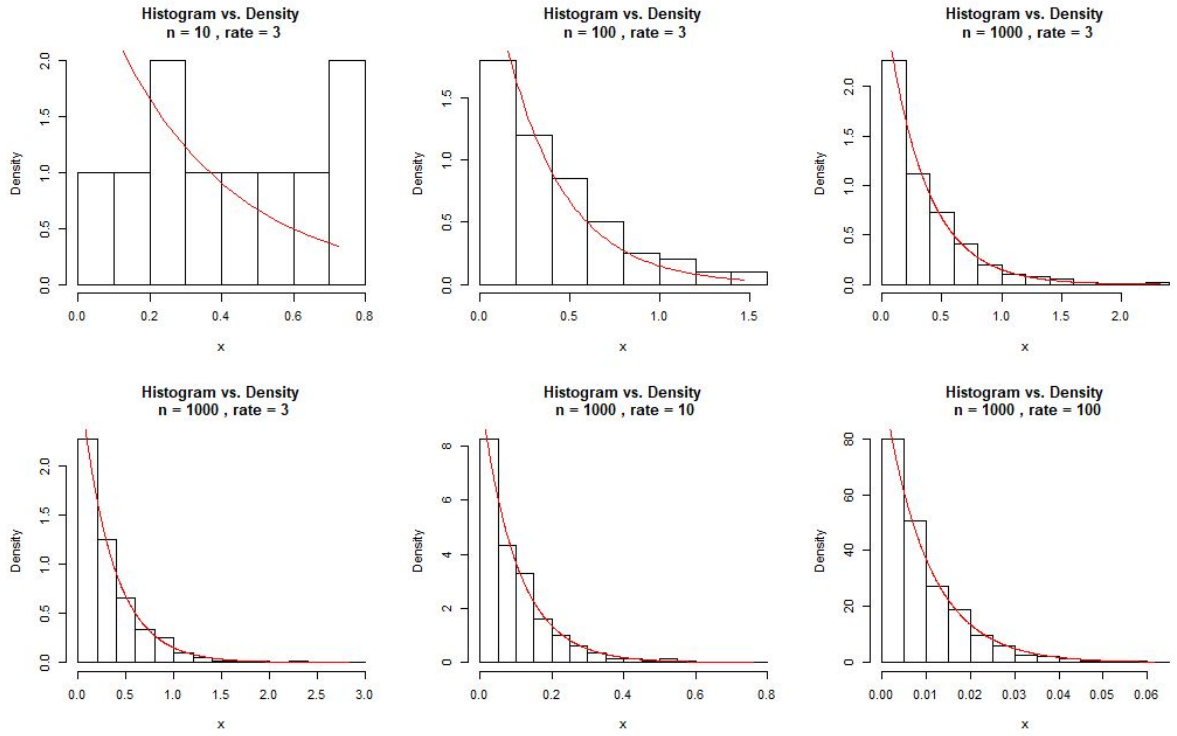
Leon Lušić (2670440)

Assignment 1

Exercise 1.3

The function ***exp*** is set to take two parameters (***n*** and ***rate***), as described in the assignment. The function first creates a sample of size ***n*** from the exponential distribution with rate ***rate*** and stores it into the vector ***x***. Next, a histogram of the vector ***x*** is plotted. After executing the plot, another vector, named ***points***, of equally spaced points between 0 and the largest value in ***x*** is created. The length of this vector is set to be equal to the length of ***x***. For each point in ***points***, the probability density function at that point is calculated. The result is plotted on top of the existing histogram using the function ***lines()***.

In order to study the difference between the histogram and the true density, five different combinations of ***n*** and ***rate*** have been chosen. These are: (10, 3), (100, 3), (1000, 3), (1000, 10), (1000, 100). Naturally, the first number indicates the value of ***n*** while the second indicates the ***rate***. These are the resulting plots:



As can be seen, the increase in ***n***, as well as the increase in ***rate*** both result in the increasingly closer resemblance between the histogram and the true density.

Exercise 1.4

The idea behind the code is to initially prepare the data for analysis and only then perform the appropriate univariate and bivariate analyses.

Data preparation

First, the values in the ***date*** column are converted to ***Date*** objects. This allows us to create two new data frames (***dec*** and ***jan***), from the initial data frame. The contents of these two data frames will be the ***iso_code*** and ***new_cases_smoothed_per_million*** for December 2020 and January 2021. Naturally, the December values correspond to the ***dec*** vector and January values to the ***jan*** vector. After this is done, the ***new_cases_smoothed_per_million*** values are converted from ***character*** datatype to *double* as they represent numbers and should be treated as such. Lastly, we assign new names to the values ***dec_cases*** and ***jan_cases*** to ***dec$new_cases...*** and ***jan$new_cases...*** in order to keep code cleaner and more readable. This concludes the data preparation part and we may move on to data analysis.

Data analysis

Univariate analysis - numerical

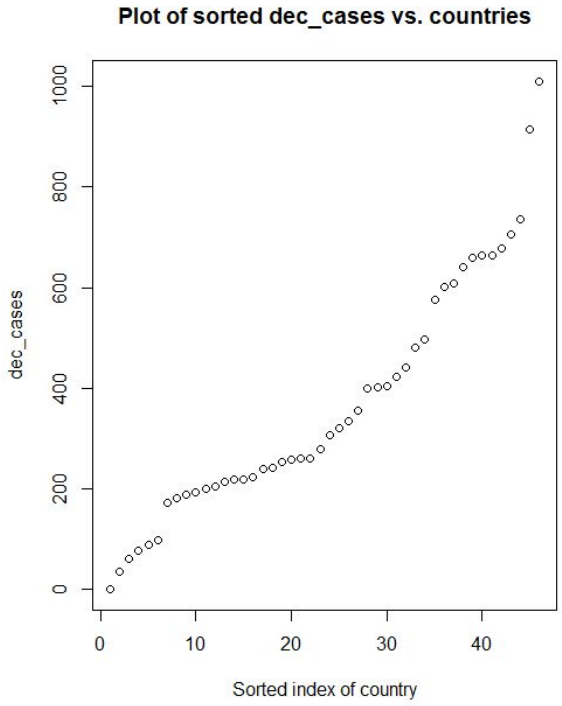
Firstly, we made use of the already built-in functions ***summary()*** and ***sd()***. The first of the two provides the five-number summary of the data, alongside the sample mean. To accompany these values the ***sd()*** function also provides the standard deviation for both of the samples. The resulting values are as follows:

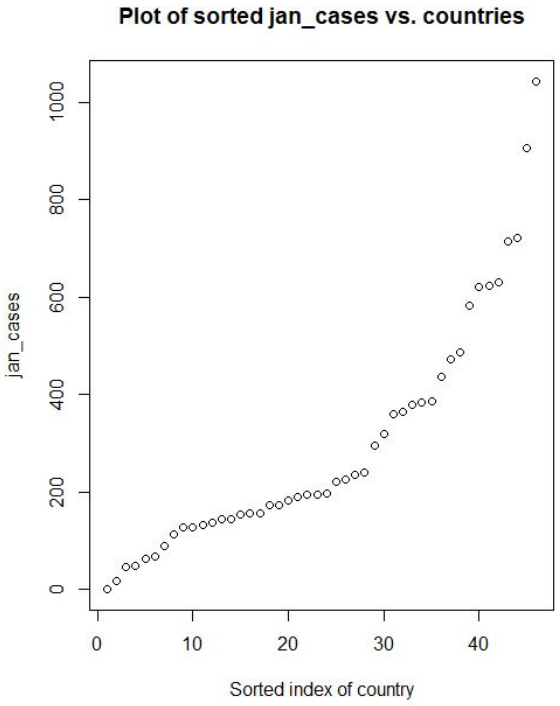
Sample	Min.	1st quartile	Median	Mean	3rd quartile	Max.	Standard deviation
December cases	0	206	294	369	556	1011	237.71
January cases	0	139	195	297	386	1044	240.61

As can be seen from the table, the cases dropped by 20-35% across all categories, except for ***Max.***, which stayed approximately equal (even though it actually increased). The standard deviation stayed roughly the same, which indicates overall decrease in cases over all the European countries in the dataset. This makes sense as late December/early January was the time when a lot of countries introduced new measures intended to decrease the number of new cases - which apparently worked out.

Univariate and bivariate analysis - graphical

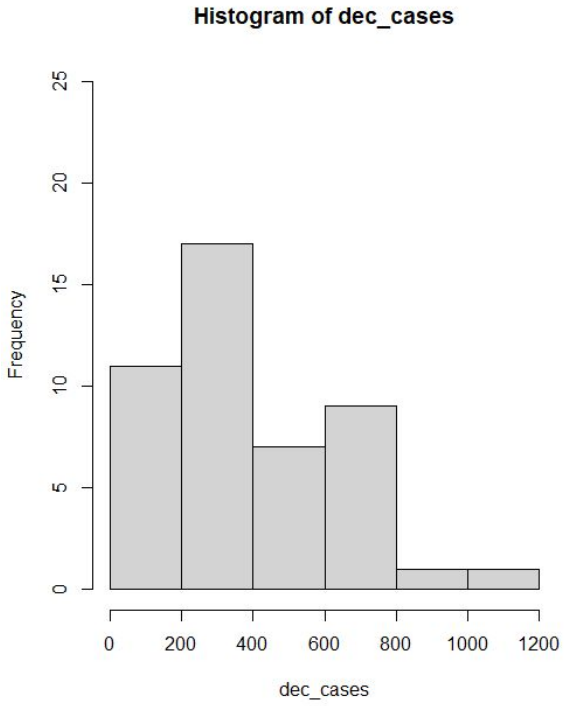
Next up, we may decide to visually compare the two samples. Firstly, we may look at the plots of sorted vectors ***dec_cases*** and ***jan_cases***:

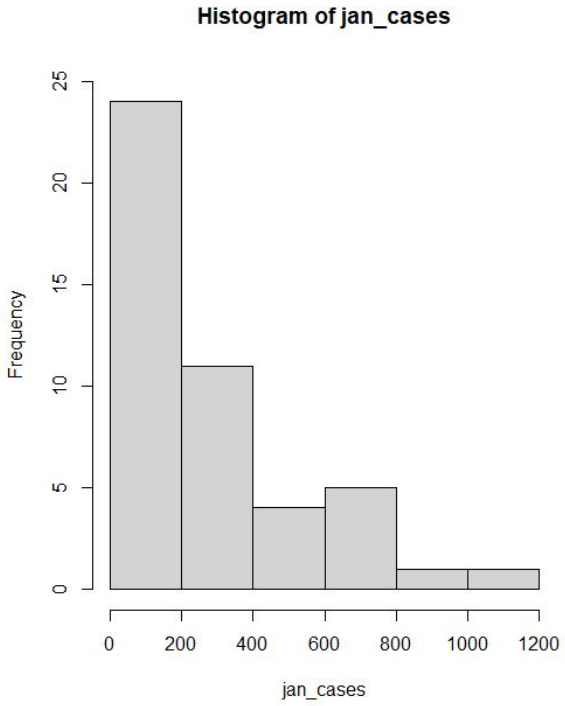




Firstly, note that the indices (*x*-axis values) of countries do not correspond to indices in ***dec_cases*** (or ***jan_cases***, depending on the plot), but to the new indices in the sorted arrays. With that in mind, it can be seen that the graphs indicate a similar pattern. However, the January graph shows higher concentration of new cases near the *x*-axis while the cases in the December graph are more distant. Naturally, this indicates an overall decrease in cases across almost all the countries (in the dataset), although the few countries do not exhibit such decrease, as can be seen from the lingering high-points to the very right in both graphs.

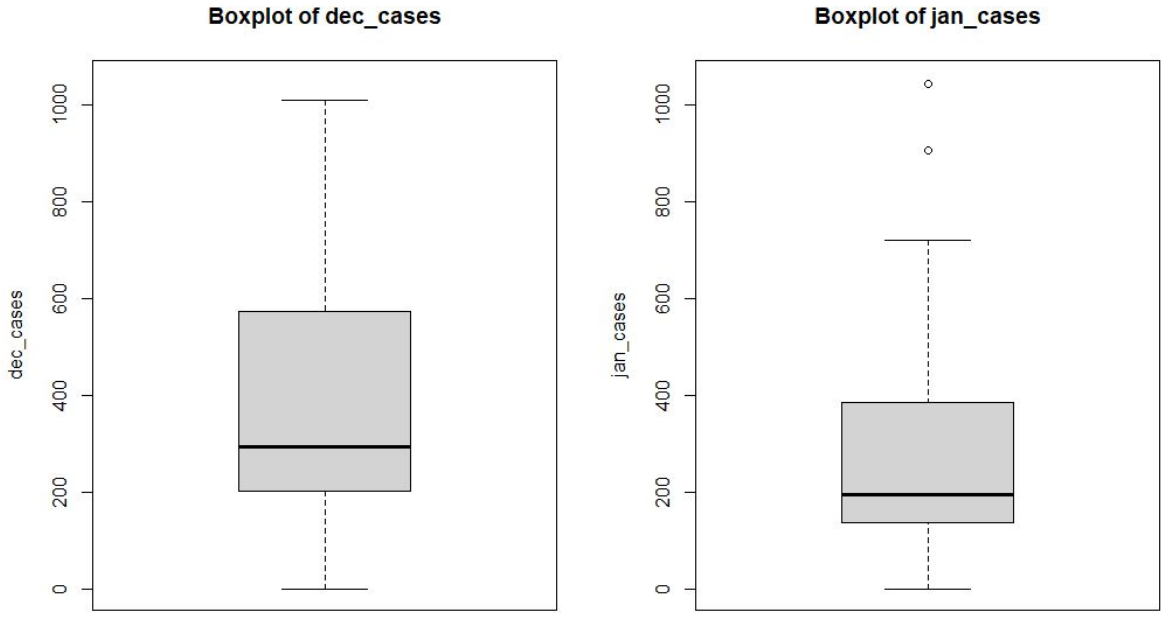
Next up, let us have a look at the histogram representation of the samples:





These plots indicate a very noticeable trend; namely, the distribution of the new cases became a lot more skewed towards zero. The graph on the left shows a peak between 200 and 400 new cases, although not very sharp one, while the graph on the right very clearly indicates a peak between 0 and 200 new cases with the height seemingly twice that of the second highest column. Lastly, note that the columns between 200 and 800 cases all decreased in height, while the two columns between 800 and 1200 cases stayed intact. This further supports our claim that most of the countries saw a fairly significant decrease in the number of new cases, except for the few countries which form the last three columns.

We may also look at the boxplots of the two samples in order to get an even better idea of the distribution of our data, as well as how it changed in a month:



As can be seen from the boxplots, once again, our claim is solidified. All the points in the five-number summary saw a decrease (except for the ***Min.***), while the dataset overall became more concentrated. We can see this graphically as the box, along with its whiskers shrank notably.

Bivariate analysis - numerical

Lastly, let us have a look at the numerical bivariate analysis between the two samples. Only three new values have been calculated, as we already have the bivariate mean (since we have both of the individual means). The three quantities are the *correlation*, *Spearman rank correlation*, and *Kendall rank correlation*. Their values are the following:

(regular) correlation	Spearman rank correlation	Kendall rank correlation
0.298	0.443	0.316

As can be seen from the table, the two samples are barely correlated. They are although none of the coefficients are significantly close to 0, they are even further away from 1 (or even -1, for that matter) which indicates that the two samples are fairly independent of each other.

Appendix

Exercise 1.3 R code

```
exp <- function(n, rate) {
x = rexp(n, rate)
title = paste("Histogram vs. Density\n", "n =", n, ", rate =",
rate)
hist(x, freq = FALSE, col="white", border="black",main=title)
points = seq(0, max(x), length=length(x))
lines(points, dexp(points,rate), col="red")
}

par(mfrow=c(2,3))

exp(10, 3)
exp(100, 3)
exp(1000, 3)
exp(1000, 3)
exp(1000, 10)
exp(1000, 100)
```

Exercise 1.4 R code

```
df = read.table("owid-covid-data.csv", sep=";",
                header = TRUE, stringsAsFactors = FALSE)

df$date = as.Date(df$date, "%d-%m-%y")

dec = subset(df, df$date == as.Date("20-12-2020", "%d-%m-%y") &
             df$continent == "Europe",
             select = c(iso_code, new_cases_smoothed_per_million))
jan = subset(df, df$date == as.Date("20-01-2021", "%d-%m-%y") &
             df$continent == "Europe",
             select = c(iso_code, new_cases_smoothed_per_million))

dec$new_cases_smoothed_per_million =
  as.numeric(gsub(",", ".", dec$new_cases_smoothed_per_million))
jan$new_cases_smoothed_per_million =
  as.numeric(gsub(",", ".", jan$new_cases_smoothed_per_million))
```

```
dec_cases = dec$new_cases_smoothed_per_million
jan_cases = jan$new_cases_smoothed_per_million

summary(dec_cases)
sd(dec_cases)

summary(jan_cases)
sd(jan_cases)

par(mfrow=c(1,2))

plot(sort(dec_cases), main="Plot of sorted dec_cases vs.
countries",
     xlab="Sorted index of country", ylab="dec_cases")
plot(sort(jan_cases), main="Plot of sorted jan_cases vs.
countries",
     xlab="Sorted index of country", ylab="jan_cases")

hist(dec_cases, ylim=c(0,25))
hist(jan_cases, ylim=c(0,25))

boxplot(dec_cases, main="Boxplot of dec_cases", ylab="dec_cases",
ylim=c(0,1050))
boxplot(jan_cases, main="Boxplot of jan_cases", ylab="jan_cases",
ylim=c(0,1050))

par(mfrow=c(1,1))

plot(dec_cases, jan_cases)

cor(dec_cases, jan_cases)
cor(dec_cases, jan_cases, method="spearman")
cor(dec_cases, jan_cases, method="kendall")
```