Italian Vaccination Program Forecasting

Alessandro Lombardi

1 Introduction

This project aims to build an ARIMA model to forecast the daily covid-19 vaccination rate in Italy.
ARIMA stands for Auto Regressive Integrated Moving Average, and is a common approach used in time series forecasting.
ARIMA will be compared with a polynomial regression, using MAPE (mean average percentage error) as evaluation function.
We will use cross validation to observe the performance of the model, by using MAPE as a measure of how the prediction fits the actual daily number of doses.
The original data is provided by Our World in Data, and is available in csv format at:

https://www.kaggle.com/arthurio/italian-vaccination

This csv file contains day-by-day number of first and second doses, with the level of granularity expressed by:

1. Age-group (16-19, 20-30, 30-40,40-50,50-60,60-70,70-80,80-90,90+)
2. Italian Region (there are 21 different regions)
3. Date of vaccination (starting from December 27th 2020)
4. Vaccine Supplier (Pfizer, Astrazeneca, Moderna and Janssen)

Section 2 will try to explore the dataset in order to see how the above features 1. 2. 3 and 4, together with male-female distribution, affect the vaccine distribution rate.
Section 3 will build the (optimal) ARIMA model.
Section 4 will evaluate the predicted vaccine rate with the test set, and compare the resulting MAPE with a polynomial interpolation regression

2 Data Exploration

This section will focus on a portion of the features of the original dataset, to see how they affect the vaccine distribution in Italy.
In particular, we will focus on:

- Age-range
- Regions
- Vaccine supplier
- Sex

The re-arranged dataset looks like this:

administration_date	supplier	age_range	females
2021-01-01	Pfizer/BioNTech	20-29	49
2021-01-01	Pfizer/BioNTech	30-39	97
2021-01-01	Pfizer/BioNTech	40-49	110
2021-01-01	Pfizer/BioNTech	50-59	114
2021-01-01	Pfizer/BioNTech	60-69	43
2021-01-01	Pfizer/BioNTech	16-19	2

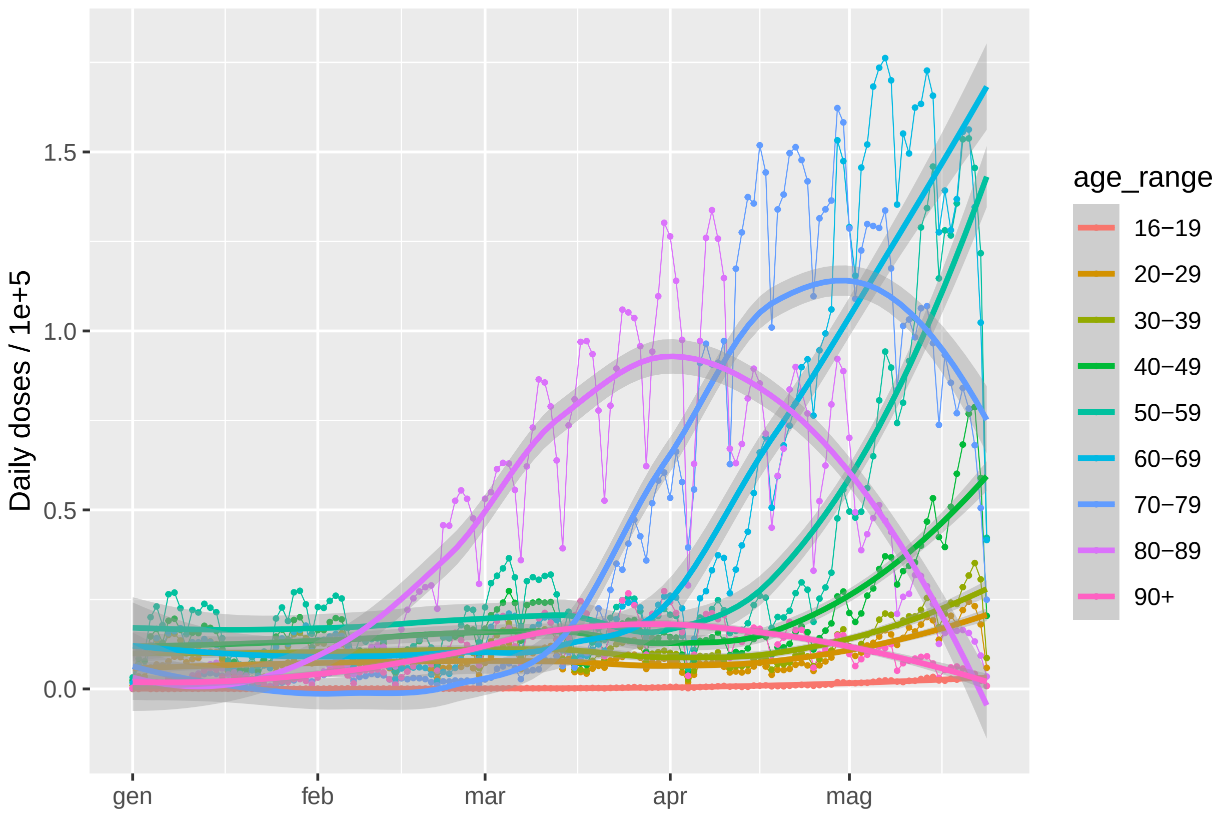
males	first_dose	second_dose	region_name
27	76	0	Campania
92	189	0	Campania
102	212	0	Campania
156	270	0	Campania
129	172	0	Campania
0	2	0	Lazio

In following paragraphs and sections we generically write *Dose* to indicate the total daily doses, given by the sum of first and second doses:

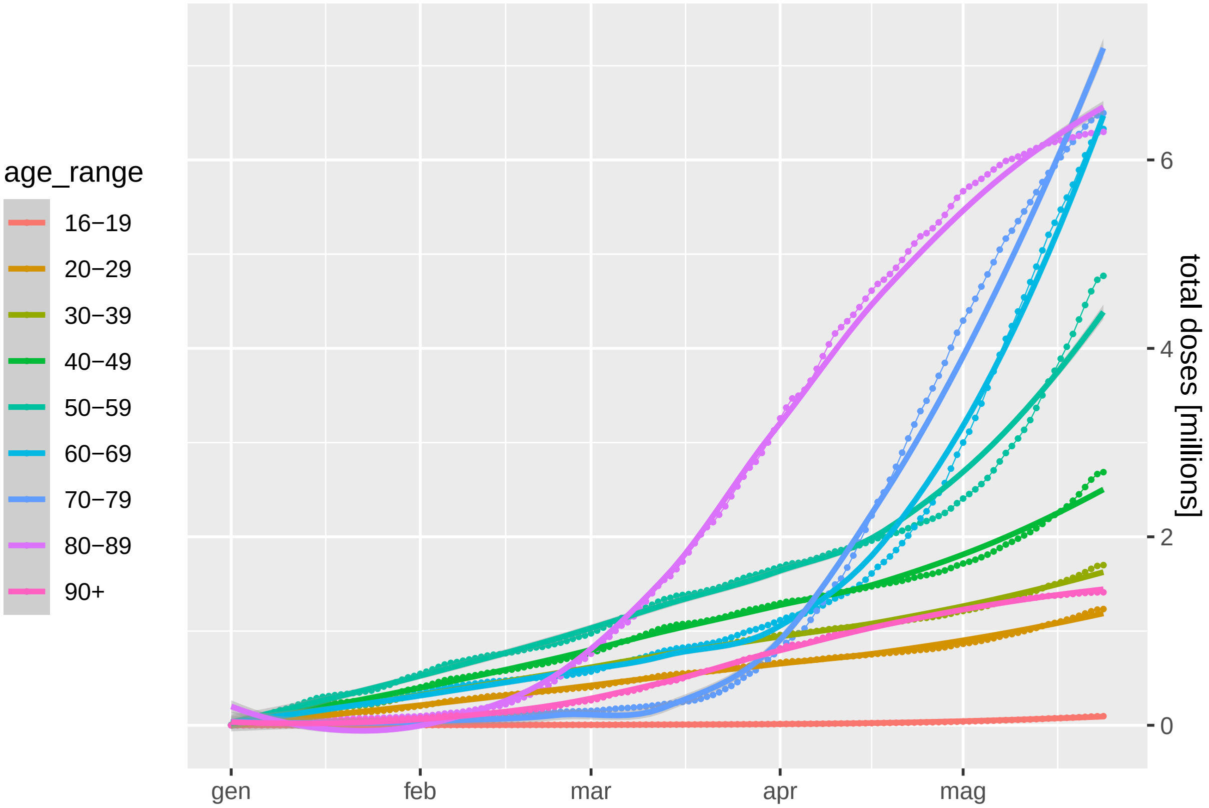
$$Dose = first\_dose + secod\_dose$$

2.1 Vaccination rate by age-range

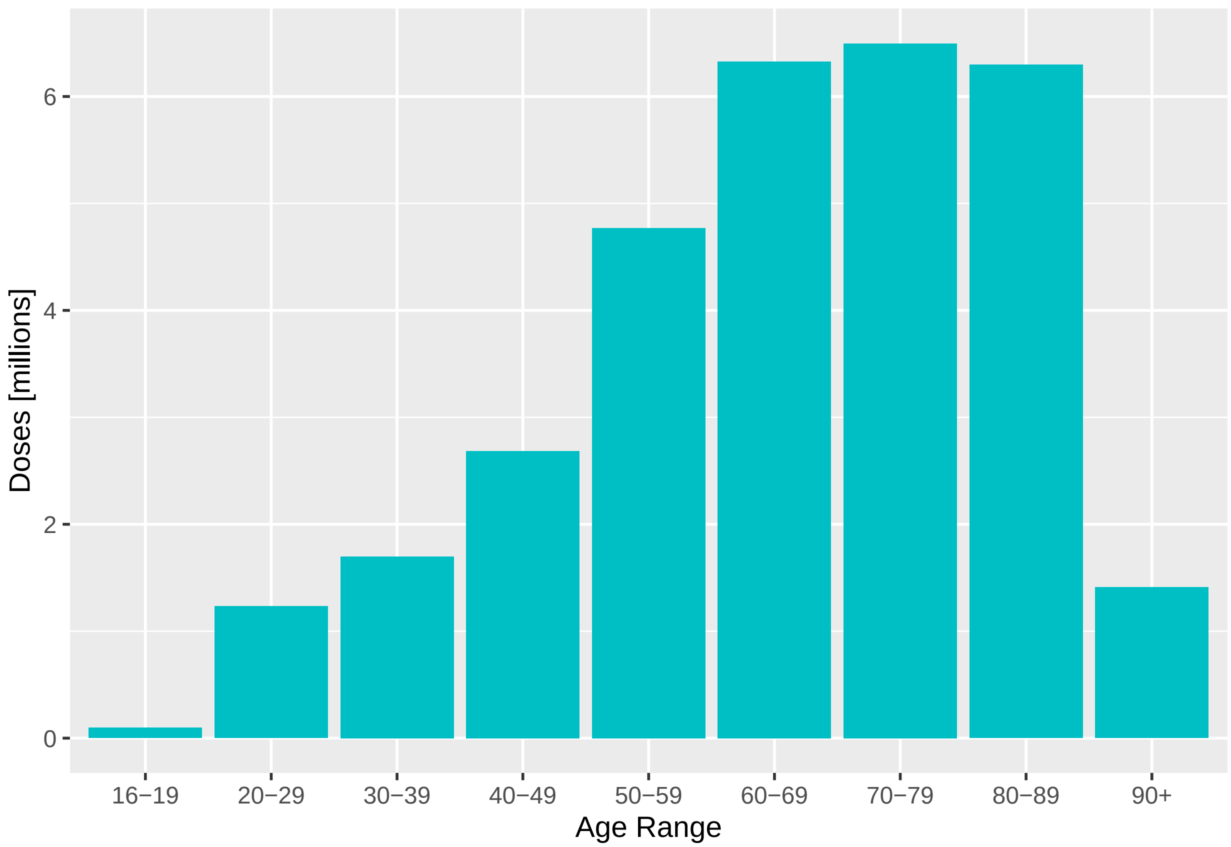
In Italy, the first people that received a vaccine shot in early 2021 were health-care workers, immediately followed by older people (> 80 years old).
The plot below shows how the vaccine daily rate evolved over the last months:



We can see that by now (late May 2021) most of the people older than 70 years old have already received the vaccine.
We also note an increasing trend for all the remaining age groups.
If we sum the vaccine doses day by day we can plot the cumulative vaccine trend by age group:



Finally, this bar-plot summarizes the total up-to-date vaccine distribution by age group:



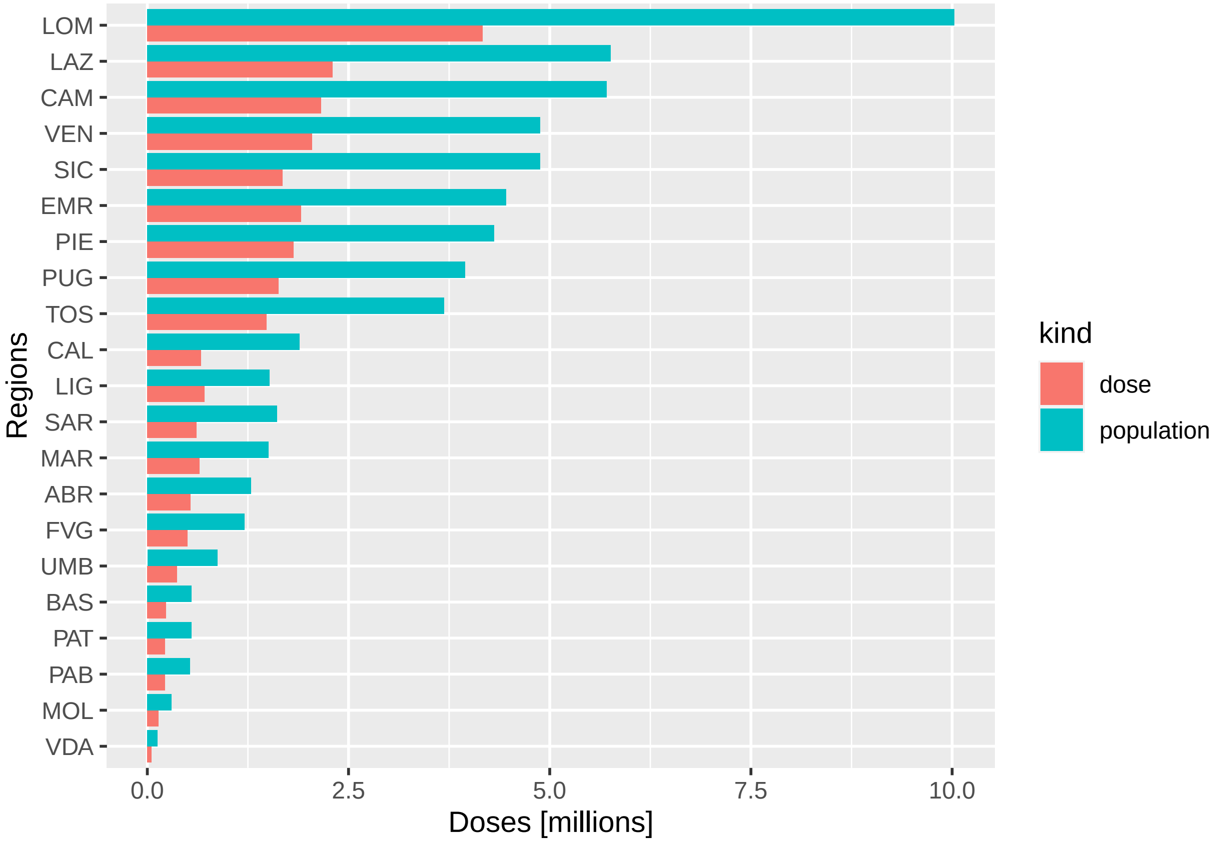
2.2 Vaccination rate by region and area

Italy is divided into 21 local regions (Abruzzo, Basilicata, Calabria, Campania, Emilia ROmagna, Friuli-venezia-giulia, Lazio, Liguria, Lombardia, Marche, Molise, Provincia di Bolzano, Provincia di Trento, Piemonte, Puglia, Sardegna, Sicilia, Toscana, Umbria, Valle d'Aosta, Veneto).

The plot below shows:

- the current number of vaccine shots by region, in red
- the total population of the region, in blue

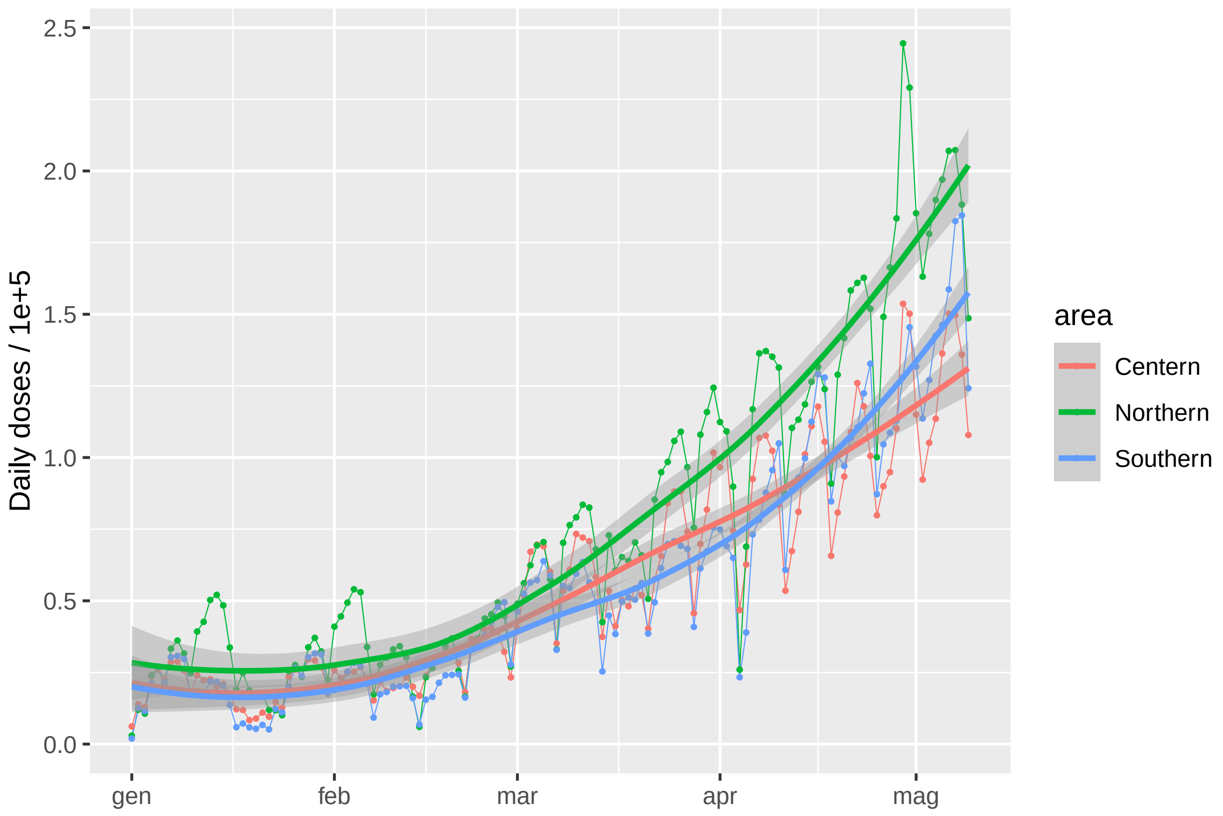
Considering that most of vaccines requires two shots to receive a complete vaccination, at the end of the vaccination program an ideal scenario would be to have the red bar to be twice the blue bar in length.



Different regions have different populations. In order to see if the vaccination trend is homogeneous across all the country, we can categorize the Italian regions to fall into three different areas:

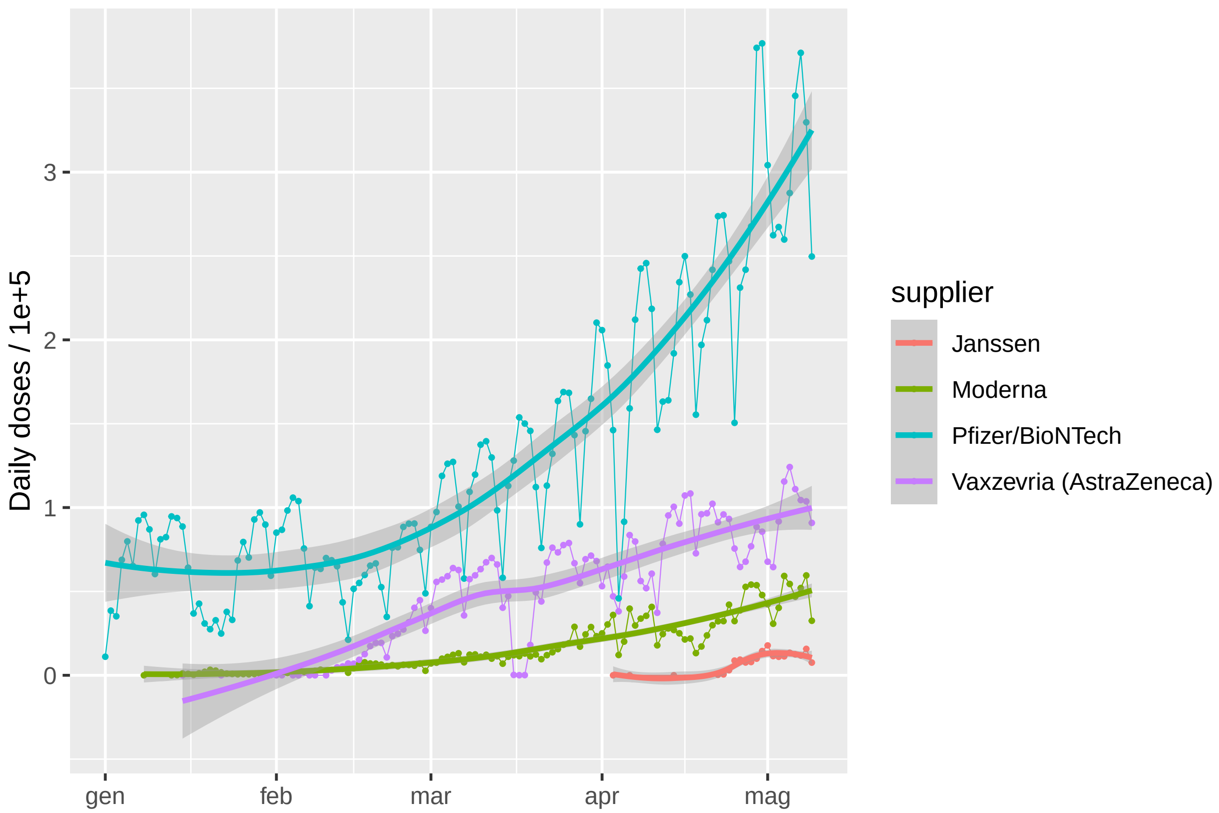
- Northern Italy
- Central Italy
- Southern Italy

The plot below shows that the trend of daily vaccine shots is homogeneous across the three different zones (Northern Italy is slightly higher, but we need to keep in mind that most of Italian people live in this zone)



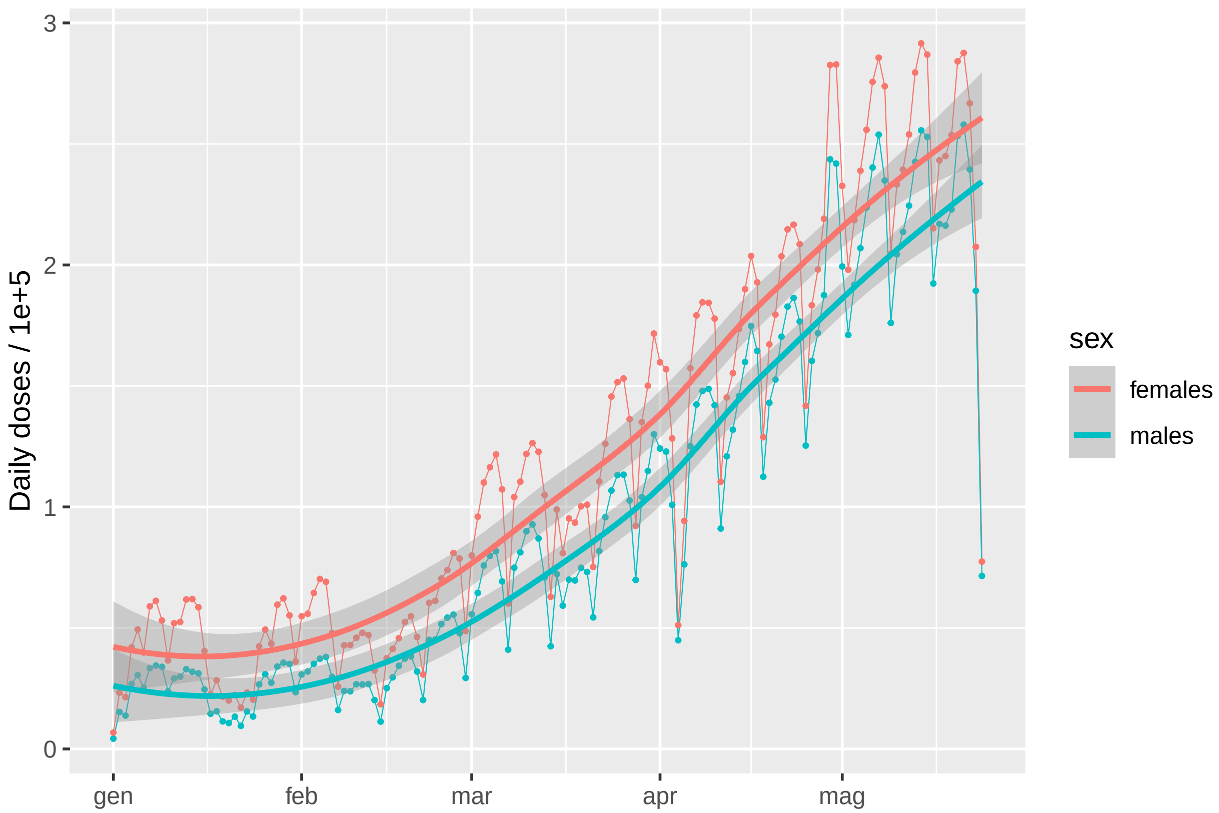
2.3 Vaccination rate by vaccine supplier

Most of the doses come from Pfizer/Biontech, followed by Astrazeneca and Moderna. Janssen is starting only now (May 2021) to show a positive trend. We can see that from the plot:



2.4 Vaccination rate by sex

The following plot shows that the female daily vaccine rate trend is slightly higher compared to the male trend.
This is probably due to the fact that the first people who received a shot fall on older category groups, where females are generally more then males (and this is probably true also in other European countries).



3 Data Normalization and Modeling

ARIMA stands for *Auto Regressive Integrated Moving Average.*
To implement ARIMA regression we will use the function ARIMA(data.frame(), order =c(p,d,q)) from the package *"tseries".*
The arguments p, d and q are integer numbers, and the goal will be to find the optimal set of (p,d,q) that minimizes the evaluation function MAPE, but at the same time limiting over-training.

The term *Auto Regressive* means that we can estimate Y_t, or the output of the series at time t, as a linear combination of its previous p lags:

$$(1)\ \ Y_t = \beta_0 + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + ... + \beta_p Y_{t-p} + \epsilon_t$$

where:

- $\beta_0 ... \beta_p$ are constant terms
- $Y_{t-1} ... Y_{t-p}$ are the p lagged values of Y_t
- ϵ_t is the stochastic error, or white noise, with average $\mu = 0$ and variance $\sigma^2 = const$

The term *Moving average* means that we can represent the output of the series Y_t as a linear combination of its previous q lagged errors:

$$(2)\;\; Y_t = \beta_0 + \epsilon_t + \phi_1\epsilon_{t-1} + \phi_2\epsilon_{t-2} + ... + \phi_q\epsilon_{t-q}$$

where:

- $\beta_0, \phi1 ... \phi_p$ are constant terms
- $\epsilon_{t-1} ... \epsilon_{t-p}$ are the q lagged errors of Y_t
- ϵ_t is the white noise at time t, with average $\mu = 0$ and variance $\sigma^2 = const$

By combining (1) and (2) we can write the final ARIMA equation as:

$$(3)\;\; Y_t = \beta_0 + \epsilon_t + \phi_1\epsilon_{t-1} + \phi_2\epsilon_{t-2} + ... + \phi_q\epsilon_{t-q} + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + ... + \beta_p Y_{t-p}$$

An ARIMA model can be used only if the time series is stationary, meaning that its average and standard deviation remain constant over time.
If the series does not respect this condition, it can still be made stationary. For example, we can compute the difference of the series: $d_k = Y_{k+1} - Y_k$ for every k in $\{1, 2, ..., max(t) - 1\}$
The term *Integrated* literally means that an ARIMA model can be the result of the integral of a differentiated time series.

3.1 Vaccination rate as a time series

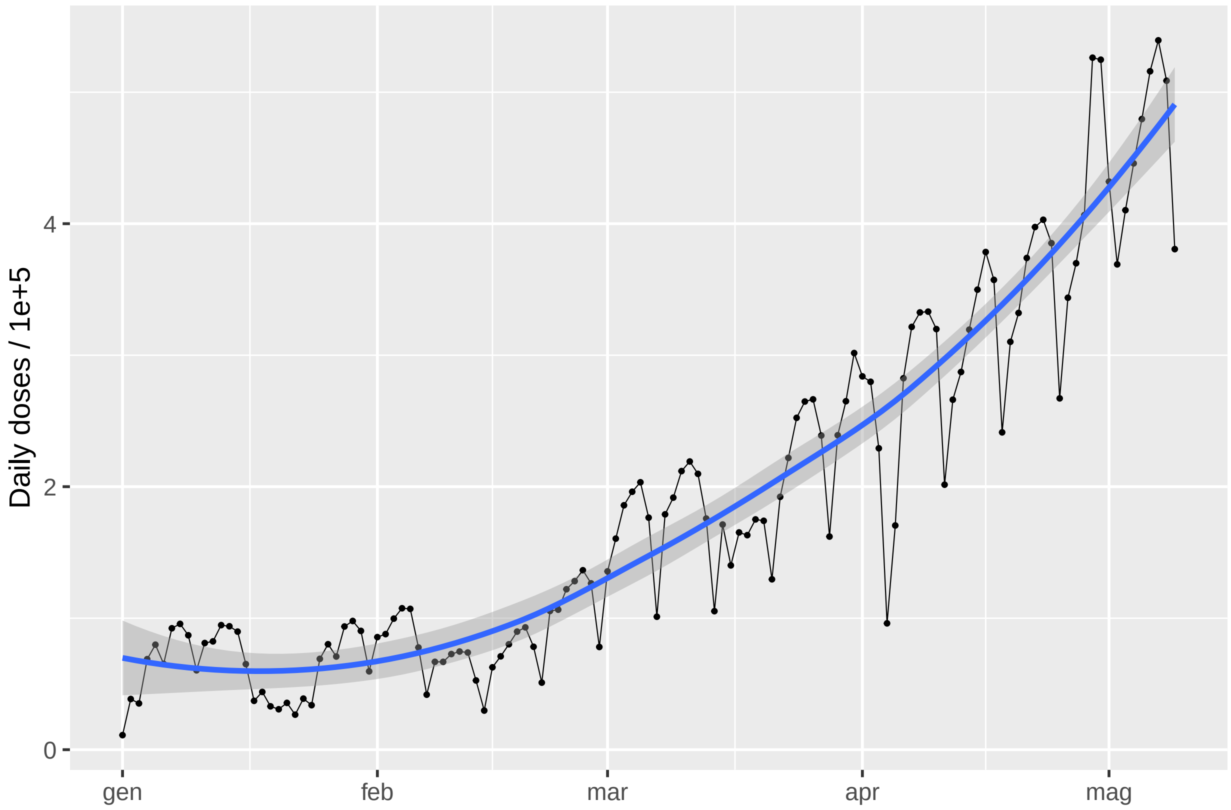
A time series is a list of observations at different times. The time frequency can be yearly, monthly, daily or even less than a day.
For this project we focus on the daily vaccination rate. The code below transform the train_set into a time series:

```r
daily_doses <- train_set %>%
  group_by(administration_date) %>%
  summarise(dose = sum(dose))



knitr::kable(head(daily_doses))
```

administration_date	dose
2021-01-01	11063
2021-01-02	38528
2021-01-03	35172
2021-01-04	68837
2021-01-05	79833
2021-01-06	65210

In other words, we simply group by the administration_date and summarize the doses as the sum of doses. In this way we lose the information about the other features (vaccine supplier, age range, sex, regions), and we try to build an ARIMA model to predict the future daily vaccination rate using only the data from the past observations of the time series.

The train set now represents the daily vaccine shots from day 1 (Jan 1st 2021) up to *max(administration_date) - 14*:



3.2 the ARIMA model

3.2.1 the *d* parameter

In order to apply ARIMA, a time series must be stationary.
We use *augmented dickey fuller test (adf.test)* to see if the time series is stationary. If the resulting p-value is less than 0.05 we reject the null hypothesis (series non stationary) and we do accept the alternative hypothesis (the time series is stationary):

```
daily_doses <- daily_doses %>% pull(dose)
adf.test(daily_doses, alternative = "stationary")
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  daily_doses
## Dickey-Fuller = -1.5124, Lag order = 5, p-value = 0.7792
## alternative hypothesis: stationary
```

Since the p-value is higher than 0.05 the time series is not stationary.
So, we re-write the time series as its first difference, using the *diff* function, and apply *adf.test* again to see if the series gets stationary:

```r
first_diff = diff(daily_doses)
adf.test(first_diff, alternative = "stationary")
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  first_diff
## Dickey-Fuller = -13.624, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

From the ADF test we can infer that the first difference seems to be enough to make the series stationary. Therefore the d parameter in ARIMA(p,d,q) can be set to 1:

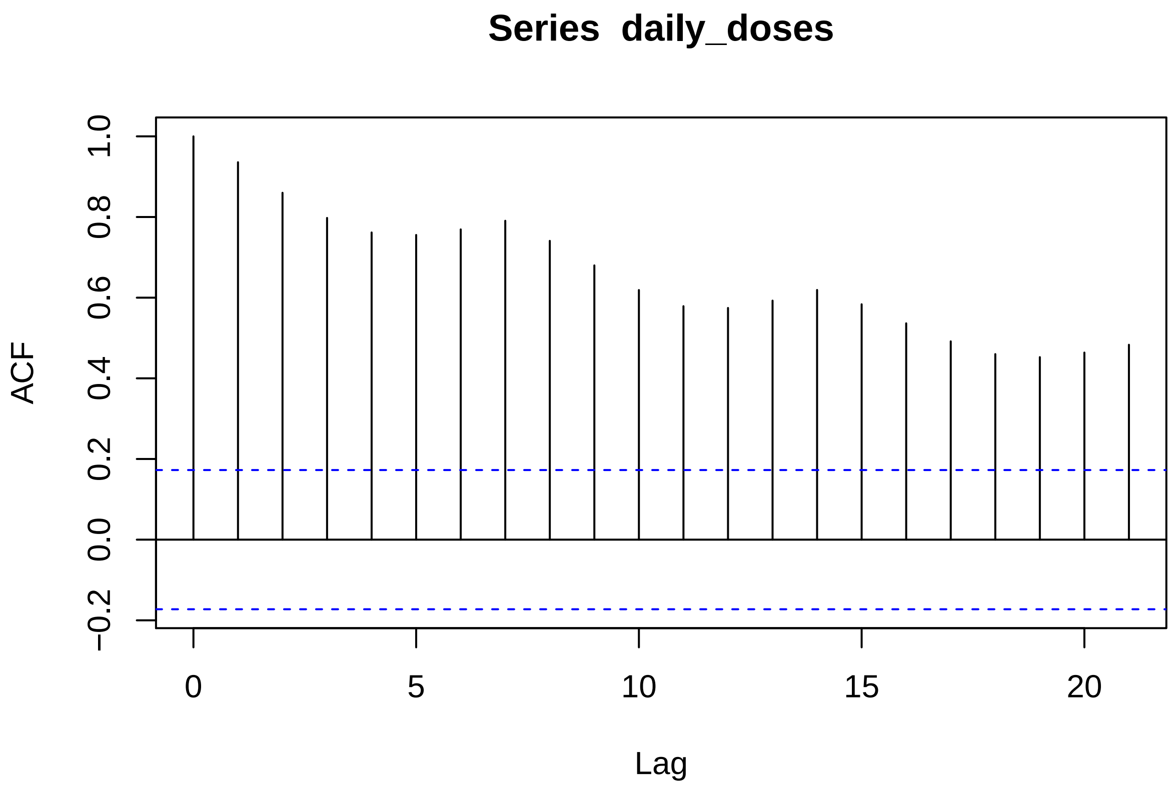
$d = 1$

Another way to set the d parameter is to look at the autocorrelation function.
Autocorrelation of order k is the correlation between the time series and the time series obtained by delaying the series k times.
For example:

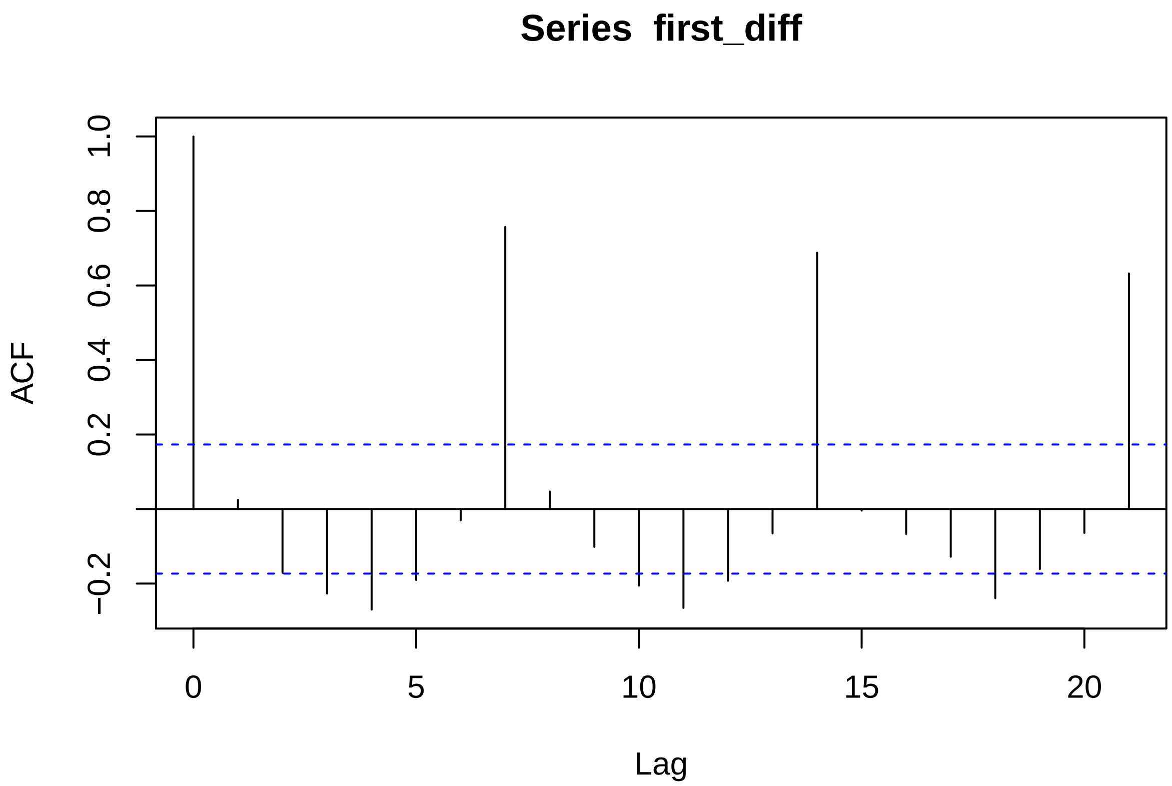
- when k = 0, the autocorrelation is equal to 1, as it represents the correlation between the time series and an exact copy of itself.
- when k = 1, the autocorrelation is the correlation between the time series $(y_0, y_1, ..., y_{n-1})$ and the 1-step-translated version of the series $(y_1, y_2, ..., y_n)$

The autocorrelation plot or *correlogram* plots the auto-correlations at different values of k (typically from k = 0 to k = 20).

If we plot the correlogram of the original series we see that the auto-correlations are all above the confidence interval (dashed line), meaning that the series is not stationary:



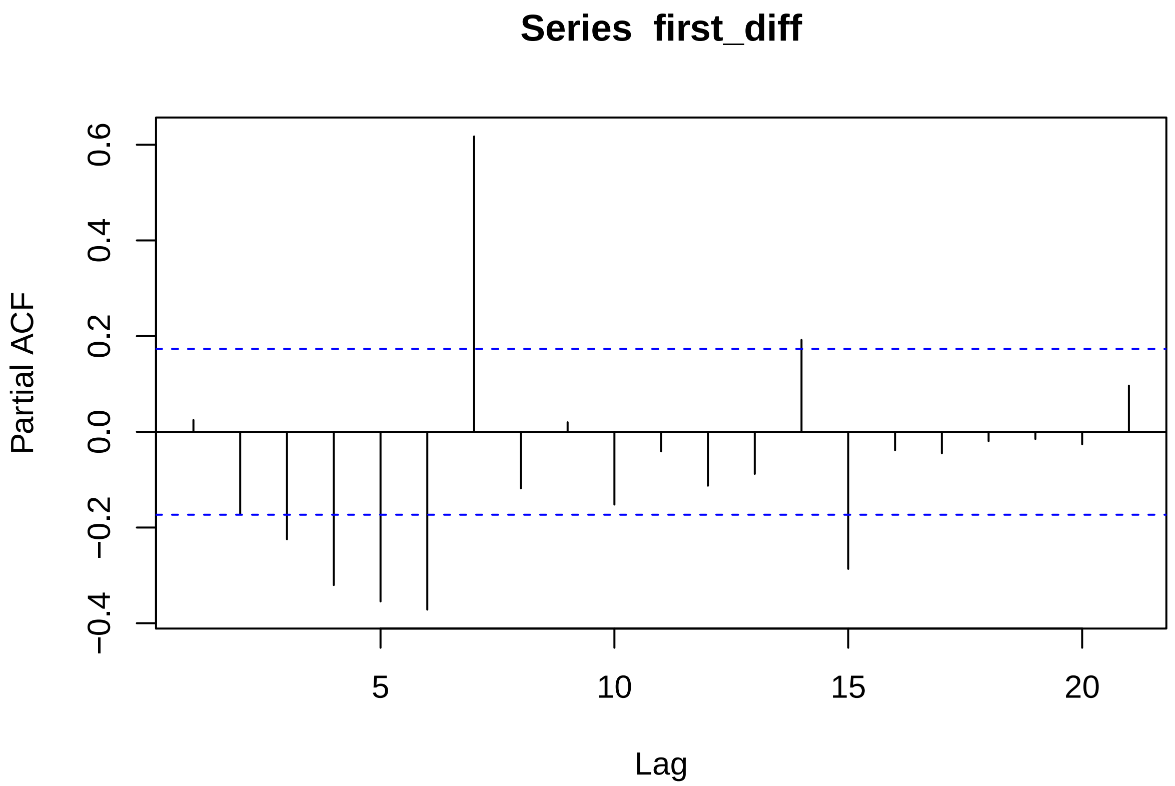
If we plot the correlogram for the first-difference series, we note that for k = 1 the correlation is way below the confidence interval, a symptom that the series may be now stationary.



3.2.2 the p parameter

The p parameter, or the number of lagged observations required to build the AR part of the model, can be picked by looking at the *partial autocorrelation function (PACF)*.
PACF of order k represents the correlation between the observation at time t Y_t and the observation at time Y_{t-k} without considering the observations in between $Y_{t-1}, ..., Y_{t-k+1}$.
Again, the k values for which the correlations are below the confidence interval may be good candidates as p numbers to be used in ARIMA(p,d,q). We save the p candidates in the PACF data frame (see code below).

```
PACF = pacf(first_diff)
```

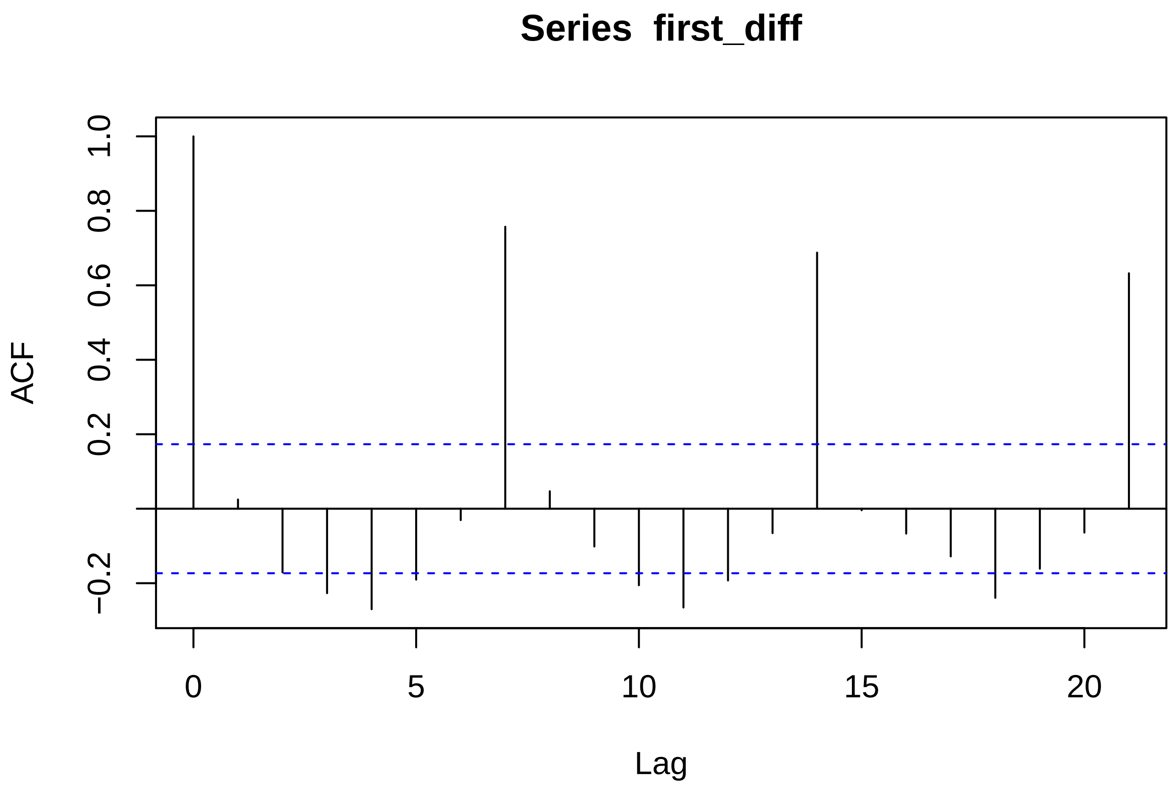


```
PACF = data.frame(correlation = PACF$acf) %>%
  mutate(n= row_number()) %>%
  filter(abs(correlation) < 0.1)
```

3.2.3 the *q* parameter

The *q* parameter, or the number of lagged errors required to build the MA part of the model, can be picked by looking at the *autocorrelation function (ACF)*.
Again, the k values for which the correlations are below the confidence interval may be good candidates as *q* numbers to be used in ARIMA(p,d,q). We save the q candidates in the ACF data frame (see code below).

```
ACF = acf(first_diff)
```



```
ACF = data.frame(correlation = ACF$acf) %>%
  mutate(n= row_number()) %>%
  filter(abs(correlation) < 0.1)
```

3.3 Fitting the best ARIMA model

We now need to loop through all the p,q combinations (we have already set d = 1) to get the best ARIMA regression.
We use the *sqldf* library to write a *cross join* query to get p,q combinations:

```
PQ <- sqldf('SELECT PACF.n, ACF.n FROM  PACF CROSS JOIN ACF') %>%
     setNames(c("P","Q"))
```

The below *while-loop* examines all the p-q combinations from the PQ dataset to build an ARIMA(p,d,q) model and provide the relative AIC value.
AIC stands for Akaike Information Criterion and it is used to perform a multi-objetcive optimization.
In fact, we need:

1. to find the optimal set of p,q parameters to minimize the error between the ARIMA and the test set
2. at the same time, to limit over-training.

The lowest AIC value provides the optimal p-q values respecting both conditions 1. and 2. For some p-q combinations the ARIMA() function may raise an error due to non-stationarity: the try-catch construct allows the while-loop to continue even when any error is raised.

```r
temp = data.frame()
i = 1
PREV = -1
while (i < length(PQ$P)){

    tryCatch({
      model <- arima(daily_doses, order = c(PQ$P[i],1,PQ$Q[i]))

    }, warning = function(w) {
    }, error = function(e) {
    }, finally = {})

  AIC_VALUE <- ifelse(PREV == model$aic,1000000,model$aic)
  temp <- bind_rows(temp,data.frame(AIC_VALUE ,PQ$P[i] , PQ$Q[i]))

  PREV = model$aic

  i = i + 1

}

temp <- temp %>% setNames(c("AIC","P","Q")) %>%  filter(AIC < 10000) %>%
        mutate(n = row_number())

p <-temp$P[which.min(temp$AIC)]
q <-temp$Q[which.min(temp$AIC)]
```

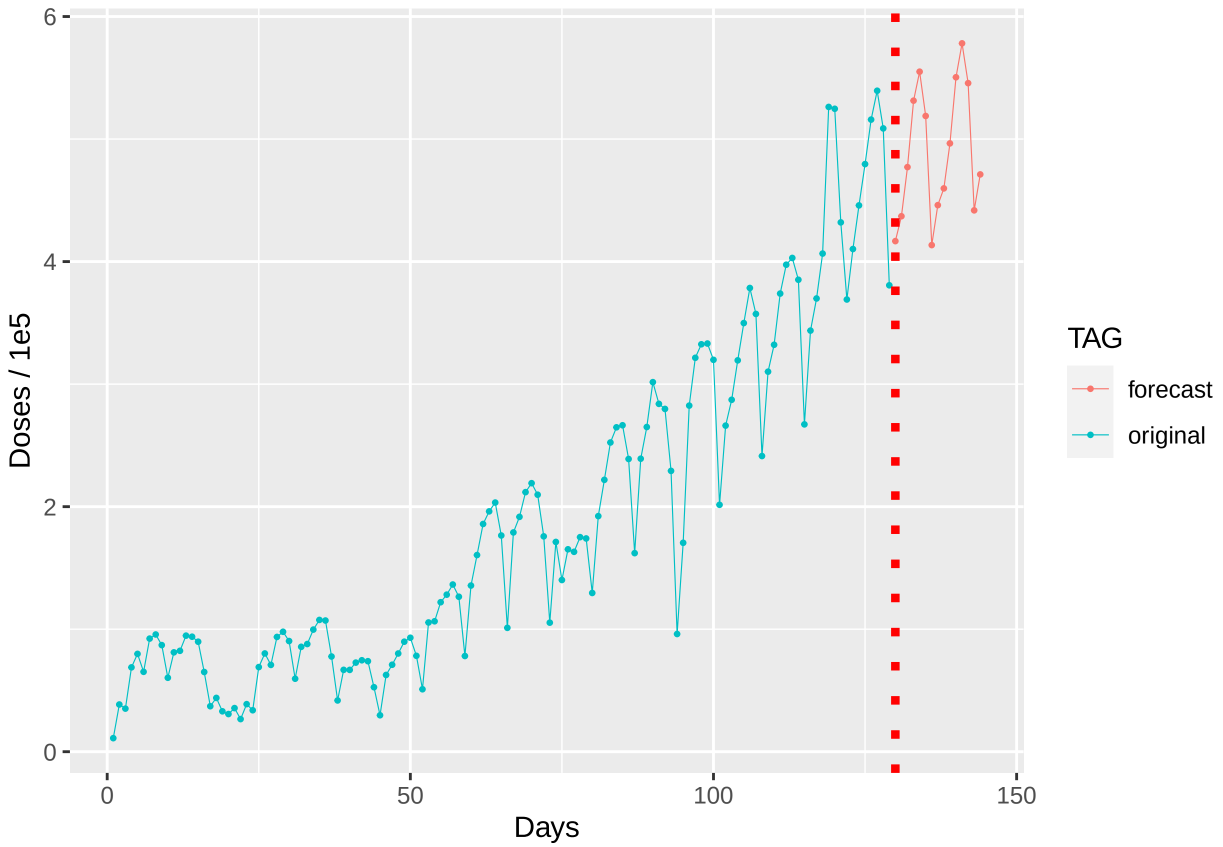
The lowest AIC is achieved when $p = 9$ and $q = 9$:

BEST_AIC	p	q
2954.925	9	9

We can finally build the optimal model:

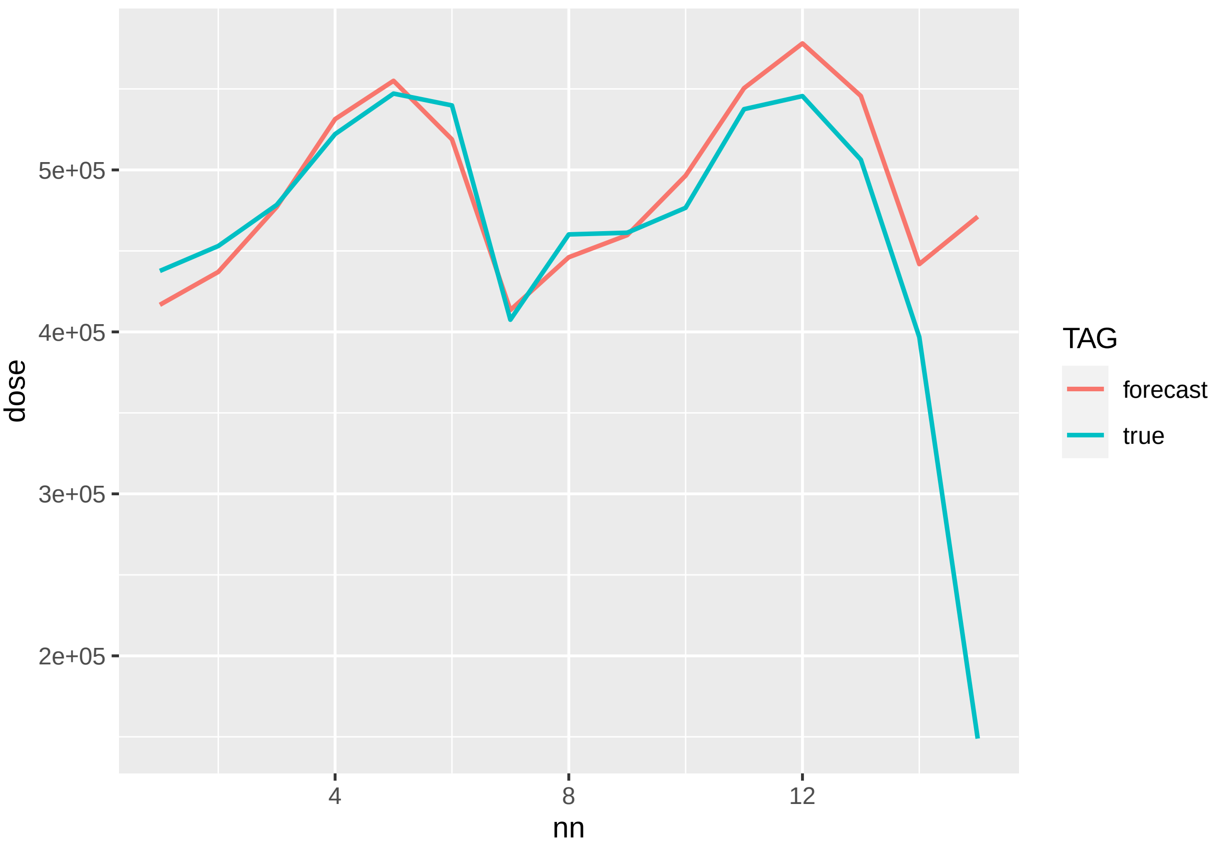
```r
p <-temp$P[which.min(temp$AIC)]
q <-temp$Q[which.min(temp$AIC)]

model <- arima(daily_doses, order = c(p,1,q))
```

The plot below shows the original series and the forecast, separated by a red dashed v-line (y axis unit is hundreds of thousands doses per day):



4 Forecasting and Evaluation

We plot the true daily doses from the test set together with the forecast to see how well the model fits the original series:



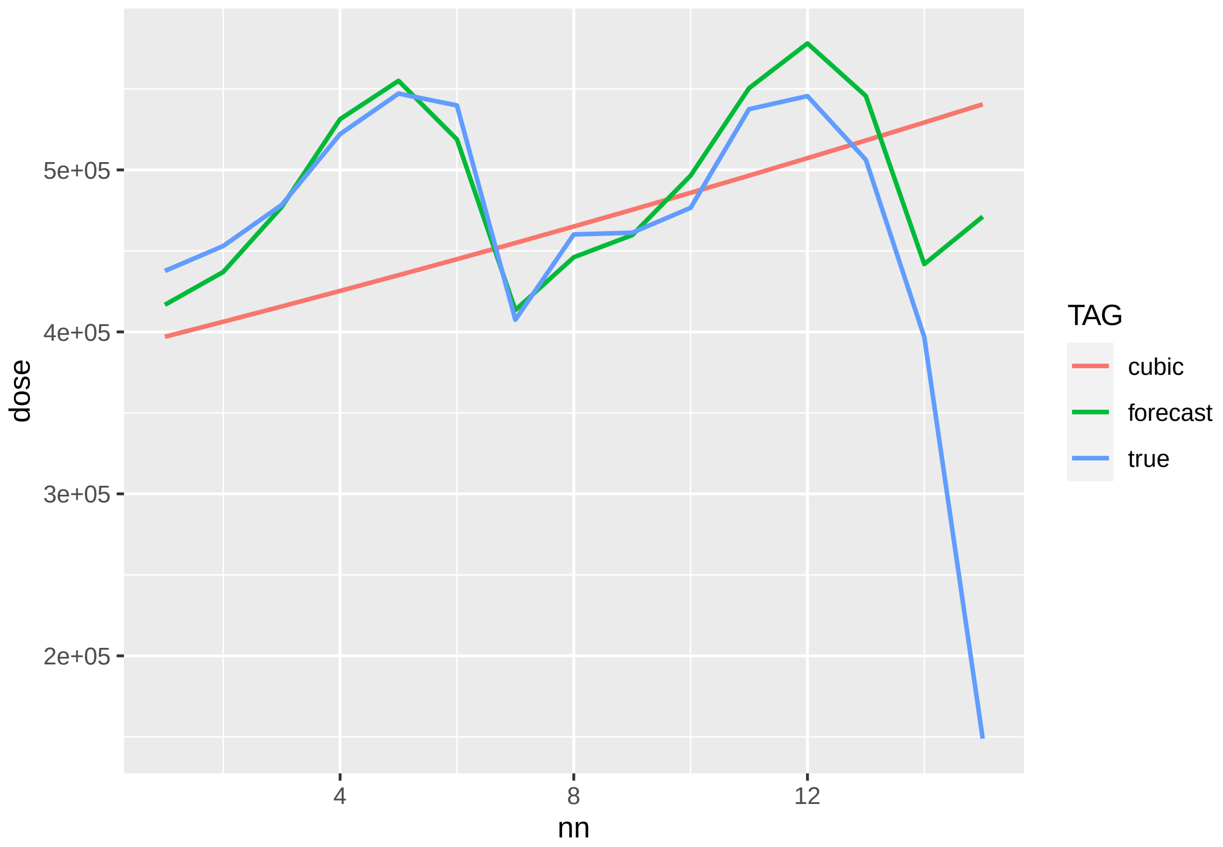
To express a measure for the fitness of the model we use *Mean average percentage error (MAPE)*, from the MLmetrics library:

Method	MAPE
ARIMA(9,1,9)	0.0792449

We compare this result with a more simple polynomial (cubic) regression. In other words, we express the time series as:

$Y = an^3 + bn^2 + cn + d$

and use *lm()* to find the optimal residuals a, b, c, d. The plot below draws the original series, together with the ARIMA forecast and the cubic regression:



Again, we use *MAPE* the measure the fitness of the model. We note that ARIMA provides a significantly better result (8% against 16%):

Method	MAPE
ARIMA(9,1,9)	0.0792449
Cubic polynomial	0.1591014

5 Conclusion

Vaccination forecasting may help Italian central and local authorities to further improve vaccine distribution. A further improvement may be achieved by increasing the level of granularity from the features that, for the sake of simplicity, have been initially discarded. For example, the forecast may be subdivided by considering how the vaccine distribution is evolving among the different Italian regions, or how the vaccines are distributed among the age-groups or, finally, how the distribution depends on the four different vaccine suppliers (Pfizer-Biontech, Janssen, Astrazeneca and Moderna).