Impact of COVID-19 vaccination on maternal deaths: a time series approach

2022

Contents

Abstract

At the end of 2019, the outbreak of COVID-19 began. Later, it was declared by the WHO as a pandemic. In Brazil, vaccination against COVID-19 started in January 2021, only for people belonging to the groups classified as priority, namely: health professionals, institutionalized people aged 60 or over or with disabilities and the village indigenous populations. During this period, there was no relevant concern regarding the maternal population in the country on the part of the bodies responsible for vaccine administration policies. This population was classified as a priority group only at the beginning of May 2021, almost 4 months after vaccination in the country was started.

It was possible to notice that, from the month of May onwards, the number of deaths of pregnant and postpartum women began to show a consistent downward trend, while the number of pregnant women and postpartum women vaccinated with the first dose of the vaccine against COVID-19 began to increase. Until that month, approximately 33 million Brazilians had been vaccinated, a fact that could also be influencing the downward trend in deaths of pregnant and postpartum women.

The objective of this study is to verify which factor had the greatest impact on the fall in maternal population deaths: vaccination focused specifically on this population or the fact that the general population was already being vaccinated, which would confer a protective effect. The analysis consists of a time series approach, in which daily information on deaths and vaccination of the general population and pregnant and postpartum women in 2021 are considered. Impulse-response functions were used in order to verify the impact of vaccination series on maternal deaths series. The analysis of these functions showed an immediate impact of the first dose of the vaccine in pregnant women on the deaths of pregnant and postpartum women.

Data

For the analysis in this document, data of pregnant and postpartum women reported in the Influenza Epidemiological Surveillance Information System were used, which is a national surveillance database used to monitor respiratory infections in Brazil, known as SIVEP-Gripe. In addition, vaccination data against COVID-19 in Brazil were also used. The datasets were obtained on February 1, 2022, they can be found at https://opendatasus.saude.gov.br/dataset.

The data were analyzed using the free-software R https://www.R-project.org in version 4.1.3. Next, we present and load the libraries used in the data analysis process.

Packages

```r
loadlibrary <- function(x) {
  if (!require(x, character.only = TRUE)) {
    install.packages(x, dependencies = T)
    if (!require(x, character.only = TRUE))
      stop("Package not found")
  }
}

packages <-
  c(
    "tseries",
    "itsmr",
    "tsibble",
    "lubridate",
    "data.table",
    "tidyr",
    "dplyr",
    "readr",
    "forecast",
    "TSA",
    "vars",
    "MTS",
    "hrbrthemes",
    "smooth",
    "RcppRoll",
    "ggplot2",
    "patchwork"
  )
lapply(packages, loadlibrary)
```

Our objective is to analyze data for the period of 2021, which is when vaccination started in Brazil. We will filter the data in order to obtain the vaccination series of the general Brazilian population and the maternal population, as well as the series of deaths of the maternal population. All these time series were considered between 01/03/2021 and 01/01/2022, these being the beginning and the end of the first epidemiological week of 2021.

Loading the data

```
data <- read_csv('dados_SIVEP_Gripe_Brasil_01-02-22.csv')

## Sample period: 01/03/2021 - 01/01/2022
icu_deaths <- data[,c('dt_sint','n_icu','n_deaths','n_cases')]
icu_deaths$dt_sint <- as.Date(icu_deaths$dt_sint, "%m/%d/%Y")
n <- nrow(icu_deaths)

########### Vaccines (total) 2021-2022 ################
vaccines <- read_csv('Vacinados_Brasil_01-02-22.csv')

### Sample period: 01/01/2021 - 01/14/2022
vaccines_2022 <- vaccines[427:1661,]
vaccines_2022$vaccine_application_date <-
  as.Date(vaccines_2022$vaccine_application_date, "%m/%d/%Y")
vaccines_2022 <- vaccines_2022 %>%
  transmute(date = vaccine_application_date,
            dose_vaccine,
            cases = n_cases)

## Filtering: vaccinated with the first dose of the vaccine
first_dose <- vaccines_2022 %>%
  filter(dose_vaccine == '1a dose') %>%
  group_by(date) %>%
  summarise(one_dose = sum(cases))

## Filtering: vaccinated with at least the second dose of the vaccine
second_dose <- vaccines_2022 %>%
  filter(dose_vaccine != '1a dose') %>%
  group_by(date) %>%
  summarise(immunized = sum(cases))

#######################################################

temp <-  icu_deaths %>%
  left_join(first_dose, by = c("dt_sint" = "date")) %>%
  left_join(second_dose, by = c("dt_sint" = "date")) %>%
  transmute(date = dt_sint,
            icu = n_icu,
            deaths = n_deaths,
            cases = n_cases,
            first_dose.BR = one_dose,
            immunized.BR = immunized)
```

```
############### Vaccines pregnant women 2022 ################

vaccines_pregnant_women <-
  read_csv('Vacinados_GestaPuerp_Brasil_01-02-22.csv')

vaccines_pregnant_women_2022 <- vaccines_pregnant_women[5:790,]
vaccines_pregnant_women_2022$vaccine_application_date <-
  as.Date(vaccines_pregnant_women_2022$vaccine_application_date,
          "%m/%d/%Y")

vaccines_pregnant_women_2022 <- vaccines_pregnant_women_2022 %>%
  transmute(date = vaccine_application_date,
            dose_vaccine,
            cases = n_cases)

first_dose_pregnant_women <- vaccines_pregnant_women_2022 %>%
  filter(dose_vaccine == '1a dose') %>%
  group_by(date) %>%
  summarise(one_dose_pregnant_women = sum(cases))

second_dose_pregnant_women <- vaccines_pregnant_women_2022 %>%
  filter(dose_vaccine != '1a dose') %>%
  group_by(date) %>%
  summarise(immunized_pregnant_women = sum(cases))

temp_pregnant_women <-  icu_deaths %>%
  left_join(first_dose_pregnant_women,
            by = c("dt_sint" = "date")) %>%
  left_join(second_dose_pregnant_women,
            by = c("dt_sint" = "date")) %>%
  transmute(date = dt_sint,
            n_icu,
            n_deaths,
            n_cases,
            first_dose.GES = one_dose_pregnant_women,
            immunized.GES = immunized_pregnant_women) %>%
  mutate(first_dose.GES = if_else(is.na(first_dose.GES),
                                  0,
                                  first_dose.GES)) %>%
  mutate(immunized.GES = if_else(is.na(immunized.GES),
                                 0,
                                 immunized.GES))

# Selecting variables:

data_S <- temp
data_S$first_dose.GES <- temp_pregnant_women$first_dose.GES
data_S$immunized.GES <- temp_pregnant_women$immunized.GES

# All series in the 2021 period
series <- data_S[ ,c('date',
                     'deaths',
                     'first_dose.BR',
                     'immunized.BR',
                     'first_dose.GES',
                     'immunized.GES')]
```

Time series concepts while analyzing the data

In order to pass on to the reader the basic concepts for understanding the analyses, I will present and explain in a simplified way each point that I find important for the understanding of this document as a whole, while performing the univariate analyzes of each time series.

To begin our study, we must understand what a time series is. In a simplified way, a time series is nothing more than a set of data generated by a random process and observed over time. As an example of a time series, we can think of the number of pregnant women vaccinated daily throughout the year of 2021.

To carry out time series analysis, we need to respect some concepts on which its theory is based. The first thing we need to check, is whether the series in question is stationary. In practical terms, we say that a series is stationary if it has a constant mean and variance over time.

I will show the step by step until obtaining a stationary series using the data of this study, and in the course of the text some important new concepts will also be addressed.

Maternal population vaccinated with the first dose of the COVID-19 vaccine

In order to verify if the series is stationary, let's take a look at some basic graphs.

```
plot.ts(ts(series$first_dose.GES),
        xlab = "Time (Days)",
        ylab = "Number of cases (first dose)")
```

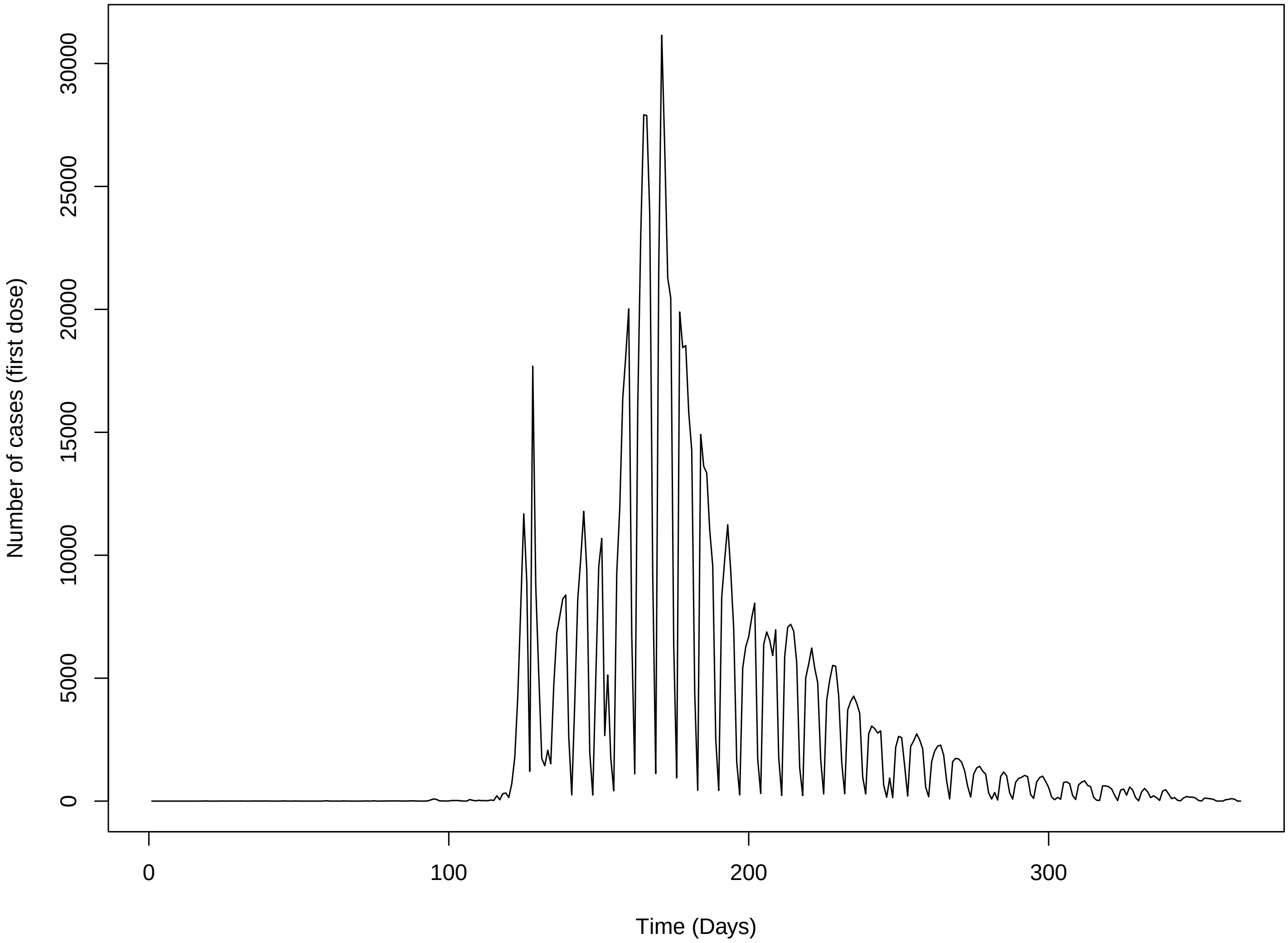


Figure 1: daily time series of pregnant women vaccinated with the first dose of the vaccine in 2021.

With the time series graph, we can get a sense of its dynamics over time. In Figure 1 for example, we can see that the number of pregnant women being vaccinated with the first dose of the vaccine, only started to increase around day 117 of the year 2021, before that, the number of cases were close to zero. As a whole, this time series clearly does not have a constant mean and variance over time.

Before we talk about autocorrelation, we should have at least a brief understanding of the concept of lag. As stated earlier, a time series is a set of observations over time that were generated by some random process. A random or stochastic process, is a sequence of random variables over time, so a time series is an observation of a random process. Consider X_t a sequence of random variables where t represents the time in days, as the series in question is the number of cases of daily vaccination of pregnant women, the number of pregnant women vaccinated on day 1 is an observation of the variable X_1, the number of pregnant women vaccinated on day 2 is an observation of the variable X_2, and so on. Given this, the lag is nothing more than the size of the "jump" we make between these variables of the random process that generated the time series. For example, when we do an autocorrelation with a lag of 1, it means that we are looking at the autocorrelation of X_1 with X_2, X_2 with X_3, and so on. If the lag is equal 2, then the autocorrelation is being calculated between X_1 with X_3, X_2 with X_4, and so on. Thus, the autocorrelation is being calculated between the series and itself lagged by h units of time, where in our case, h is the number of days.

The autocorrelation graph is very useful to find moving average components when the time series is stationary. Moving average components are values used in the modeling process that help us to eliminate cyclical variations, consequently, as the autocorrelation graph informs us of these components, through it we can identify possible cycles in the time series.

```
acf(ts(series$first_dose.GES),
    xlab = "Lag (days)",
    ylab = "ACF",
    main = "",
    lag.max = 50)
```

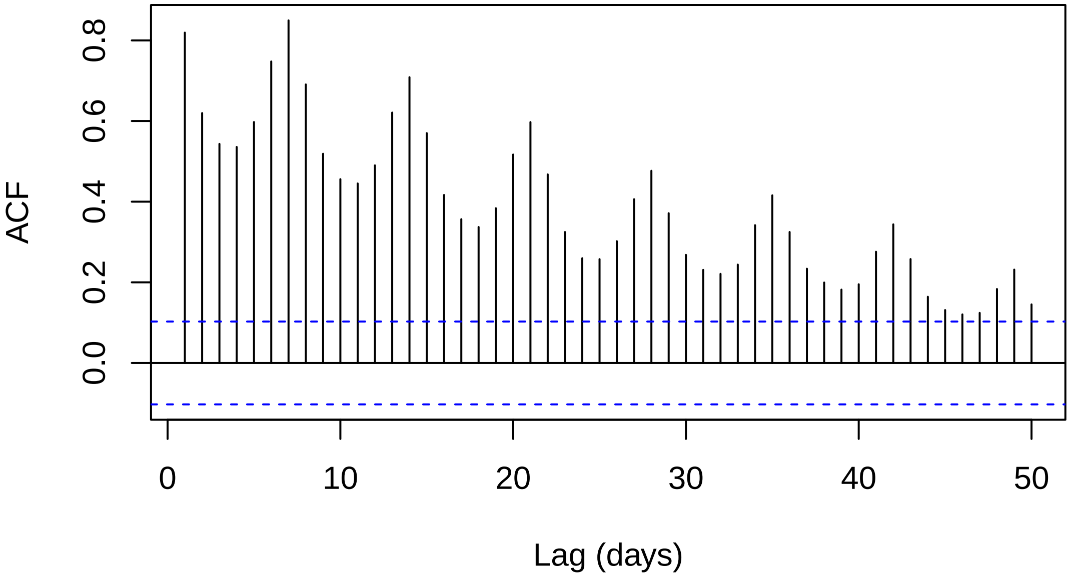


Figure 2: autocorrelation graph (ACF) for the time series of pregnant women vaccinated with the first dose of the vaccine in the year 2021. The confidence bands of 95% are presented in blue color.

In Figure 2, we can notice a natural behavior of a time series that is not stationary, that's because when a series is stationary, most of the graph's bars must be within the confidence bands. In addition, it is possible to notice a repetitive behavior every 7 bars of the graph, which suggests cycles in the series.

The partial autocorrelation graph is somehow similar to the autocorrelation graph, but instead of being useful to find moving average components, it is useful to find autoregressive components in a stationary time series, which are also used in the modeling process. Autoregressive components are related to the regression of the variable on its own past values, similar to a linear regression model. However, the covariate will be the lagged response variable itself. In the modeling process, we want to use the smallest possible autoregressive component so we can obtain the most parsimonious model. To achieve that, we can incorporate moving averages components into the model.

```
pacf(ts(series$first_dose.GES),
     xlab = "Lag (days)",
     ylab = "PACF",
     main = "",
     lag.max = 50)
```

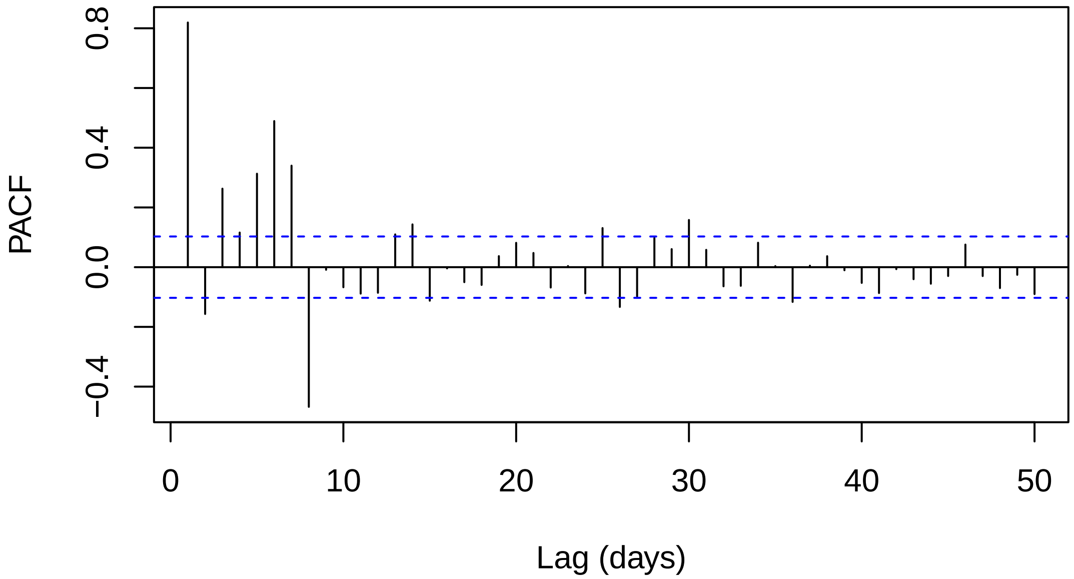


Figure 3: partial autocorrelation graph (PACF) for the time series of pregnant women vaccinated with the first dose of the vaccine in the year 2021. The confidence bands of 95% are presented in blue color.

The Figure 3 is a example of a partial autocorrelation graph using the time series of pregnant women vaccinated with the first dose of the vaccine. In practice, as for now our objective is only to achieve stationarity, the partial autocorrelation graph will not be very important seems it doens't tell us much if the series is not stationary.

```
TSA::periodogram(ts(series$first_dose.GES),
                 xlab = "Frequency",
                 ylab = "Periodogram")
```

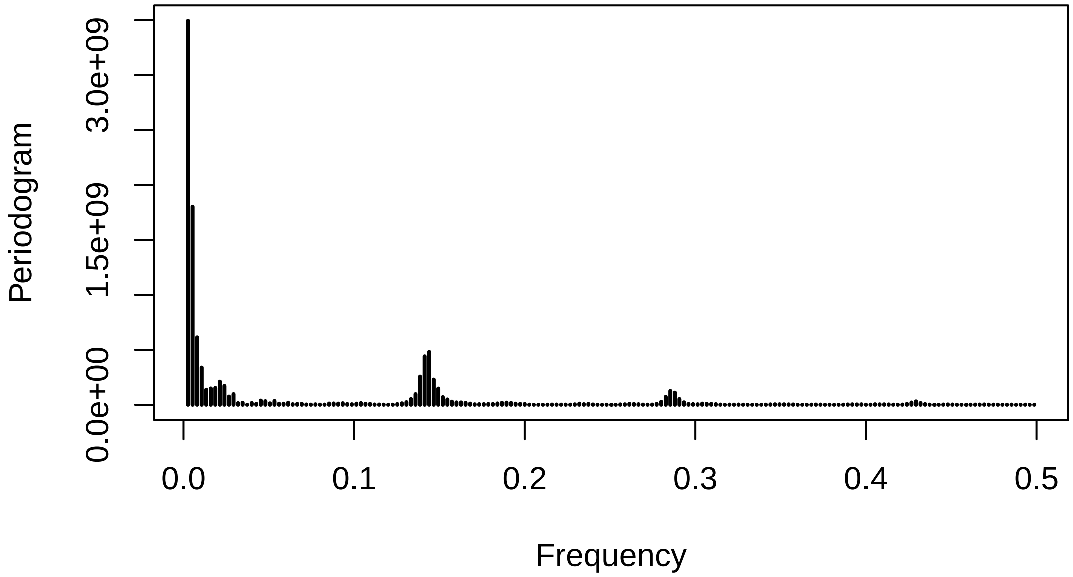


Figure 4: periodogram for the time series of pregnant women vaccinated with the first dose of the vaccine.

Another useful tool to find cycles in a time series is the periodogram, which uses frequencies to identify the possible cycles. In Figure 4, we have a periodogram for our time series where we can notice some big jumps, which suggests that the series has cycles in it, as we saw also in the autocorrelation graph.

We saw before that the time series of pregnant women vaccinated with the first dose of the vaccine has a dynamic that starts to appear in fact, around the day 117 (115 in the data) of the year, which correspond to 04/27/2021, before that, the number of cases seems to be really close to zero, since the vaccination actually started after 04/27/2021 by the time series information. Because of that, for all the time series in this study, we are going to consider the information after the moment the vaccination started.

All the graphs we have seen so far, suggest that the series is not stationary, to confirm this suspicion we need to apply a stationarity test. One of the most used tests in time series to verify stationarity in small samples is the Augmented Dickey-Fuller Test, where the null hypothesis is that the series is not stationary.

```
## Point where the times series dynamic starts
start_vac <- 115

## Point where the time series ends
end_vac <- 364

adf.test(series$first_dose.GES[start_vac:end_vac])
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  series$first_dose.GES[start_vac:end_vac]
## Dickey-Fuller = -2.7011, Lag order = 6, p-value = 0.2806
## alternative hypothesis: stationary
```

With 5% of significance, the test points out evidences that the time series is not stationary. Having confirmed that the series is not stationary, we say that we have a degree of integralization, which allows us to derive the series one time so we can try to achieve stationarity.

```r
## 1 differentiation
firstdose_dif1 <- diff(ts(series$first_dose.GES[start_vac:end_vac]))

par(mfrow=c(2,2))
plot.ts(firstdose_dif1,
        xlab = "Time",
        ylab = "Number of cases (first dose)")
TSA::periodogram(firstdose_dif1,
                 xlab = "Frequency",
                 ylab = "Periodogram")
acf(firstdose_dif1,
    xlab = "Lag (days)",
    ylab = "ACF",
    main = "",
    lag.max = 50)
pacf(firstdose_dif1,
     xlab = "Lag (days)",
     ylab = "PACF",
     main = "",
     lag.max = 50)
```

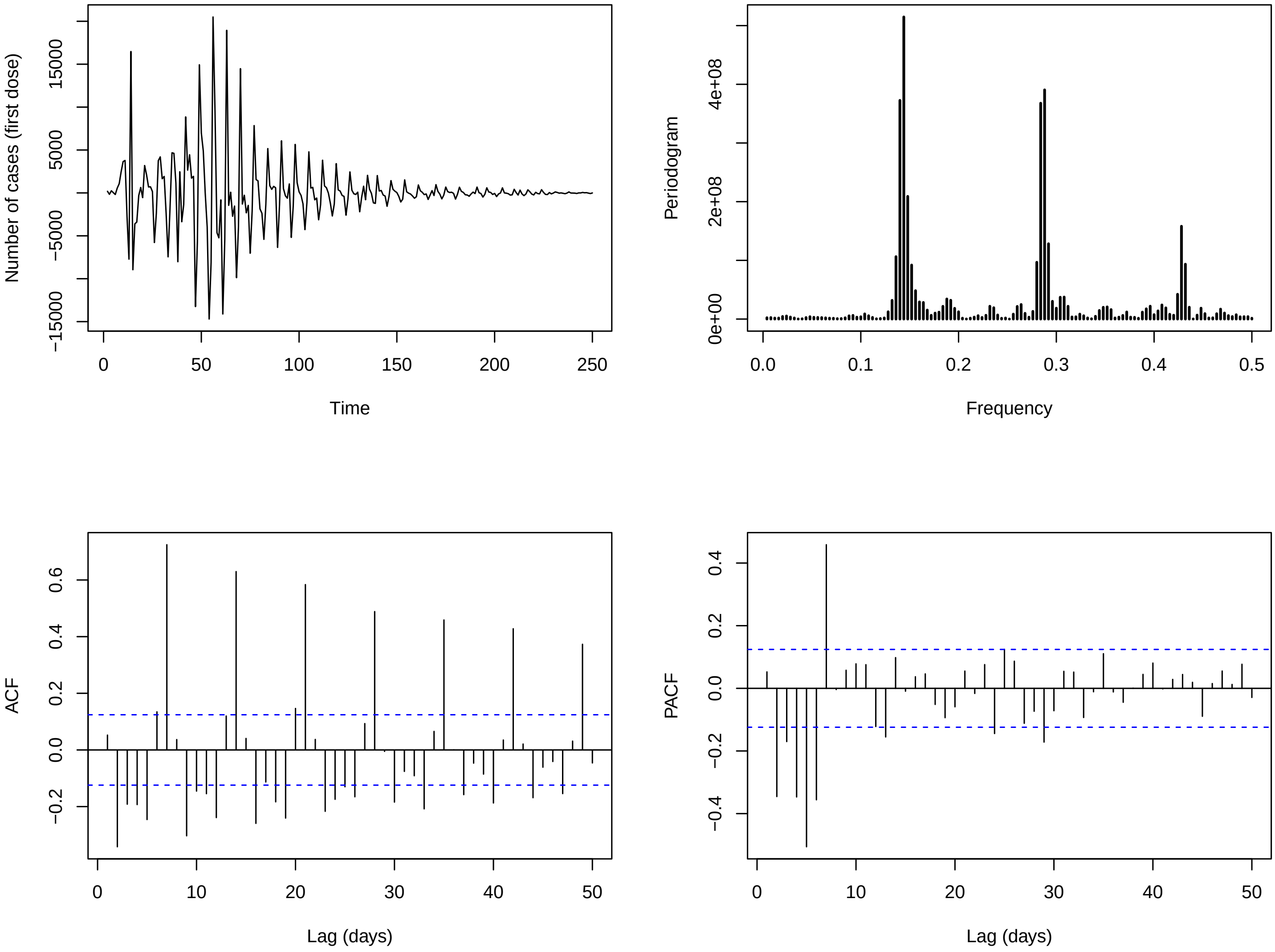


Figure 5: basic graphs in the analysis of time series applied to the series of pregnant and postpartum women vaccinated with the first dose of the COVID-19 vaccine considering one differentiation.

We can see that the jumps we saw earlier in the periodogram are now more evident. In the autocorrelation graph, we can clearly notice a repeating behavior that happens every 7 days, which means that we have a period of 7. Now, let's inform this period to our differentiation process and check the time series.

```
## Considering the 7 period information
firstdose_dif7 <- diff(firstdose_dif1, 7)

par(mfrow=c(2,2))
plot.ts(firstdose_dif7,
        xlab = "Time",
        ylab = "Number of cases (first dose)")
TSA::periodogram(firstdose_dif7,
                 xlab = "Frequency",
                 ylab = "Periodogram")
acf(firstdose_dif7,
    xlab = "Lag (days)",
    ylab = "ACF",
    main = "",
    lag.max = 50)
pacf(firstdose_dif7,
     xlab = "Lag (days)",
     ylab = "PACF",
     main = "",
     lag.max = 50)
```

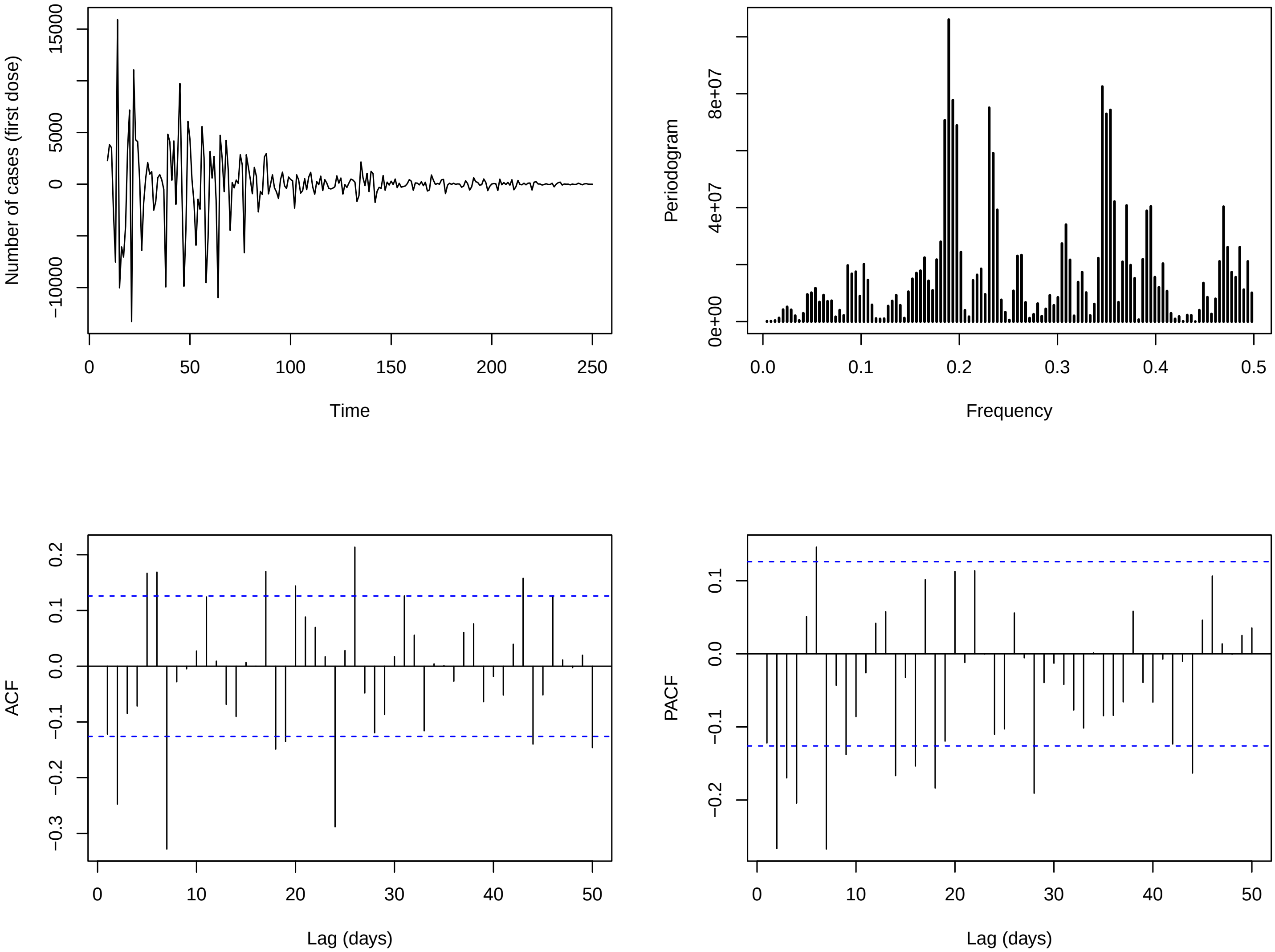


Figure 6: basic graphs in the analysis of time series applied to the series of pregnant and postpartum women vaccinated with the first dose of the COVID-19 vaccine, considering seasonal differentiation of seven.

Now we can observe a more expected behavior of a stationary series when we look at the periodogram and the autocorrelation graphs. However, the variance still seems to be a little unstable, to reduce this instability we will apply a logarithmic transformation to the data and add them to the mean, which is a procedure widely used in the area of economics to stabilize the variance of time series.

```r
## Summary of the time series
summary(ts(series$first_dose.GES[start_vac:end_vac]))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##     0.0   291.5  1352.0  4062.3  5618.8 31148.0
```

```r
## ## Adding 4000
firstdose_log <-
  log(ts(series$first_dose.GES[start_vac:end_vac]) + 4000)
firstdose_dif1 <- diff(firstdose_log)
firstdose_dif7 <- diff(firstdose_dif1, 7)

par(mfrow=c(2,2))
plot.ts(firstdose_dif7,
        xlab = "Time",
        ylab = "Number of deaths")
TSA::periodogram(firstdose_dif7,
                 xlab = "Frequency",
                 ylab = "Periodogram")
acf(firstdose_dif7,
    xlab = "Lag (days)",
    ylab = "ACF",
    main = "",
    lag.max = 10,
    xaxt = "n")
axis(1, at = 1:10)
pacf(firstdose_dif7,
     xlab = "Lag (days)",
     ylab = "PACF",
     main = "",
     lag.max = 10,
     xaxt = "n")
axis(1, at = 1:10)
```

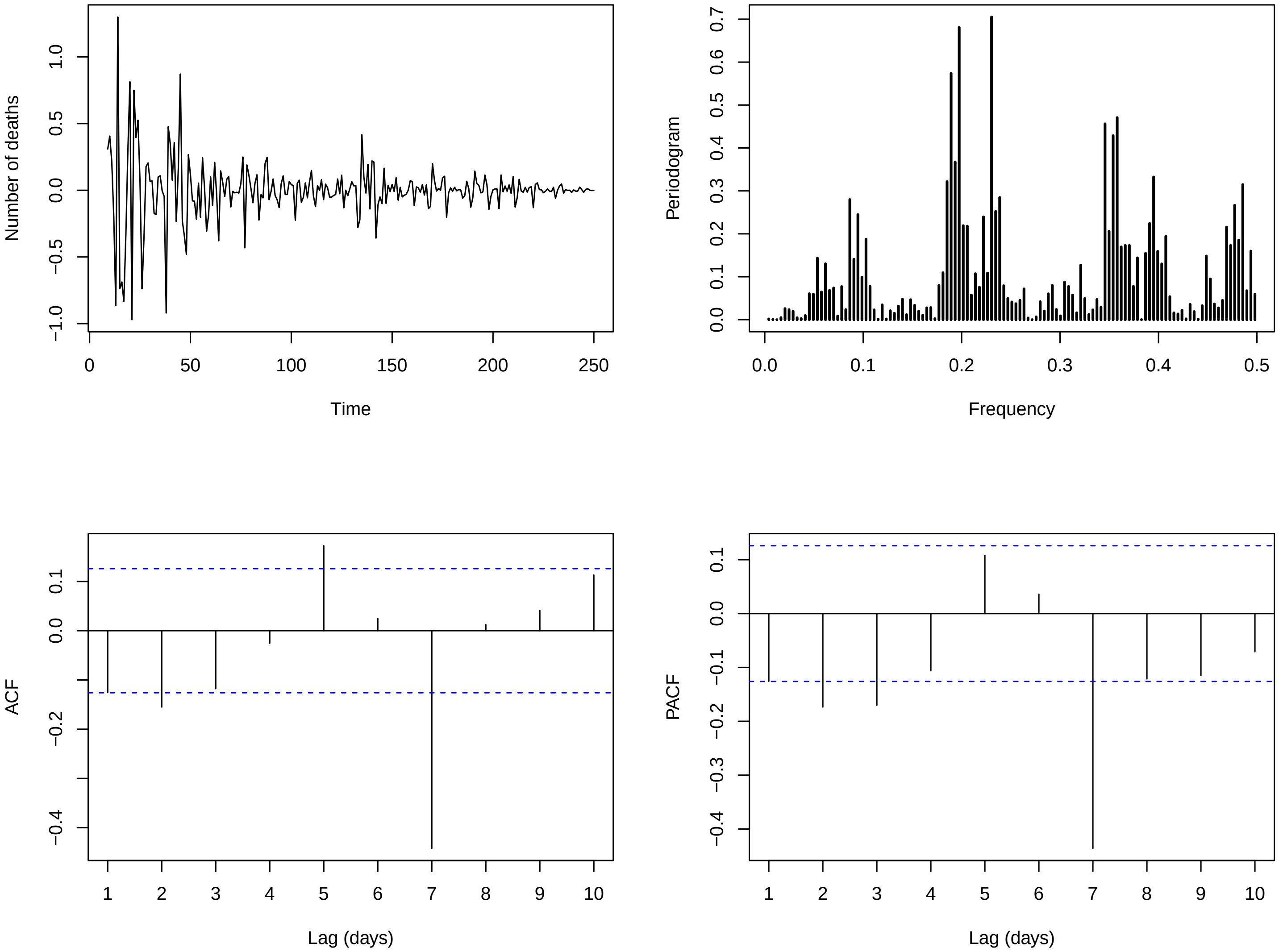


Figure 7: basic graphs in the analysis of time series applied to the series of pregnant and postpartum women vaccinated with the first dose of the COVID-19 vaccine, considering seasonal differentiation of seven and logarithmic transformation of the data with the addition of the mean.

Applying the Augmented Dickey-Fuller Test one more time.

```
adf.test(firstdose_dif7)
```

```
## 
##  Augmented Dickey-Fuller Test
## 
## data:  firstdose_dif7
## Dickey-Fuller = -10.505, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

The test points out to a stationary time series.

Daily deaths of the maternal population

As we have done so far, we are going to verify if the time series of deaths of pregnant and postpartum women is stationary. First, let's take a look at basic time series analysis graphs.

```
par(mfrow=c(2,2))
plot.ts(ts(series$deaths[start_vac:end_vac]),
        xlab = "Time",
        ylab = "Number of deaths")
TSA::periodogram(ts(series$deaths[start_vac:end_vac]),
                 xlab = "Frequency",
                 ylab = "Periodogram")
acf(ts(series$deaths[start_vac:end_vac]),
    xlab = "Lag (days)",
    ylab = "ACF",
    main = "",
    lag.max = 50)
pacf(ts(series$deaths[start_vac:end_vac]),
     xlab = "Lag (days)",
     ylab = "PACF",
     main = "",
     lag.max = 50)
```

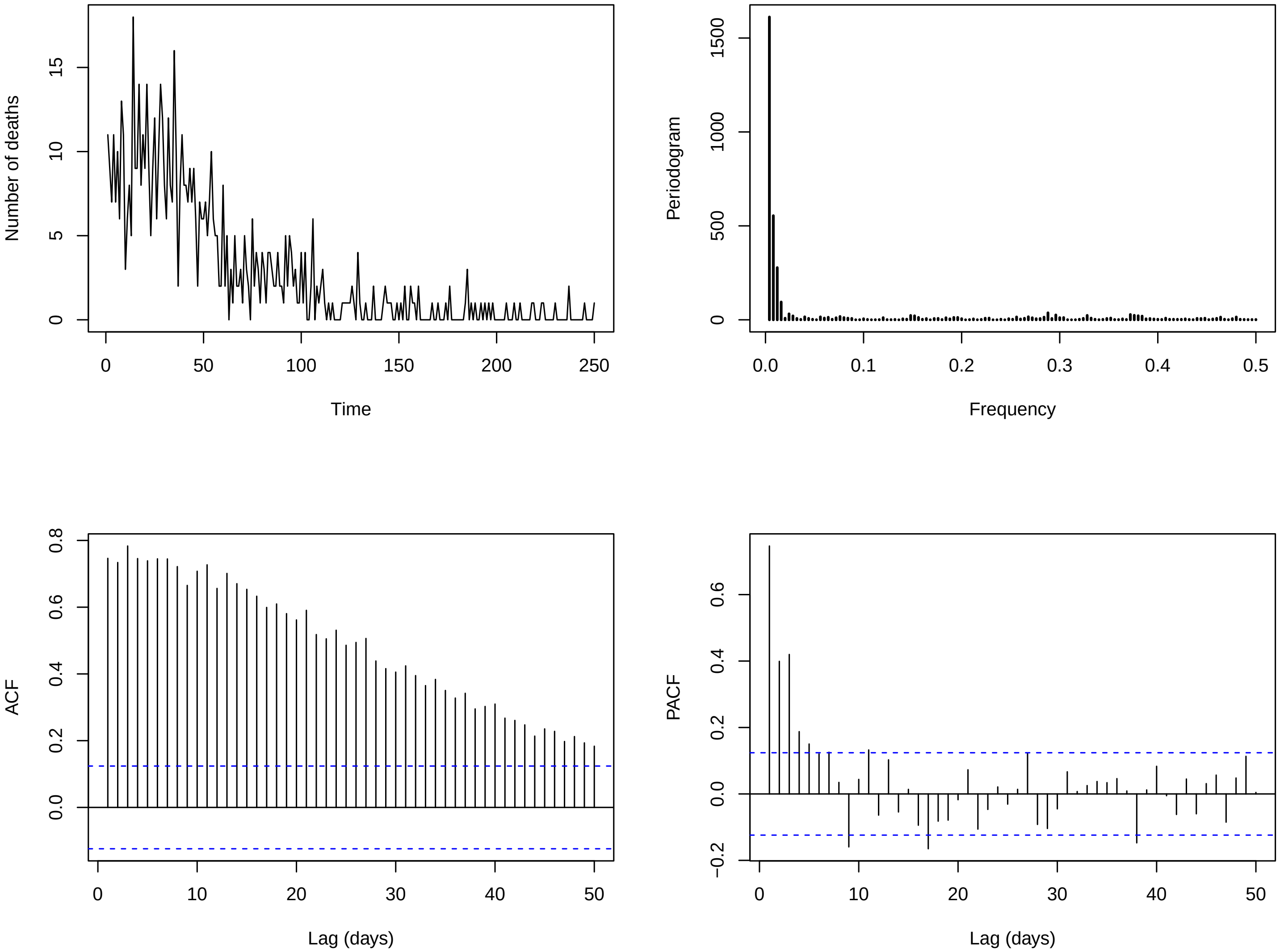


Figure 8: basic graphs in the analysis of time series applied to the series of pregnant and postpartum women deaths.

At first, the autocorrelation graphs and the periodogram do not suggest cycles in their behavior that are evident. The time series graph suggests that the mean and variance are not constant over time, which goes against stationarity. To verify this assumption, we applied a Augmented Dickey-Fuller Test.

```
adf.test(series$deaths[start_vac:end_vac])
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  series$deaths[start_vac:end_vac]
## Dickey-Fuller = -1.4739, Lag order = 6, p-value = 0.7973
## alternative hypothesis: stationary
```

The Augmented Dickey-Fuller Test showed evidence of non-stationarity. We can apply a differentiation for better results.

```
deaths_dif1 <- diff(ts(series$deaths[start_vac:end_vac]))

par(mfrow=c(2,2))
plot(deaths_dif1,
     xlab = "Time",
     ylab = "Number of deaths")
TSA::periodogram(deaths_dif1,
                 xlab = "Frequency",
                 ylab = "Periodogram")
acf(deaths_dif1,
    xlab = "Lag (days)",
    ylab = "ACF",
    main = "",
    lag.max = 50)
pacf(deaths_dif1,
     xlab = "Lag (days)",
     ylab = "PACF",
     main = "",
     lag.max = 50)
```

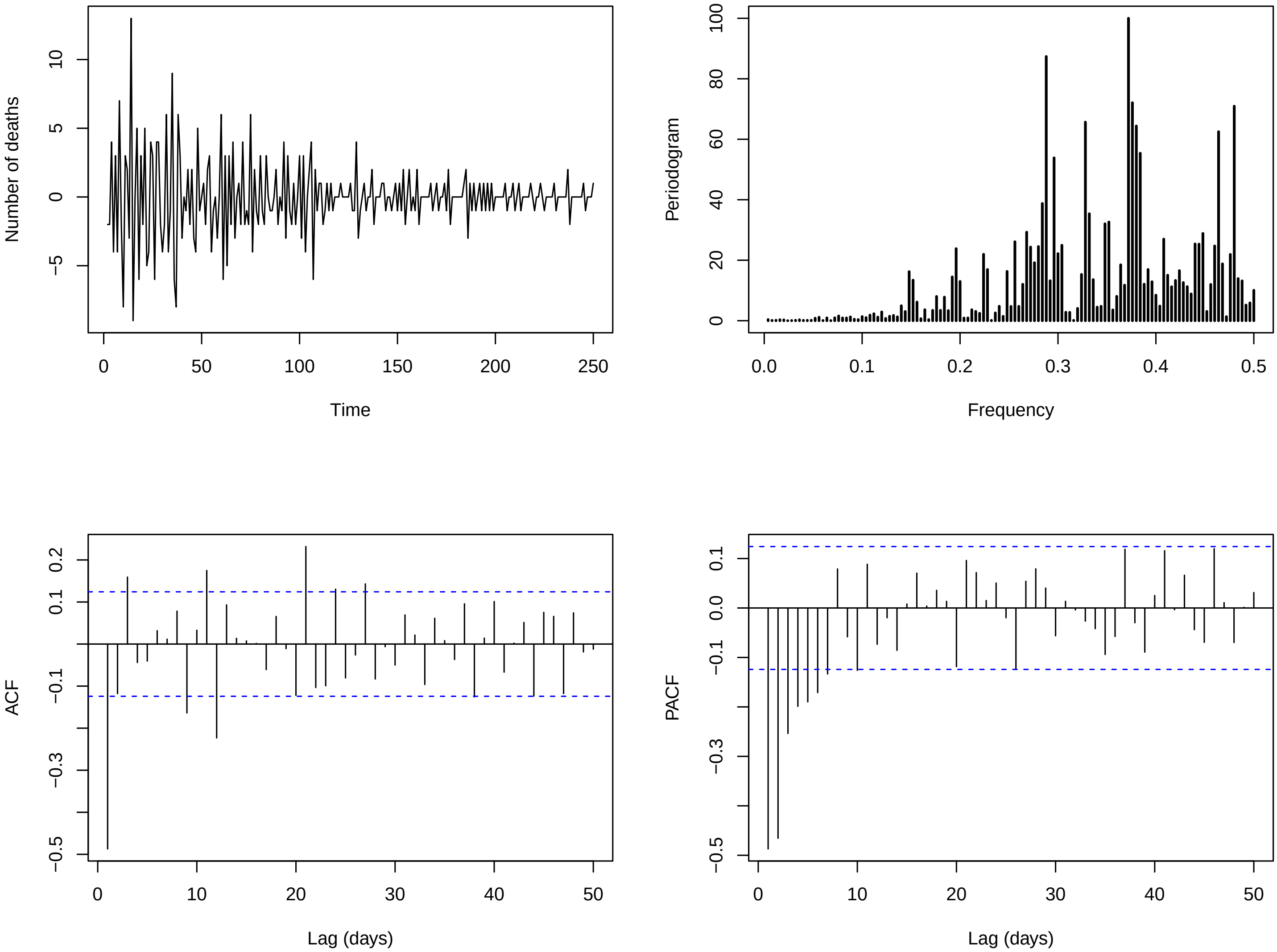


Figure 9: basic graphs in the analysis of time series applied to the series of pregnant and postpartum women deaths, considering one differentiation.

Even after a differentiation, we can observe a non-constant variance resulting from the dynamics of the time series. To deal with this non-constant variance, we can apply a logarithmic transformation to stabilize it, but since we have many observations where zero deaths occurred (95 out of 250 observations) we can change the scale by adding the mean to the data.

```
## Summary
summary(ts(series$deaths[start_vac:end_vac]))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.000   1.000   2.752   4.000  18.000
```

```
## Adding the mean and applying log transformation
deaths_log <- log(ts(series$deaths[start_vac:end_vac]) + 3)
deaths_dif1 <- diff(deaths_log)
```

```r
par(mfrow=c(2,2))
plot.ts(deaths_dif1,
        xlab = "Time",
        ylab = "Number of deaths")
TSA::periodogram(deaths_dif1,
                 xlab = "Frequency",
                 ylab = "Periodogram")
acf(deaths_dif1,
     xlab = "Lag (days)",
     ylab = "ACF",
     main = "",
     xaxt = "n",
     lag.max = 10)
axis(1, at = 1:10)
pacf(deaths_dif1,
     xlab = "Lag (days)",
     ylab = "PACF",
     main = "",
     xaxt = "n",
     lag.max = 10)
axis(1, at = 1:10)
```

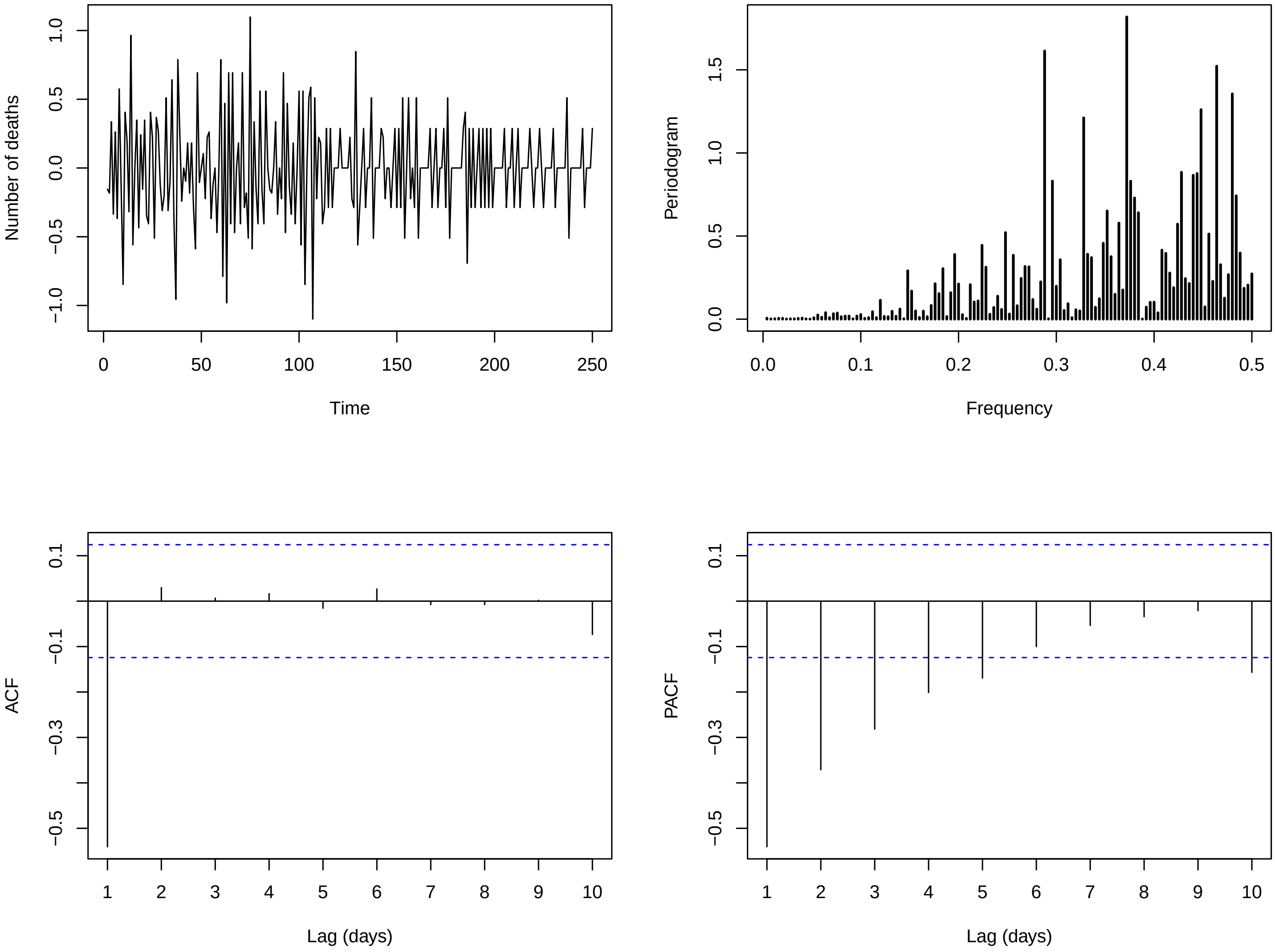


Figure 10: basic graphs in the analysis of time series applied to the series of pregnant and postpartum women deaths, considering one differentiation and logarithmic transformation of the data with the addition of the mean.

Now we have the suspicion of stationarity. To verify it, we can apply the Augmented Dickey-Fuller Test again.

```
adf.test(deaths_dif1)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  deaths_dif1
## Dickey-Fuller = -9.3195, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

The test points out to the rejection of the non-stationarity hypothesis, confirming the suspicions we had before.

Brazilian people vaccinated with the first dose of the COVID-19 vaccine

These data consider the general population of Brazil. we will follow the same steps as before until we obtain stationarity.

```r
par(mfrow=c(2,2))
plot.ts(ts(series$first_dose.BR[start_vac:end_vac]),
        xlab = "Time",
        ylab = "Number of cases (Brazilians first dose)")

TSA::periodogram(ts(series$first_dose.BR[start_vac:end_vac]),
                 xlab = "Frequency",
                 ylab = "Periodogram")
acf(ts(series$first_dose.BR[start_vac:end_vac]),
    xlab = "Lag (days)",
    ylab = "ACF",
    main = "",
    lag.max = 250)
pacf(ts(series$first_dose.BR[start_vac:end_vac]),
     xlab = "Lag (days)",
     ylab = "PACF",
     main = "",
     lag.max = 250)
```

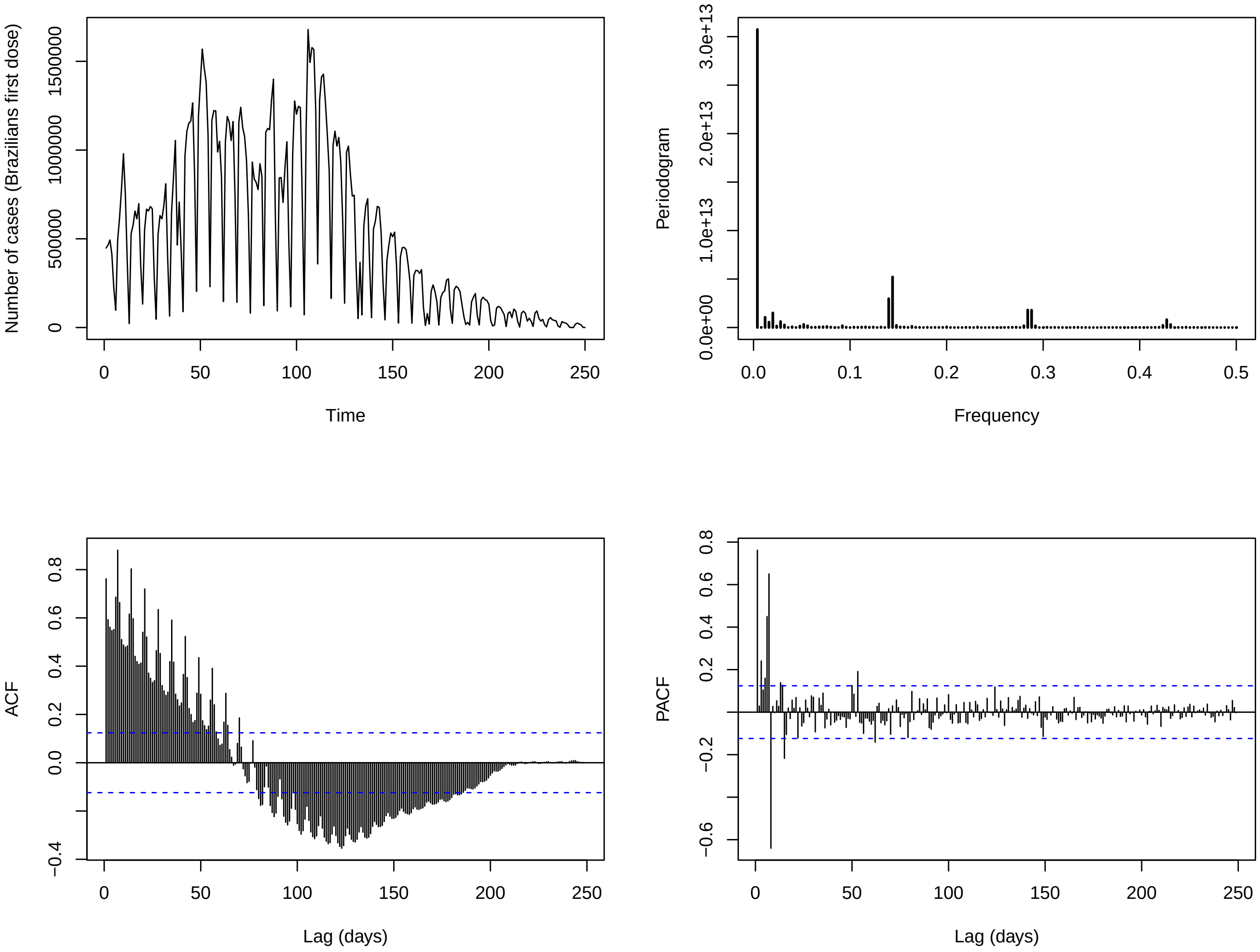


Figure 11: basic graphs in the analysis of time series applied to the series of brazilian people vaccinated with the first dose of the COVID-19 vaccine.

The time series graph suggests non-stationarity. In addition, we can see jumps in the bars of the autocorrelation graphs and the periodogram. We used the Augmented Dickey-Fuller Test to confirm non-stationarity.

```
adf.test(ts(series$first_dose.BR[start_vac:end_vac]))
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  ts(series$first_dose.BR[start_vac:end_vac])
## Dickey-Fuller = -2.3627, Lag order = 6, p-value = 0.4231
## alternative hypothesis: stationary
```

As the test shows evidence of non-stationarity, we will consider a differentiation of 1 as a first step, and check what happens to the series.

```r
## 1 differentiation
firstdoseBR_dif1 <-
  diff(ts(series$first_dose.BR[start_vac:end_vac]))

par(mfrow=c(2,2))
plot.ts(firstdoseBR_dif1,
        xlab = "Time",
        ylab = "Number of cases (Brazilians first dose)")
TSA::periodogram(firstdoseBR_dif1,
                 xlab = "Frequency",
                 ylab = "Periodogram")
acf(firstdoseBR_dif1,
    xlab = "Lag (days)",
    ylab = "ACF",
    main = "",
    lag.max = 50)
pacf(firstdoseBR_dif1,
     xlab = "Lag (days)",
     ylab = "PACF",
     main = "",
     lag.max = 50)
```

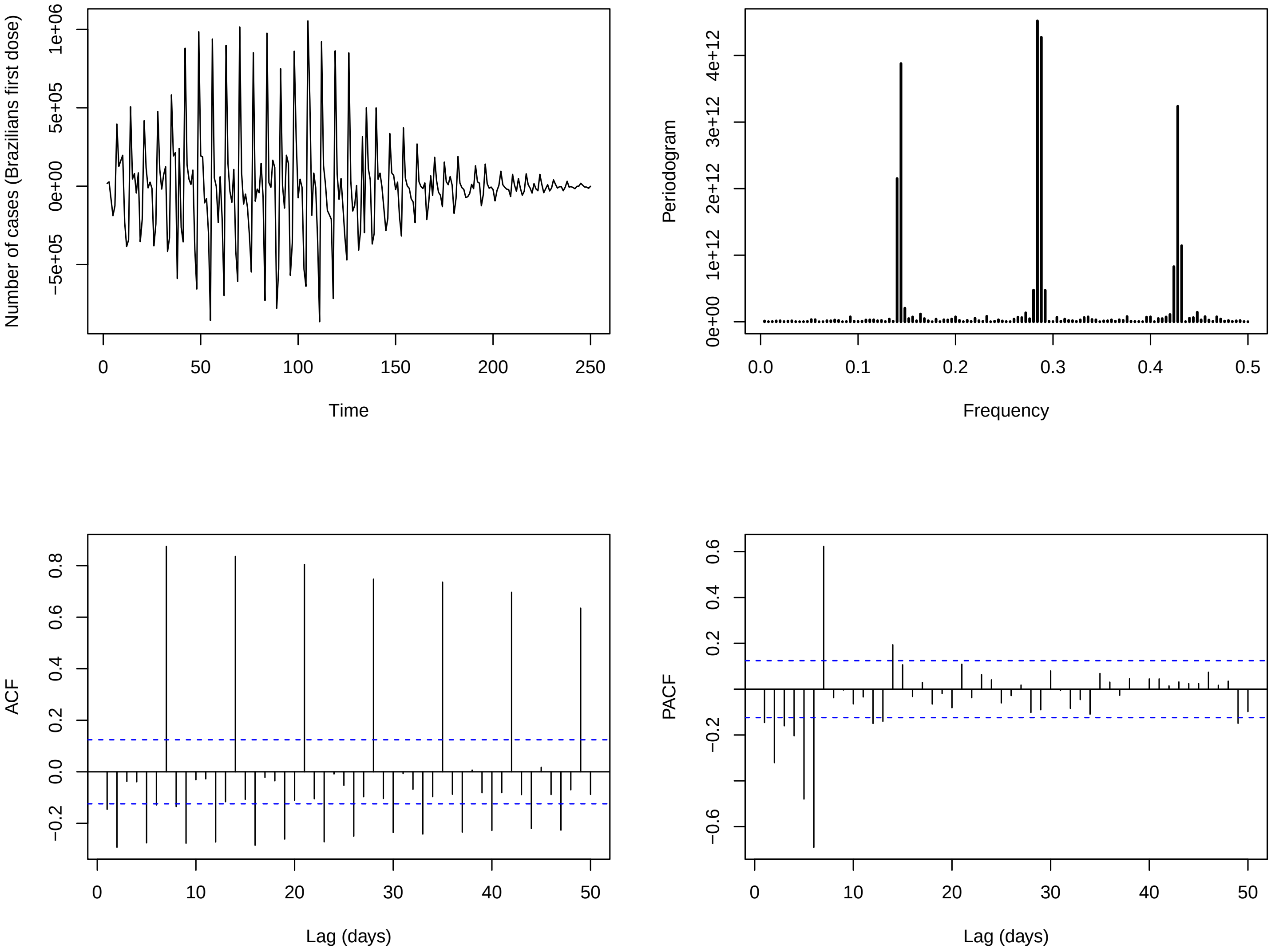


Figure 12: basic graphs in the analysis of time series applied to the series of brazilian people vaccinated with the first dose of the COVID-19 vaccine, considering one differentiation.

The jumps between graph bars are now more evident. Also, it is possible to identify a period of 7. Now, we need to include this information in the differentiation.

```
## Considering the 7 period information
firstdoseBR_dif7 <- diff(firstdoseBR_dif1, 7)

par(mfrow=c(2,2))
plot.ts(firstdoseBR_dif7,
        xlab = "Time",
        ylab = "Number of cases (Brazilians first dose)")
TSA::periodogram(firstdoseBR_dif7,
                 xlab = "Frequency",
                 ylab = "Periodogram")
acf(firstdoseBR_dif7,
    xlab = "Lag (days)",
    ylab = "ACF",
    main = "",
    lag.max = 50)
pacf(firstdoseBR_dif7,
     xlab = "Lag (days)",
     ylab = "PACF",
     main = "",
     lag.max = 50)
```

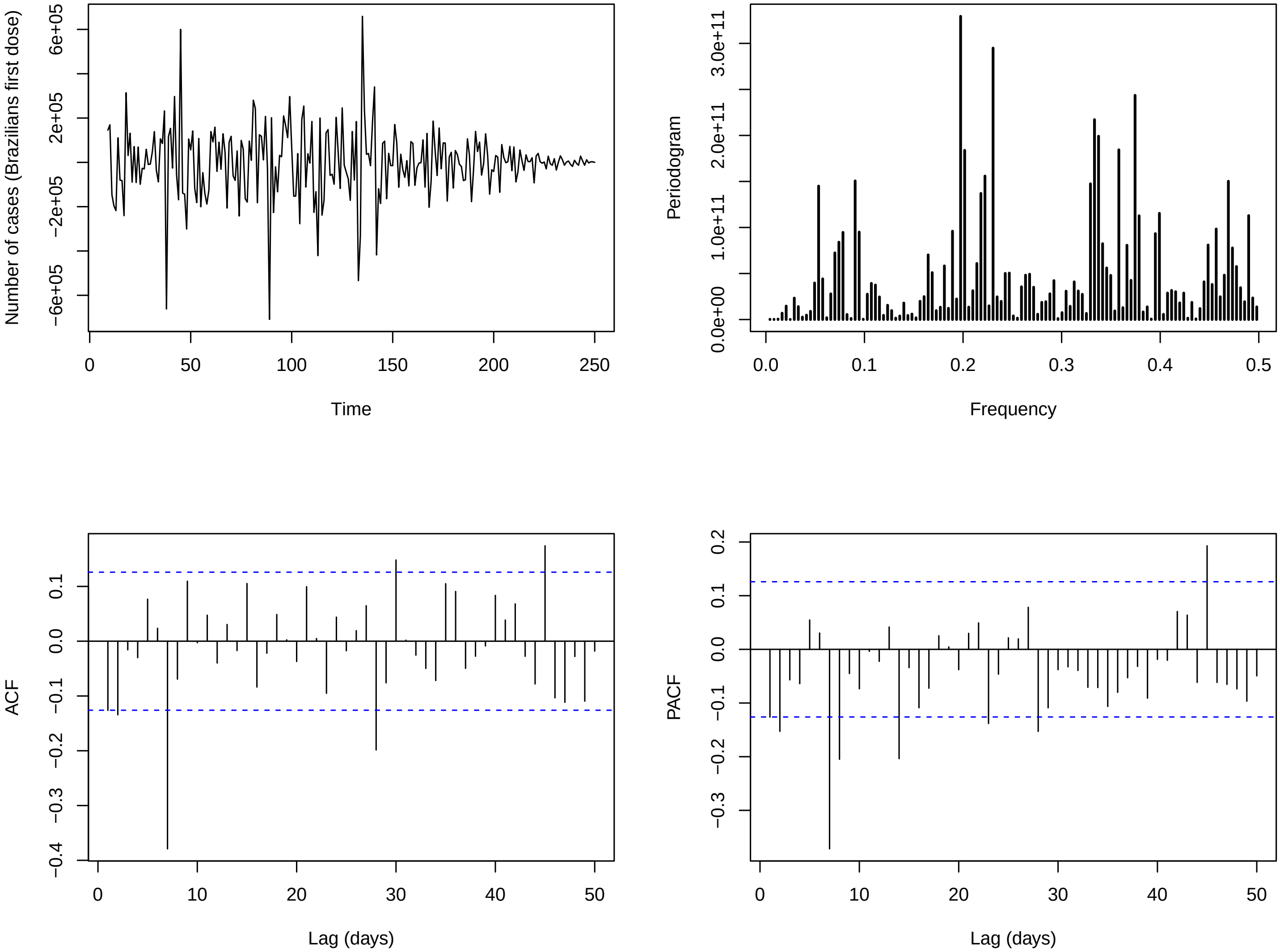


Figure 13: basic graphs in the analysis of time series applied to the series of brazilian people vaccinated with the first dose of the COVID-19 vaccine, considering seasonal differentiation of seven.

Now, we apply the logarithmic transformation to the data.

```
firstdoseBR_log <-
  log(ts(series$first_dose.BR[start_vac:end_vac]))
firstdoseBR_dif1 <- diff(firstdoseBR_log)
```

```
firstdoseBR_dif7 <- diff(firstdoseBR_dif1, 7)

par(mfrow=c(2,2))
plot.ts(firstdoseBR_dif7,
        xlab = "Time",
        ylab = "Number of cases (Brazilians first dose)")
TSA::periodogram(firstdoseBR_dif7,
                 xlab = "Frequency",
                 ylab = "Periodogram")
acf(firstdoseBR_dif7,
    xlab = "Lag (days)",
    ylab = "ACF",
    main = "",
    lag.max = 10,
    xaxt = "n")
axis(1, at = 1:10)
pacf(firstdoseBR_dif7,
     xlab = "Lag (days)",
     ylab = "PACF",
     main = "",
     lag.max = 10,
     xaxt = "n")
axis(1, at = 1:10)
```

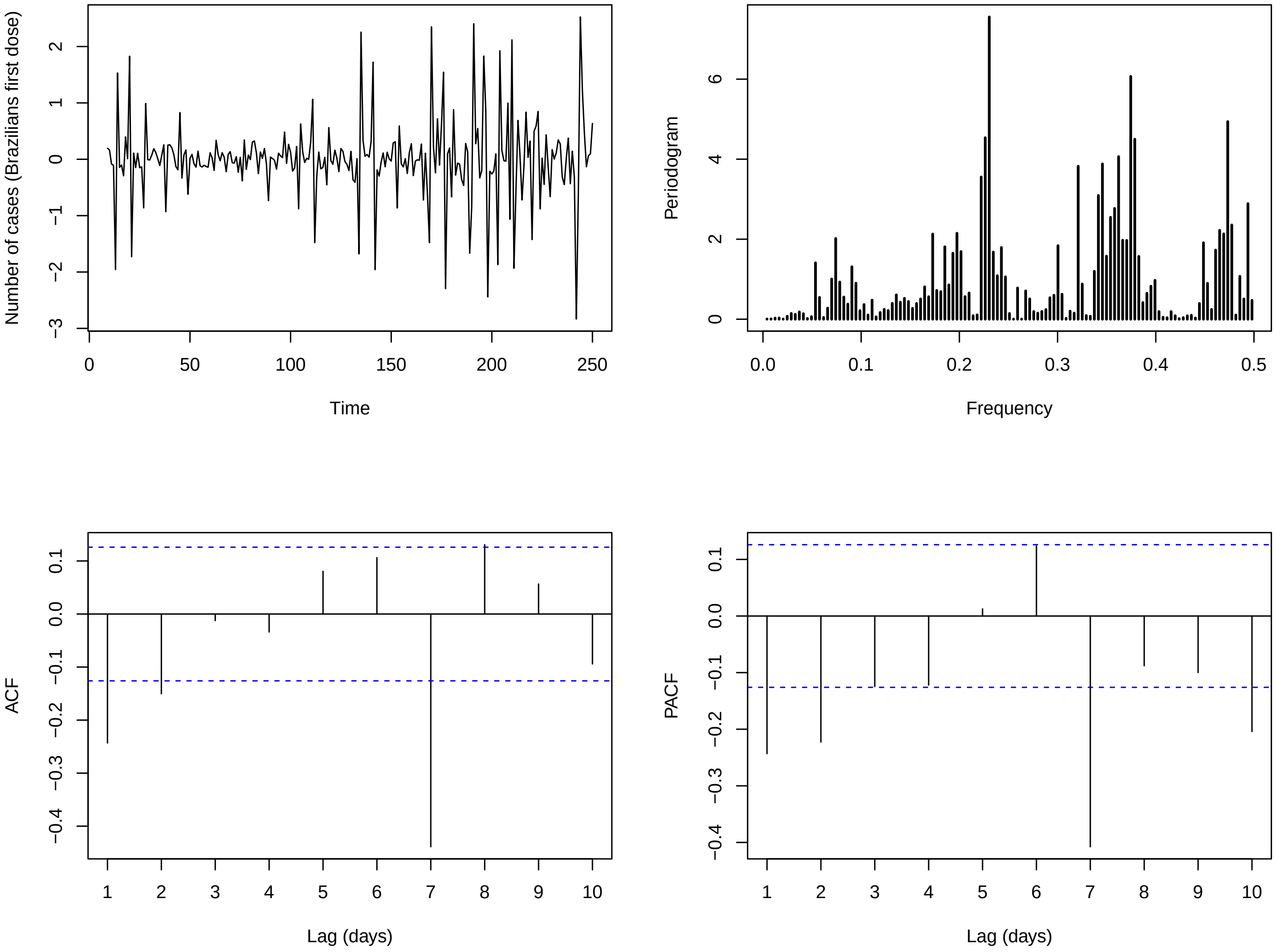


Figure 14: basic graphs in the analysis of time series applied to the series of brazilian people vaccinated with the first dose of the COVID-19 vaccine, considering seasonal differentiation of seven and logarithmic transformation of the data.

Applying the Augmented Dickey-Fuller Test one more time.

```
adf.test(firstdoseBR_dif7)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  firstdoseBR_dif7
## Dickey-Fuller = -9.8798, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

Now we have that the time series is stationary.

Immunized maternal population

These data consider pregnant and postpartum women who took the second dose of the vaccine against COVID-19 or a booster dose.

```r
par(mfrow=c(2,2))
plot.ts(ts(series$immunized.GES[start_vac:end_vac]),
        xlab = "Time",
        ylab = "Number o cases (immunized)")
TSA::periodogram(ts(series$immunized.GES[start_vac:end_vac]),
                 xlab = "Frequency",
                 ylab = "Periodogram")
acf(ts(series$immunized.GES[start_vac:end_vac]),
    xlab = "Lag (days)",
    ylab = "ACF",
    main = "",
    lag.max = 250)
pacf(ts(series$immunized.GES[start_vac:end_vac]),
     xlab = "Lag (days)",
     ylab = "PACF",
     main = "",
     lag.max = 250)
```

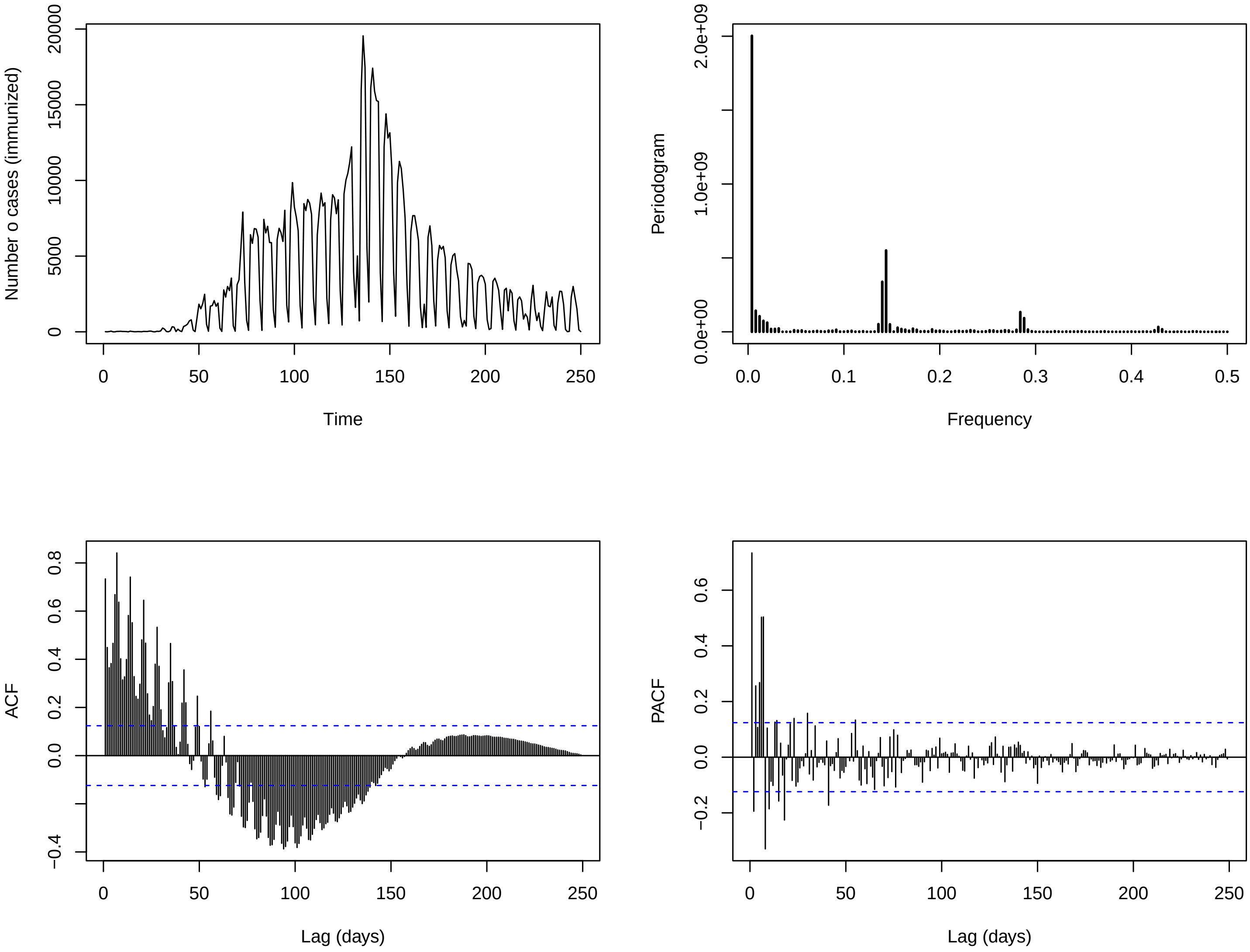


Figure 15: basic graphs in the analysis of time series applied to the series of immunized maternal population.

We clearly see that the time series of immunized pregnant women is not stationary as the mean and variance seems to be non-constant. Also, we can notice jumps through the bars when we look at the autocorrelation and periodogram graphs, which suggests that maybe there is a cicle to be considered.

Checking the non-stationarity assumption of the time series.

```
adf.test(ts(series$immunized.GES[start_vac:end_vac]))
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  ts(series$immunized.GES[start_vac:end_vac])
## Dickey-Fuller = -0.81152, Lag order = 6, p-value = 0.9597
## alternative hypothesis: stationary
```

Let’s apply 1 differentiation and verify the time series.

```
## 1 differentiation
immunized_dif1 <- diff(ts(series$immunized.GES[start_vac:end_vac]))

par(mfrow=c(2,2))
plot.ts(immunized_dif1,
        xlab = "Time",
        ylab = "Number of cases (immunized)")
TSA::periodogram(immunized_dif1,
                 xlab = "Frequency",
                 ylab = "Periodogram")
acf(immunized_dif1,
    xlab = "Lag (days)",
    ylab = "ACF",
    main = "",
    lag.max = 50)
pacf(immunized_dif1,
     xlab = "Lag (days)",
     ylab = "PACF",
     main = "",
     lag.max = 50)
```

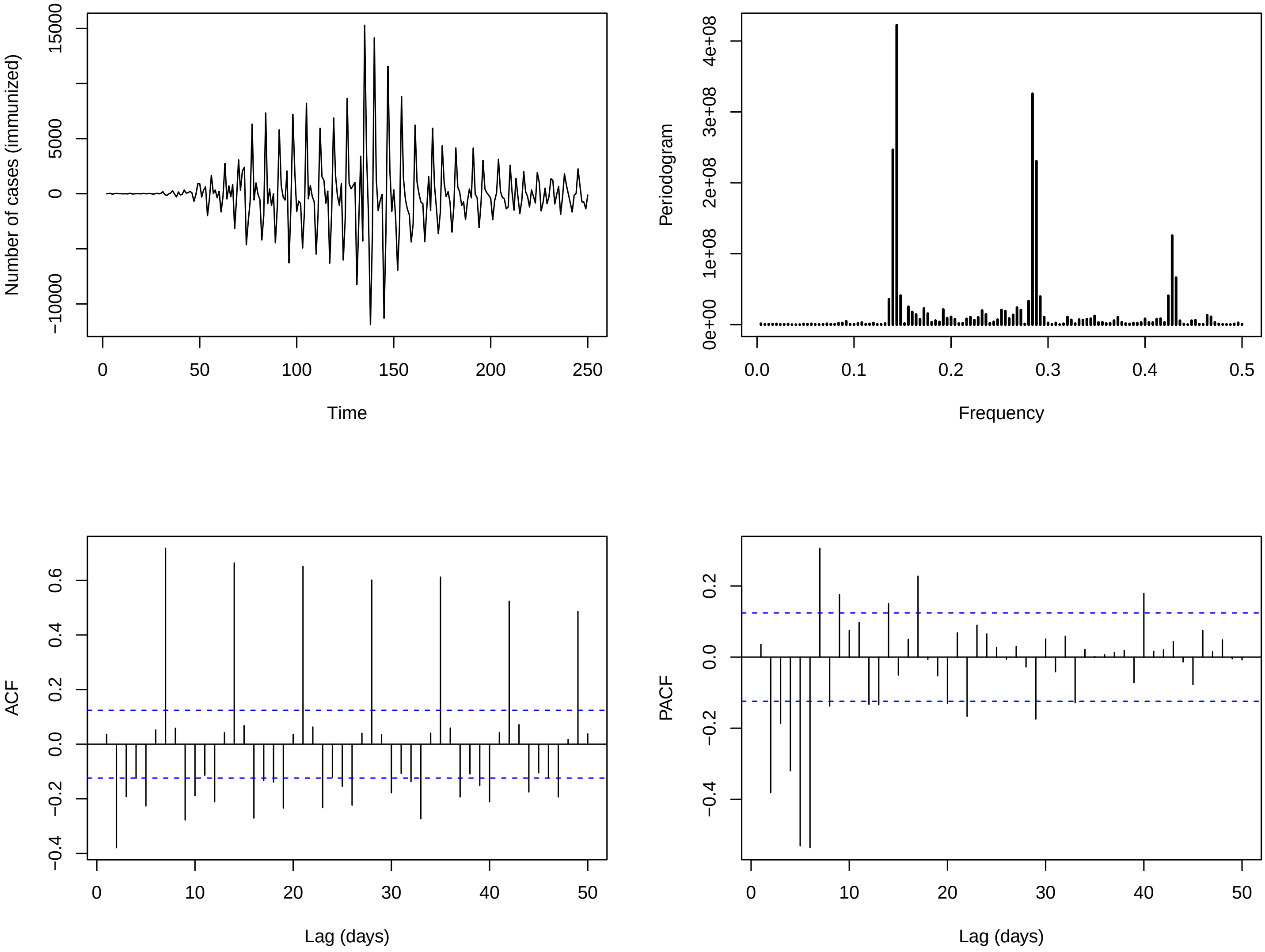


Figure 16: basic graphs in the analysis of time series applied to the series of immunized maternal population, considering one differentiation.

We can see that the jumps between the bars are now more evident, and we can see a period of 7 days when we look at the autocorrelation graph. Now, we need to inform this cycle in the differentiation process.

```
immunized_dif7 <-
  diff(ts(immunized_dif1), 7)

par(mfrow=c(2,2))
plot.ts(immunized_dif7,
        xlab = "Time",
        ylab = "Number of cases (immunized)")
TSA::periodogram(immunized_dif7,
                 xlab = "Frequency",
                 ylab = "Periodogram")
acf(immunized_dif7,
    xlab = "Lag (days)",
    ylab = "ACF",
    main = "",
    lag.max = 50)
pacf(immunized_dif7,
     xlab = "Lag (days)",
     ylab = "PACF",
     main = "",
     lag.max = 50)
```

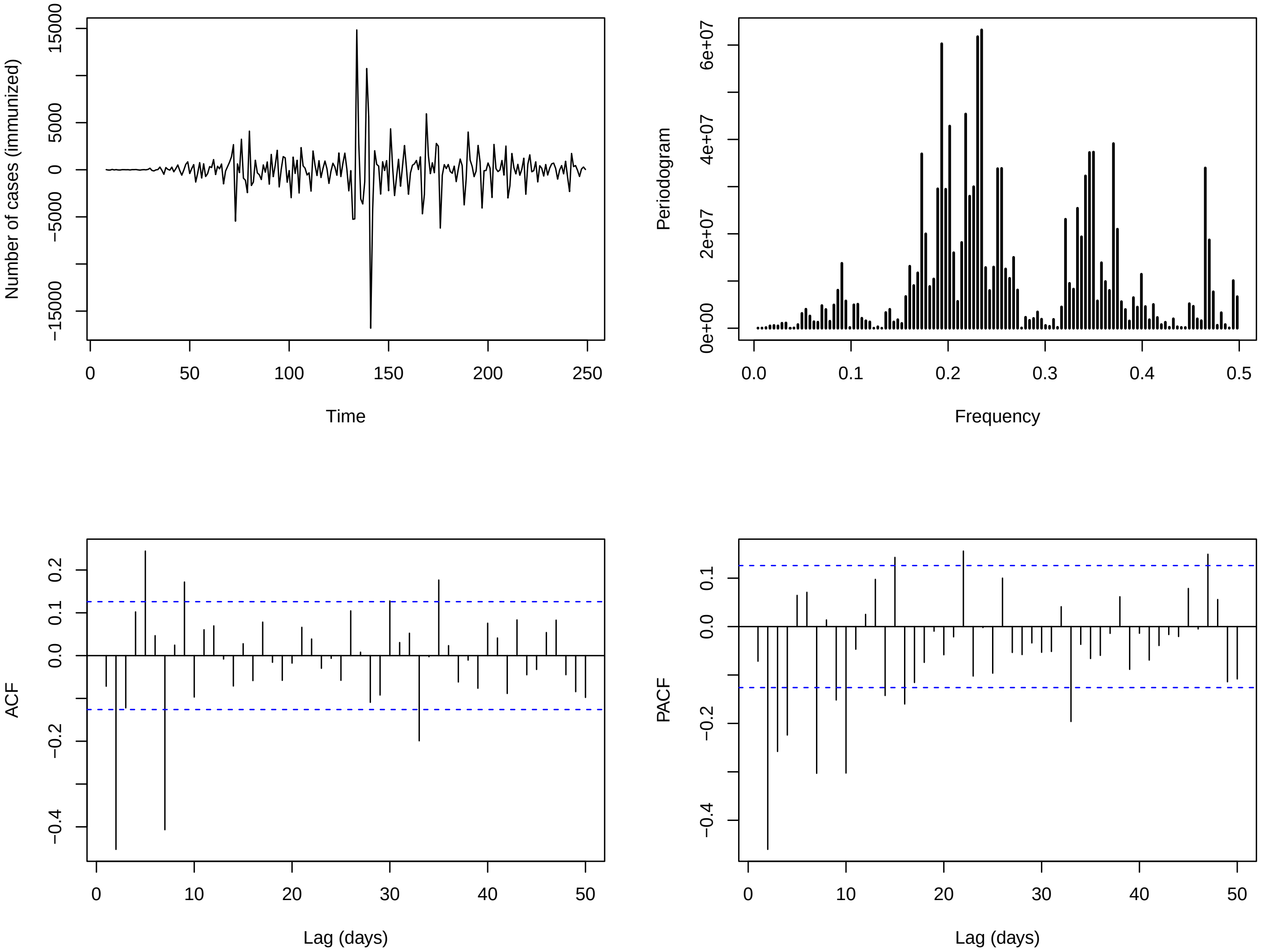


Figure 17: basic graphs in the analysis of time series applied to the series of immunized maternal population, considering seasonal differentiation of seven.

Now, we apply the data transformation.

```
## Summary of the time series
summary(ts(series$immunized.GES[start_vac:end_vac]))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0   280.2  2052.0  3568.7  5949.0 19552.0
```

```
## ## Adding 3500
immunized_log <-
  log(ts(series$immunized.GES[start_vac:end_vac]) + 3500)
immunized_log_dif1 <- diff(immunized_log)
immunized_dif7 <- diff(immunized_log_dif1, 7)

par(mfrow=c(2,2))
plot.ts(immunized_dif7,
        xlab = "Time",
        ylab = "Number of cases (immunized)")
TSA::periodogram(immunized_dif7,
                 xlab = "Frequency",
                 ylab = "Periodogram")
acf(immunized_dif7,
    xlab = "Lag (days)",
    ylab = "ACF",
    main = "",
    lag.max = 10,
    xaxt = "n")
axis(1, at = 1:10)
pacf(immunized_dif7,
     xlab = "Lag (days)",
     ylab = "PACF",
     main = "",
     lag.max = 10,
     xaxt = "n")
axis(1, at = 1:10)
```

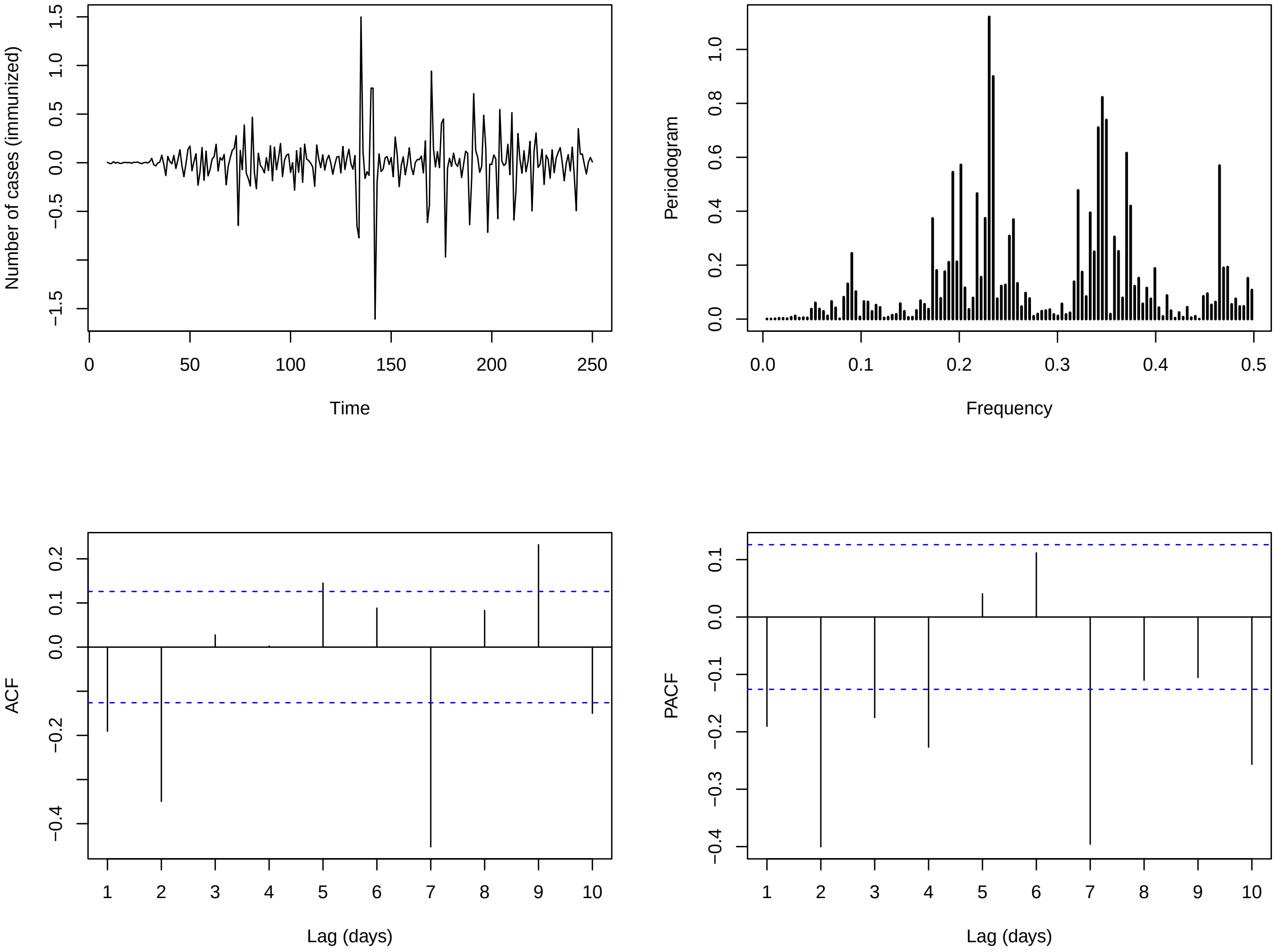


Figure 18: basic graphs in the analysis of time series applied to the series of immunized maternal population, considering seasonal differentiation of seven and logarithmic transformation of the data adding the mean.

Now, we verify one more time if the time series is stationary.

```
adf.test(immunized_dif7)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  immunized_dif7
## Dickey-Fuller = -9.7464, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

The Augmented Dickey-Fuller Test points out that the series is stationary.

Immunized brazilian population

These data consider brazilians in general who took the second dose of the vaccine against COVID-19 or a booster dose.

First, we analyze the basic graphs of the time series.

```r
par(mfrow=c(2,2))
plot.ts(ts(series$immunized.BR[start_vac:end_vac]),
        xlab = "Time",
        ylab = "Number of cases (Immunized Brazilian People)")

TSA::periodogram(ts(series$immunized.BR[start_vac:end_vac]),
                 xlab = "Frequency",
                 ylab = "Periodogram")
acf(ts(series$immunized.BR[start_vac:end_vac]),
    xlab = "Lag (days)",
    ylab = "ACF",
    main = "",
    lag.max = 250)
pacf(ts(series$immunized.BR[start_vac:end_vac]),
     xlab = "Lag (days)",
     ylab = "PACF",
     main = "",
     lag.max = 250)
```

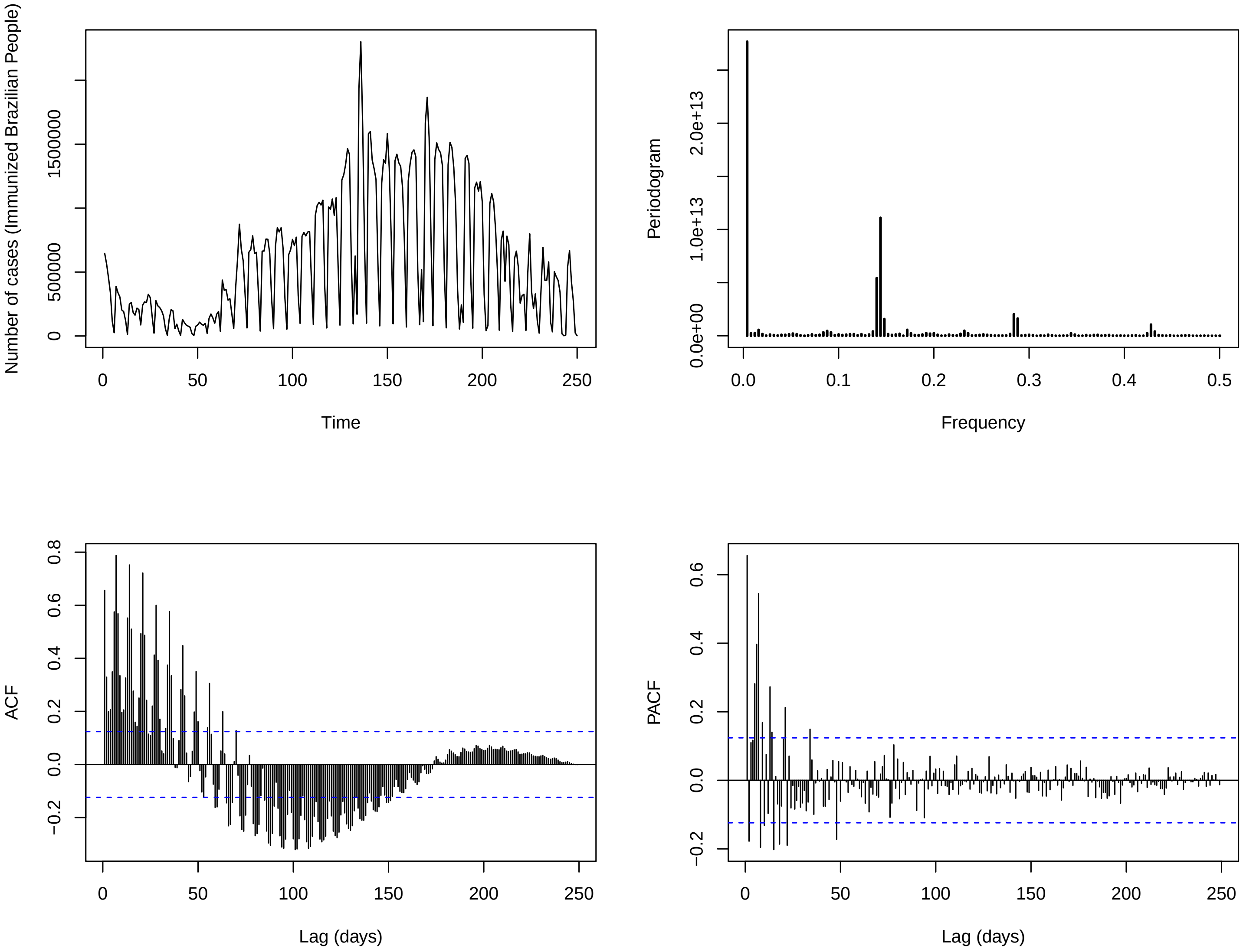


Figure 19: basic graphs in the analysis of time series applied to the series of immunized brazilian population.

The time series seems to be not stationary, as the mean and the variance looks not constant over time. We also can notice jumps through the bars in the autocorrelation and periodogram graphs. Let's try to apply one differentiation.

```
## 1 differentiation
immunizedBR_dif1 <- diff(ts(series$immunized.BR[start_vac:end_vac]))

par(mfrow=c(2,2))
plot.ts(immunizedBR_dif1,
        xlab = "Time",
        ylab = "Number of cases (Immunized Brazilian People)")
TSA::periodogram(immunizedBR_dif1,
                 xlab = "Frequency",
                 ylab = "Periodogram")
acf(immunizedBR_dif1,
    xlab = "Lag (days)",
    ylab = "ACF",
    main = "",
    lag.max = 50)
pacf(immunizedBR_dif1,
     xlab = "Lag (days)",
     ylab = "PACF",
     main = "",
     lag.max = 50)
```

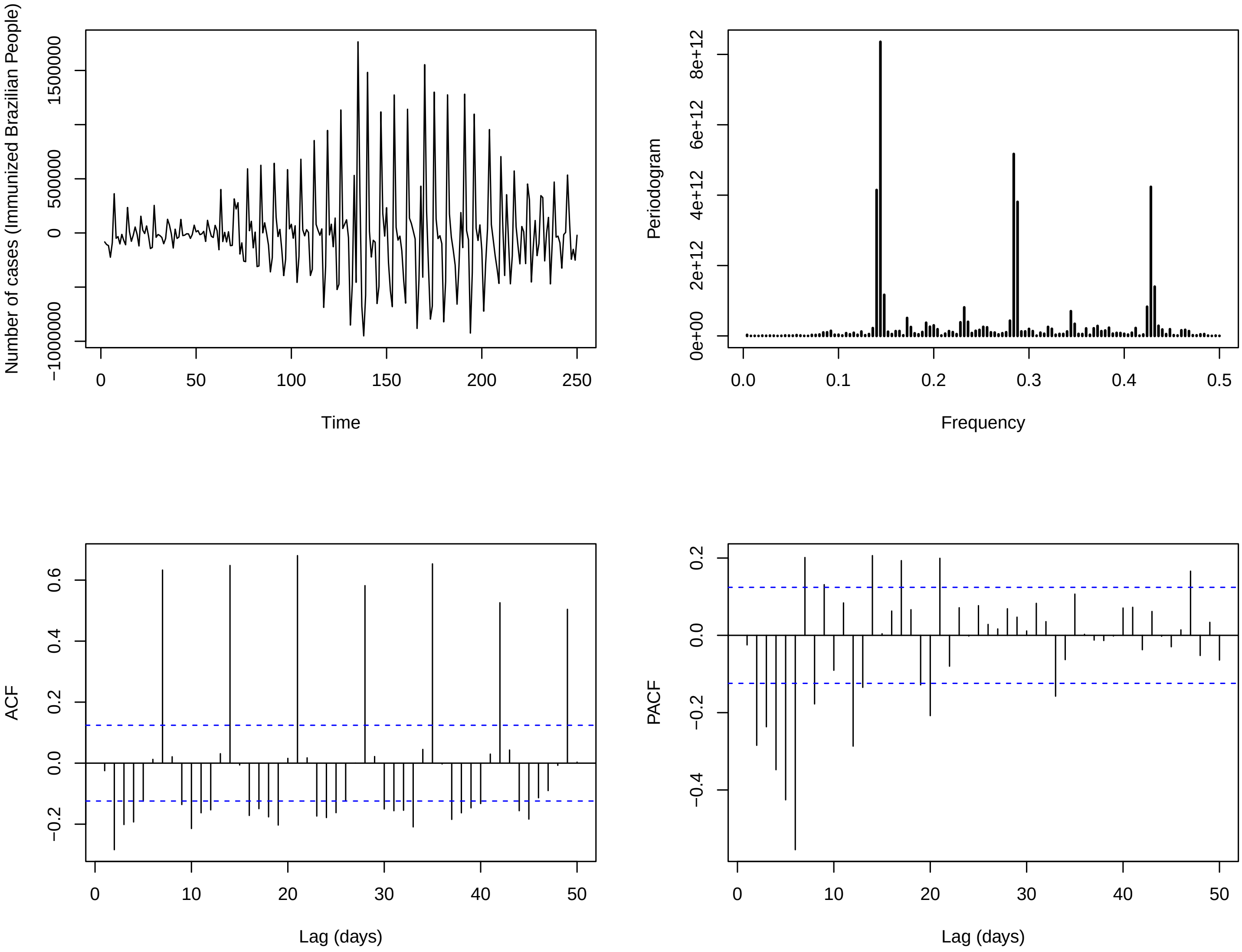


Figure 20: basic graphs in the analysis of time series applied to the series of immunized brazilian population considering one differentiation.

The jumps between the bars of the autocorrelation and periodogram graphs became more evident after the differentiation, now we can notice a period of 7 days as in the previous time series. We need to inform this period in the differentiation process.

```
immunizedBR_dif7 <- diff(immunizedBR_dif1, 7)

par(mfrow=c(2,2))
```

```
plot.ts(immunizedBR_dif7,
        xlab = "Time",
        ylab = "Number of cases (Immunized Brazilian People)")
TSA::periodogram(immunizedBR_dif7,
                 xlab = "Frequency",
                 ylab = "Periodogram")
acf(immunizedBR_dif7,
    xlab = "Lag (days)",
    ylab = "ACF",
    main = "",
    lag.max = 50)
pacf(immunizedBR_dif7,
     xlab = "Lag (days)",
     ylab = "PACF",
     main = "",
     lag.max = 50)
```

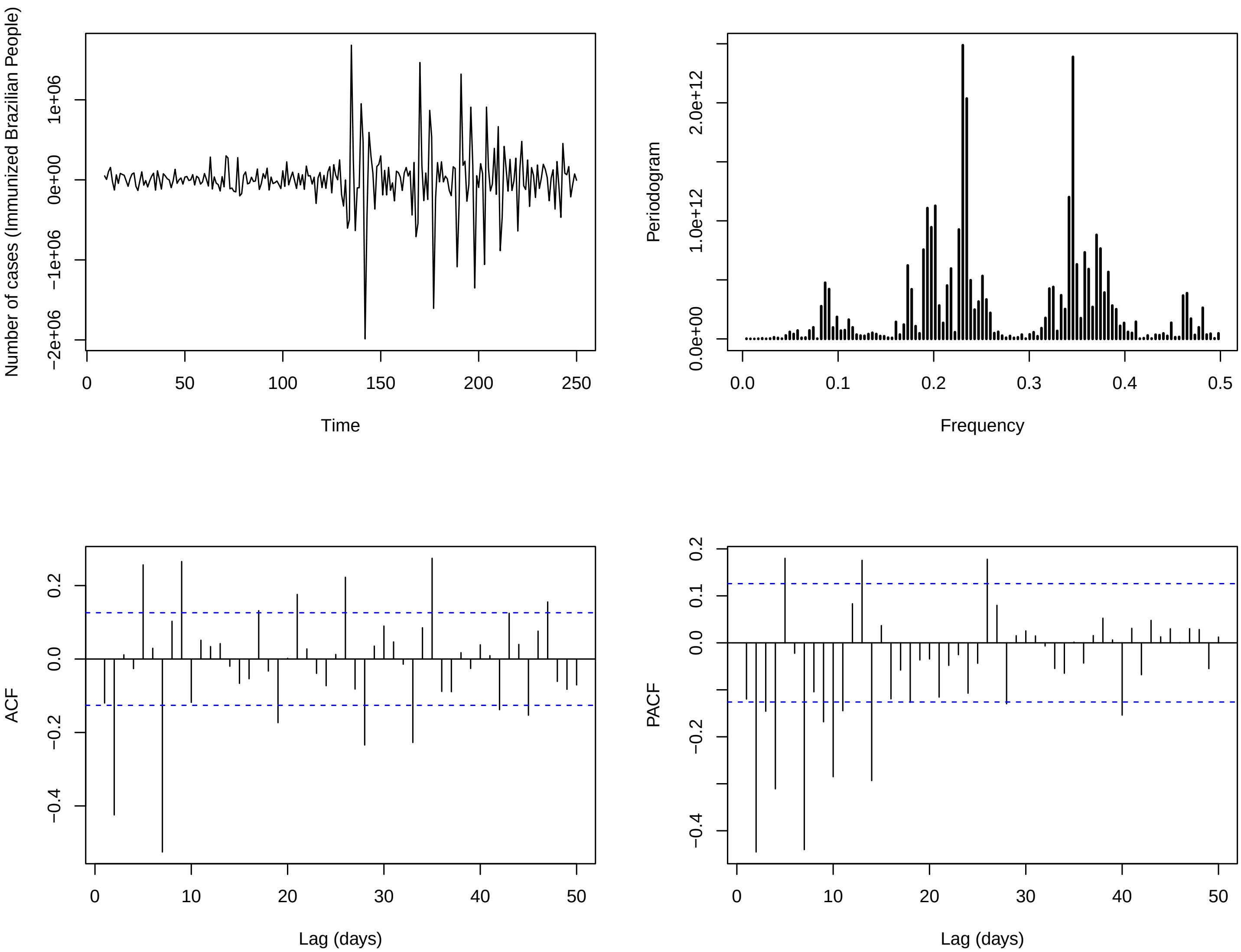


Figure 21: basic graphs in the analysis of time series applied to the series of immunized brazilian population considering

seasonal differentiation of seven.

Applying the transformation to the data.

```
immunizedBR_log <-
  log(ts(series$immunized.BR[start_vac:end_vac]))
immunizedBR_dif1 <- diff(immunizedBR_log)
immunizedBR_dif7 <- diff(immunizedBR_dif1, 7)

par(mfrow=c(2,2))
plot.ts(immunizedBR_dif7,
        xlab = "Time",
        ylab = "Number of cases (Immunized Brazilian People)")
TSA::periodogram(immunizedBR_dif7,
                 xlab = "Frequency",
                 ylab = "Periodogram")
acf(immunizedBR_dif7,
    xlab = "Lag (days)",
    ylab = "ACF",
    main = "",
    lag.max = 10,
    xaxt = "n")
axis(1, at = 1:10)
pacf(immunizedBR_dif7,
     xlab = "Lag (days)",
     ylab = "PACF",
     main = "",
     lag.max = 10,
     xaxt = "n")
axis(1, at = 1:10)
```

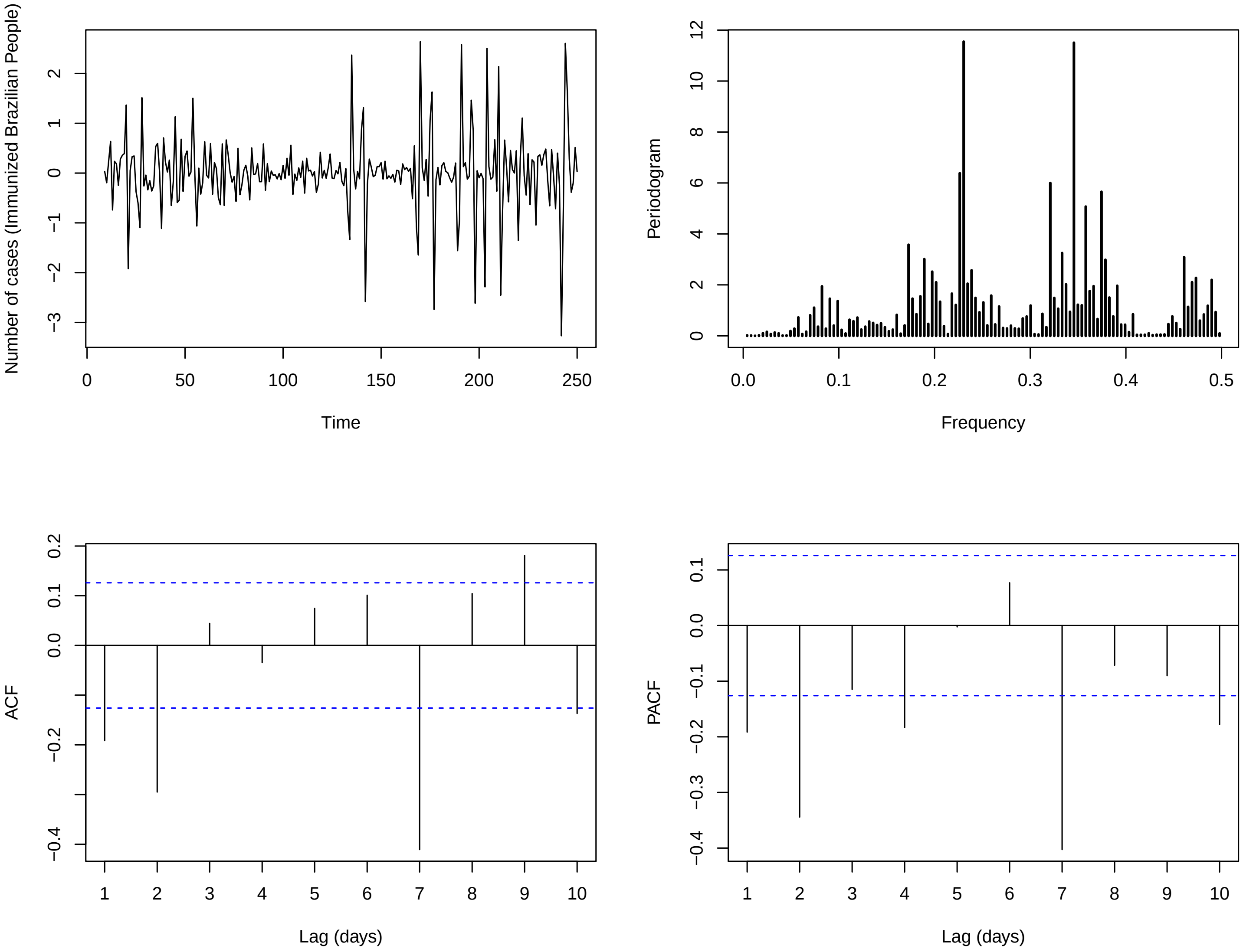


Figure 22: basic graphs in the analysis of time series applied to the series of immunized brazilian population considering seasonal differentiation of seven and logarithmic transformation of the data.

Now, we need to verify if the time series is stationary using the Augmented Dickey-Fuller Test.

```
adf.test(immunizedBR_dif7)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  immunizedBR_dif7
## Dickey-Fuller = -10.484, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

The Augmented Dickey-Fuller Test pointed out that the time series is now stationary.

Impact of COVID-19 vaccination on maternal deaths

Let's take a look again at our time series, and as we have seen, we are considering the period after the vaccination of pregnant and postpartum women did actually start.

```
plot.ts(series[start_vac:end_vac,-1],
        main = "Time series")
```

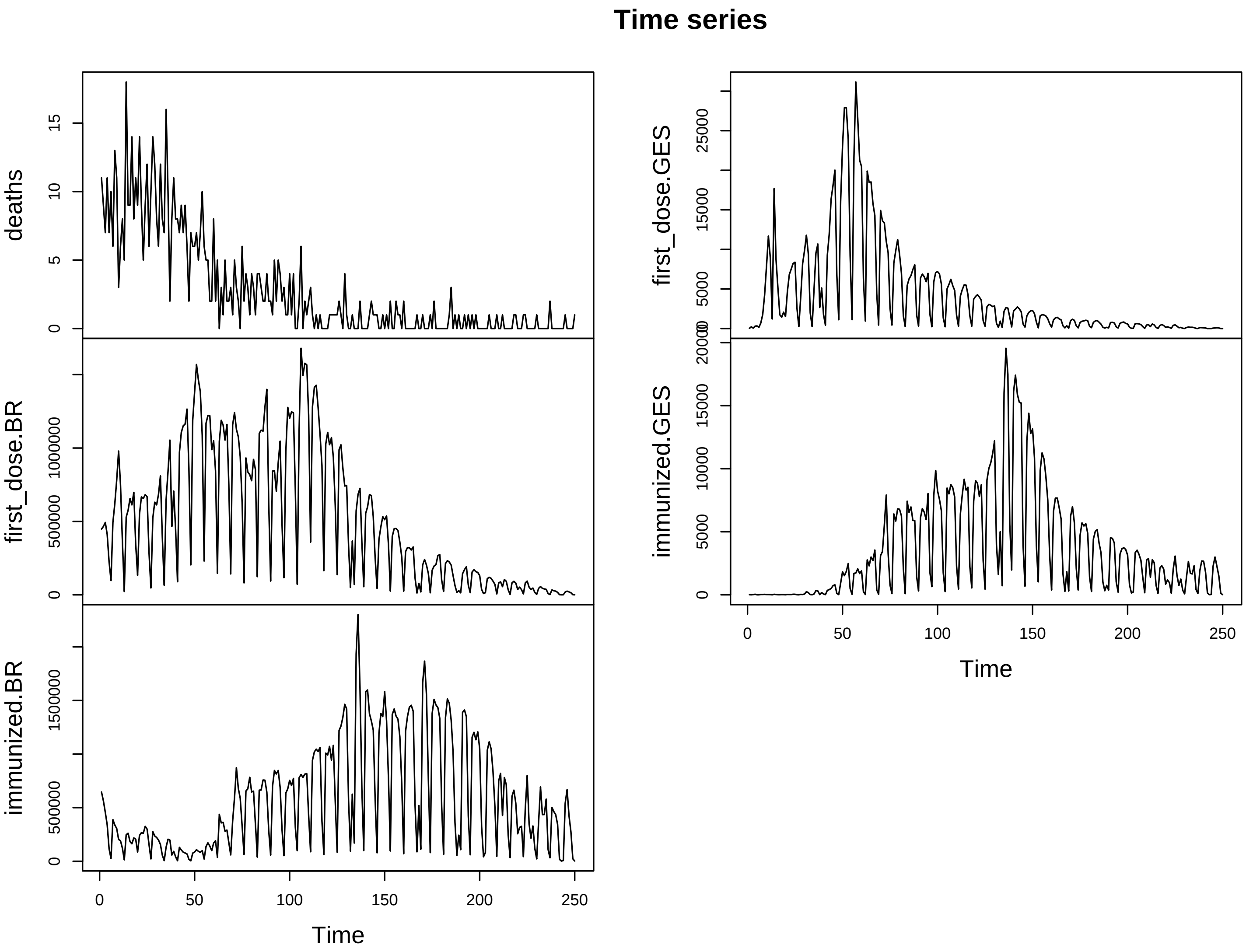


Figure 23: graphs of all time series.

For the following analyses, we need all the time series in Figure 23 to be stationary. Thus, the entire procedure performed during the univariate analysis of the series will be considered.

```
series_vac <- series[start_vac:end_vac,]
series_vac[,'deaths'] <- series_vac[,'deaths'] + 3.
series_vac[,'first_dose.GES'] <- series_vac[,'first_dose.GES'] + 4000.
series_vac[,'immunized.GES'] <- series_vac[,'immunized.GES'] + 3500.
```

```r
lseries <- log(series_vac[,-1])

df.diff  <-  base::diff(as.matrix(lseries), lag = 1)
df.diff0  <-  base::diff(df.diff[,-1],lag = 7)
df.diff3 <- cbind(df.diff[8:249,1], df.diff0)
colnames(df.diff3) <- c("deaths",
                        "first_dose.BR",
                        "immunized.BR",
                        "first_dose.GES",
                        "immunized.GES")
```

The analysis of the impact between the vaccination series and the series of deaths of pregnant and postpartum women will be performed using impulse response functions(IRF). This analysis consists of using vaccination series to create structural shocks in the deaths series so that we can visualize the impact over time. An initial impulse is performed, and then, through graphs we visualize the intensity and distance over time that this impact is maintained. Furthermore, confidence intervals are constructed for such functions, so that they are considered as statistical tests and not just graphs with suggestive capabilities.

To build impulse response functions, we must first build a vector autoregressive model or VAR. Earlier, we discussed autoregressive components, and talked about how these components are related to models where the covariates are the response variable itself, however, lagged. This type of model is called an autoregressive model or AR model. VAR models are nothing more than the multivariate version of the AR models.

Model

The first step in modeling VAR models is to identify the lag of the autoregressive component, and then, use it to build the model.

```r
# identifies the autoregressive lag
VARselect(df.diff3,
          lag.max = 7,
          type = 'const',
          season = 7,
          exogen = NULL)$selection
```

```
## AIC(n)  HQ(n)  SC(n) FPE(n)
##      7      2      1      7
```

```r
# Using lag = 2 to build a VAR model
var1 <-  vars::VAR(df.diff3,
                   p = 2,
                   type = 'none',
                   season = NULL,
                   exogen = NULL)
```

Once this is done, we need to do a diagnostic analysis of the model. This step is important so we can be confident that the model is meeting the theoretical concepts.

Diagnostics

First, we verify the serial autocorrelation of the model residuals with the Multivariate Portmanteau Test. For this test, the null hypothesis is that the residuals are not autocorrelated, which is desired in models with the objective of obtaining predictions. Although, it is not our objective to obtain predictions, the application of this test is part of the good practices of multivariate analysis in time series. It is worth remembering that the Multivariate Portmanteau Test is an asymptotic test.

```
(model.serial  <- vars::serial.test(var1))
```

```
##
##  Portmanteau Test (asymptotic)
##
## data:  Residuals of VAR object var1
## Chi-squared = 451.32, df = 350, p-value = 0.0002025
```

At the 5% significance level, the test indicates that there is some degree of autocorrelation in the residuals.

In univariate analysis, residuals of a good predictive model ideally need to be white noise. Being white noise means that the variables of the random process of the time series are independent and identically distributed with zero mean. In multivariate analysis, what is usually done is to analyze the stability of the residuals of the series, in order to identify possible structural breaks through tests known as CUSUM.

```
Stability1 <- stability(var1, type = "OLS-CUSUM")
plot(Stability1)
```

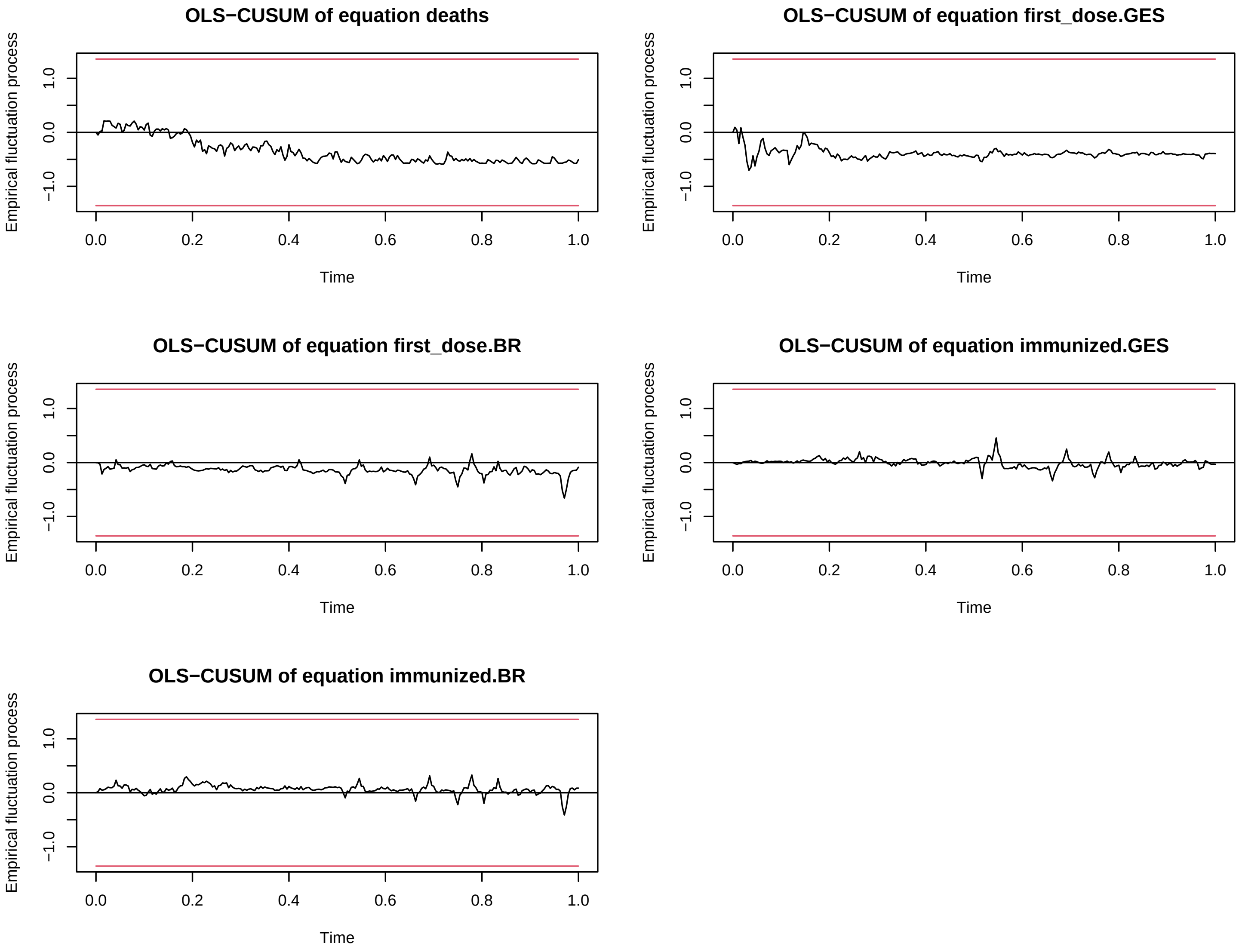


Figure 24: structural stability test of residuals with 95% confidence bands.

In Figure 24, we can see that there is no structural irregularity since no residual curve exceeds the confidence bands.

Another point that we must check is the normality of the residuals. Since we are in a multivariate problem, we want to test whether the residuals follow a Gaussian or a multivariate normal distribution. When we refer only as normal distribution, we imply that it is a univariate distribution, which is not our case, so there is an importance in making this distinction. Although it is an important step, it will not be a requirement for the construction of impulse response functions, since their confidence intervals will be built using bootstrap.

```
normality.test(var1, multivariate.only = TRUE)
```

```
## $JB
##
##  JB-Test (multivariate)
##
## data:  Residuals of VAR object var1
## Chi-squared = 599.85, df = 10, p-value < 2.2e-16
##
##
## $Skewness
##
##  Skewness only (multivariate)
##
## data:  Residuals of VAR object var1
## Chi-squared = 16.175, df = 5, p-value = 0.006363
##
##
## $Kurtosis
##
##  Kurtosis only (multivariate)
##
## data:  Residuals of VAR object var1
## Chi-squared = 583.68, df = 5, p-value < 2.2e-16
```

The normality test applied is called Jarque-Bera, which is a common test to be applied in time series. The test points with a 5% significance level that the residuals do not follow a Gaussian distribution.

Granger Causality

An interesting tool in time series is the Granger Causality Test. Consider X and Y two time series, we say that X Granger causes Y if the lagged values of X contribute significantly to determining the present values of Y, regardless of past values of Y . In other words, past values of X are relevant to predicting Y. With this, we are able to get a sense of whether the past values of the vaccination series had any influence on the values of the series of deaths that we observed.

The Granger Causality Test will be applied in a multivariate way, so that it will tell us if the series we are putting as a cause is Granger causing the values of the other series in the model. The null hypothesis of the test, is that the series we are putting as a cause, Granger cause the other series. In practice, If the null hypothesis is rejected, the test does not say exactly whether the series we are putting as a cause, Granger causes a specific series.

```
# First dose BR as a cause
c1 <- causality(var1, cause = "first_dose.BR")
c1$Granger
```

```
##
##  Granger causality H0: first_dose.BR do not Granger-cause deaths
##  immunized.BR first_dose.GES immunized.GES
##
## data:  VAR object var1
## F-Test = 3.795, df1 = 8, df2 = 1150, p-value = 0.0002076
```

```
# Immunized BR as a cause
c2 <- causality(var1, cause = "immunized.BR")
c2$Granger
```

```
##
##  Granger causality H0: immunized.BR do not Granger-cause deaths
##  first_dose.BR first_dose.GES immunized.GES
##
## data:  VAR object var1
## F-Test = 0.92248, df1 = 8, df2 = 1150, p-value = 0.4967
```

```
# First dose pregnant women as a cause
c3 <- causality(var1, cause = "first_dose.GES")
c3$Granger
```

```
##
##  Granger causality H0: first_dose.GES do not Granger-cause deaths
##  first_dose.BR immunized.BR immunized.GES
##
## data:  VAR object var1
## F-Test = 2.0541, df1 = 8, df2 = 1150, p-value = 0.03752
```

```
# Immunized pregnant women as a cause
c4 <- causality(var1, cause = "immunized.GES")
c4$Granger
```

```
##
##  Granger causality H0: immunized.GES do not Granger-cause deaths
##  first_dose.BR immunized.BR first_dose.GES
##
## data:  VAR object var1
## F-Test = 2.1054, df1 = 8, df2 = 1150, p-value = 0.03266
```

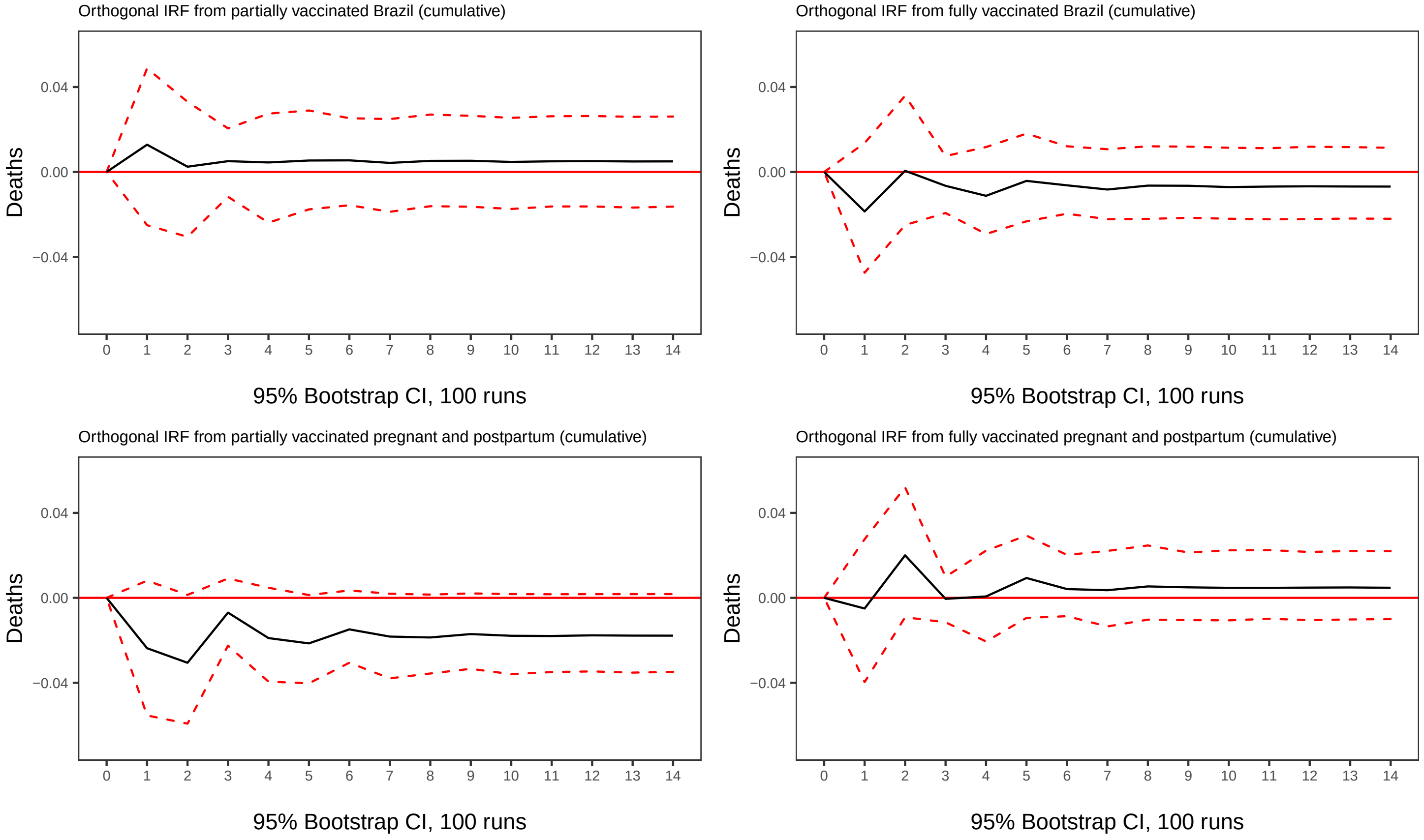
The Granger Causality Tests applied above indicate that among the 4 vaccination series we are studying, only the series of the immunized Brazilian population does not Granger causes the observed values of the other series, including the series of deaths. The test points out that the other series of vaccinations have some causal Granger effect, but does not specify in which series.

Impulse response functions(IRF)

In order to be able to assess more clearly the possible impact of vaccination series on deaths, we will now build impulse response graphs. We have 4 graphs where each one will have a series of vaccinations that will serve as an impulse to carry out the structural shock, and in all of them we will visualize the response in the deaths of pregnant and postpartum women.

```
# impulse response functions
feir1 <- irf(var1,
             impulse = "first_dose.BR",
             response = c("deaths"),
             n.ahead = 14,
             ortho = TRUE,
             runs = 100,
             cumulative = TRUE,
             boot = TRUE,
             ci = 0.95)

feir2 <- irf(var1,
             impulse = "immunized.BR",
             response = c("deaths"),
             n.ahead = 14,
             ortho = TRUE,
             runs = 100,
             cumulative = TRUE,
             boot = TRUE,
             ic = 0.95,
             plot.type = "multiple")

feir3 <- irf(var1,
             impulse = "first_dose.GES",
             response =c("deaths"),
             n.ahead = 14,
             ortho = TRUE,
             runs = 100,
             cumulative = TRUE,
             boot = TRUE,
             ic = 0.95)

feir4 <- irf(var1,
             impulse = "immunized.GES",
             response = c("deaths"),
             n.ahead = 14,
             ortho = TRUE,
             runs = 100,
             cumulative = TRUE,
             boot = TRUE,
             ic = 0.95)

# extract_varirf() extracts IRF estimates so we can use them freely
# https://raw.githubusercontent.com/anguyen1210/var-tools/master/R/extract_varirf.R

# Orthogonal Impulse Response from partially vaccinated Brazil (cumulative)
feir1_varirf <- extract_varirf(feir1)
plot1 <- feir1_varirf %>%
  ggplot(aes(x = period,
             y = irf_first_dose.br_deaths,
             ymin = lower_first_dose.br_deaths,
             ymax = upper_first_dose.br_deaths)) +
  geom_hline(yintercept = 0, color = "red") +
  geom_ribbon(fill = NA,
              color = "red",
              linetype = "dashed") +
  geom_line() +
  theme_bw() +
  ggtitle("Orthogonal IRF from partially vaccinated Brazil (cumulative)")+
  ylab("Deaths")+
  xlab("\n95% Bootstrap CI, 100 runs") +
  scale_x_continuous(breaks = 0:14) +
  coord_cartesian(ylim = c(- 0.07, 0.06)) +
  theme(panel.grid.minor = element_line(color = 'white'),
        panel.grid.major = element_line(color = 'white'),
        plot.title = element_text(size = 7.7),
        axis.text = element_text(size = 7))

# Orthogonal Impulse Response from Fully vaccinated Brazil (cumulative)
feir2_varirf <- extract_varirf(feir2)
plot2 <- feir2_varirf %>%
  ggplot(aes(x = period,
             y = irf_immunized.br_deaths,
             ymin = lower_immunized.br_deaths,
             ymax = upper_immunized.br_deaths)) +
  geom_hline(yintercept = 0, color = "red") +
  geom_ribbon(fill = NA,
              color = "red",
              linetype = "dashed") +
  geom_line() +
  theme_bw() +
  ggtitle("Orthogonal IRF from fully vaccinated Brazil (cumulative)")+
  ylab("Deaths")+
  xlab("\n95% Bootstrap CI, 100 runs") +
  scale_x_continuous(breaks = 0:14) +
  coord_cartesian(ylim = c(- 0.07, 0.06)) +
  theme(panel.grid.minor = element_line(color = 'white'),
        panel.grid.major = element_line(color = 'white'),
        plot.title = element_text(size = 7.7),
        axis.text = element_text(size = 7))

# Orthogonal Impulse Response from Partially vaccinated pregnant and postpartum (cumulative)
feir3_varirf <- extract_varirf(feir3)
plot3 <- feir3_varirf %>%
  ggplot(aes(x = period,
             y = irf_first_dose.ges_deaths,
             ymin = lower_first_dose.ges_deaths,
             ymax = upper_first_dose.ges_deaths)) +
  geom_hline(yintercept = 0, color = "red") +
  geom_ribbon(fill = NA,
              color = "red",
              linetype = "dashed") +
  geom_line() +
  theme_bw() +
  ggtitle("Orthogonal IRF from partially vaccinated pregnant and postpartum (cumulative)")+
  ylab("Deaths")+
  xlab("\n95% Bootstrap CI, 100 runs") +
  scale_x_continuous(breaks = 0:14) +
  coord_cartesian(ylim = c(- 0.07, 0.06)) +
  theme(panel.grid.minor = element_line(color = 'white'),
        panel.grid.major = element_line(color = 'white'),
        plot.title = element_text(size = 7.7),
        axis.text = element_text(size = 7))

# Orthogonal Impulse Response from Fully vaccinated pregnant and postpartum (cumulative)
```

```
feir4_varirf <- extract_varirf(feir4)
plot4 <- feir4_varirf %>%
  ggplot(aes(x = period,
             y = irf_immunized.ges_deaths,
             ymin = lower_immunized.ges_deaths,
             ymax = upper_immunized.ges_deaths)) +
  geom_hline(yintercept = 0, color = "red") +
  geom_ribbon(fill = NA,
              color = "red",
              linetype = "dashed") +
  geom_line() +
  theme_bw() +
  ggtitle("Orthogonal IRF from fully vaccinated pregnant and postpartum (cumulative)")+
  ylab("Deaths")+
  xlab("\n95% Bootstrap CI, 100 runs") +
  scale_x_continuous(breaks = 0:14) +
  coord_cartesian(ylim = c(- 0.07, 0.06)) +
  theme(panel.grid.minor = element_line(color = 'white'),
        panel.grid.major = element_line(color = 'white'),
        plot.title = element_text(size = 7.7),
        axis.text = element_text(size = 7))

(plot1 | plot2) / (plot3 | plot4)
```


Orthogonal IRF from partially vaccinated Brazil (cumulative)

Orthogonal IRF from fully vaccinated Brazil (cumulative)

Orthogonal IRF from partially vaccinated pregnant and postpartum (cumulative)

Orthogonal IRF from fully vaccinated pregnant and postpartum (cumulative)

Figure 25: Orthogonal Impulse Response Functions (black line) over 14 days for each vaccination series with 95% confidence bands (dashed red curves) calculated using bootstrap. The “partially vaccinated” refer to the series of the first dose of the

COVID-19 vaccine of their respective group (Brazil or pregnant and postpartum women). The "fully vaccinated" refer to populations (Brazil or pregnant and postpartum women) who took a second dose or booster, which we have called immunized.

In Figure 25, we can visualize the impact of each vaccination series on the deaths of pregnant and postpartum women. We can notice that, for all the vaccination series we had an immediate impact on the series of deaths, however, we see that the impact generated by the series of vaccination with the first dose of the vaccine of the maternal population (partially vaccinated pregnant and postpartum) was reasonably greater than the others. Note that, after the initial "jump", the IRF of the vaccination with the first dose series of the maternal population remained the furthest away from zero over time. This shows that the first dose of the vaccine against COVID-19 in pregnant and postpartum women in Brazil, when compared to the vaccination of the general population and the second dose or booster of the maternal population, was the factor that, not only had the greatest immediate impact on the fall of deaths in the month of May in 2021, as it was also the factor that held back the downward behavior more intensely over time.