Employment Rate Analysis in Italy

Lorenzo Saracino - Biagio Buono - Leonardo Puricelli

2024-02-19

Contents

Introduction

This work deals with the analysis of the employment rate in Italy spanning from January 2004 to December 2023. The dataset used for this analysis is sourced from the Italian National Institute of Statistics (ISTAT) and is accessible *here*.

The dataset includes 240 observations, each corresponding to a monthly measurement of the employment rate. The temporal scope of the data allows for a detailed exploration of employment trends, capturing fluctuations, patterns, and potential influencing factors over the last two-decade period. The data provided by ISTAT are a valuable resource for understanding the dynamics of the Italian labor market.

As we delve into the analysis, we will explore various aspects of the employment rate, uncovering insights that may contribute to a better understanding of the economic landscape during the specified timeframe.

```
y = read_excel("/Users/biagiobuono/Documents/Time series analysis/Project/employment.xlsx")
y = y$'Tasso di occupazione'
summary(y)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   54.76   56.35   57.46   57.57   58.54   61.94
```

The employment rate during the treated period ranges between 54.76 and 61.94, and it has a mean equal to 57.57.

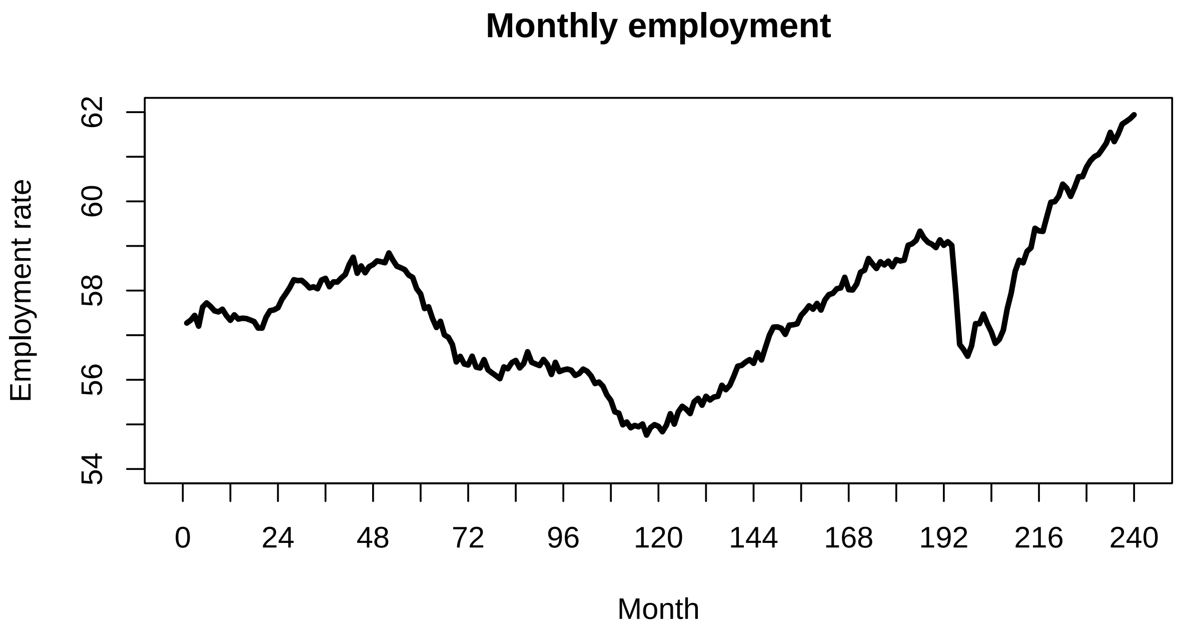


Figure 1: Graphical representation of the time series.

The plot of the time series shows non-stationarity since the data do not exhibit a constant mean over time.

We can obtain a zero-mean process, that now varies in the interval (-3, 4.5), and allows us to identify patterns, trends and outliers more effectively.

```
# Mean-adjusted data
z = y - mean(y)
n = length(z)
range(z)
```

```
## [1] -2.809925  4.370745
```

The stationarity of a time series can be assessed through the Augmented Dickey-Fuller test: the more negative the statistic is, the stronger the rejection of the hypothesis that the series is non stationary.

```
adf.test(z)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  z
## Dickey-Fuller = -0.54199, Lag order = 6, p-value = 0.9794
## alternative hypothesis: stationary
```

With a p-value of 0.9794, we fail to reject the null hypothesis, confirming our earlier suggestion that the series is non-stationary. When dealing with a non-stationary series, it's essential to transform it to achieve stationarity before continuing with analysis. A common approach is differencing, which involves subtracting each observation from its previous one. The first-order differencing ($\nabla Z_t = Z_t - Z_{t-1}$) makes the series less susceptible to long-term variations, focusing instead on short-term fluctuations. If the series is not stationary yet, higher-order differencing may be required, since this operation can be applied iteratively until the resulting series is stationary.

```
diff.z=rep(NA,times=n)

for (t in 2:n) {
  diff.z[t]=z[t]-z[t-1]
}
```

Figure 2 shows that the data now vary around a constant zero mean, except for observations corresponding to the COVID-19 period, which exhibits exceptional and unexpected changes. The ADF test also produces a p-value that is smaller than the threshold, suggesting that the series follows a stationary behavior.

```
adf.test(diff.z[2:n])
```

```
## Warning in adf.test(diff.z[2:n]): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff.z[2:n]
## Dickey-Fuller = -4.3239, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

Model Specification

Model specification is the first fundamental step to analyze a time series. It consists in defining the structure of a model, that may be appropriate to understand the temporal pattern of a given observed series, and determining its optimal order.

After identifying significant negative autocorrelation at lag 12 in *Figure 3*, we will start modeling the data with an AutoRegressive model of order 12. Subsequently, we will conduct a thorough evaluation, comparing the performance of this model against alternative options to validate our choice and assess its predictive capabilities.

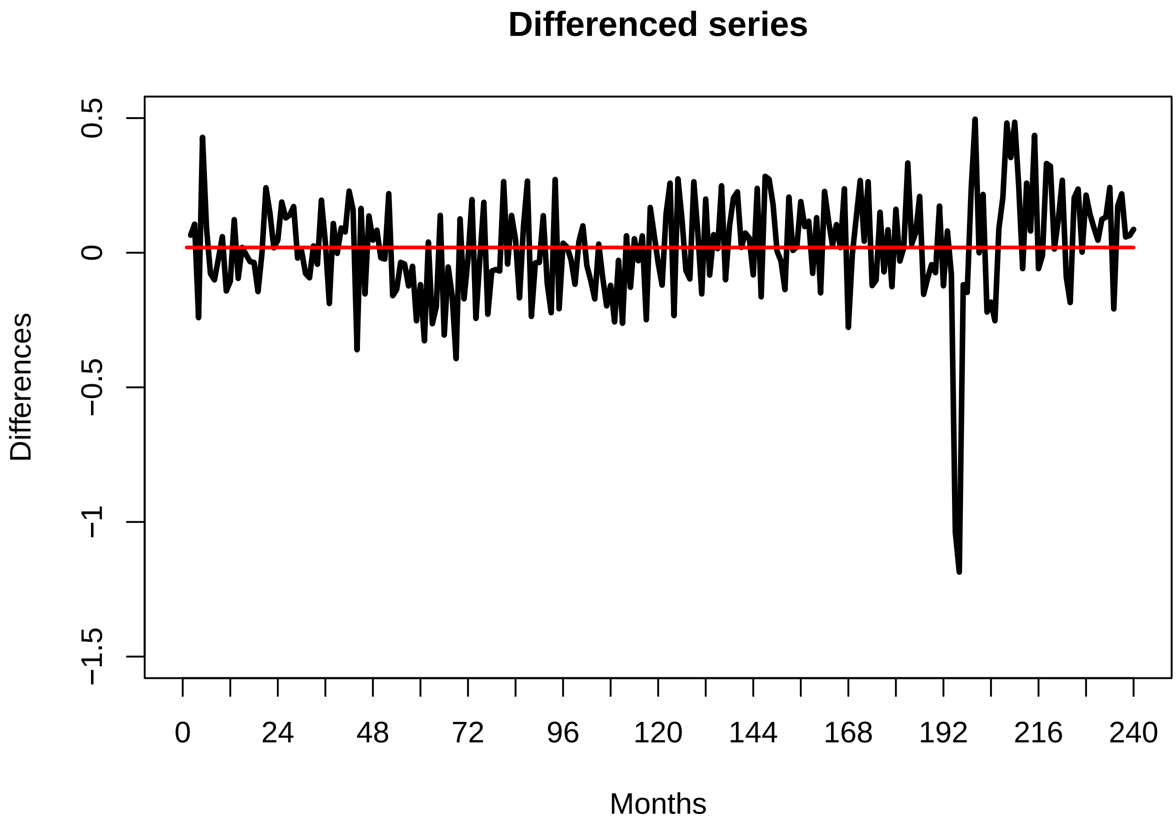


Figure 2: Graphical representation of the differenced time series.

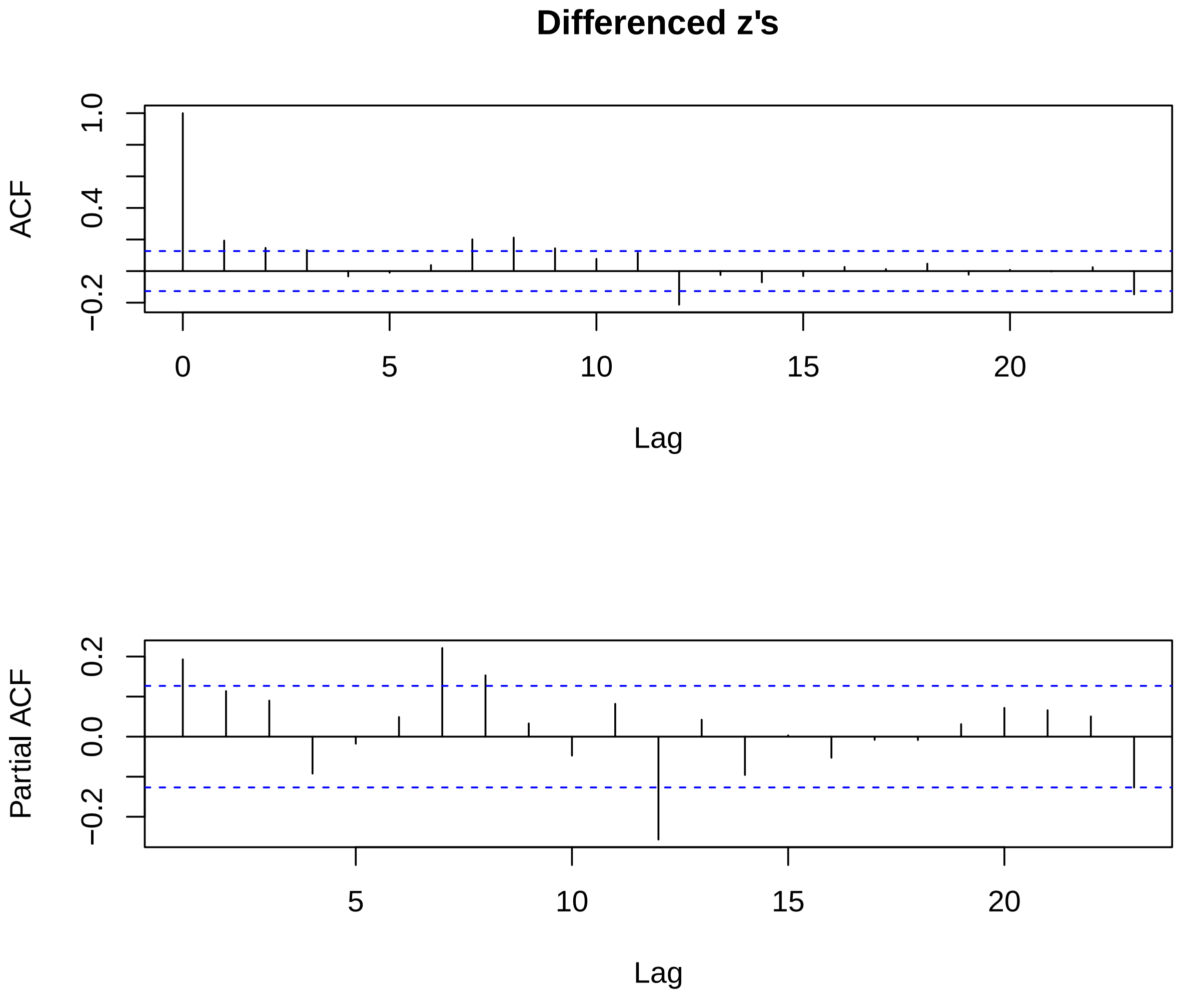


Figure 3: Top: ACF plot. Bottom: PACF plot.

Model Fitting

Since the residuals are not normally distributed, as we will observe later in the discussion, the unknown parameters are estimated through the Conditional Sum of Squares method. It aims to minimize the sum of the squared differences between observed and predicted values, conditioned on the previous observed value. The estimates of the coefficients and the variance are the following:

```
ARI121 = arima(z, order = c(12, 1, 0), include.mean = FALSE, method = "CSS")
```

	Parameter	Value
ar1	est.phi1	0.1642666
ar2	est.phi2	0.1096950
ar3	est.phi3	0.1337712
ar4	est.phi4	-0.0898751
ar5	est.phi5	-0.0102343
ar6	est.phi6	0.0032878
ar7	est.phi7	0.2004366
ar8	est.phi8	0.0984919
ar9	est.phi9	0.0628662
ar10	est.phi10	-0.0279382
ar11	est.phi11	0.1249927
ar12	est.phi12	-0.2613164
1	est.sigma2	0.0307131

The estimated parameters are used to derive the fitted values, which are then employed for further analysis.

First of all, we fit the model to the differenced time series, obtaining the following results:

```
diff.z.fit=rep(NA,times=n)

for (t in 13:n) {
  diff.z.fit[t]= est.phi1*diff.z[t-1] + est.phi2*diff.z[t-2] +
                 est.phi3*diff.z[t-3] + est.phi4*diff.z[t-4] +
                 est.phi5*diff.z[t-5] + est.phi6*diff.z[t-6] +
                 est.phi7*diff.z[t-7] + est.phi8*diff.z[t-8] +
                 est.phi9*diff.z[t-9] + est.phi10*diff.z[t-10] +
                 est.phi11*diff.z[t-11] + est.phi12*diff.z[t-12]
}
```

Nevertheless, in order to observe the model's behavior directly on the starting series, we exploit the rationale of the differencing operation to derive fitted values for the z's:

$$\nabla Z_t = \hat{\phi}1\nabla Z_{t-1} + \hat{\phi}2\nabla Z_{t-2} + \hat{\phi}3\nabla Z_{t-3} + \hat{\phi}4\nabla Z_{t-4} + \hat{\phi}5\nabla Z_{t-5} + \hat{\phi}6\nabla Z_{t-6} +$$

$$\hat{\phi}_7\nabla Z_{t-7} + \hat{\phi}_8\nabla Z_{t-8} + \hat{\phi}_9\nabla Z_{t-9} + \hat{\phi}_{10}\nabla Z_{t-10} + \hat{\phi}_{11}\nabla Z_{t-11} + \hat{\phi}_{12}\nabla Z_{t-12}$$

that can be rewritten as

$$Z_t - Z_{t-1} = \phi1(Z_{t-1} - Z_{t-2}) + \phi2(Z_{t-2} - Z_{t-3}) + \phi3(Z_{t-3} - Z_{t-4}) + \phi4(Z_{t-4} - Z_{t-5}) + \phi5(Z_{t-5} - Z_{t-6}) +$$

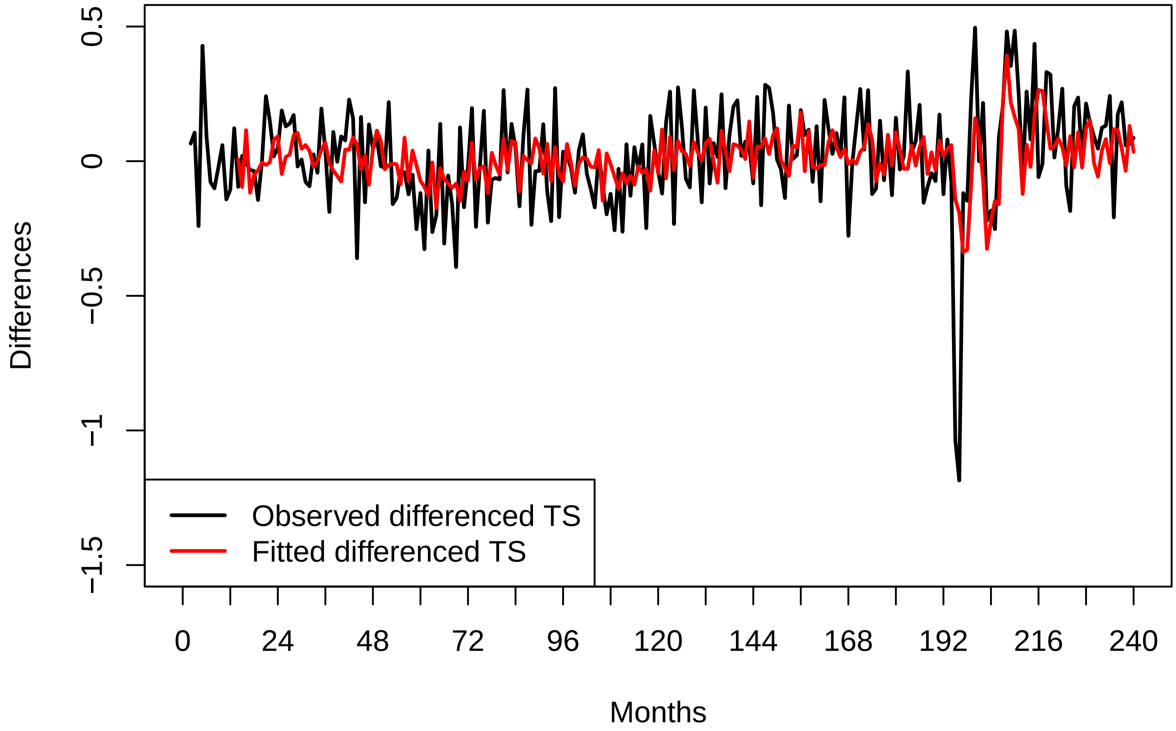


Figure 4: Observed differenced time series vs Fitted differenced time series

$$\phi_6(Z_{t-6} - Z_{t-7}) + \phi_7(Z_{t-7} - Z_{t-8}) + \phi_8(Z_{t-8} - Z_{t-9}) + \phi_9(Z_{t-9} - Z_{t-10}) + \phi_{10}(Z_{t-10} - Z_{t-11}) +$$

$$\phi_{11}(Z_{t-11} - Z_{t-12}) + \phi_{12}(Z_{t-12} - Z_{t-13})$$

from which we can derive the next formula.

```
z.fit = rep(NA, times = n)

for (t in 14:(n+1)) {
  z.fit[t]=z[t-1]*(1+est.phi1) + z[t-2]*(est.phi2-est.phi1) +
           z[t-3]*(est.phi3-est.phi2)+ z[t-4]*(est.phi4-est.phi3) +
           z[t-5]*(est.phi5-est.phi4) + z[t-6]*(est.phi6-est.phi5)+
           z[t-7]*(est.phi7-est.phi6) + z[t-8]*(est.phi8-est.phi7) +
           z[t-9]*(est.phi9-est.phi8)+ z[t-10]*(est.phi10-est.phi9) +
           z[t-11]*(est.phi11-est.phi10) + z[t-12]*(est.phi12-est.phi11) -
           z[t-13]*est.phi12
}
```

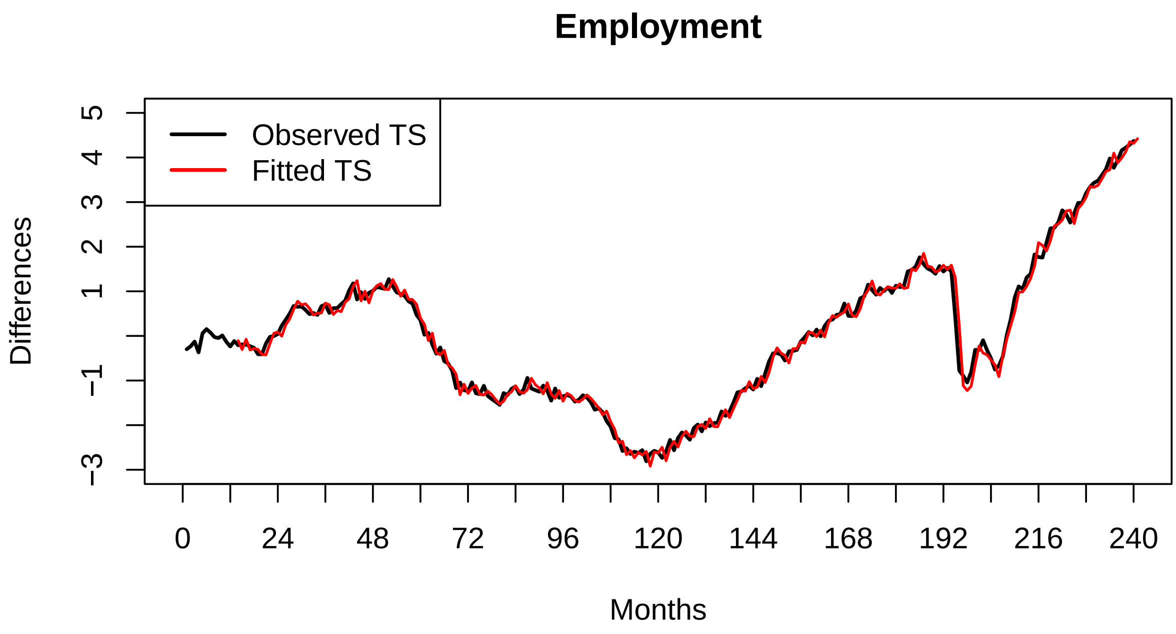


Figure 5: Observed time series vs Fitted time series.

We can obtain an initial assessment of the model's performance by analyzing the plots above.

Figure 5 shows that fitted values seem to resemble well the general trend of the data with some slight divergences.

Figure 6 instead represents a scatterplot of the observed values from the series and those obtained with the chosen model. Generally, if the points of observed values and those of fitted values follow the diagonal line, that passes through the origin and has a slope of 1, it indicates that the model adequately captures the data trend, as the plot actually suggests.

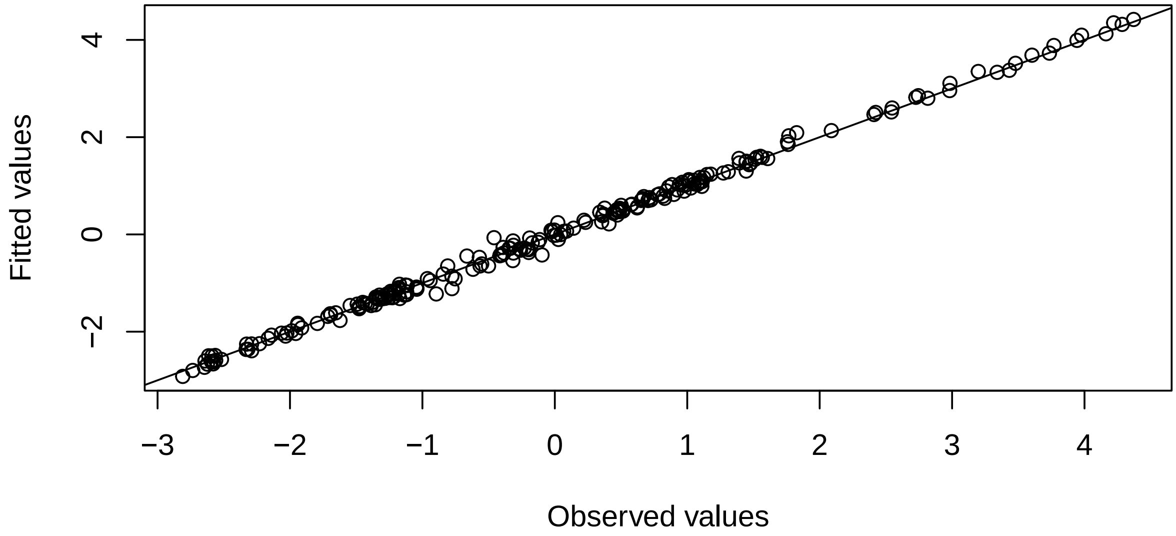


Figure 6: Observed values vs Fitted values.

Model Comparison

Model comparison consists in selecting the most suitable model that effectively captures the temporal dependencies and patterns in the data. We use different methods to identify it, that are the Akaike Information Criterion (AIC), the Bayesian Information Criterion (BIC), the Root Mean Squared Error (RMSE) and the Mean Absolute Error (MAE). The model that minimizes the most statistics is then considered as the most favorable choice.

```
orders = list(c(12,1,0), c(12,1,3), c(12,1,8), c(7,1,8), c(7,1,12), c(12,1,12))

aic_values = numeric(length = length(orders))
bic_values = numeric(length = length(orders))
res = list()
RMSE_values = numeric(length = length(orders))
MAE_values = numeric(length = length(orders))

for (i in seq_along(orders)) {
  model = arima(z, order = orders[[i]], include.mean = FALSE)
  res[[i]] = c(model$residuals[14:240])
}

for (i in seq_along(orders)) {
  model = arima(z, order = orders[[i]], include.mean = FALSE)
  aic_values[i] = AIC(model)
  bic_values[i] = BIC(model)
  RMSE_values[i] = sqrt(mean(res[[i]]^2))
  MAE_values[i] = mean(abs(res[[i]]))
}
```

```
model_data = data.frame(Order = sapply(orders,
            function(x) paste0("(", paste(x, collapse = ","), ")")), AIC = aic_values, BIC = bic_values
kable(model_data, caption = "Comparison of different ARIMA models", align = "c")
```

Table 2: Comparison of different ARIMA models

Order	AIC	BIC	RMSE	MAE
(12,1,0)	-127.9712	-82.77715	0.1753118	0.1319322
(12,1,3)	-125.5808	-69.95738	0.1739121	0.1304446
(12,1,8)	-123.9371	-50.93138	0.1694910	0.1267061
(7,1,8)	-117.8450	-62.22161	0.1766635	0.1307416
(7,1,12)	-118.8200	-49.29078	0.1716254	0.1286066
(12,1,12)	-118.6998	-31.78823	0.1669210	0.1257671

The optimal model for our dataset is determined to be an ARIMA model with a lagged AutoRegressive component of order 12 and a differencing order of 1, whose estimated parameters are exhibited above.

```
## Best Model: ARIMA ( 12,1,0 )
```

Model Diagnostics

Model diagnostics is a crucial step before using the model to make forecasts. This phase consists in testing the goodness of fit of the model and finding appropriate alternatives or modifications if the fit is poor, by employing the analysis of the residuals.

Residuals are calculated as the difference between the actual observations and the fitted values. If the model is correctly specified and the parameters estimates are reasonably close to the true values, then the residuals should behave like independent, identically distributed normal random variables with zero mean and common standard deviation.

```
z.res = rep(0, times = n)

for (t in 14:n) {
  z.res[t] = z[t] - z.fit[t]
}

range(z.res[14:n])
```

```
## [1] -0.9939128  0.3615927
```

The following plots and tests are used to do three main important checks on the residuals.

First of all, we can observe that the residuals lie around a zero horizontal line with no specific trends, meaning that the model is adequate.

The autocorrelation plot instead shows that the residuals are not correlated and do not exhibit any temporal structure.

Normality of the residuals is assessed by analyzing the histogram of the residuals, that shows some observations that are far from the most part, and the q-q plot, suggesting some departures from normality distribution.

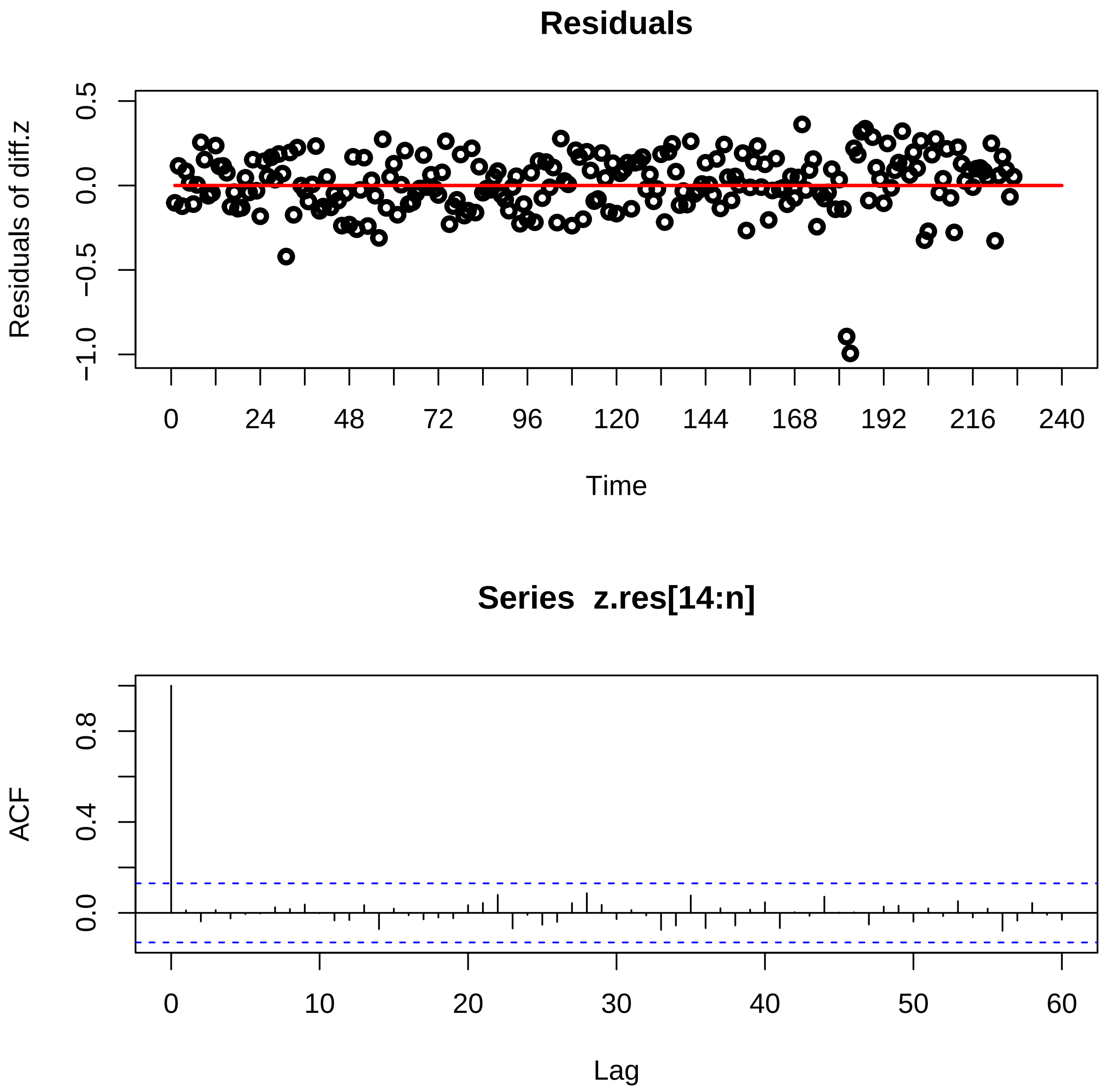


Figure 7: Top: Residuals over time. Bottom: ACF plot of the residuals.

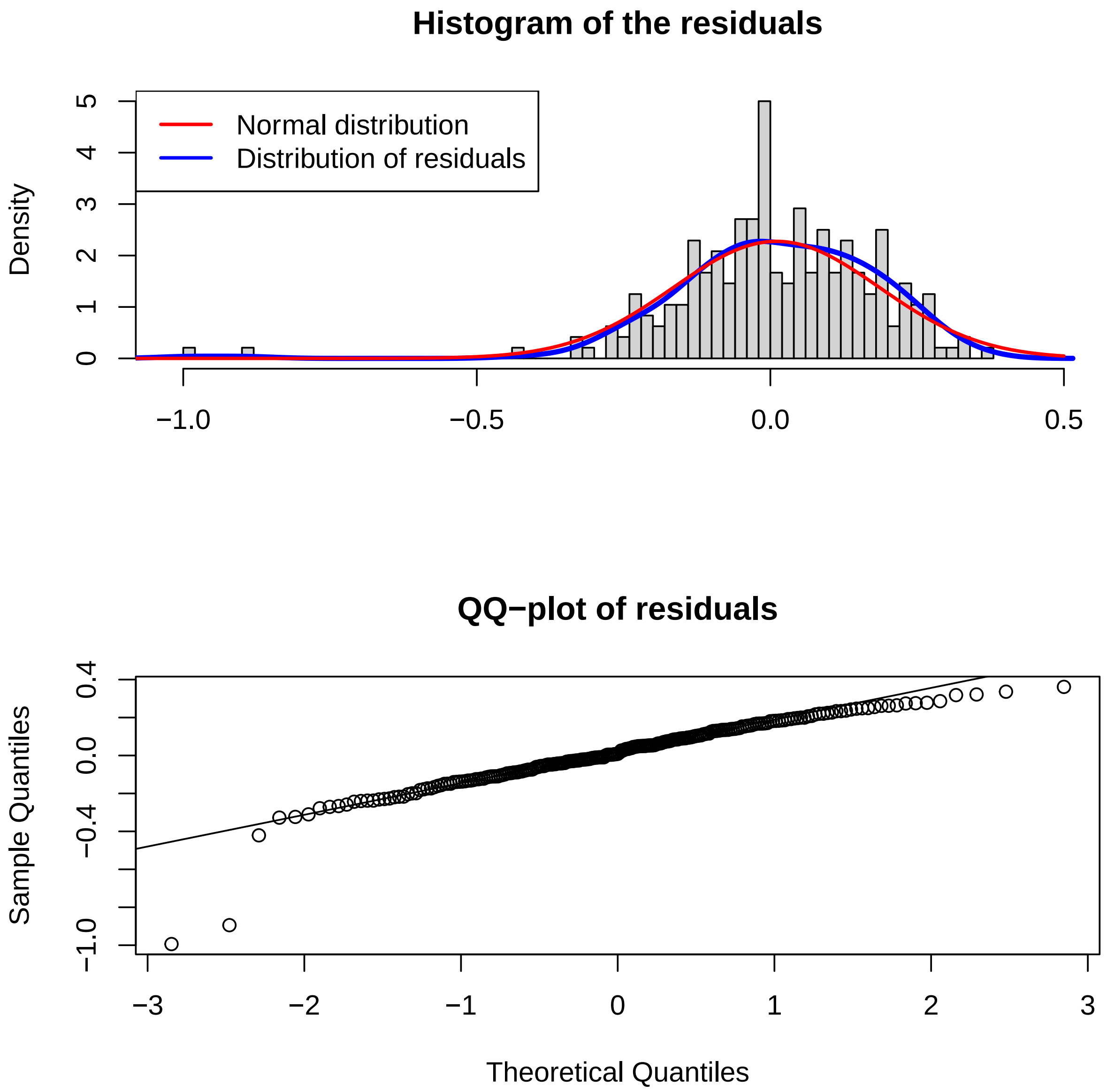


Figure 8: Top: Histogram of the residuals. Bottom: Q-Q plot of the residuals.

Moreover, two widely used statistical tests to verify the distribution of the residuals are the Shapiro-Wilk and Jarque Bera tests.

```
shapiro.test(z.res[14:n])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  z.res[14:n]
## W = 0.91648, p-value = 5.406e-10
```

```
jarque.bera.test(z.res[14:n])
```

```
##
##  Jarque Bera Test
##
## data:  z.res[14:n]
## X-squared = 452.39, df = 2, p-value < 2.2e-16
```

Both the tests show that the residuals do not follow a Normal distribution, since the associated p-values are widely smaller than the threshold.

Forecasting

One of the primary goals of building a model for a time series is to be able to forecast the values for that series at future times. Therefore, we try to predict the employment rate for the next 24 months.

```
pr = 24
nn = n + pr
z.for <- predict(ARI121,n.ahead=pr)$pred
```

In *Figure 9*, the initial time series is represented in black, while the future values are depicted in green, with a black dashed vertical line marking the separation. Over the subsequent 24 months, the forecasts indicate a stabilization of the employment rate around the most recent observations in the series.

The uncertainty of the forecasts can be evaluated by computing the confidence intervals. To do this, we need to calculate the ψ vector, that is used to determine the variance of the forecast errors. It is initialized with a length equal to the number of the forecasts and its first element is set equal to 1, while subsequent ones are computed using the AutoRegressive coefficients obtained from the ARIMA model.

Next, we compute the variance of the errors at each forecasting step, storing it in the *var.er* vector.

```
phi = ARI121$model$phi

psi = rep(NA,pr)
psi[1] = 1

for(j in 2:pr){
       candidate = 0
       for(p in 1:length(phi)){
               if(j > p){
```

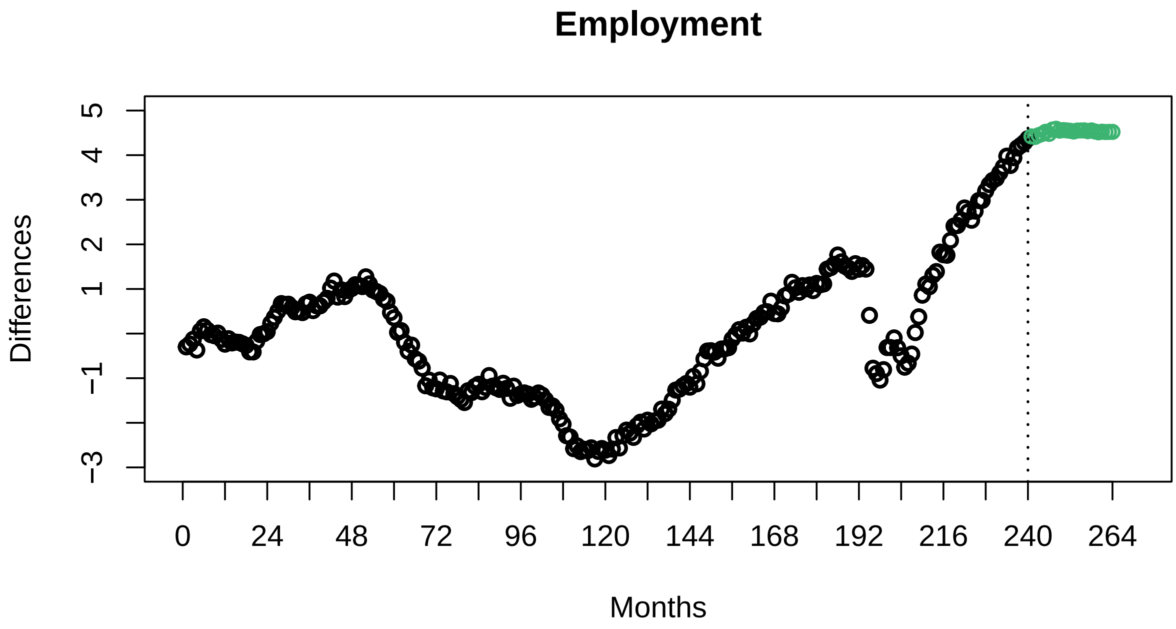


Figure 9: Forecasts of the differenced series.

```
                  candidate = candidate + psi[j-p] * phi[p]
   }
}
       psi[j]=candidate
}
var.er=rep(NA,times=pr)
var.er=rep(0,times=pr)
var.er[1]=est.sigma2

for (j in 2:pr) {
 var.er[j]=est.sigma2*(psi[1]+sum(psi[2:(j)]^2))
}
```

```
left_ci.er=rep(NA,times=pr)

for (t in 1:pr) {
        left_ci.er[t]=z.for[t]-qnorm(.975)*sqrt(var.er[t])
}

right_ci.er=rep(NA,times=pr)

for (t in 1:pr) {
       right_ci.er[t]=z.for[t]+qnorm(.975)*sqrt(var.er[t])
}

left_ci.er = c(rep(NA,240),left_ci.er)
right_ci.er = c(rep(NA,240),right_ci.er)
```

```
plot(z.for,type="p",pch=16,ylim=c(4,5),
main='Confidence interval of predictions',ylab='z')
lines(left_ci.er,type="l",col="blue",lwd=2)
lines(right_ci.er,type="l",col="blue",lwd=2)
```

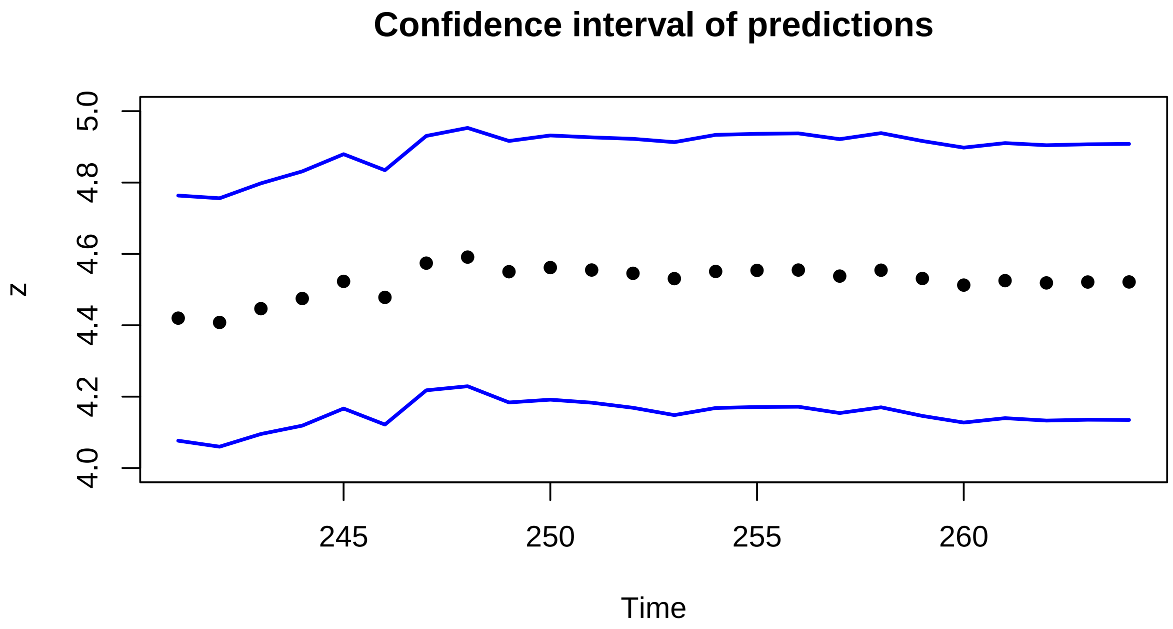


Figure 10: Forecasts confidence interval.

Figure 10 represents the 95% confidence interval, that contains the forecasted values of the time series in the range between 4.1 and 4.8.

However, we shall recall that the time series includes some observations that correspond COVID-19 period, that led to economic recessions in many countries, as a result of lockdown measures, and to widespread job losses as businesses struggled to stay afloat or were forced to close temporarily or permanently.

Therefore, since this event had an unexpected and extraordinary impact on the employment rate in many countries, including Italy, an interesting analysis can also be conducted on the series leading up to January 2020, just before the explosion of the pandemic, in order to see how the series behaves during ordinary times.

Pre COVID-19 series

The additional upcoming analysis will dive into the data from January 2004 to January 2020, allowing us to try to understand how the job market would have gone in the years 2020-2021 if the pandemic had never occured.

In the following all the steps that have been done for the entire series are now repeated for the shortened one.

```
y_precovid = y[1:192]
```

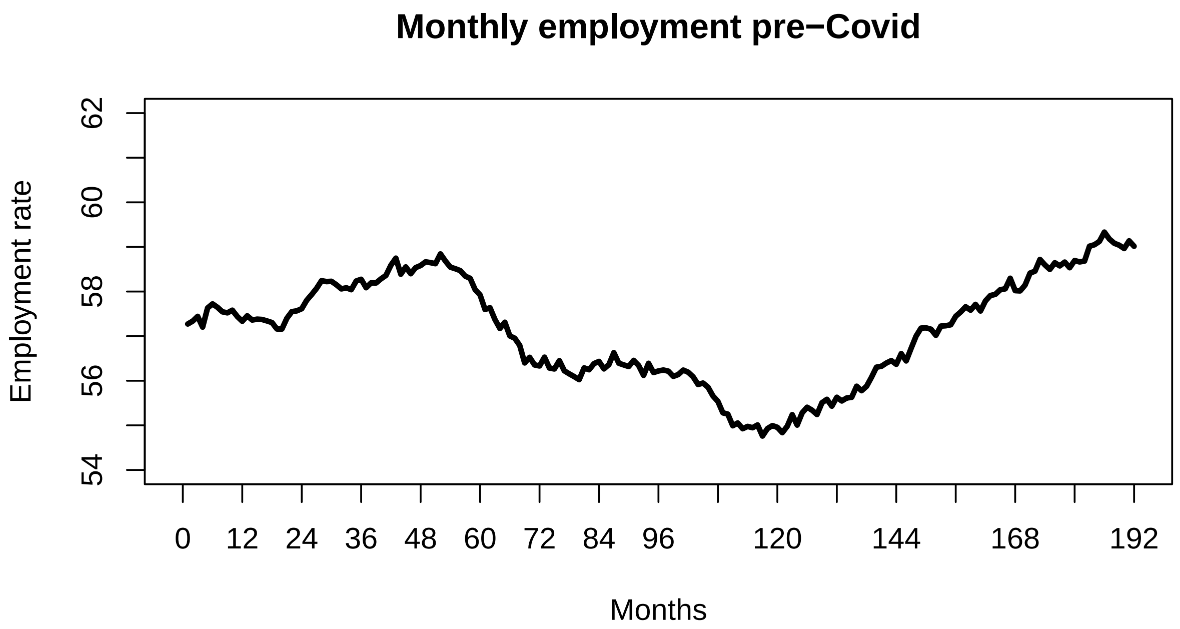


Figure 11: Graphical representation of the time series.

```
# Mean-adjusted data
z_precovid = y_precovid - mean(y_precovid)
n = length(z_precovid)
adf.test(z_precovid)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  z_precovid
## Dickey-Fuller = -0.80415, Lag order = 5, p-value = 0.9596
## alternative hypothesis: stationary
```

Looking at Dickey-Fuller test, the series remains non-stationary despite removing COVID-19 observations. Again, we make the series stationary using first order differencing.

```
diff.z_precovid=rep(NA,times=n)

for (t in 2:n) {
  diff.z_precovid[t]=z_precovid[t]-z_precovid[t-1]
}
```

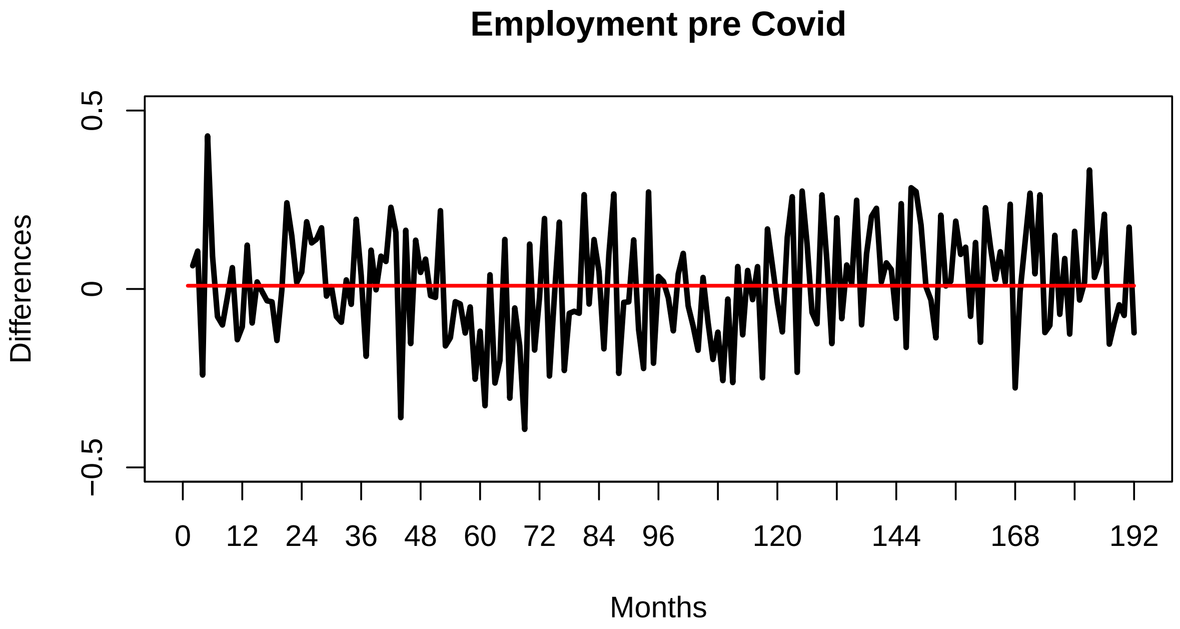


Figure 12: Graphical representation of the differenced time series.

```
adf.test(diff.z_precovid[2:n])
```

```
## Warning in adf.test(diff.z_precovid[2:n]): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff.z_precovid[2:n]
## Dickey-Fuller = -4.2411, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

Figure 12 shows that the data now vary around a constant zero mean and the ADF test produces a p-value that is smaller than the threshold, suggesting that the series follows a stationary behavior.

Model Specification

Looking at the autocorrelation plots, we can make the same considerations as before about the order of the model that will be used to analyze the data, leading us to choose an Auto Regressive model of order 12 for the differenced series.

Model Fitting

The model parameters are now estimated using the Maximum Likelihood Estimator, because, as we will observe later in the analysis, residuals follow a normal distribution, satisfying one of the key assumptions of the method. The estimates of the unknown parameters are as follows:

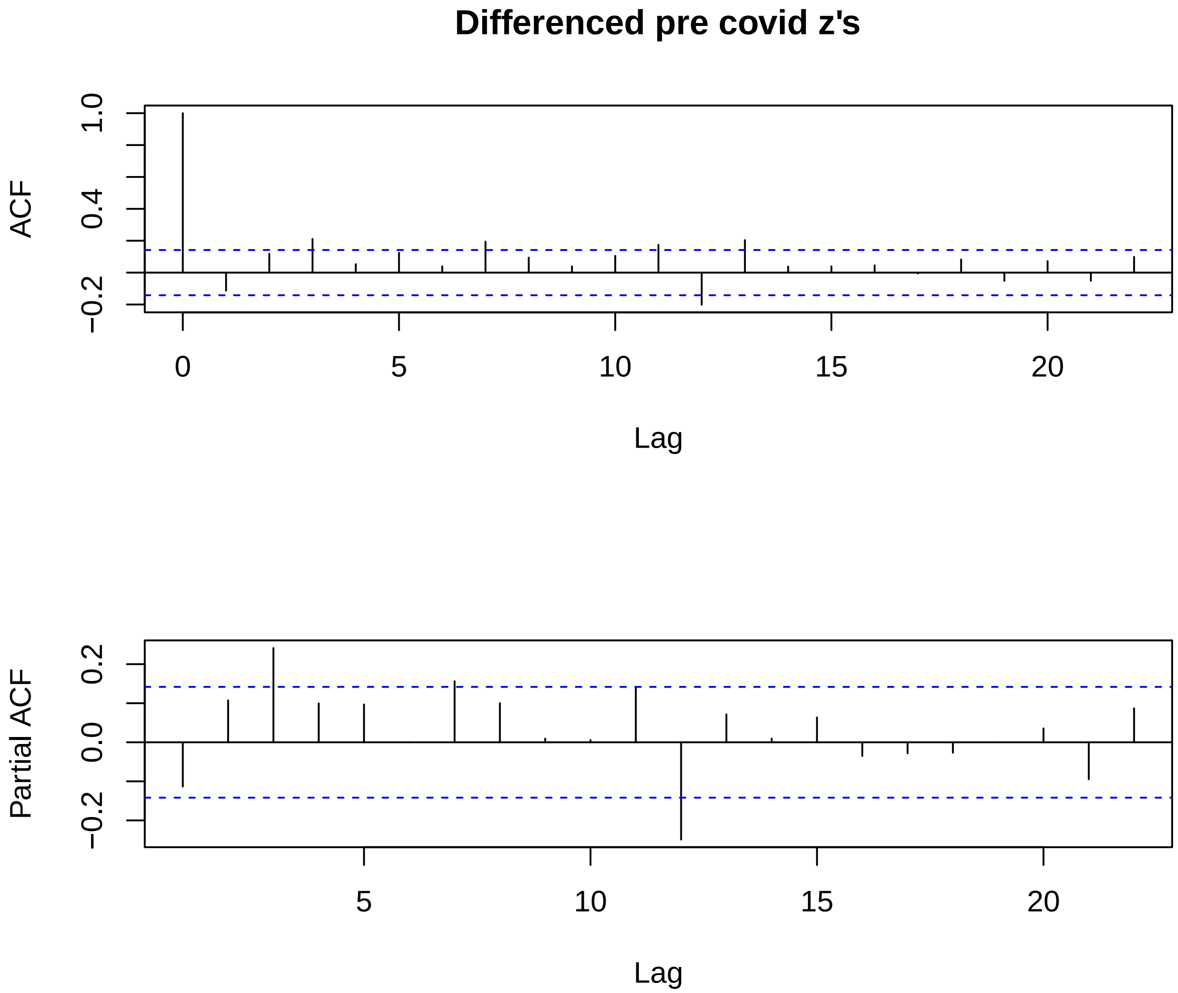


Figure 13: Top: ACF plot. Bottom: PACF plot.

```
ARI121.precovid = arima(z_precovid, order = c(12, 1, 0),
                        include.mean = FALSE, method = "ML")
```

	Parameter	Value
ar1	est.phi1	-0.1374293
ar2	est.phi2	0.0836135
ar3	est.phi3	0.2025089
ar4	est.phi4	0.0409274
ar5	est.phi5	0.0976037
ar6	est.phi6	0.0140462
ar7	est.phi7	0.1911774
ar8	est.phi8	0.0863801
ar9	est.phi9	0.0487984
ar10	est.phi10	0.0576039
ar11	est.phi11	0.1016739
ar12	est.phi12	-0.2768705
1	est.sigma2	0.0184676

Once more, we fit the model to the differenced time series and then, leveraging the rationale behind the differencing operation, we derive the fitted values for the starting series, obtaining the following results:

```
diff.z_precovid.fit=rep(NA,times=n)

for (t in 13:n) {
  diff.z_precovid.fit[t] =
    est.phi1*diff.z_precovid[t-1] + est.phi2*diff.z_precovid[t-2] +
    est.phi3*diff.z_precovid[t-3] + est.phi4*diff.z_precovid[t-4] +
    est.phi5*diff.z_precovid[t-5] + est.phi6*diff.z_precovid[t-6] +
    est.phi7*diff.z_precovid[t-7] + est.phi8*diff.z_precovid[t-8] +
    est.phi9*diff.z_precovid[t-9] + est.phi10*diff.z_precovid[t-10] +
    est.phi11*diff.z_precovid[t-11] + est.phi12*diff.z_precovid[t-12]
}
```

```
z_precovid.fit = rep(NA, times = n)

for (t in 14:(n+1)) {
  z_precovid.fit[t]=
    z_precovid[t-1]*(1+est.phi1) + z_precovid[t-2]*(est.phi2-est.phi1) +
    z_precovid[t-3]*(est.phi3-est.phi2)+ z_precovid[t-4]*(est.phi4-est.phi3) +
    z_precovid[t-5]*(est.phi5-est.phi4) + z_precovid[t-6]*(est.phi6-est.phi5)+
    z_precovid[t-7]*(est.phi7-est.phi6) + z_precovid[t-8]*(est.phi8-est.phi7) +
    z_precovid[t-9]*(est.phi9-est.phi8)+ z_precovid[t-10]*(est.phi10-est.phi9) +
    z_precovid[t-11]*(est.phi11-est.phi10) + z_precovid[t-12]*(est.phi12-est.phi11) -
    z_precovid[t-13]*est.phi12
}
```

Model's performance in predicting the observed values is aligned with the findings of the previous analysis, as highlighted by *Figure 15* and *Figure 16*. Moreover, *Figure 14*, illustrating the fitted values for the differenced series, does not exhibit now values that diverge in a consistent manner from the observed values.

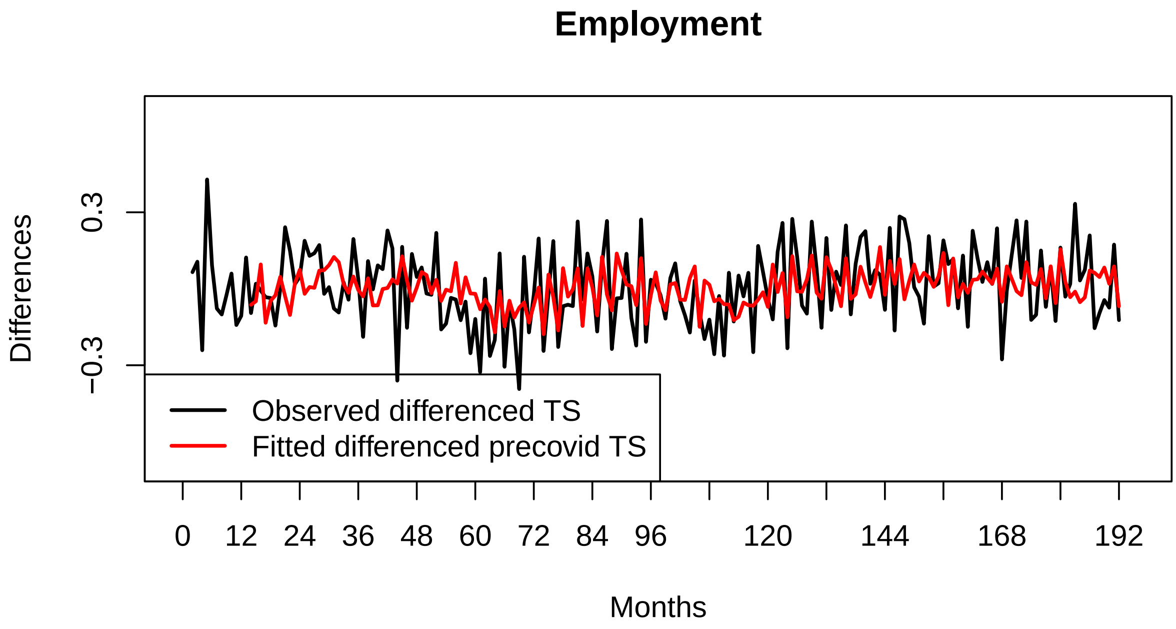


Figure 14: Observed differenced time series vs Fitted differenced time series.

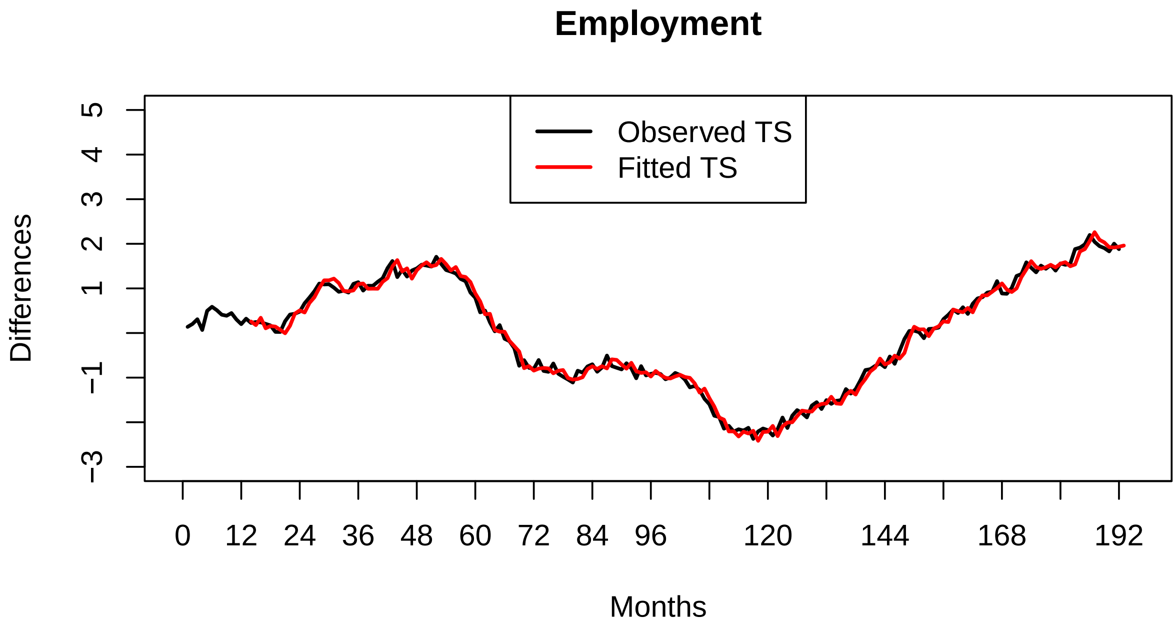


Figure 15: Observed time series vs Fitted time series.

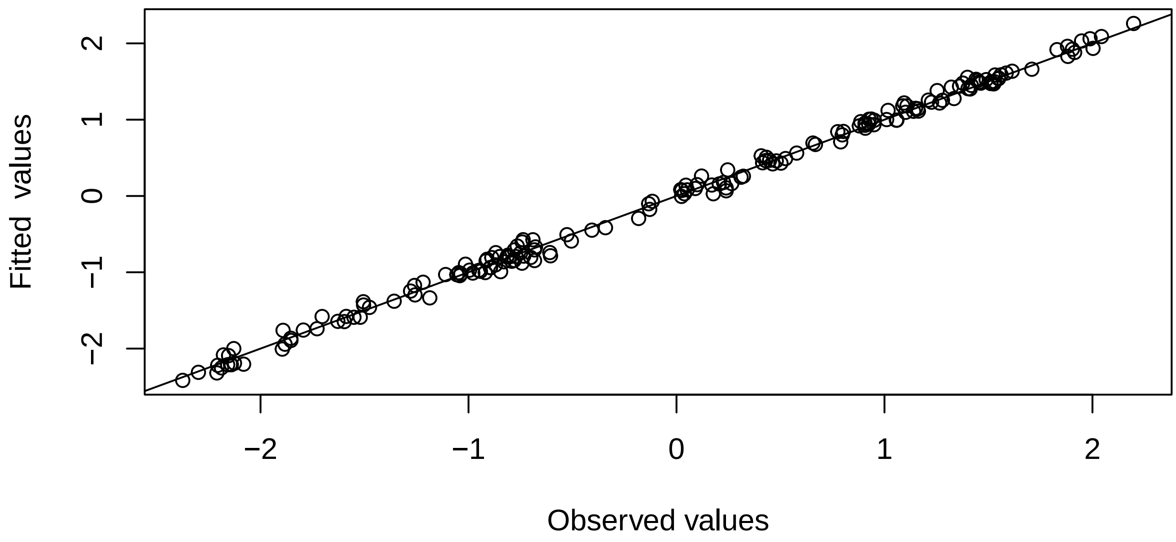


Figure 16: Observed values vs Fitted values.

Model Comparison

Again, the model that minimizes the most statistics is the ARIMA with an AutoRegressive component of order 12 and a differencing order of 1, which continues to be the optimal model even for the shortened series.

Table 4: Comparison of different ARIMA models

Order	AIC	BIC	RMSE	MAE
(12,1,0)	-183.1545	-140.94323	0.1368716	0.1125675
(12,1,3)	-181.9325	-129.98013	0.1342789	0.1110735
(12,1,8)	-181.3412	-113.15371	0.1292475	0.1058178
(7,1,8)	-173.2290	-121.27662	0.1372204	0.1118004
(7,1,12)	-175.1179	-110.17740	0.1318594	0.1076876
(12,1,12)	-176.1624	-94.98678	0.1278810	0.1038754

```
## Best Model: ARIMA ( 12,1,0 )
```

Model Diagnostics

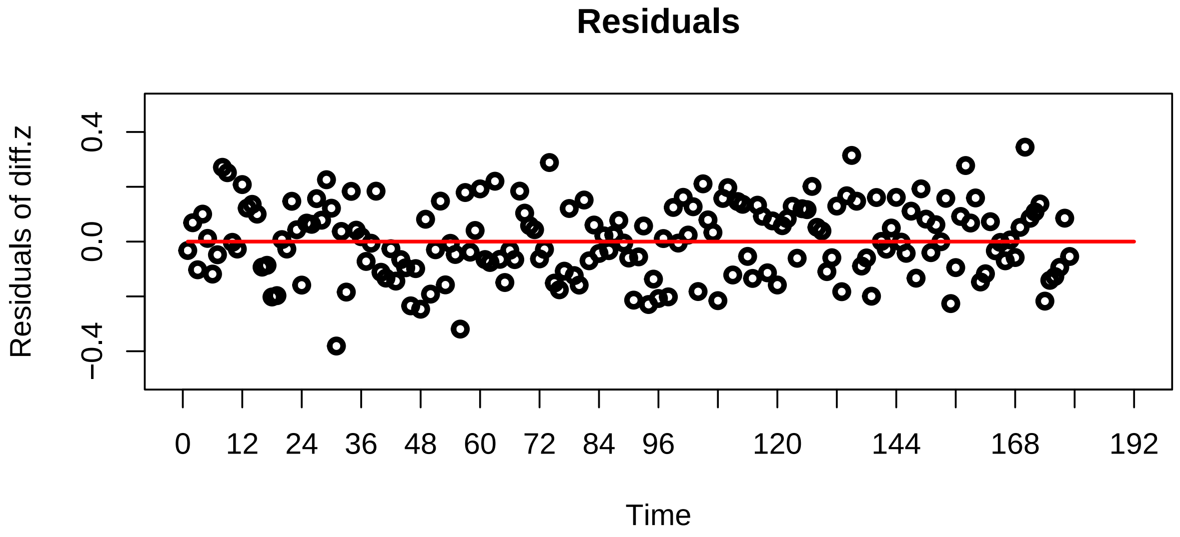
The residuals of the model remain constant around the zero horizontal line, without showing any trend or pattern. Additionally, there are no longer extreme values that deviate significantly from the main cluster of residuals.

The ACF plot shows again that the residuals have no significant correlation and do not exhibit temporal structure.

```
z_precovid.res = rep(0, times = n)

for (t in 14:n) {
  z_precovid.res[t] = z_precovid[t] - z_precovid.fit[t]
}
range(z_precovid.res[14:n])
```

```
## [1] -0.3810254  0.3445093
```



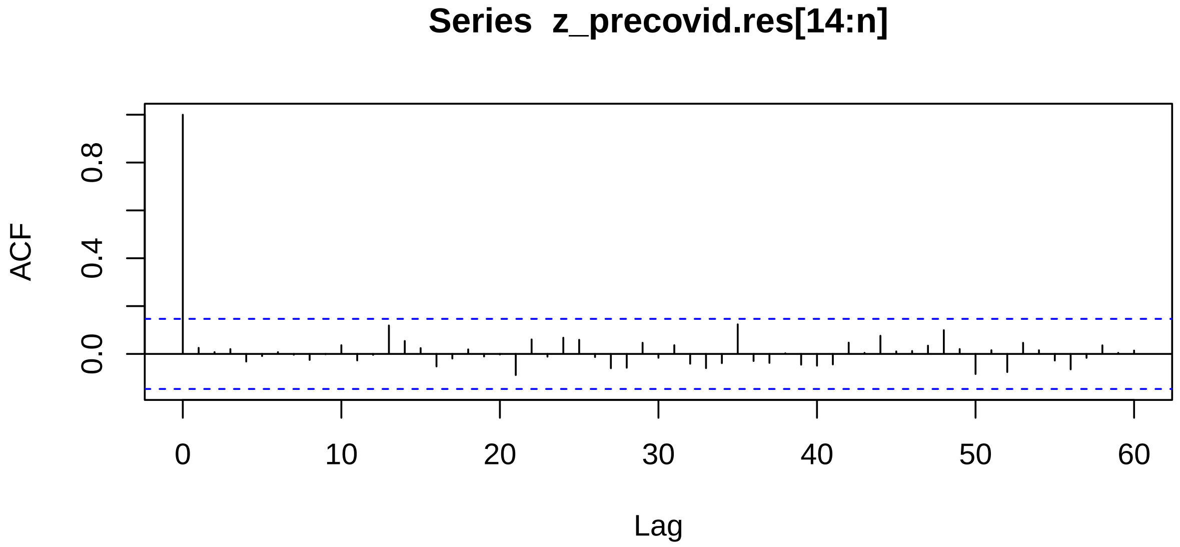


Figure 17: Top: Residuals over time. Bottom: ACF plot of the residuals.

The key distinction from the residuals of the previous model is that they now resemble quite well a normal distribution, as illustrated in the following plots.

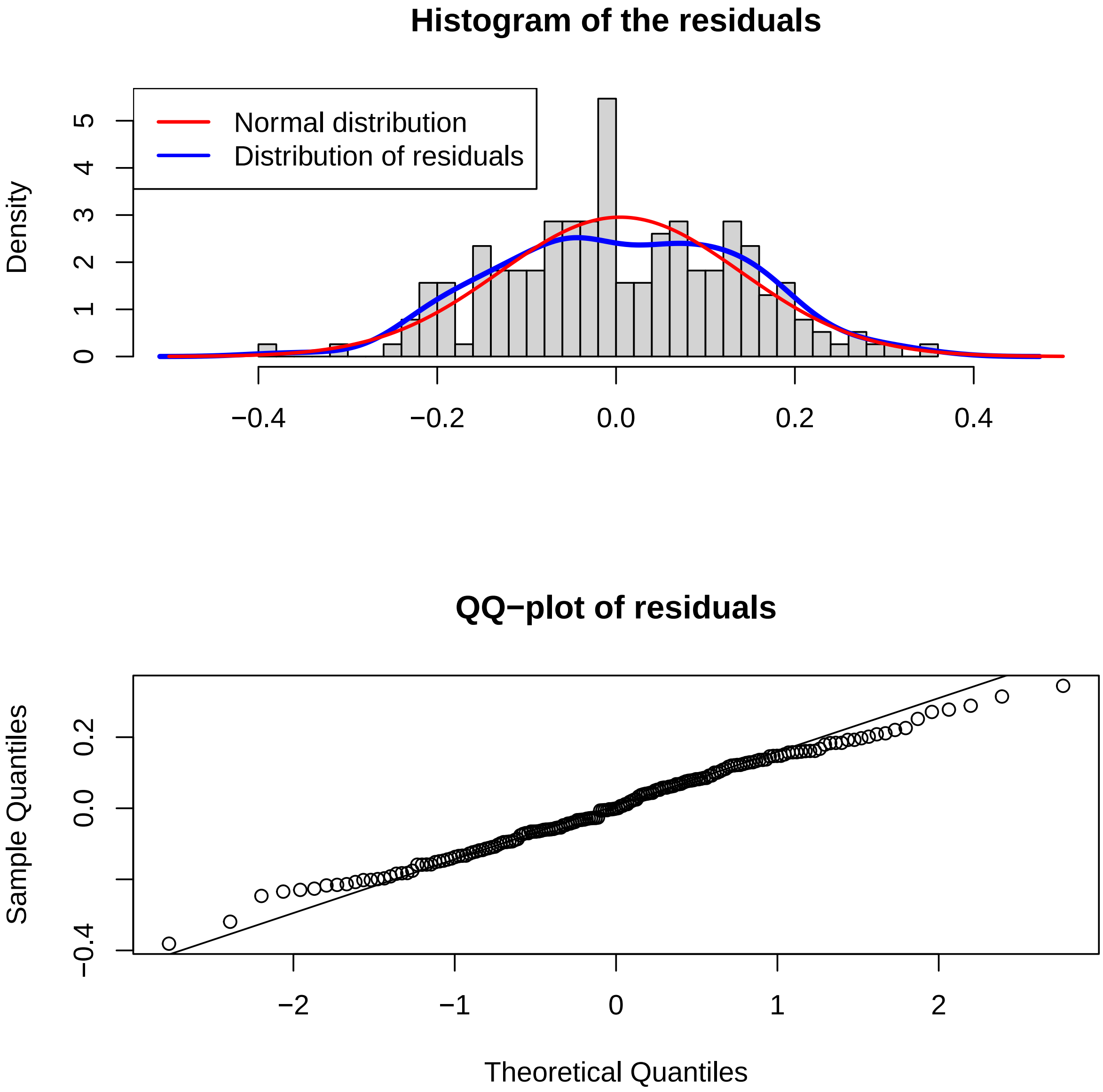


Figure 18: Top: Histogram of the residuals. Bottom: Q-Q plot of the residuals.

Normality tests further confirm the normality distribution of the residuals since the associated p-values are greater than 0.05.

```
shapiro.test(z_precovid.res[14:n])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  z_precovid.res[14:n]
## W = 0.99286, p-value = 0.5308
```

```
jarque.bera.test(z_precovid.res[14:n])
```

```
##
##  Jarque Bera Test
##
## data:  z_precovid.res[14:n]
## X-squared = 1.4985, df = 2, p-value = 0.4727
```

Forecasting

Finally, we try to predict the employment rate for the two years following 2020, in a hypothetical scenario in which COVID-19 never happened.

```
pr = 24
nn = n + pr
z_precovid.for = predict(ARI121.precovid,n.ahead=pr)$pred
```

In the *Figure 19* the forecasts closely resemble those obtained for the entire time series, differing slightly only due to a minor decline in the employment rate projected for the truncated time series. This suggests that, according to our model, the forecasted values for the years 2020-2021 significantly deviated from the actual outcomes. Such deviation was predictable, given the unforeseen and unprecedented nature of the COVID-19 pandemic, which led to an employment decline that was not foreseeable under normal circumstances.

The 95% confidence interval for the truncated series seems to be slightly narrower than the previous one, suggesting that the forecasts are computed using more reliable coefficients.

This additional analysis has provided insights into the employment rate's behavior in the absence of extraordinary events. Surprisingly, the findings reveal that the shortened series follows the same model as the whole series, except for an enhancement in the distribution of residuals, which exhibit a normal distribution when excluding the COVID-19 period, allowing us to use more reliable methods in our analysis.

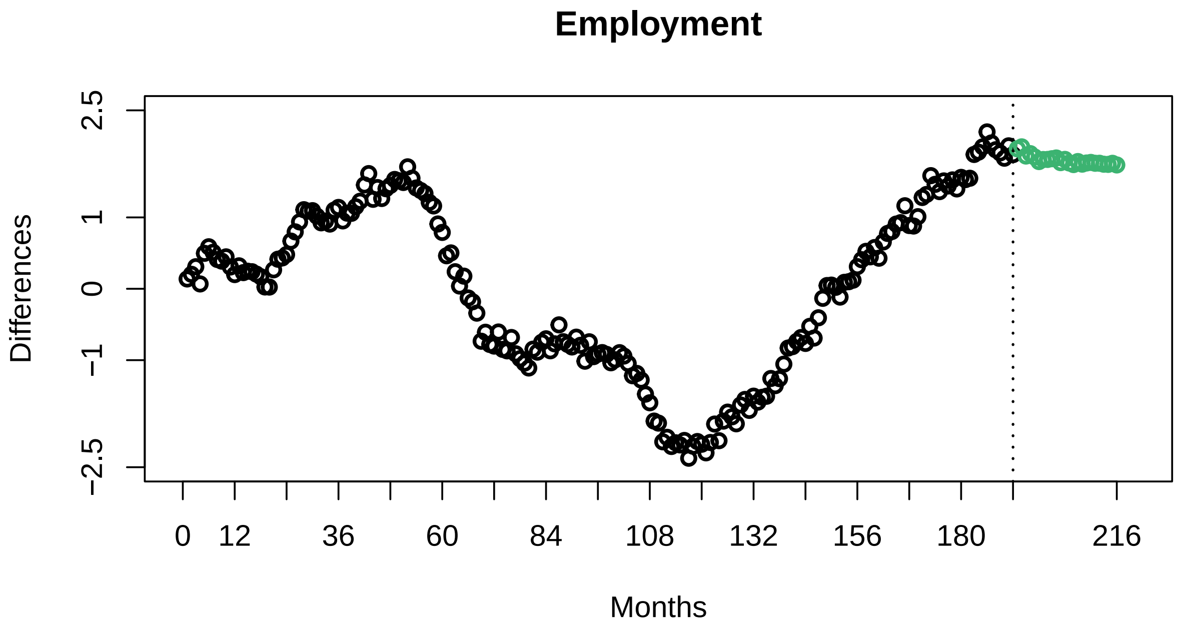


Figure 19: Forecasts of the differenced series.

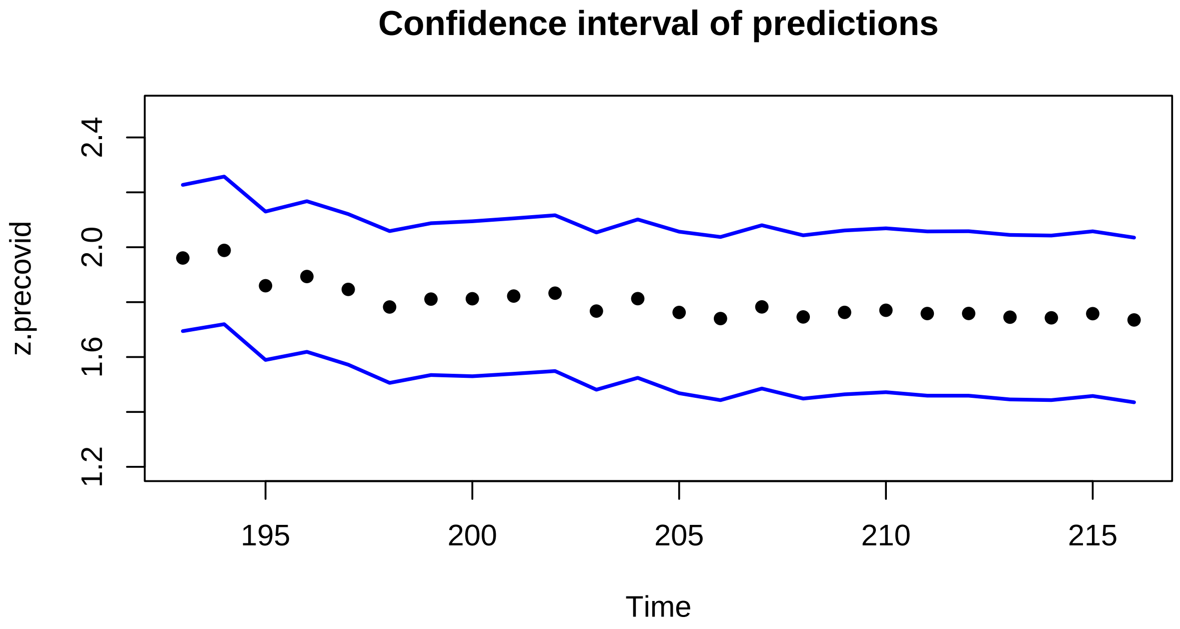


Figure 20: Forecasts confidence interval.