Project 1 Time Series

Carlos Perez, Elaheh Kimia

2023-04-23

Envirомment

```
library(pacman)
p_load(data.table, fixest, multcomp,stargazer, ggplot2, multcomp, knitr, stats, tidyr, haven, rlang, dpl
```

Theoretical part: Properties of the moving average process

Let $\mu \in \mathbb{R}, \sigma^2 > 0$ and let $Z \sim WN(\mu, \sigma^2)$. Let Y be the process defined by

$$Y_t = \sum_{j=0}^{q} \theta_j Z_{t-j},$$

for some coefficients $\theta_1, ..., \theta_q \in \mathbb{R}$, $q \in \mathbb{N}$, with $\theta_1 = 1$.

A. Show that for any $(t,h) \in \mathbb{Z}^2$, $\mathbb{E}[Y_t] = \mathbb{E}[Y_{t+h}]$.

Using the noise characteristics of $\mathbb{E}[Z_t] = \mu$ and $Var(Z_t) = \sigma^2$ we have:

$$\mathbb{E}[Y_t] = \mathbb{E}[\sum_{j=0}^{q} \theta_j Z_{t-j}] = \sum_{j=0}^{q} \theta_j \mathbb{E}[Z_{t-j}] = \mu \sum_{j=0}^{q} \theta_j$$

On the other side we have:

$$\mathbb{E}[Y_{t+h}] = \mathbb{E}[\sum_{j=0}^{q} \theta_j Z_{t-j+h}] = \sum_{j=0}^{q} \theta_j \mathbb{E}[Z_{t-j+h}] = \mu \sum_{j=0}^{q} \theta_j$$

$\Rightarrow \mathbb{E}[Y_t] = \mathbb{E}[Y_{t+h}]$, which means the moving average process Y has a constant mean over passing time.

B. Show that $(t,s,h) \in \mathbb{Z}^3, Cov(Y_t, Y_{t+h}) = Cov(Y_s, Y_{s+h})$.

By using the properties of white noise process and the linearity of the covariance we want to show what is mentioned above:

$$Cov(Y_t, Y_{t+h}) = Cov(\sum_{j=0}^{q} \theta_j Z_{t-j}, \sum_{k=0}^{q} \theta_k Z_{t+h-j})$$

$$\sum_{j=0}^{q}\sum_{k=0}^{q}\theta_j\theta_k Cov(Z_{t-j}, Z_{t+h-j})$$

In WN process $Cov(Z_i, Z_j) = 0 : \forall\ i \neq j \Rightarrow$ if $j = k \Rightarrow j - k = 0 \Rightarrow Z_{t-j}$ and Z_{t+h-k} independent $\Rightarrow Cov(Z_{t-j}, Z_{t+h-j}) = 0$

$$\Longrightarrow \theta_0^2 Cov(Z_t, Z_{t+h}) = \sigma^2$$

if $j = k \Rightarrow j - k = 0 \Rightarrow Z_{t-j}$ and Z_{t+h-k} are the same random variables.

$$\Longrightarrow Cov(Z_{t-j}, Z_{t+h-k}) = Var(Z_{t-j}) = \sigma^2$$

$$\Longrightarrow Cov(Y_t, Y_{t+h}) = \theta_0^2\sigma^2 + \sum_{j=1}^{q}\theta_j^2\sigma^2$$

We consider a similar approach for $Cov(Y_s, Y_{s+h})$

$$Cov(Y_s, Y_{s+h}) = \theta_0^2\sigma^2 + \sum_{j=1}^{q}\theta_j^2\sigma^2$$

C. Show that Y is stationary and give the autocovariance function.

To show that Y is stationary, we have to show that its mean and ACVF are independent of time.

In part (A) we showed that $\mathbb{E}[Y_t] = \mu\sum_{j=0}^{q} : \forall\ t \Rightarrow Y$ is constant over time.

Now, let's consider ACVF:

$$\gamma(h) = Cov(Y_t, Y_{t+h}) : \forall\ h \in Z$$

Using calculation of part(b) we have:

$$\gamma(h) = Cov(Y_t, Y_{t+h}) = \theta_0^2\sigma^2 + \sum_{j=1}^{q}\theta_j^2\sigma^2 : \forall\ h \in Z$$

which is independent of time.

Additionally we can show that the $Var(Y_t) < \infty$

$$Var(Y_t) = Var(\sum_{j=0}^{q}\theta_j Z_{t-j}) = \theta_j^2\sum_{j=0}^{q}Var(Z_{t-j}) = \theta_j^2\sigma^2 < \infty$$

Stationarity of Y is satisfied.

D. Prove that if Z is a Gaussian process, then Y_t is independent of Y_{t+h} for any $t \in \mathbb{Z}$ and $|h| > q$.

Since Z is a Gaussian process, then any linear combination of Z, such as Y, is also a Gaussian process:

Let's consider Y_t and $Y_{t+h} : \forall\ t \in Z$ and $|h| > q$m then to show their independence"

$$Y_t = \theta_0 Z_t + \theta_1 Z_{t-1} + ... + \theta_q Z_{t-q} \&\ Y_{t+h} = \theta_0 Z_{t+h} + \theta_1 Z_{t+h-1} + ... + \theta_q Z_{t+h-q}$$

$$f(Y_t, Y_{t+h}) = f(Z_t, Z_{t-1}, ..., Z_{t-q}, Z_{t+h}, Z_{t+h-1}, ..., Z_{t+h-q})$$

Where $f(Y_t, Y_{t+h})$ has a multivariate normal distribution because Z is a Gaussian process.

To proof the independence of Y_t and Y_{t+h}, we show that their covariance matrix is diagonal.

$$Cov(Y_t, Y_{t+h}) = Cov(\sum_{j=0}^{q} \theta_j Z_{t-j}, \sum_{k=0}^{q} \theta_k Z_{t+h-j}) \sum_{j=0}^{q} \sum_{k=0}^{q} \theta_j \theta_k Cov(Z_{t-j}, Z_{t+h-j})$$

Let's consider $|i - j| > q \Rightarrow Z_{t-i}$ and Z_{t+h-j} are separated by more than q time steps.

This means that are uncorrelated, then:

$$Cov(Y_t, Y_{t+h}) = \sum_{i=0}^{q} \theta_i \theta_{i+h-t} Cov(Z_{t-i}, Z_{t+h-i}) \Rightarrow |i| > q, |i + h - t| > q, |t - i| > q, |t + h - i|$$

$\Rightarrow Z_{t-i}, Z_{t+h-i}$ are uncorrelated $\Rightarrow Cov(Y_t, Y_{t+h}) = 0 \Rightarrow Y_t$ and Y_{t+h} are uncorrelated $\Rightarrow Y_t$ and Y_{t+h} are independent : $\forall\ t \in Z$ & $|h| > q$.

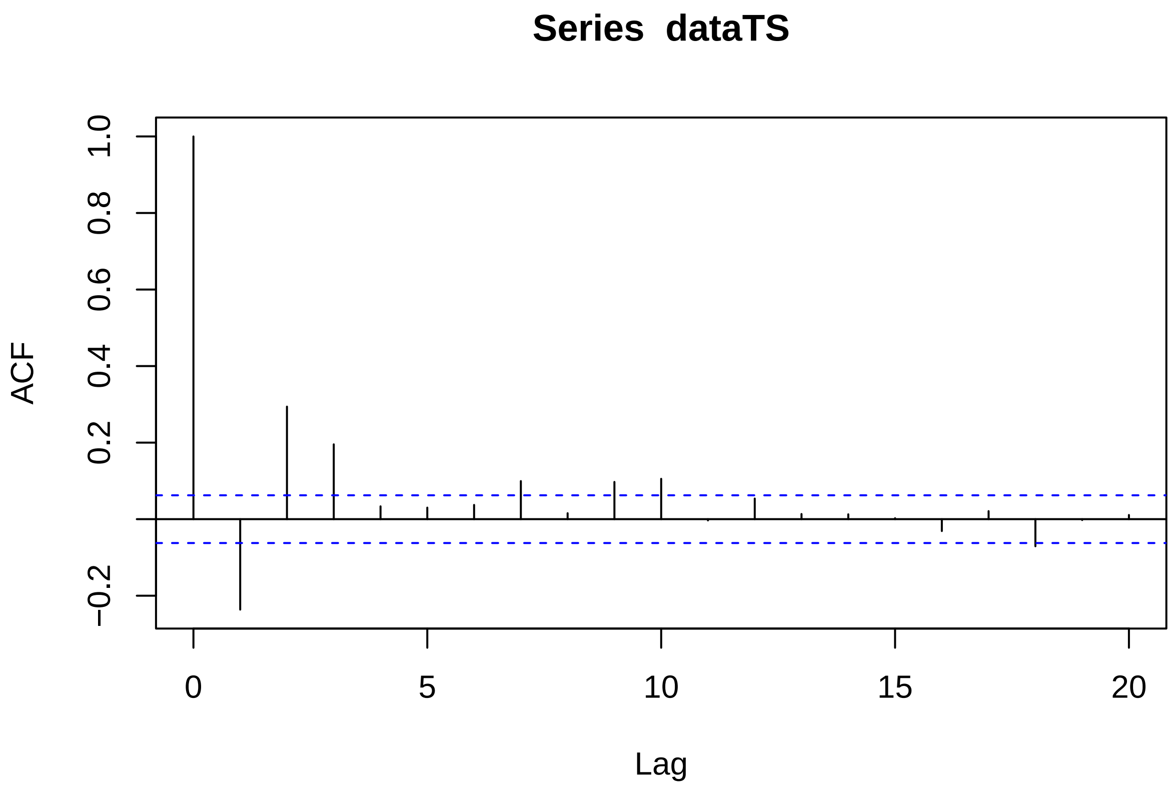
Practical part: 007 – The missing data

Exercise 1

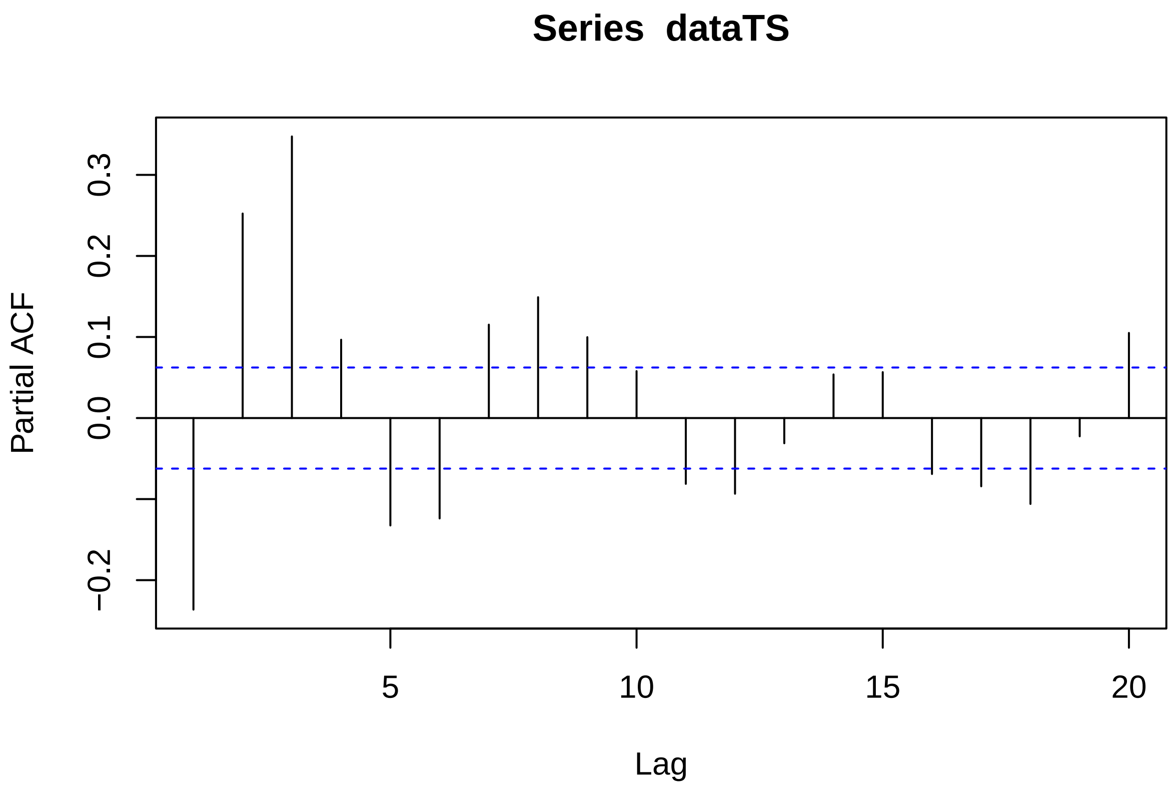
```r
#Create the data set from the .csv file
data <- read.csv("data/data.csv")

#Modify the dataset so it is treated as time series data
dataTS <- ts(data$datamissing, freq = 1)

#Run the acf function and plot
acf(dataTS, lag.max = 20, na.action = na.pass)
```



```
#Run the pacf function and plot
pacf(dataTS, lag.max = 20, na.action = na.pass)
```



We Know that if the ACF shuts off after a certain value q in the plot, it could be a good fit for the MA(q) model. We have $q = 2$ for the given moving average equation. However, when we consider our ACF plot with h=20, we observe that it loses significance at $q = 3$ with a significant drop in PAC at the same lag. Therefore, we suggest defining a moving average process with an order of $q = 3$, i.e., MA(3).

Exercise 2.a

First we create a function that gives you the indices of the missing data in the vector X.

```r
find_missing <- function(X) {
  return(which(is.na(X)))
}
```

Second we create a function that describes the theoretical ACF.

$$
\begin{aligned}
\gamma(h) &= \mathrm{cov}(X_t, X_{t+h}) \\
&= E[(Z_t + \theta_1 Z_{t-1} + \theta_2 Z_{t-2})(Z_{t+h} + \theta_1 Z_{t+h-1} + \theta_2 Z_{t+h-2})] \\
&= E[Z_t Z_{t+h}] + \theta_1 E[Z_{t-1} Z_{t+h}] + \theta_2 E[Z_{t-2} Z_{t+h}] \\
&\quad + \theta_1 E[Z_t Z_{t+h-1}] + \theta_1^2 E[Z_{t-1} Z_{t+h-1}] \\
&\quad + \theta_1 \theta_2 E[Z_{t-2} Z_{t+h-1}] + \theta_2 E[Z_t Z_{t+h-2}] \\
&\quad + \theta_1 \theta_2 E[Z_{t-1} Z_{t+h-2}] + \theta_2^2 E[Z_{t-2} Z_{t+h-2}]
\end{aligned}
$$

$$= \begin{cases} \sigma^2(1+\theta_1^2+\theta_2^2) & \text{if } h=0 \\ \sigma^2(\theta_1+\theta_1\theta_2) & \text{if } h=\pm 1 \\ \sigma^2\theta_2 & \text{if } h=2 \end{cases}$$

Then the autocorrelation function would be:

$$= \begin{cases} 1 & \text{if } h=0 \\ \frac{(\theta_1+\theta_1\theta_2)}{(1+\theta_1^2+\theta_2^2)} & \text{if } h=\pm 1 \\ \sigma^2\theta_2 & \text{if } h=2 \end{cases}$$

```
#We create the function that describes the theoretical form of the ACF for the different instances of q
acf_MA2 <- function(h, theta1=-0.5, theta2=0.6, sigma2=0.4) {
if (h == 0) {
acf <- 1
} else if (abs(h) == 1) {
acf <- (theta1 + theta1*theta2) / (1 + theta1^2 + theta2^2)
} else if (abs(h) == 2) {
acf <- theta2 / (1 + theta1^2 + theta2^2)
} else {
acf <- 0
}
return(acf)
}
```

Third we create a function that correctly computes Γ given the parametric expression of Equation (2) and a function that computes γ.

For Equation 1 of the Corollary we have that:

$$a_0 = \mu(1 - \sum_{i=1}^{n} ai)$$

However, since the mean of Xt is assumed to be 0, the $a_1 = 0$. Therefore, we just have to create the function for the Γa matrix and the γ vector

```
#We create the function that creates the Gamma matrix
GAMMAM <- function(t) {
  n <- length(t)
  GAMMA <- matrix(nrow = n, ncol = n)
  for (i in 1:n) {
    for (j in 1:n) {
      h <- abs((n+1-j)-(n+1-i))
      GAMMA[i,j] <- acf_MA2(h)
    }
  }
  return(GAMMA)
}

#We create the function that creates the gamma vector
GAMMAV <- function(t) {
  n <- length(t)
  gama <- matrix(nrow = n, ncol = 1)
  for (i in 1:n) {
    h <- i
    gama[i,1] <- acf_MA2(i)
  }
return(gama)
}
```

Now we run the functions:

```
#Assign the t times that have missing values to missing_idx
missing_idx <- find_missing(dataTS)

# Compute the gamma matrix and vector for the missing data points
Gamma <- GAMMAM(dataTS)
gamma <- GAMMAV(dataTS)


# Solve the system of equations to obtain the best linear predictor
a <- solve(Gamma, gamma)

#create the Xq_all list
n <- length(dataTS)
XqV <- numeric(n)
for (t in 1:n) {
  Xq <- dataTS[which(!is.na(dataTS) & pmax(1, t - 2) <= seq_along(dataTS) & seq_along(dataTS) <= pmin(n
  XqV[t] <- mean(Xq)
}

#Create the linear combination between the vector a and the list of random variables Xq and assign the
BltV <- numeric(length(missing_idx))
for (i in 1:length(missing_idx)) {
  t <- missing_idx[i]
  Xq <- XqV[t]
  a_t <- a[t]
  BltV[i] <- sum(a_t * rev(Xq[1:length(a_t)]))
}

# Create a copy of data$datamissing
data_imputed <- dataTS

# Replace missing values with BltV
data_imputed[missing_idx] <- BltV
```

Exercise 2.b

```
#Create the data set that has the mean of Xq as a replacement for the missing values
datar_mean <- dataTS
datar_mean[missing_idx] <- mean(XqV)

#Calcualte and compare the Mean Squared errors for both datasets
MSE1 <- sqrt((1/abs(length(missing_idx))*(sum(data$data-data_imputed)^2)))
MSE1
```

```
## [1] 0.4253417
```

```
MSE2 <- sqrt((1/abs(length(missing_idx))*(sum(data$data-datar_mean)^2)))
MSE2
```

```
## [1] 0.4004448
```

```
#Compare to see which mean squared eeror is best.
MSE1 > MSE2
```

```
## [1] TRUE
```

Here the MSE of our approximation is bigger than the MSE using only the mean of the collection of random variables X_q. The main reason for this is that using $q = 2$ does not reflect the actual correlation of the value with previous lags and therefore the estimation could be biased. Also looking at the plots of the ACF and PACF we can notice that the values of the θ_1 and θ_2 may not be correct. Using the mean of X_q to make an approximation of the values may result on a lower MSE since all values are within a certain range, but it is not the most effective method. At least compared to a correct computation of the best linear predictor (b_t^l).

Exercise 3

As per exercise 1, the best value for q would be 3, since the autocorrelation function drops below significance aster lag 3 and we see a a significant drop on the partial autocorrelation function at the same lag. Therefore rewriting exercise 2.a and 2.b with $q = 3$ we would have:

$$X_t = Z_t + \theta_1 Z_{t-1} + \theta_2 Z_{t-2} + \theta_3 Z_{t-3}$$

We calculate the value of θ_3:

```
# Fit a moving average model of order 3
ma_model <- arima(dataTS, order = c(0, 0, 3))

# View the estimated parameters, including theta3
ma_model$coef
```

```
##           ma1           ma2           ma3     intercept
## -0.3454561061  0.5146839456  0.3067457851 -0.0001317118
```

We compute the function that describes the theoretical ACF function with $q = 3$.

$$\gamma(h) = Cov(X_t, X_{t-h}) = Cov(Z_t + \theta_1 Z_{t-1} + \theta_2 Z_{t-2} + \theta_3 Z_{t-3}, Z_{t-h} + \theta_1 Z_{t-h-1} + \theta_2 Z_{t-h-2} + \theta_3 Z_{t-h-3})$$

$$\begin{cases} (1 + \theta_1^2 + \theta_2^2 + \theta_3^2)\sigma^{,} & \text{if } h = 0 \\ (\theta_1 + \theta_1\theta_2 + \theta_2\theta_3)\sigma^2 & \text{if } h = \pm 1 \\ (\theta_2 + \theta_1\theta_3)\sigma^2 & \text{if } h = \pm 2 \\ \theta_3\sigma^2 & \text{if } h = \pm 3 \\ 0 & \text{if } |h| > 3 \end{cases}$$

$$\rho(k) = \frac{\gamma_k}{\gamma_0}\begin{cases} 1 & \text{if } k = 0 \\ \frac{\theta_1+\theta_1\theta_2+\theta_2\theta_3}{1+\theta_1^2+\theta_2^2+\theta_3^2} & \text{if } k = \pm 1 \\ \frac{\theta_2}{1+\theta_1^2+\theta_2^2+\theta_3^2} & \text{if } k = \pm 2 \\ \frac{\theta_3}{1+\theta_1^2+\theta_2^2+\theta_3^2} & \text{if } k = \pm 3 \\ 0 & \text{if } |k| > 3 \end{cases}$$

```
acf_MA3 <- function(h, theta1=-0.5, theta2=0.6, theta3=0.3, sigma2=0.4) {
if (h == 0) {
 acf3 <- 1
  } else if (abs(h) == 1) {
    acf3 <- (theta1 + theta1*theta2 + theta1*theta2*theta3) / (1 + theta1^2 + theta2^2 + theta3^2)
  } else if (abs(h) == 2) {
    acf3 <- (theta2 + theta1*theta3) / (1 + theta1^2 + theta2^2 + theta3^2)
  } else if (abs(h) == 3) {
    acf3 <- theta3 / (1 + theta1^2 + theta2^2 + theta3^2)
  } else {
    acf3 <- 0
  }
  return(acf3)
}
```

With the autocorrelation function rewriting we have to rewrite the functions that create the Γ and γ matrices:

```
GAMMAM3 <- function(t) {
  n <- length(t)
  GAMMA3 <- matrix(nrow = n, ncol = n)
  for (i in 1:n) {
    for (j in 1:n) {
      h <- abs((n+1-j)-(n+1-i))
      GAMMA3[i,j] <- acf_MA3(h)
    }
  }
  return(GAMMA3)
}

GAMMAV3 <- function(t) {
  n <- length(t)
  gama3 <- matrix(nrow = n, ncol = 1)
  for (i in 1:n) {
    h <- i
    gama3[i,1] <- acf_MA3(1)
  }
return(gama3)
}
```

Now we create the new X_q with $q = 3$

```
n <- length(data$datamissing)
XqV3 <- numeric(n)
for (t in 1:n) {
  Xq3 <- dataTS[which(!is.na(data$datamissing) & pmax(1, t - 3) <= seq_along(data$datamissing) & seq_al
  XqV3[t] <- mean(Xq3)
}
```

We compute the fucntions:

```
missing_idx <- find_missing(dataTS)

# Compute the gamma matrix and vector for the missing data points
Gamma3 <- GAMMAM3(dataTS)
gamma3 <- GAMMAV3(dataTS)

# Solve the system of equations to obtain the best linear predictor
a3 <- solve(Gamma3, gamma3)

#Create the linear combination between the vector a and the list of random variables Xq
BltV3 <- numeric(length(missing_idx))
for (i in 1:length(missing_idx)) {
  t <- missing_idx[i]
  Xq3 <- XqV3[t]
  a_t3 <- a3[t]
  BltV3[i] <- sum(a_t3 * rev(Xq3[1:length(a_t3)]))
}

# Create a copy of data$datamissing
data_imputed3 <- dataTS

# Replace missing values with BltV
data_imputed3[missing_idx] <- BltV3
```

Now we create the alternative case with the mean of X_q, compute the mean squared errors and compare:

```
#Create the data set that has the mean of Xq as a replacement for the missing values
datar_mean3 <- dataTS
datar_mean3[missing_idx] <- mean(XqV3)

#Calculate and compare the Mean Squared errors for both datasets
MSE1.3 <- sqrt((1/abs(length(missing_idx))*(sum(data$data-data_imputed3)^2)))
MSE1.3
```

```
## [1] 0.09022282
```

```
MSE2.3 <- sqrt((1/abs(length(missing_idx))*(sum(data$data-datar_mean3)^2)))
MSE2.3
```

```
## [1] 0.3938672
```

```
#Compare to see which mean squared error is best.
MSE1.3 > MSE2.3
```

```
## [1] FALSE
```

Here we have a better computation of the best linear predictor, which takes into account all significant lags that determine the correlation at t, this is done by selecting $q = 3$ as the new value and drafting a new ACF. Compared to the previous root-mean squared error the new prediction only has an error o of 0.09022282 which is much lower than 0.4253417 and additionally lower than the new mean prediction which has a RMSE of 0.3938672. Here we can see that a correct computation of the acf with the correct significant lags can make a really good prediction of the missing values of the dataset.

Therefore, Agent 007 should be able to open the vault and save the world!!