

Comparing Predictions of Autoregressive models

Predictions

In this short note, we compare the predictions of the random walk model with an AR(1) process. We shall do the following:

1. Compute the autocorrelation function using R.
2. Compute the RMSE of the two models and choose the best model according to the predictive performance.

To illustrate this test, we focus on the monthly U.S. inflation series computed from the CPI.

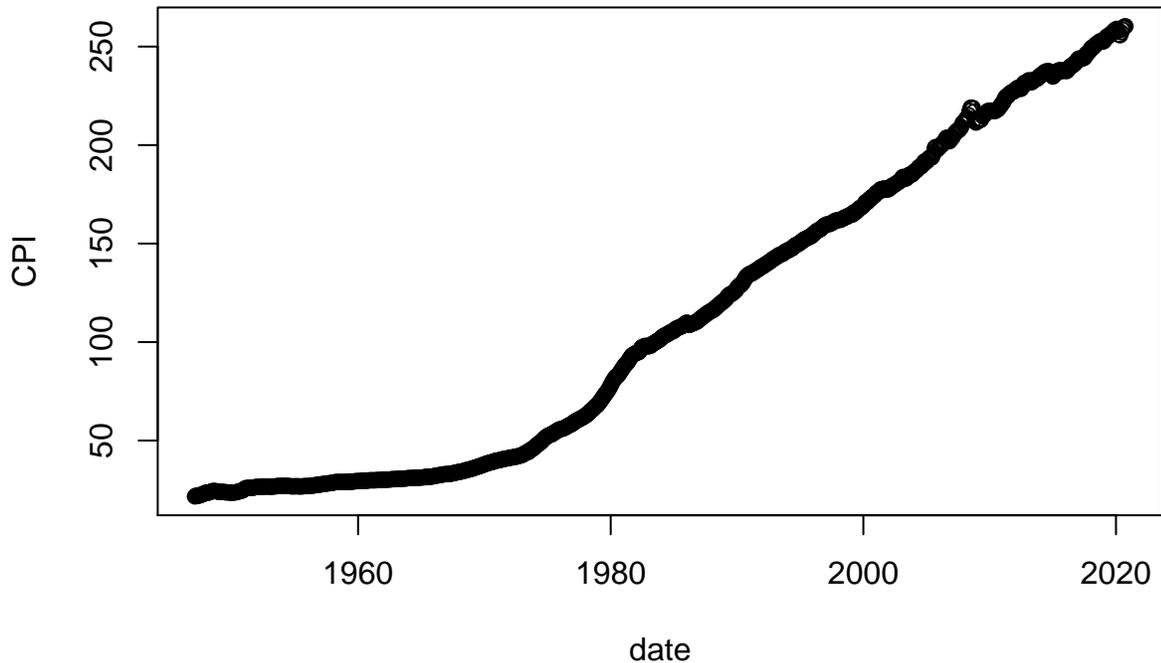
Estimation of the AR(1) model

We begin by loading the CPI data.

```
url = "https://raw.githubusercontent.com/adufays/GDP_expectancy/main/CPIAUCSL.csv"
CPI_data = read.csv(url, sep=",")
```

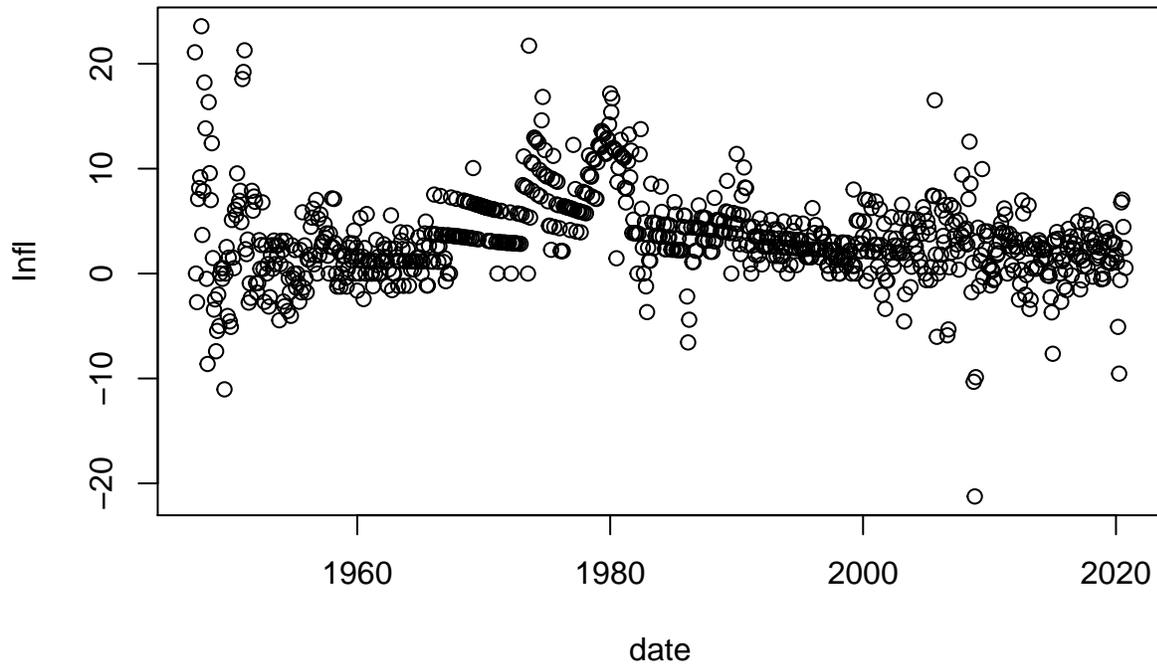
1. We make a plot of the CPI. As can be seen from the graph, the series is not stationary due to an exponential trend.

```
DATE = CPI_data$DATE
CPI = CPI_data$CPIAUCSL
date = as.Date(DATE, format="%Y-%m-%d")
plot(date, CPI)
```



Since the CPI exhibits a trend, it is not stationary and we cannot directly model it using an auto-regressive model. We need to stationarize the series. To do so, compute the annualized percentage inflation as follows, $\text{Infl}_t = 1200 \frac{CPI_t - CPI_{t-1}}{CPI_{t-1}}$. Since the CPI starts in February 1947, our inflation series should start in March 1947 because the Inflation formula requires to divide (and subtract) by the previous CPI. So, the sample size of our Inflation series is now equal to $T = 884$.

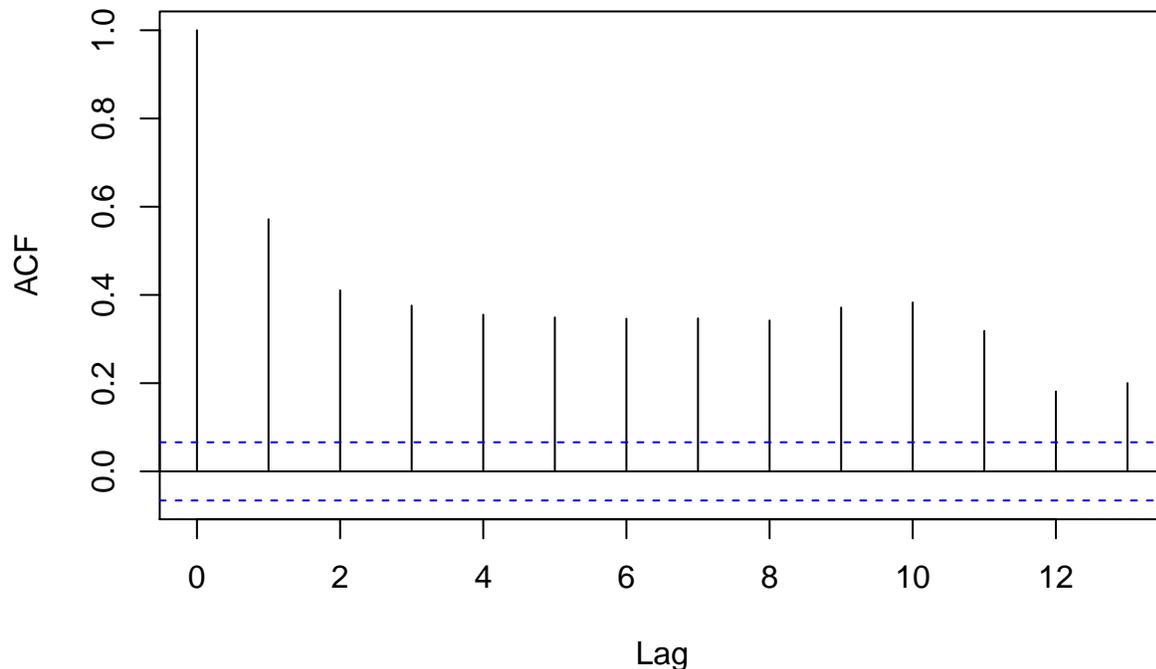
```
T = length(CPI)
Infl = 1200*(CPI[2:T]-CPI[1:(T-1)])/CPI[1:(T-1)]
DATE_Infl = DATE[2:T]
date = as.Date(DATE_Infl,format="%Y-%m-%d")
plot(date,Infl)
```



The Inflation series does not exhibit a trend and seems reasonably stationary. We could potentially use an AR model to fit the data if the series presents some linear persistence. To test this hypothesis, we first plot the autocorrelation function from lag 1 to 12. Then, we perform a Box-Pierce test based on the same number of lags to see if we reject the Null of no autocorrelation.

```
max_lag = 13  
acf_val = acf(Infl, lag.max = max_lag) ## R function for the acf
```

Series Infl



```
Box.test(Infl,lag = 12) ## R function

##
## Box-Pierce test
##
## data: Infl
## X-squared = 1467.7, df = 12, p-value < 2.2e-16

val = acf_val$acf
T = length(Infl)
BP = T*sum(val[2:max_lag]^2)
crit = qchisq(0.95,max_lag-1)
BP_test = data.frame(cbind(BP,crit))
(BP_test)

##      BP      crit
## 1 1467.744 21.02607
```

The autocorrelation values of lag 1 to 12 are all larger than the critical value of the graph. So, we suspect that we will reject the Null hypothesis of no autocorrelation. By performing the Box-Pierce test, this is the conclusion we reach. We also performed the Box-Pierce test manually and we reach the same conclusion.

To take this autocorrelation into account, we estimate an AR(1) model. We then perform a statistical test to see if the residuals of the model still present autocorrelations using a Box-Pierce test based on the autocorrelation values of the first 12 lags.

```
## Estimation of the AR(1) model
T = length(Infl)
X = array(1,dim=c(T-1,2))
```

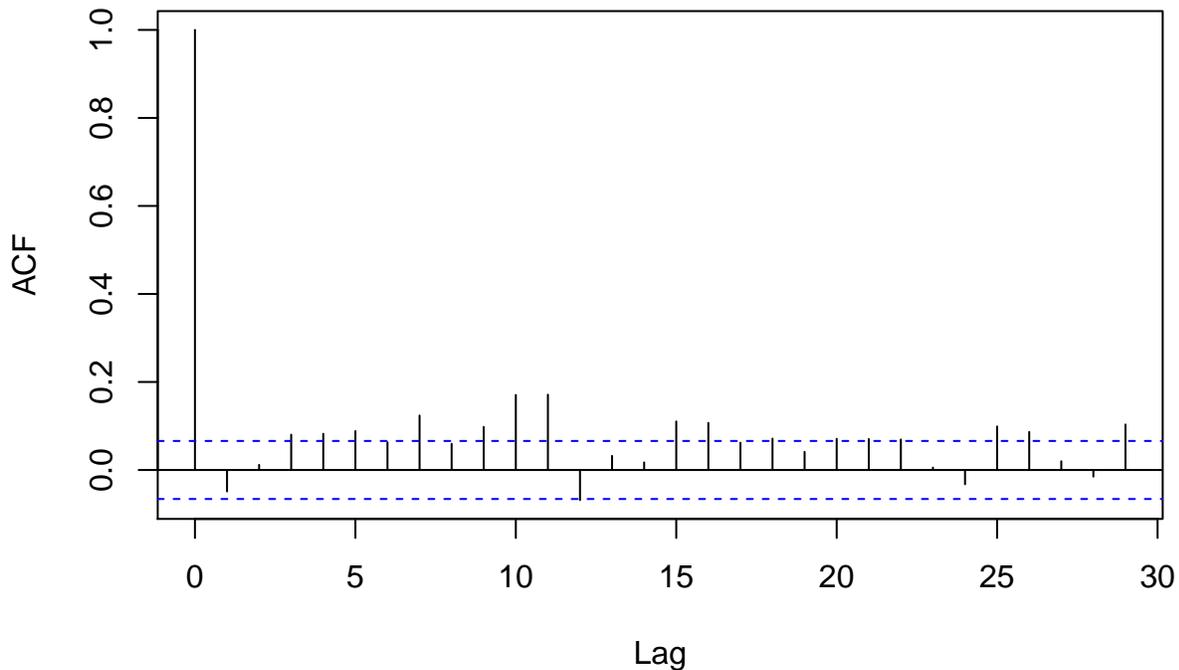
```

y = Infl[2:T]
X[,2] = Infl[1:(T-1)]
beta = solve(t(X)%*%X)%*%t(X)%*%y

## Test on the residuals
residuals = y - X%*%beta
acf(residuals)

```

Series 1



```
Box.test(residuals,lag = 12) ## R function
```

```

##
## Box-Pierce test
##
## data: residuals
## X-squared = 105.18, df = 12, p-value < 2.2e-16

```

We observe that the AR(1) model captures a lot of persistence. Nevertheless, it seems that the AR(1) model is not flexible enough to capture all the persistence exhibited by the Inflation series. This is confirmed by the Box-Pierce test that rejects the Null hypothesis of no autocorrelation on the residuals. As the purpose of this empirical exercise is about forecasting, we do not pursue the search of the best model.

Predictions

We discard the last 50% of your observations. They will now be used for evaluating the forecasts. Using an AR(1) model and for the last 50% of our observations, we make forecasts for horizon 1 to 6 months (observation 443 is the first to be predicted). To do so, we re-estimate our model parameters after every forecast.

```

t_star = 0.5*T
nb_pred = T-t_star
T_y = length(y)
nb_horizon = 6
y_pred_AR1 = array(0,dim=c(nb_pred,nb_horizon))
y_pred_RW = array(0,dim=c(nb_pred,nb_horizon))
y_pred = numeric(nb_horizon)
y_true = array(0,dim=c(nb_pred,nb_horizon))
for(t in 1:nb_pred){
  t_cur = t_star-1 + t

  ## AR(1)
  X_cur = X[1:t_cur,]
  y_cur = y[1:t_cur]
  beta_ols = solve(t(X_cur)%*%X_cur)%*%t(X_cur)%*%y_cur
  y_pred[1] = beta_ols[1] + beta_ols[2]*y_cur[t_cur]
  if(t_cur+1<=T_y){
    y_true[t,1] = y[t_cur+1]
  }
  for(h in 2:nb_horizon){
    y_pred[h] = beta_ols[1] + beta_ols[2]*y_pred[h-1]
    if(t_cur+h<=T_y){
      y_true[t,h] = y[t_cur+h]
    }
  }
  y_pred_AR1[t,] = y_pred

  ## RW
  y_pred_RW[t,] = y_cur[t_cur]
}

```

For every horizon, we compute the mean squared error (MSE, that is the average of the squared errors) and the root mean squared error (RMSE, that is the square root of the MSE). At the end of this step, we will have 6 values of MSE and 6 values of RMSE.

```

MSE = numeric(nb_horizon)
RMSE = numeric(nb_horizon)
for(h in 1:nb_horizon){
  errors = y_true[1:(nb_pred-h),h] - y_pred_AR1[1:(nb_pred-h),h]
  MSE[h] = mean(errors^2)
  RMSE[h] = sqrt(mean(errors^2))
}

```

When we performed the AR prediction, we also carried out predictions for the random walk model (see above). Now, we summarize our random walk predictions by using the MSE and the RMSE to be able to compare the two models.

```

MSE_RW = numeric(nb_horizon)
RMSE_RW = numeric(nb_horizon)
for(h in 1:nb_horizon){
  errors = y_true[1:(nb_pred-h),h] - y_pred_RW[1:(nb_pred-h),h]
  MSE_RW[h] = mean(errors^2)
  RMSE_RW[h] = sqrt(mean(errors^2))
}

```

```

for(h in 1:nb_horizon){
print(paste("Horizon: ",h))
print(paste("The RMSE of the AR(1) model is given by ",RMSE[h]))
print(paste("The RMSE of the RW model is given by ",RMSE_RW[h]))
}

## [1] "Horizon:  1"
## [1] "The RMSE of the AR(1) model is given by  2.80447625294322"
## [1] "The RMSE of the RW model is given by  3.20843695854161"
## [1] "Horizon:  2"
## [1] "The RMSE of the AR(1) model is given by  3.26179253400898"
## [1] "The RMSE of the RW model is given by  4.20078831523195"
## [1] "Horizon:  3"
## [1] "The RMSE of the AR(1) model is given by  3.25434885964678"
## [1] "The RMSE of the RW model is given by  4.35431230761698"
## [1] "Horizon:  4"
## [1] "The RMSE of the AR(1) model is given by  3.23077277390149"
## [1] "The RMSE of the RW model is given by  4.26424214184246"
## [1] "Horizon:  5"
## [1] "The RMSE of the AR(1) model is given by  3.24718805447342"
## [1] "The RMSE of the RW model is given by  4.32328676850284"
## [1] "Horizon:  6"
## [1] "The RMSE of the AR(1) model is given by  3.2551407803651"
## [1] "The RMSE of the RW model is given by  4.28669007226263"

```

The RMSEs of the AR model are all smaller than the random walk values. We conclude that the AR(1) model produces better forecasts (in terms of RMSE) than the random walk model for horizons one to six.