



Compte-rendu TP1

Analyse de données

RÉALISATRICE: FATMA LARIBI GL3 GROUPE 2

Question 1

Enter the following vector $X = (a, b, c, d)$, then use the command `sample()` to generate a sample of 1000 entries following the distribution $p = (15\%, 25\%, 40\%, 20\%)$. Name your variable `XX`. We use the function `c()` to define the vectors

The screenshot shows the RStudio interface with the following components:

- Console:** Displays the R code and its output.

```
> x <- c("a","b","c","d")
> P <-c(0.15,0.25,0.4,0.2)
> xx <-sample(x,1000,replace=T,prob=P)
> print(xx)
```

The output shows a single row of 1000 sampled values, starting with: [1] "c" "c" "d" "c" "b" "b" "c" "a" "c" "c" "c" "d" "c".
- Environment:** Shows the current environment with the following variables:

Variable	Value
F_X	num [1:21] 0.122 0.392 0.677 0.867 0...
p	0.1
P	num [1:4] 0.15 0.25 0.4 0.2
P_X	num [1:21] 0.1216 0.2702 0.2852 0.19...
P_Y	num [1:21] 0.00674 0.03369 0.08422 0...
x	int [1:21] 0 1 2 3 4 5 6 7 8 9 ...
X	chr [1:4] "a" "b" "c" "d"
XX	chr [1:1000] "c" "c" "d" "c" "b" "b"...
- Files, Plots, Packages, Help, Viewer:** These panels are visible at the bottom of the interface.

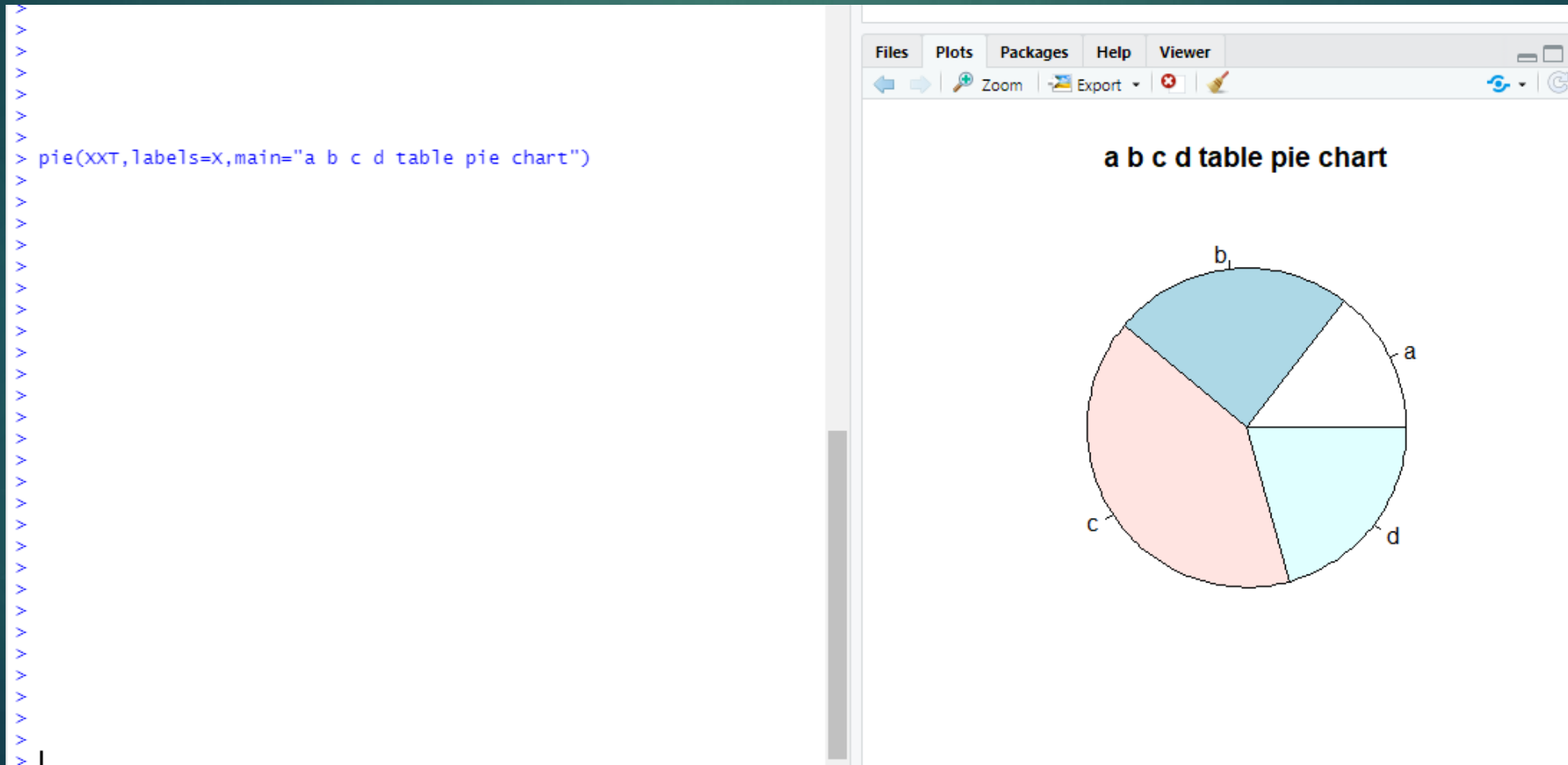
Question 2

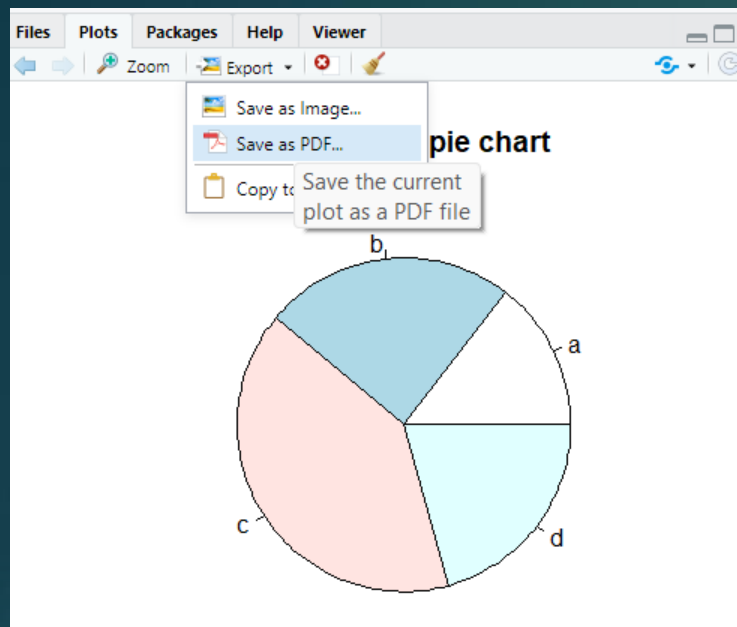
Use the command `table()` to table your statistical variable `XX` and name your table distribution `XXT`.

```
> XXT<- table(XX)
> print(XXT)
XX
  a    b    c    d
145 244 404 207
> |
```

Question 3

Use the command `pie()` to represent graphically XX into a pie chart and save your graphic in a PDF format name your graph le "XXP.pdf".





Save Plot as PDF

PDF Size: (Device Size) 5.18 × 4.97 inches

Orientation: ☐ Portrait ☒ Landscape

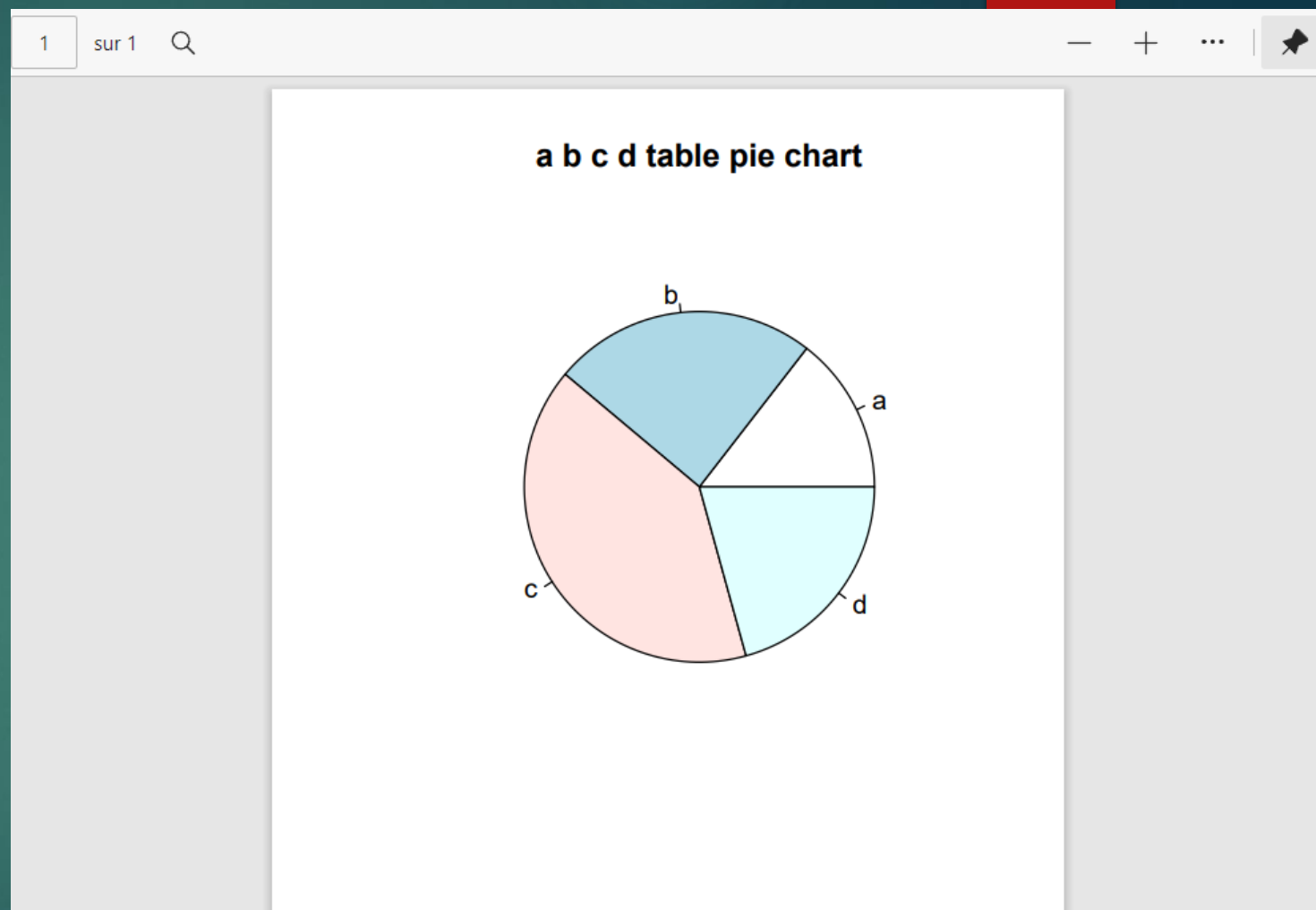
Options: ☒ Use cairo_pdf device

Directory... ~

File name: XXP

☐ View plot after saving

Preview Save Cancel



Question 4

Enter the following vector $Y = (1, 2, 3, 4)$, then use the command `sample()` to generate a sample of 1000 entries following the distribution given by $p = (25\%, 15\%, 25\%, 35\%)$, name your variable `YY`.

The screenshot shows the RStudio interface with the following components:

- Console:** Displays the R code and its output.

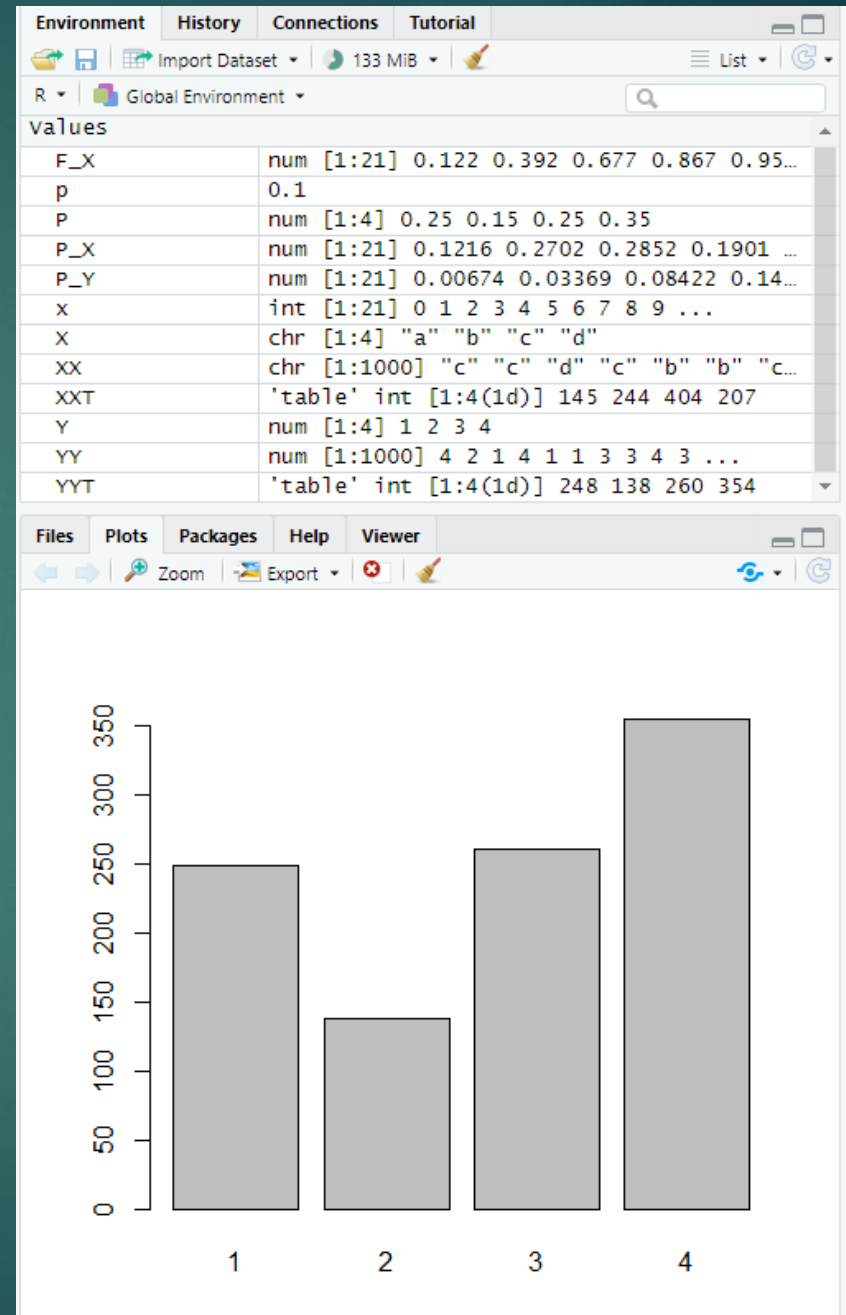
```
> Y<-c(1,2,3,4)
> P <-c(0.25,0.15,0.25,0.35)
> YY <-sample(Y,1000,replace=T,prob=P)
> print(YY)
 [1] 4 2 1 4 1 1 3 3 4 3 1 3 4 3 1 3 4 4 1 3 4 4 3 1 1 1 1 4 4 2
[31] 3 3 3 3 4 3 4 1 1 4 4 1 3 4 4 3 3 4 2 4 4 1 3 4 2 3 4 2 1 4
[61] 4 4 4 4 3 4 1 1 4 1 4 1 4 2 1 3 1 1 4 3 1 4 4 4 3 4 3 4 1 1
[91] 2 3 4 3 2 1 4 4 1 4 4 1 1 2 3 2 1 4 1 4 2 3 4 3 2 2 3 4 4 1
[121] 2 4 1 3 3 4 3 2 1 1 3 2 4 3 4 2 3 3 3 1 3 2 4 2 4 4 4 2 2 4
[151] 4 3 4 1 3 1 4 1 1 3 4 3 4 1 2 2 1 1 3 4 4 1 3 4 4 4 4 1 1 2
[181] 4 4 4 4 3 1 1 3 3 4 3 1 3 1 1 3 1 2 4 3 4 1 4 4 3 1 4 4 3 3
[211] 3 1 3 4 4 4 1 3 3 4 4 1 2 4 4 4 4 4 3 3 2 4 3 3 2 4 2 4 2 1
[241] 1 4 3 4 2 1 1 3 4 1 4 4 1 3 1 3 3 4 4 3 2 1 3 4 4 2 3 4 4 4
[271] 1 2 2 4 1 1 2 1 3 4 4 1 3 2 3 4 2 1 3 1 2 3 4 1 1 1 1 2 2 4
[301] 4 1 4 4 4 3 3 3 3 2 4 4 1 1 3 1 2 4 2 4 3 3 4 4 3 4 4 4 1
[331] 4 1 4 3 4 3 3 3 3 4 1 4 1 1 3 3 3 4 3 1 4 1 2 1 4 2 3 3 1 4
[361] 1 4 4 3 3 2 4 3 4 4 4 1 3 4 3 2 3 4 1 1 1 1 4 4 1 1 4 1 3 4
```
- Environment:** Shows the current environment with the following variables:

Variable	Value
F_X	num [1:21] 0.122 0.392 0.677 0.867 0.95...
p	0.1
P	num [1:4] 0.25 0.15 0.25 0.35
P_X	num [1:21] 0.1216 0.2702 0.2852 0.1901 ...
P_Y	num [1:21] 0.00674 0.03369 0.08422 0.14...
x	int [1:21] 0 1 2 3 4 5 6 7 8 9 ...
X	chr [1:4] "a" "b" "c" "d"
XX	chr [1:1000] "c" "c" "d" "c" "b" "b" "c"...
XXT	'table' int [1:4(1d)] 145 244 404 207
Y	num [1:4] 1 2 3 4
YY	num [1:1000] 4 2 1 4 1 1 3 3 4 3 ...

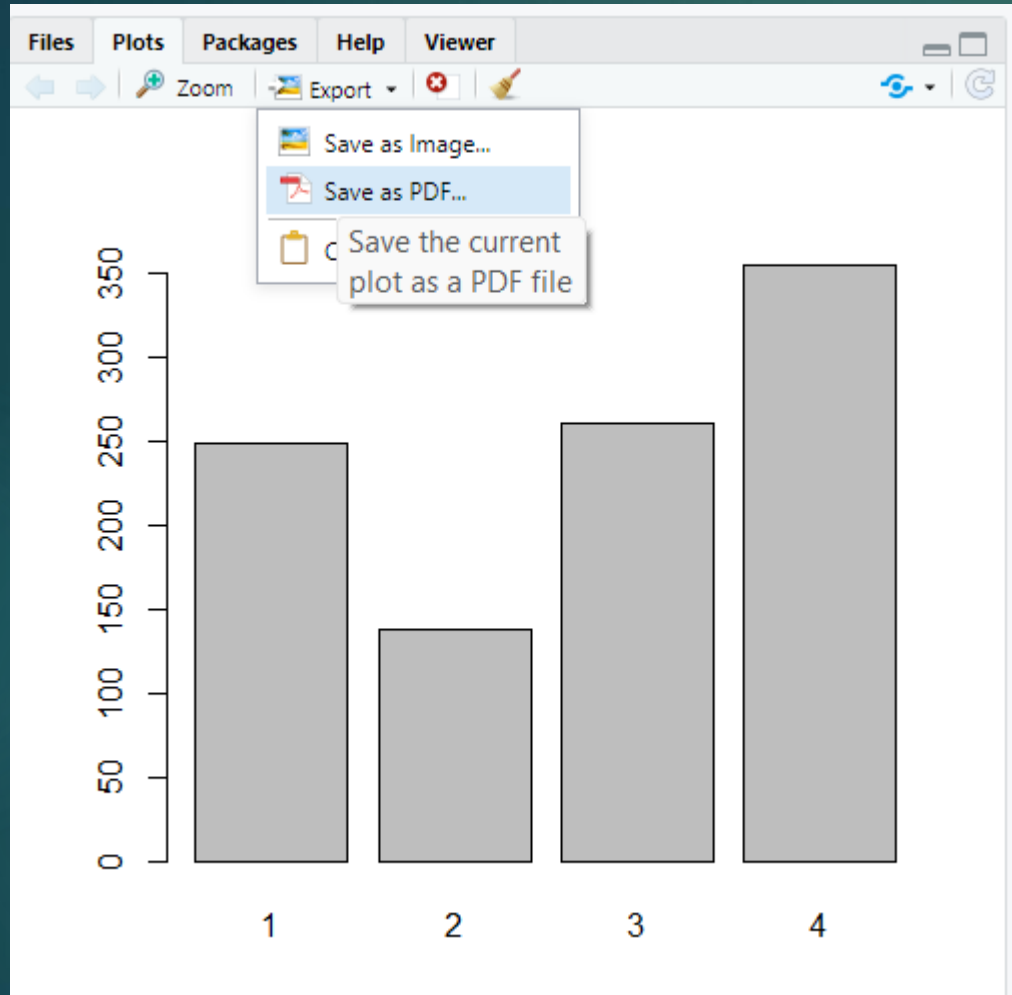
Question 5

Table your data and plot the corresponding bar graph in a pdf file that you name "YYP.pdf".

```
> YYT<-table(YY)
> print(YYT)
YY
  1   2   3   4
248 138 260 354
> barplot(YYT)
>
```



We export the bar plot the same way we did before



Save Plot as PDF

PDF Size: (Device Size) 5.18 x 4.97 inches

Orientation: ☐ Portrait ☒ Landscape

Options: ☐ Use cairo_pdf device

Directory: ~

File name: YYP

☐ View plot after saving

Preview Save Cancel

Question 6

Use the command `rnorm()` to generate 1000 entries following the normal distribution with mean $\mu = 10$ and standard deviation $\sigma = 2$ and name the obtained statistical series `Z`.

The screenshot shows the RStudio interface with the following components:

- Console:** Displays the R code and its output.

```
> μ<-10
> σ<-2
> Z<-rnorm(1000,mean=μ,sd=σ)
> print(Z)
```

The output shows a 12x6 grid of numerical values, representing the first 12 rows of the 1000 generated values.

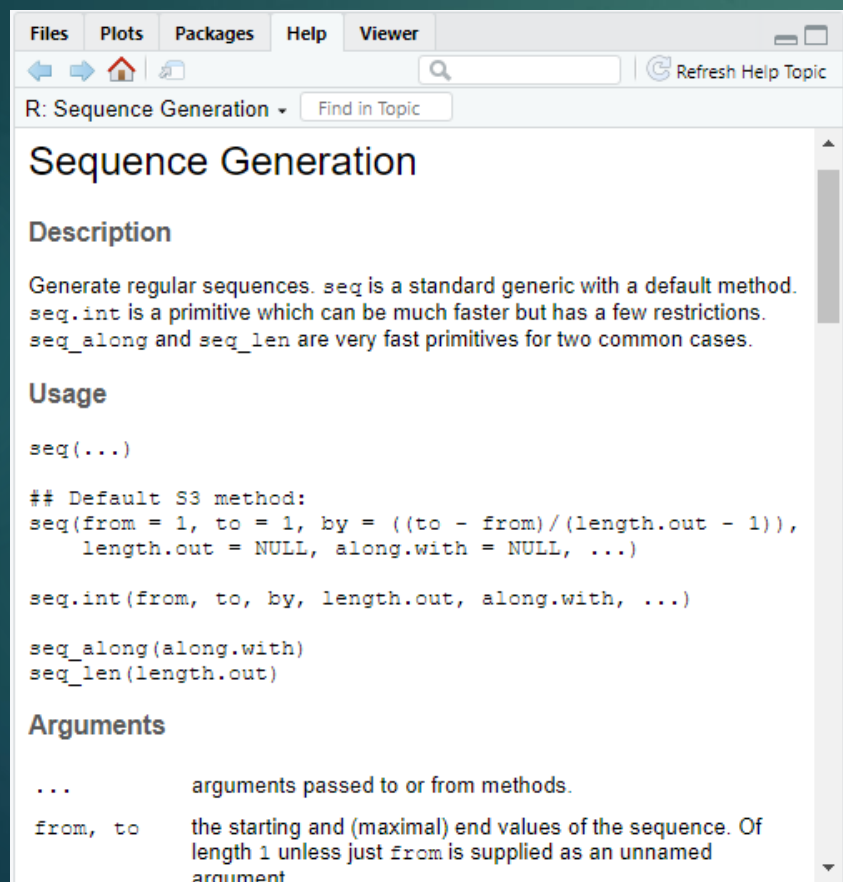
[1]	11.517771	12.804442	11.330632	6.040942	8.314738	10.717025
[7]	10.033261	9.247388	8.022577	9.184272	13.526519	8.904399
[13]	7.077351	11.604195	9.238440	6.556381	9.600900	11.247371
[19]	8.481978	7.585680	12.344353	12.470479	8.792926	10.760761
[25]	8.936111	8.510690	10.987110	4.268494	7.373002	9.266386
[31]	5.814070	11.664434	11.622564	14.357906	8.378960	8.792184
[37]	8.968660	9.098869	10.563900	8.339258	7.342528	8.617155
[43]	11.753988	8.081221	7.859772	8.532344	9.137680	9.037260
[49]	11.496659	10.970988	7.261302	11.749629	7.062979	9.338105
[55]	9.598081	9.491898	15.362611	10.654757	7.127499	10.328826
[61]	9.730150	14.326568	9.003516	10.278373	9.167846	8.091289
[67]	11.801630	9.281975	6.590132	6.291463	7.517687	8.488493
[73]	10.486284	13.234245	9.266450	10.569338	13.448321	8.568137
[79]	10.962888	7.169741	10.984217	13.586512	10.072660	6.357287
[85]	10.735821	7.385416	9.777587	8.593224	8.793244	9.678873
[91]	8.352194	6.784596	10.411804	5.149364	11.342726	7.776407
[97]	8.762813	12.009538	9.624076	13.305543	10.178571	10.889998
[103]	11.409906	7.630931	7.461900	12.344231	8.609919	9.464902
[109]	9.709203	12.238737	10.335591	9.912407	9.774405	10.869451
[115]	10.449409	8.074999	7.696495	12.642671	10.824052	10.622964
[121]	8.615316	7.514814	11.837124	9.805952	8.526229	7.991657
- Environment:** Shows the variables created in the Global Environment.

Variable	Value
P	num [1:4] 0.25 0.15 0.25 0.35
P_X	num [1:21] 0.1216 0.2702 0.2852 0.1901 ...
P_Y	num [1:21] 0.00674 0.03369 0.08422 0.14...
s	2
x	int [1:21] 0 1 2 3 4 5 6 7 8 9 ...
X	chr [1:4] "a" "b" "c" "d"
XX	chr [1:1000] "c" "c" "d" "c" "b" "b" "c..."
XXT	'table' int [1:4(1d)] 145 244 404 207
Y	num [1:4] 1 2 3 4
YY	num [1:1000] 4 2 1 4 1 1 3 3 4 3 ...
YYT	'table' int [1:4(1d)] 248 138 260 354
Z	num [1:1000] 11.52 12.8 11.33 6.04 8.31...
μ	10

Question 7

Use the command `seq()` to set up a sequence of equidistant points in the range of `Z`, name the obtained sequence "`Z_e`". Use the command `cut()` to construct classes and build up the frequency distribution by using `table` which you apply on the output of `cut()`.

Here we divided into 9 classes



The screenshot shows the R help window for the `seq` function. The title is "Sequence Generation". The description states: "Generate regular sequences. `seq` is a standard generic with a default method. `seq.int` is a primitive which can be much faster but has a few restrictions. `seq_along` and `seq_len` are very fast primitives for two common cases." The usage section lists the functions `seq(...)`, `seq.int(from, to, by, length.out, along.with, ...)`, `seq_along(along.with)`, and `seq_len(length.out)`. The arguments section explains that `...` represents arguments passed to or from methods, and `from, to` represent the starting and (maximal) end values of the sequence.

Files **Plots** **Packages** **Help** **Viewer**

R: Sequence Generation Find in Topic

Sequence Generation

Description

Generate regular sequences. `seq` is a standard generic with a default method. `seq.int` is a primitive which can be much faster but has a few restrictions. `seq_along` and `seq_len` are very fast primitives for two common cases.

Usage

```
seq(...)  
  
## Default S3 method:  
seq(from = 1, to = 1, by = ((to - from)/(length.out - 1)),  
     length.out = NULL, along.with = NULL, ...)  
  
seq.int(from, to, by, length.out, along.with, ...)  
  
seq_along(along.with)  
seq_len(length.out)
```

Arguments

... arguments passed to or from methods.

from, to the starting and (maximal) end values of the sequence. Of length 1 unless just `from` is supplied as an unnamed argument

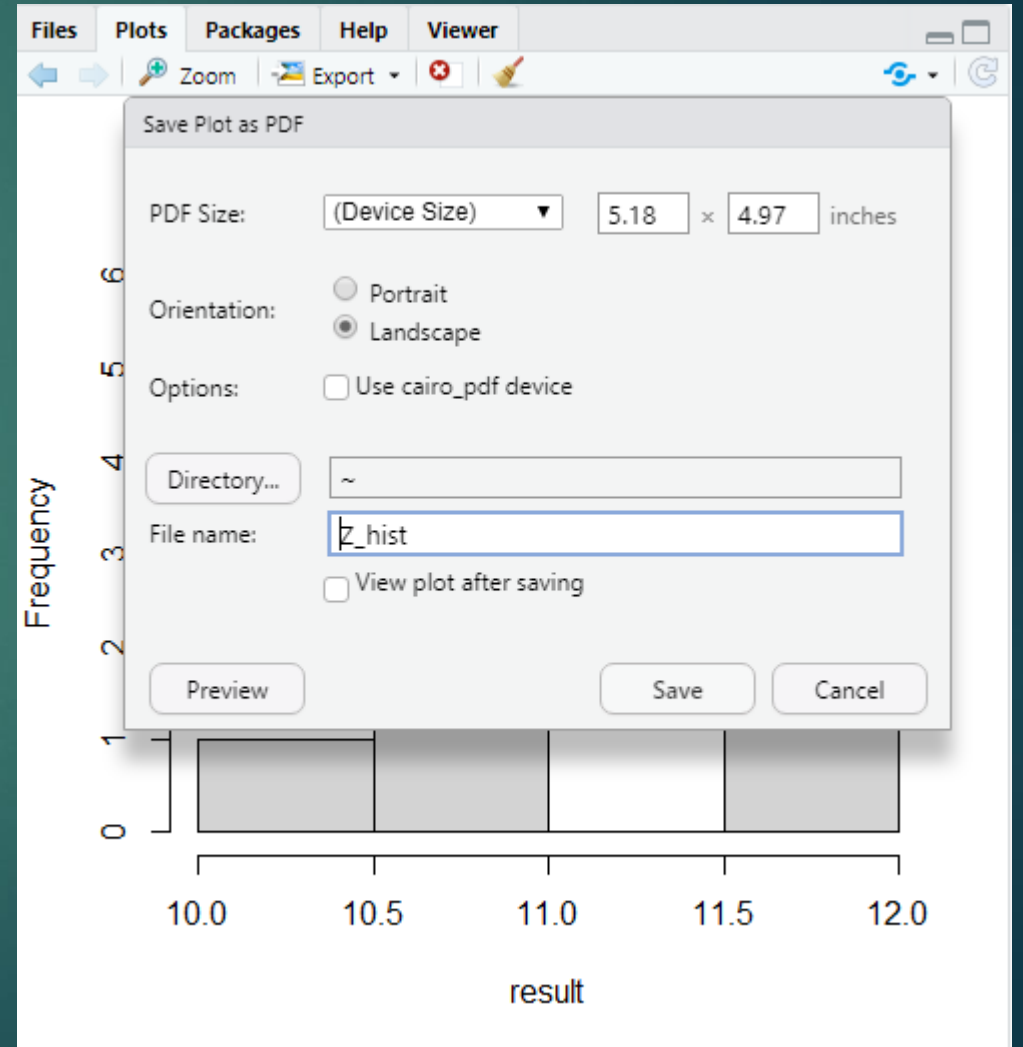
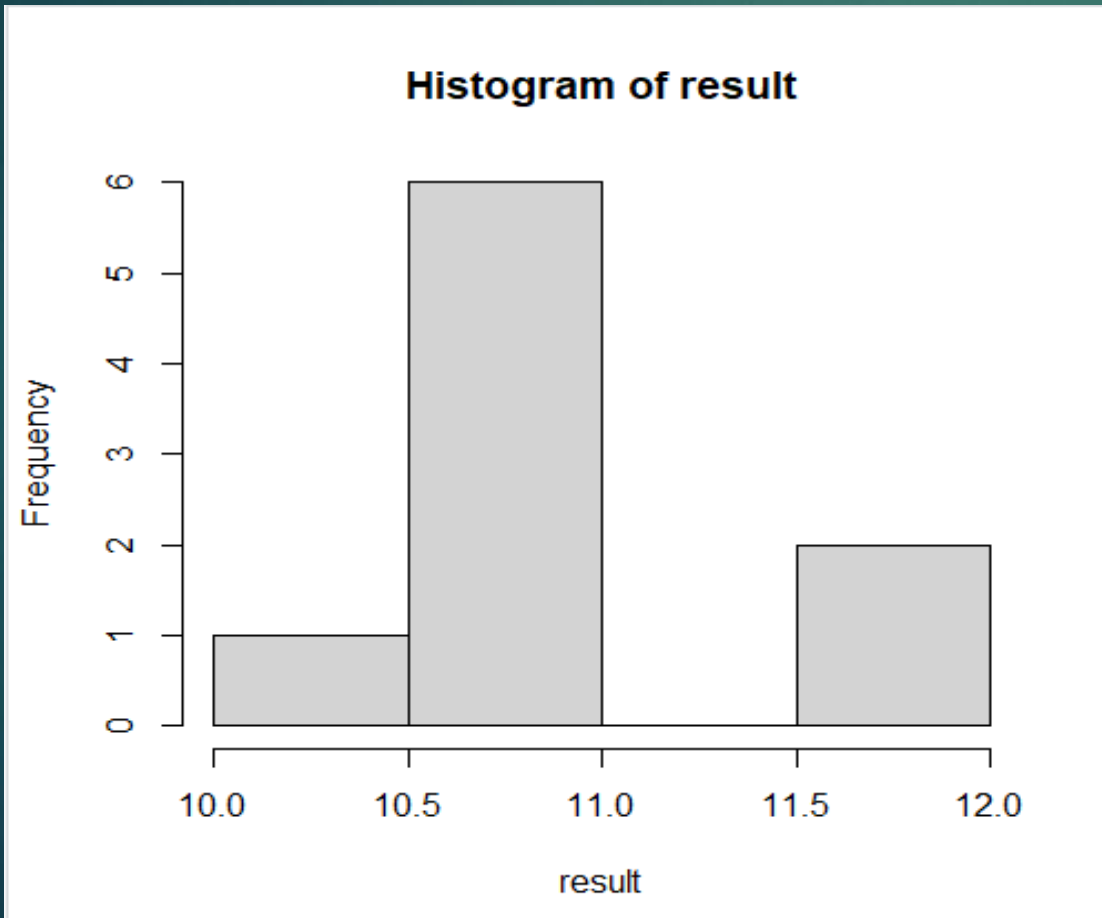
Z	num [1:1000] 11.52 12.8 11.33 6.04 8.31...
Z_e	num [1:100] 11.5 11.5 11.6 11.6 11.6 ...

```
> Z_e<-seq(from=Z[1],to=Z[1000],length.out=100)  
> classes<- cut(Z_e,9)  
> result=table(classes)  
> print(result)  
classes  
(11.5,11.8] (11.8,12.1] (12.1,12.4] (12.4,12.7] (12.7,13]  
          12          11          11          10          12  
(13,13.3] (13.3,13.6] (13.6,13.9] (13.9,14.2]  
          11          11          11          11
```

Question 8

Use the command `hist()` to draw the corresponding histogram and save the obtained graph in a pdf format file "Z_hist.pdf".

```
> hist(result)
```





Thank you for your
attention!