Excersises

Giannis Rokas

2023-12-17

Exercise 1.1:

Use *c* to create a vector that contains the names of you and two of your friends.

```
names <- c("Giannis","Nikos","Marios")
```

Exercise 1.2.

a) use your name and that of your friends from exercise 1 and use the paste command to have them appear next to each other. b) type the following: your year of birth (e.g.1992) : 2023. What do you see? Now type length(your year of birth (e.g.1992) : 2023)–what do you see? What is length?

```
paste(names)
```

```
## [1] "Giannis" "Nikos"   "Marios"
```

```
my_year_of_birth <- 1993:2023
length(my_year_of_birth)
```

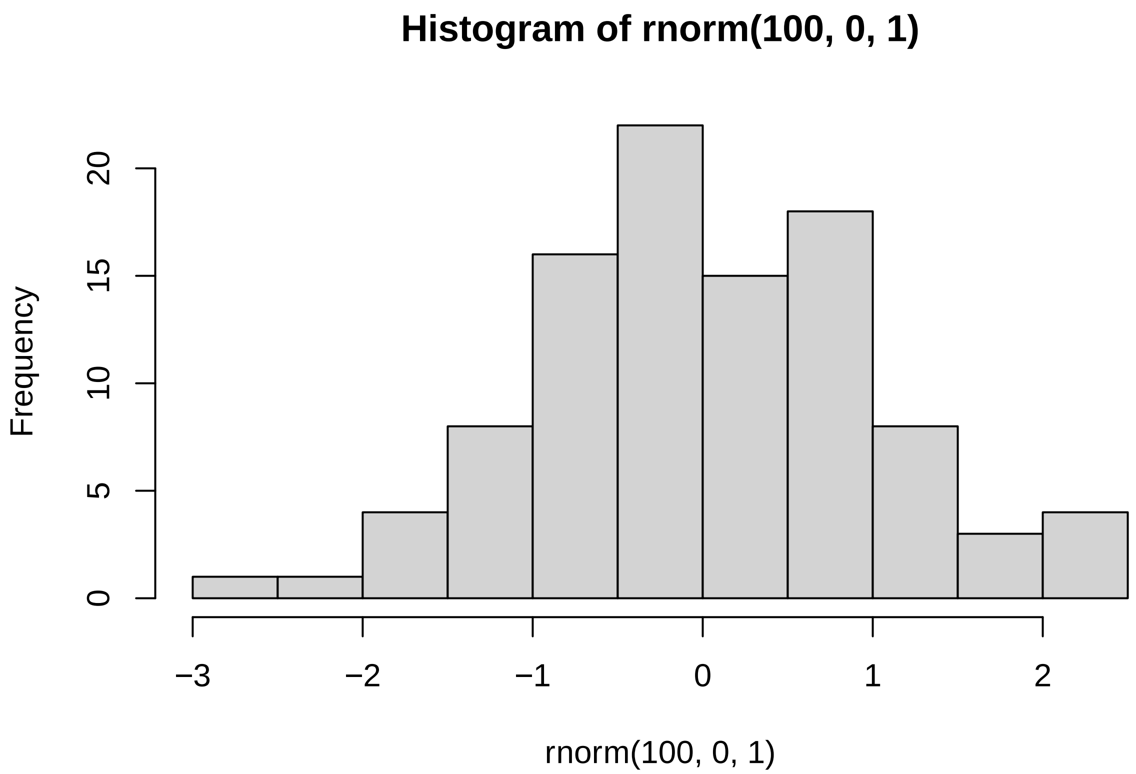
```
## [1] 31
```

```
## 31 my age
```

Exercise 1.3

a) type ?rnorm into the console. What do you see on the right? b) type test <- rnorm(100,0,1) and then type test. What do you see? c) type hist(rnorm(100,0,1)) what do you see?

```
?rnorm
## open the help page with information about this function
test <- rnorm(100,0,1)
## 100 numbers with mean=0 and sd=1
hist(rnorm(100,0,1))
```



```
## histogram of rnorm(100,0,1)
```

Exercise 1.4

a) Type set.seed(123) and generate test_1 <- rnorm(10, 0, 1). type set.seed(123) and generate test_2 <- rnorm(10, 0, 1). Type test_1 and test_2. What do you see? type test_1 == test_2, what do you get?
b) type set.seed(456) and generate test_3 <- rnorm(10, 0, 1). Type test_3. What do you see? Try test_1 == test_3. What do you get?

```
##a)
set.seed(123)
test_1 <- rnorm(10, 0, 1)

set.seed(123)
test_2 <- rnorm(10, 0, 1)
##it generates the same numbers for test_1 and test_2
##the logical equality is TRUE for all numbers

##b)
set.seed(456)
test_3 <- rnorm(10, 0, 1)
##It generates differend numbers this time
##The logical equallity is FALSE for all numbers
```

Exercise 1.5

You have created a histogram. What does this mean? What is the y-axis, what is the x-axis, what does the height of each bar mean in this case?

```r
library(tidyverse)
```

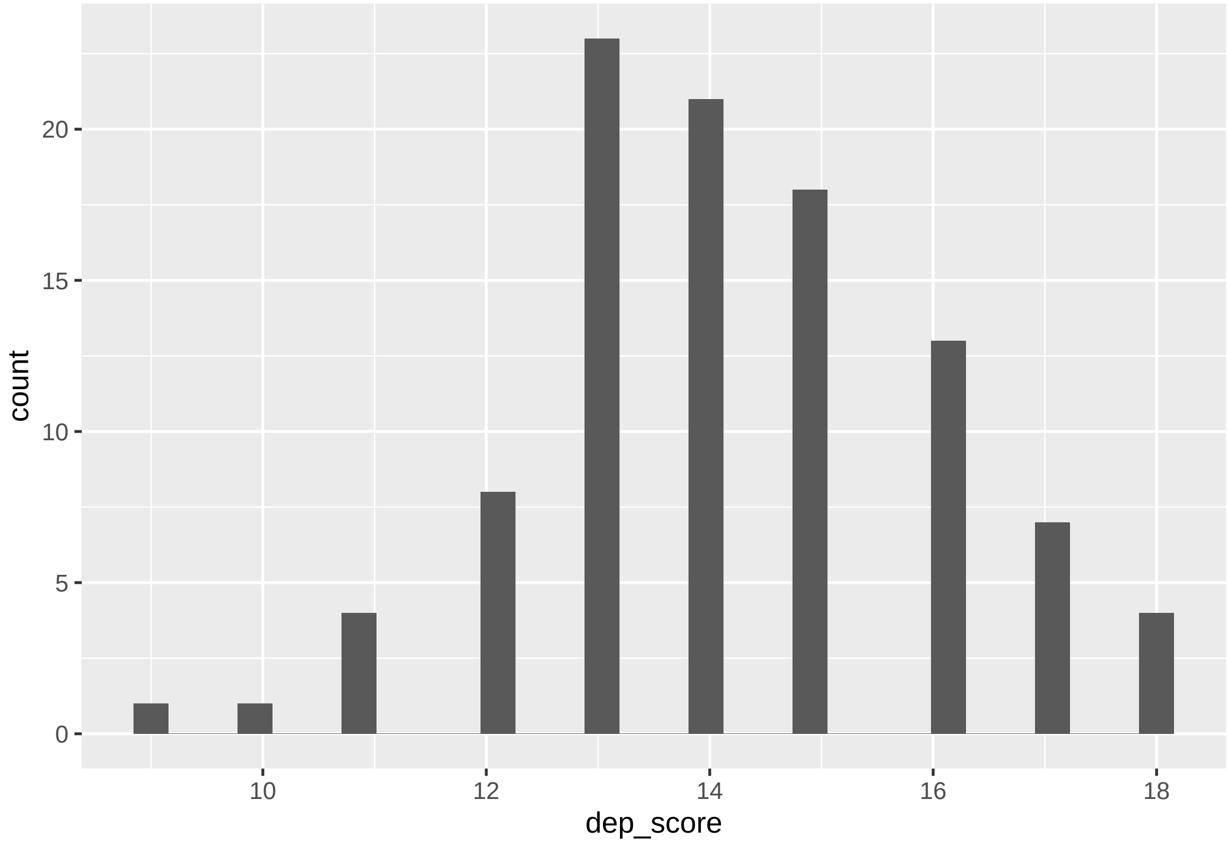
```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```r
n <- 100

set.seed(123)
aiginiteion_dep <- data.frame(
  ids = paste0("aiginiteion_",  1:n),
  dep_score = round(rnorm(n, 14, 2), 0)
)

aiginiteion_dep %>%
  ggplot(aes(x = dep_score))+
  geom_histogram(position = "identity")
```

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



```
##x= score of depression questionnaire
##y=the number of people who score this dep_score
## each bar refer to different scores of our depression questionnaire and the height of each bar visual
```

Exercise 1.6

use the *rbind* function as above, but flip around its contents, having community_dep first. What do you see when you use *head* and *tail*? What do you see when you use *dim*?

```
n <- 100

set.seed(123)
aiginiteion_dep <- data.frame(
  ids = paste0("aiginiteion_",  1:n),
  dep_score = round(rnorm(n, 14, 2), 0)
)
```

```
 set.seed(123)
community_dep <- data.frame(
  ids = paste0("community_",  1:n),
  dep_score = round(rnorm(n, 8, 2), 0)
)
```

```
combined_df <- rbind(community_dep,aiginiteion_dep)
head(combined_df)
```

```
##           ids dep_score
## 1 community_1         7
## 2 community_2         8
## 3 community_3        11
## 4 community_4         8
## 5 community_5         8
## 6 community_6        11
```

```
tail(combined_df)
```

```
##                  ids dep_score
## 195  aiginiteion_95        17
## 196  aiginiteion_96        13
## 197  aiginiteion_97        18
## 198  aiginiteion_98        17
## 199  aiginiteion_99        14
## 200 aiginiteion_100        12
```

```
dim(combined_df)
```

```
## [1] 200   2
```

```
##head= the fisrt 6 are for community
##tail= the last 6 are for aiginiteion
##dim= our dataframe has 200 rows and 2 columns
```

Exercise 1.7

change the *n* further up in your code to n <- 1000. How many rows would you expect, how many columns? Why? Check using the _dim_function. (and put it back to n = 100)

I would expect 2000 rows because now we have 1000 people from aiginiteion and 1000 from community. Columns will be the same (2) because we didn't change our variables (ids and dep__score)

Exercise 1.8

What has changed in the dataset? What command would you run to answer the question?

```
combined_df <- combined_df %>%
  mutate(location = case_when(str_detect(combined_df$ids, "aiginiteion")~ "aiginiteion",
                    str_detect(combined_df$ids, "community")~ "community") )

head(combined_df)
```

```
##           ids dep_score  location
## 1 community_1         7 community
## 2 community_2         8 community
## 3 community_3        11 community
## 4 community_4         8 community
## 5 community_5         8 community
## 6 community_6        11 community
```

```
tail(combined_df)
```

```
##                  ids dep_score    location
## 195  aiginiteion_95        17 aiginiteion
## 196  aiginiteion_96        13 aiginiteion
## 197  aiginiteion_97        18 aiginiteion
## 198  aiginiteion_98        17 aiginiteion
## 199  aiginiteion_99        14 aiginiteion
## 200 aiginiteion_100        12 aiginiteion
```

```
dim(combined_df)
```

```
## [1] 200   3
```

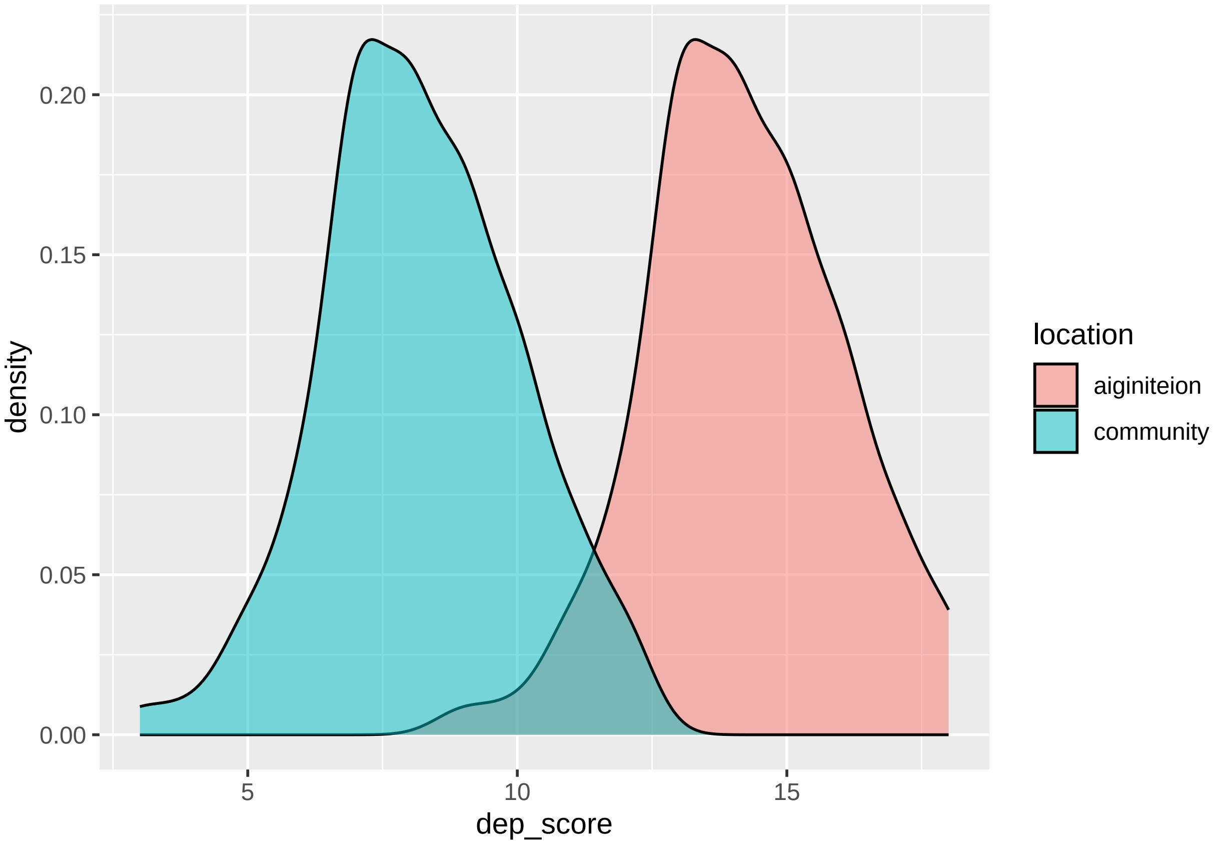
```
## the dim function. Now we have 3 colomns (and the same number od rows of course)
```

Exercise 1.9

Try to interpret the graph. What is the y-axis, what is the x-axis. What does density mean? What is the darker area in the middle?

```
combined_df %>%
  ggplot(aes(dep_score, fill = location))+
  geom_density( alpha = 0.5, bins = 100)
```

```
## Warning in geom_density(alpha = 0.5, bins = 100): Ignoring unknown parameters:
## `bins`
```



## x= the score of our depression questionnaire ## y=the proportion of our sample (in each range of dep_score) ## the darker area in the middle= the proportion of these people who have the same dep_score between the sample of aiginiteion and the sample of community

##t-test.

```
t.test(dep_score~ location, data = combined_df)
```

```
##
##  Welch Two Sample t-test
##
## data:  dep_score by location
## t = 23.431, df = 198, p-value < 2.2e-16
## alternative hypothesis: true difference in means between group aiginiteion and group community is not
## 95 percent confidence interval:
##  5.495018 6.504982
## sample estimates:
## mean in group aiginiteion   mean in group community
##                     14.21                      8.21
```