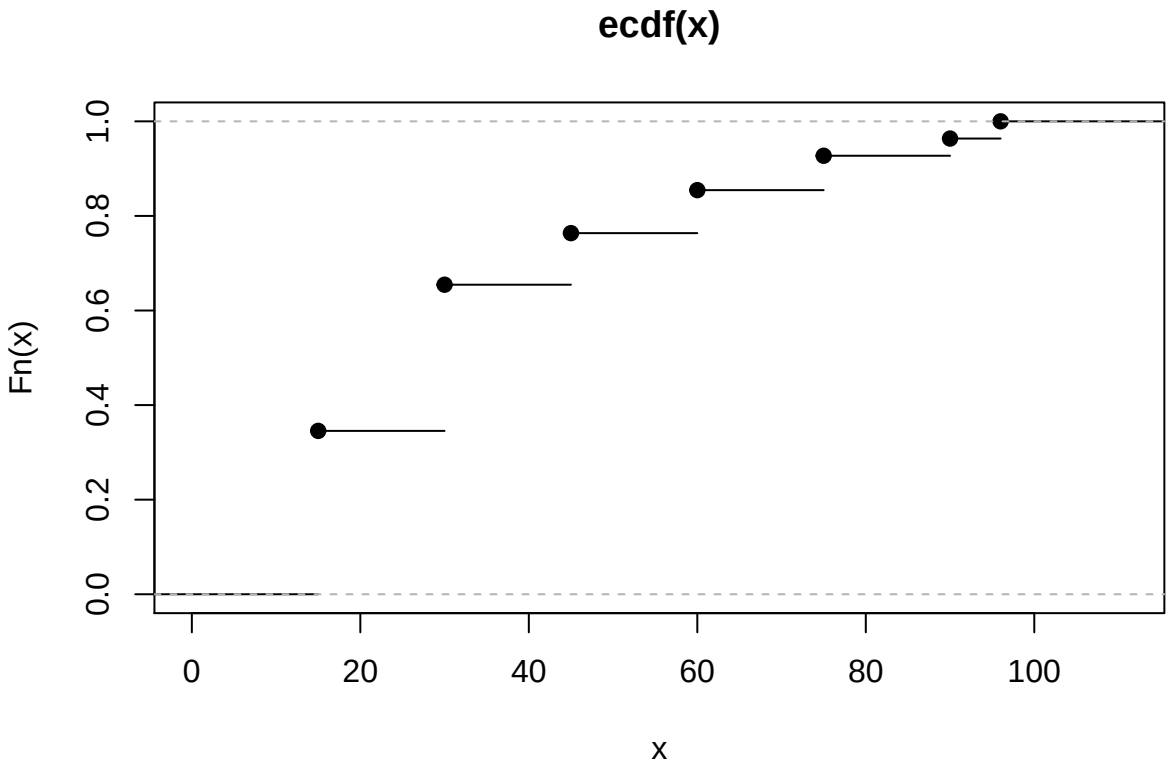
- the original data

```
plot.ecdf(goals)
```



- the grouped data.

```
plot.ecdf(as.numeric(as.character(grouped_goals)))
```



g. Consider the grouped data. Now assume that the values within each interval are distributed uniformly. Determine the proportion of first goals which occurred

- in the first half, i.e. during the first 45 min,

```
g1 <- table_frecuency_2.2[3, 7]
```

$H(X \leq 45) = F(45) = 0.77$

- in the last 10 min or during the extra time,

```
g2 <- 1 - (table_frecuency_2.2[5, 7] +
              ((table_frecuency_2.2[6, 5]/table_frecuency_2.2[6, 6])*(80 - 75)))
```

$H(x > 80 = 1 - F(80) = 1 - (F(75) + \frac{f_i}{d_j}(x - e_{j-1})) = 0.0566667$

- between the 20th and 65th min, i.e. what is H ($20 \leq X \leq 65$)?

```
g3 <- (table_frecuency_2.2[4, 7] +
              ((table_frecuency_2.2[5, 5]/table_frecuency_2.2[5, 6])*(65 - 60))) -
    (table_frecuency_2.2[1, 7] +
              ((table_frecuency_2.2[2, 5]/table_frecuency_2.2[2, 6])*(20 - 15))) +
    table_frecuency_2.2[2, 5]
```

$H(20 \leq X \leq 65) = F(65) - F(20) + f(2) = 0.74$

h. Determine the time point at which in 80 % of the matches the first goal was scored at or before this time point

```
h1 <- table_frecuency_2.2[4, 2] + ((0.8 - table_frecuency_2.2[3, 7])/
                                     (table_frecuency_2.2[4, 5]/table_frecuency_2.2[4, 6]))
```

$$F(X) = 0.8 = F(e_{j-1}) + h_j(x_p - e_{j-1})$$

$$x_p = e_{j-1} + \frac{0.8 - F(e_{j-1})}{h_j} = 50$$

Exercise 2.3 Suppose we have the following information to construct a histogram for a continuous variable with 2000 observations:

```
exercise2.3 <- data.frame("j" = c(1:4), "ej-1" = c(0, 1, 4, 7), "ej" = c(1, 4, 7, 8),
                          "dj" = c(1, 3, 3, 1), "hj" = rep(0.125, 4))
exercise2.3
```

```
##   j ej.1 ej dj    hj
## 1 1    0  1  1 0.125
## 2 2    1  4  3 0.125
## 3 3    4  7  3 0.125
## 4 4    7  8  1 0.125
```

a. Determine the relative frequencies for each interval

```
exercise2.3 <- mutate(exercise2.3, fi = exercise2.3$hj * exercise2.3$dj)
kable(exercise2.3$fi)
```

x
0.125
0.375
0.375
0.125

b. Determine the absolute frequencies

```
exercise2.3 <- mutate(exercise2.3, Fx = cumsum(exercise2.3$fi))
kable(exercise2.3$Fx)
```

x
0.125
0.500
0.875
1.000

Exercise 2.4 A university survey was conducted on 500 first-year students to obtain knowledge about the size of their accommodation (in square metres).

```
exercise2.4 <- data.frame("j" = c(1:5), "from" = c(8, 14, 22, 34, 50),
                          "to" = c(14, 22, 34, 50, 82),
                          "F(x)" = c(0.25, 0.40, 0.75, 0.97, 1))
kable(exercise2.4)
```

j	from	to	F.x.
1	8	14	0.25
2	14	22	0.40
3	22	34	0.75
4	34	50	0.97
5	50	82	1.00

a. Determine the absolute frequencies for each category.

```
exercise2.4 <- mutate(exercise2.4, fj = exercise2.4$F.x. - c(0, exercise2.4$F.x.[1:4]))
exercise2.4 <- mutate(exercise2.4, nj = exercise2.4$fj*500)
kable(exercise2.4)
```

j	from	to	F.x.	fj	nj
1	8	14	0.25	0.25	125
2	14	22	0.40	0.15	75
3	22	34	0.75	0.35	175
4	34	50	0.97	0.22	110
5	50	82	1.00	0.03	15

b. What proportion of people live in a flat of at least 34 m2 ?

```
b <- 1 - exercise2.4[3, 4]
```

$F(x > 34) = 1 - F(34) = 1 - F(34) = 0.25$

Exercise 2.5 Consider a survey in which 100 people were asked to rate on a scale from 1 to 10 how much they agree with the statement that "there is too much football on television". The results are summarized below:

```
exercise2.5 <- data.frame(score = c(1:10), responses = c(1, 3, 8, 8, 27, 30, 11, 6, 4, 2))
kable(exercise2.5)
```

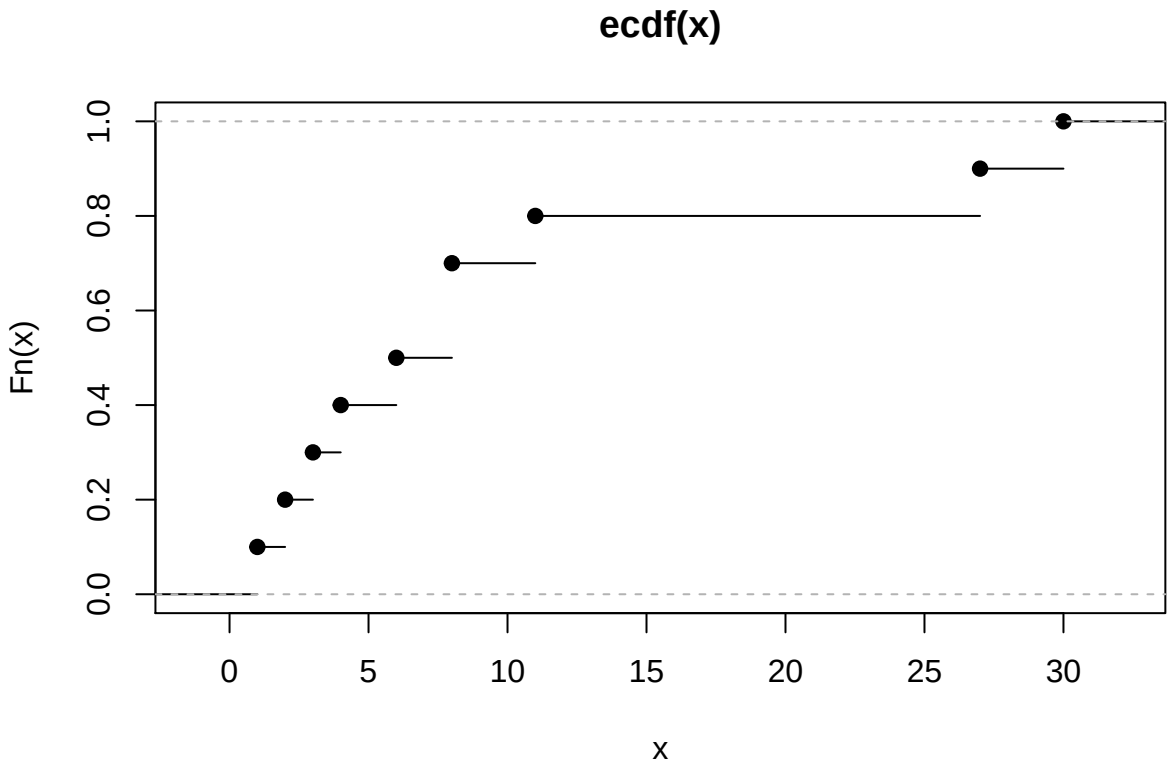
score	responses
1	1
2	3
3	8
4	8
5	27
6	30
7	11
8	6
9	4
10	2

a. Calculate and draw the ECDF of the scores.

```
exercise2.5 <- mutate(exercise2.5, fj = exercise2.5$responses/100)
exercise2.5 <- mutate(exercise2.5, Fj = cumsum(exercise2.5$fj))
kable(exercise2.5)
```

score	responses	fj	Fj
1	1	0.01	0.01
2	3	0.03	0.04
3	8	0.08	0.12
4	8	0.08	0.20
5	27	0.27	0.47
6	30	0.30	0.77
7	11	0.11	0.88
8	6	0.06	0.94
9	4	0.04	0.98
10	2	0.02	1.00

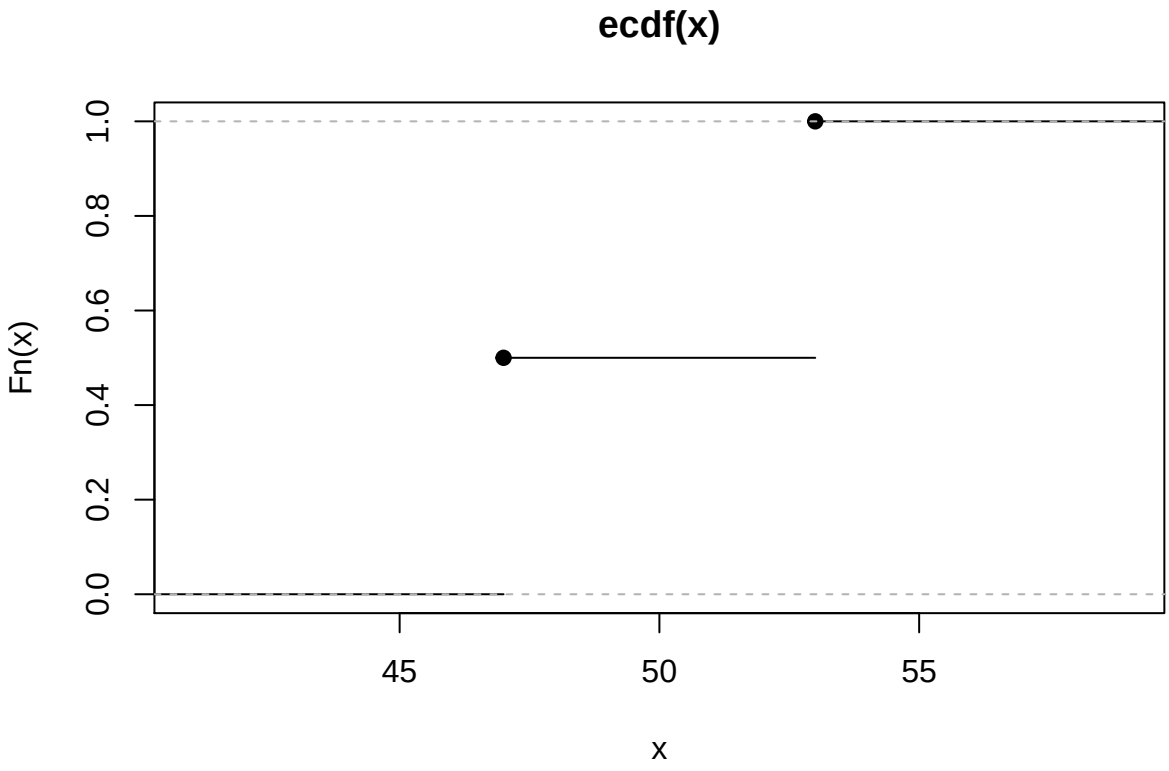
```
plot.ecdf(x = exercise2.5$responses)
```



b. Determine F(3) and F(9).

c. Consider the situation, where the data is summarized in the two categories "disagree" (score ≤ 5) and "agree" (score > 5). What would the ECDF look like under the approach outlined in (2.11)? Determine F(3) and F(9) for the summarized data.

```
groups <- c(0, 6, 11)
labels <- c("disagree", "agree")
grouped_data <- cut(exercise2.5$score, breaks = groups, labels = labels, right = FALSE)
x <- as.array(exercise2.5$responses)
dimnames(x) <- list(grouped_data)
plot.ecdf(tapply(x, grouped_data, sum))
```




```
c <- data.frame(j = c(1:2), nj = tapply(x, grouped_data, sum))
c <- mutate(c, fj = c$nj/100)
c <- mutate(c, Fj = cumsum(c$fj))

c1 <- (c$fj[1]/5)*3

c2 <- c$Fj[1] + (c$fj[2]/5)*(9-5)
```

$$F(3) = F(0) + \frac{0.47}{5}(3-0) = 0.282$$

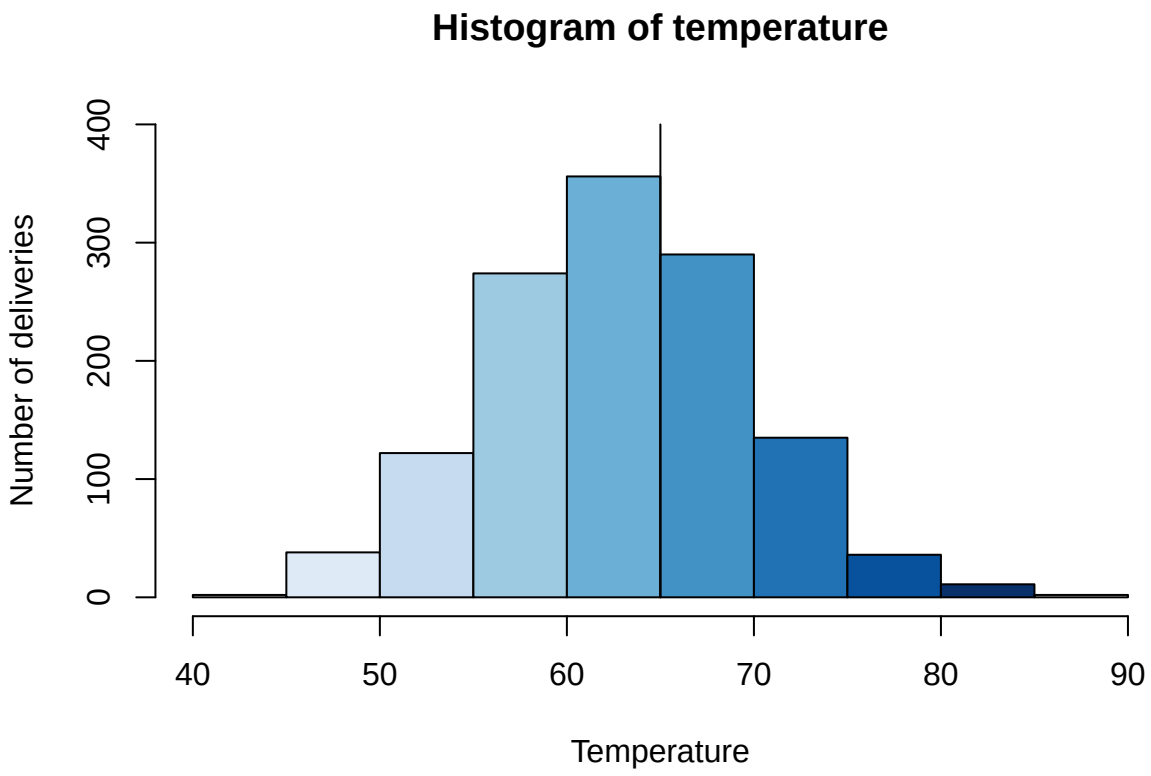
$$F(9) = F(5) + \frac{0.53}{5}(9-5) = 0.894$$

Exercise 2.6 It is possible to produce professional graphics in R. However, it is advantageous to go beyond the default options. To demonstrate this, consider Example 2.1.3 about the pizza delivery data, which is described in Appendix A.4.

a. Set the working directory in R (setwd()), read in the data (read.csv()), and attach the data. Draw a histogram of the variable "temperature". Type ?hist, and view the options. Adjust the histogram so that you are satisfied with (i) axes labelling, (ii) axes range, and (iii) colour. Now use the lines() command to add a dashed vertical line at 65 C (which is the minimum temperature the pizza should have at the time of delivery).

```
pizza_delivery <- read.csv("../../Data/pizza_delivery.csv")
attach(pizza_delivery)
```

```
hist(temperature, xlab = "Temperature", xlim = c(40, 90), ylim = c(0, 400), col = blues9,
     ylab = "Number of deliveries")
lines(c(65, 65), c(0, 400))
```

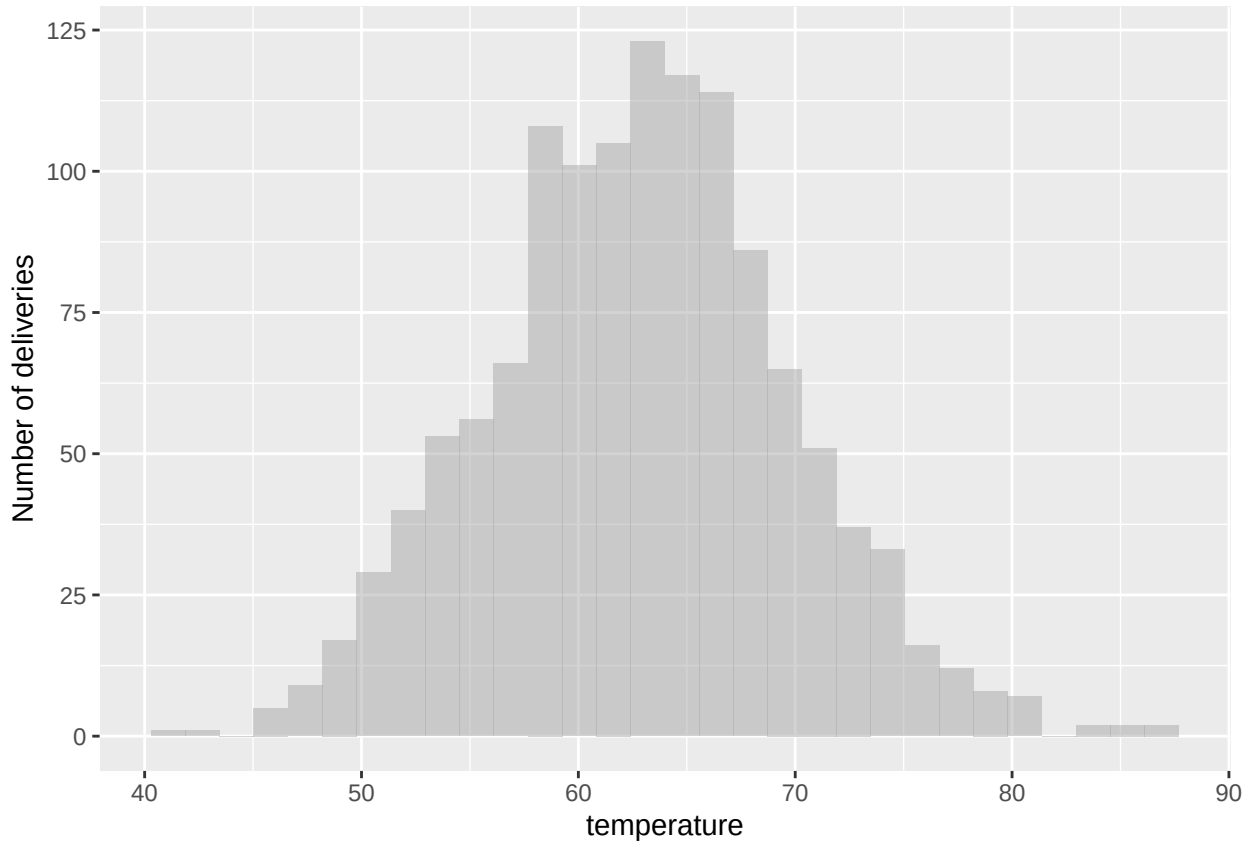


b. Consider a different approach, which constructs plots by means of multiple layers using ggplot2. You need an Internet connection to install the package using the command install.packages('ggplot2'). Browse through the help pages on http://docs.ggplot2.org/current/. Look specifically at the examples for ggplot, qplot, scale_histogram, and scale_y_continuous. Try to understand the roles of "aesthetics" and "geoms". Now, after loading the library via library(ggplot2), create a ggplot object for the pizza data, which declares "temperature" to be the x-variable. Now add a layer with geom_histogram to create a histogram with interval width of 2.5 and dark grey bars which are 50 % transparent. Change the y-axis labelling by adding the relevant layer using scale_y_continuous. Plot the graph.

```
ggplot(data = pizza_delivery, aes(x = temperature)) + geom_histogram(fill = 'darkgrey',
                                                                     alpha = 0.5, bindiwidth = 2.5) +
    scale_y_continuous("Number of deliveries")
```

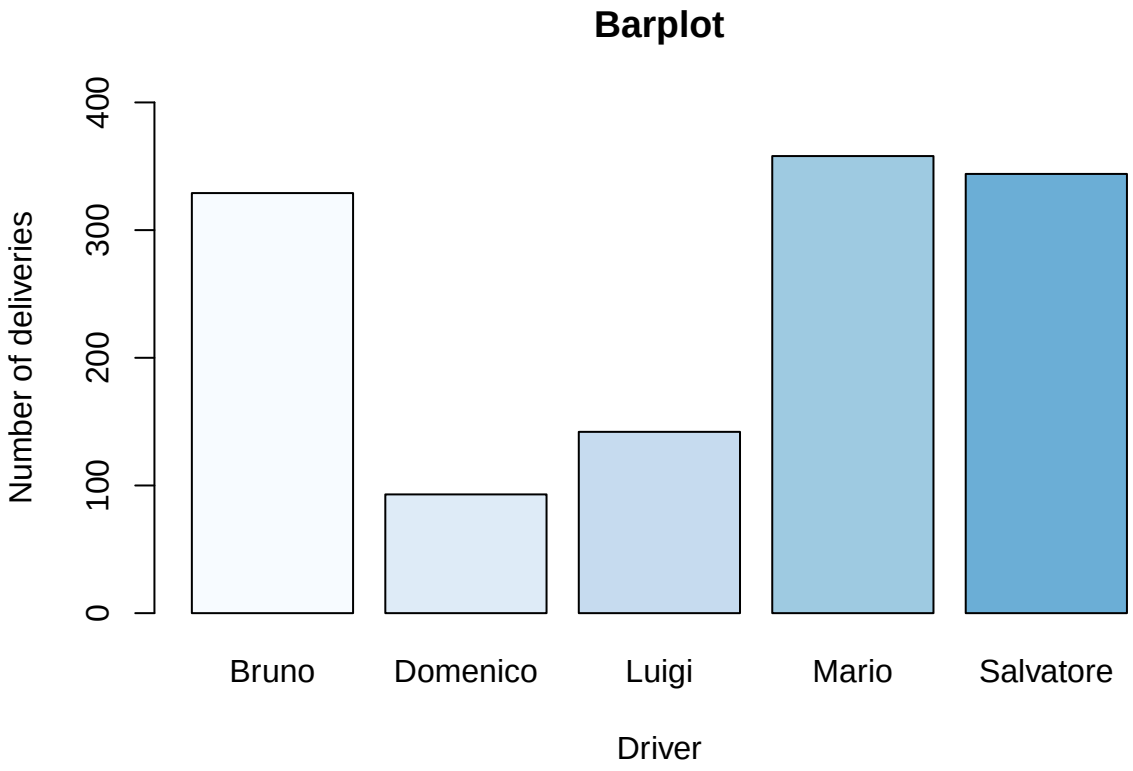
```
## Warning: Ignoring unknown parameters: bindiwidth
```

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



c. Now create a normal bar chart for the variable "driver" in R. Type ?barplot and ?par to see the options one can pass on to barchart() to adjust the graph. Make the graph look good.

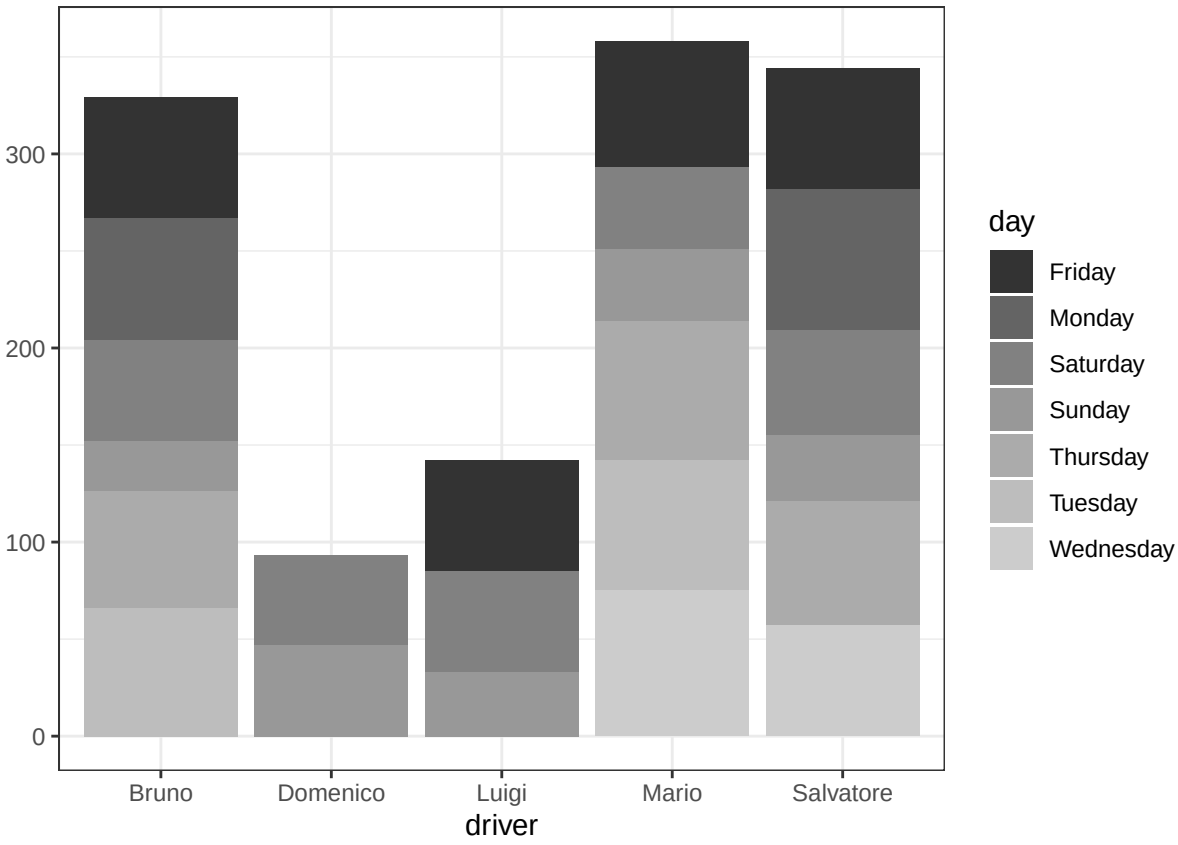
```
barplot(table(driver), ylim = c(0, 400), col = blues9, ylab = "Number of deliveries",
        xlab = "Driver", main = "Barplot")
```



d. Now create the same bar chart with ggplot2. Use qplot instead of ggplot to create the plot. Use an option which makes each bar to consist of segments relating to the day of delivery, so that one can see the number of deliveries by driver to highlight during which days the drivers delivered most often. Browse through "themes" and "scales" on the help page, and add layers that make the background black and white and the bars on a grey scale.

```
qplot(driver, data = pizza_delivery, aes=('bar'), fill=day) + scale_fill_grey() + theme_bw()
```

```
## Warning: Ignoring unknown parameters: aes
```


300
200
100
0
Bruno
Domenico
Luigi
Mario
Salvatore
driver
day
Friday
Monday
Saturday
Sunday
Thursday
Tuesday
Wednesday