Integrated processes, seasonal process, forecasting

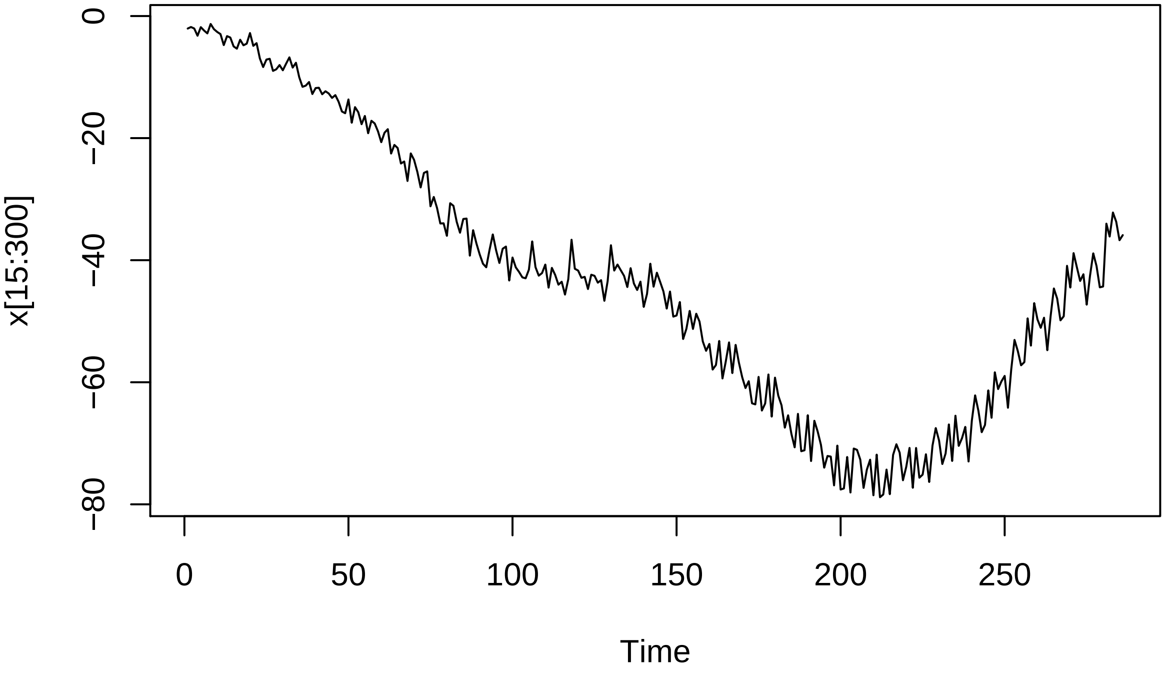
Daniel Alonso

March 8th, 2021

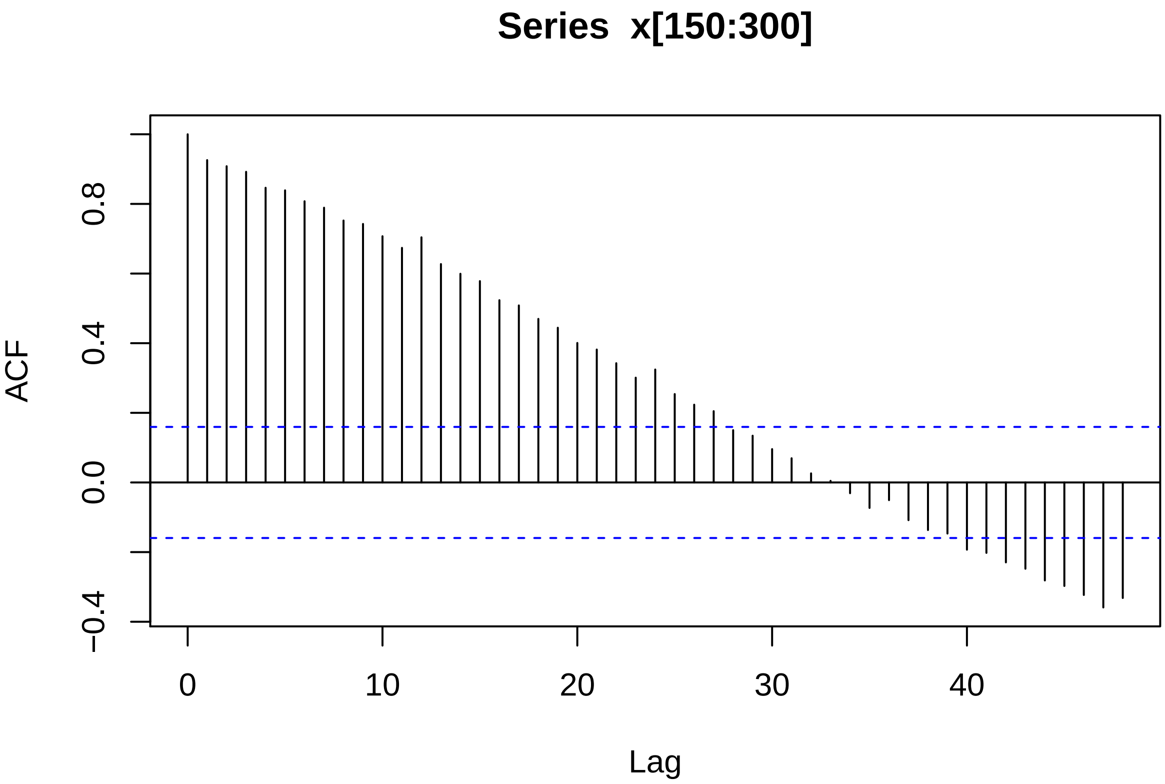
Simulation of a SARIMA (0,1,1)(0,1,1),12 process

We simulate a Seasonal ARIMA of 300 observations and components $(p, d, q) = (0, 1, 1)$, $(P, D, Q) = (0, 1, 1)$ and $m = 12$. It also has $SMA = 0.3$, and $MA = -0.3$

```
plot.ts(x[15:300])
```

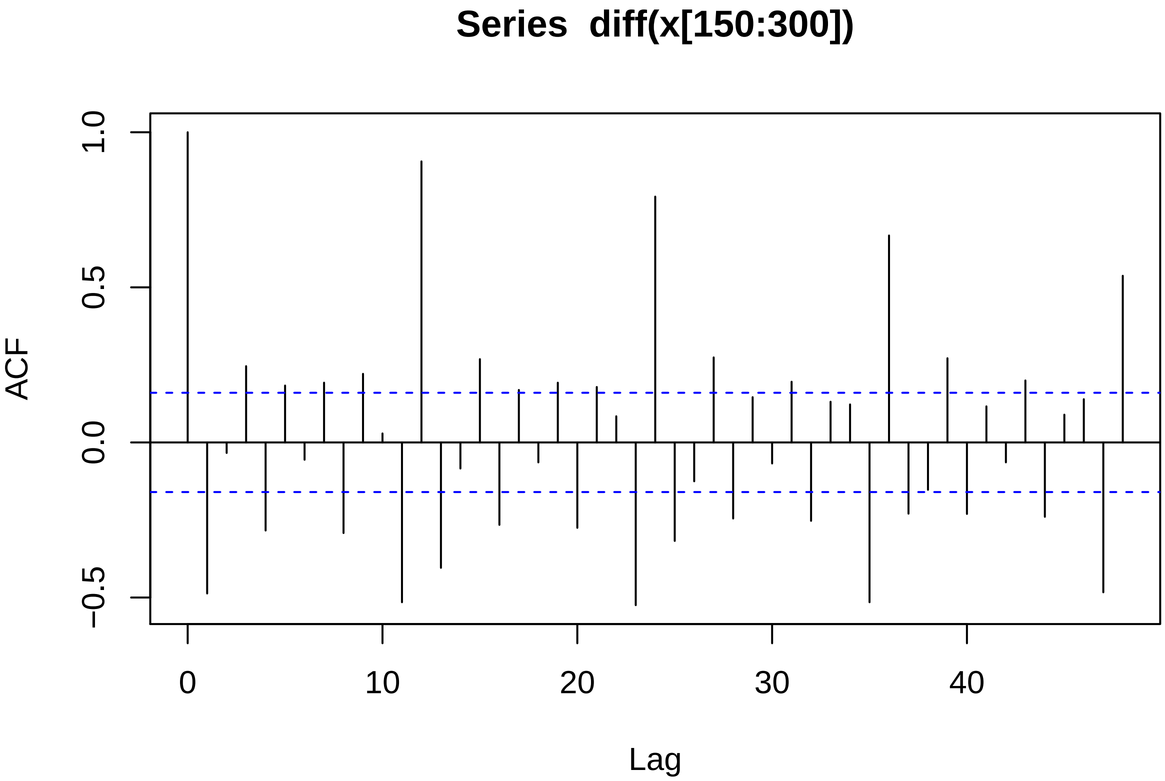


Here we can see the series has a trend upwards and exhibits seasonality (12), as defined in the simulation.

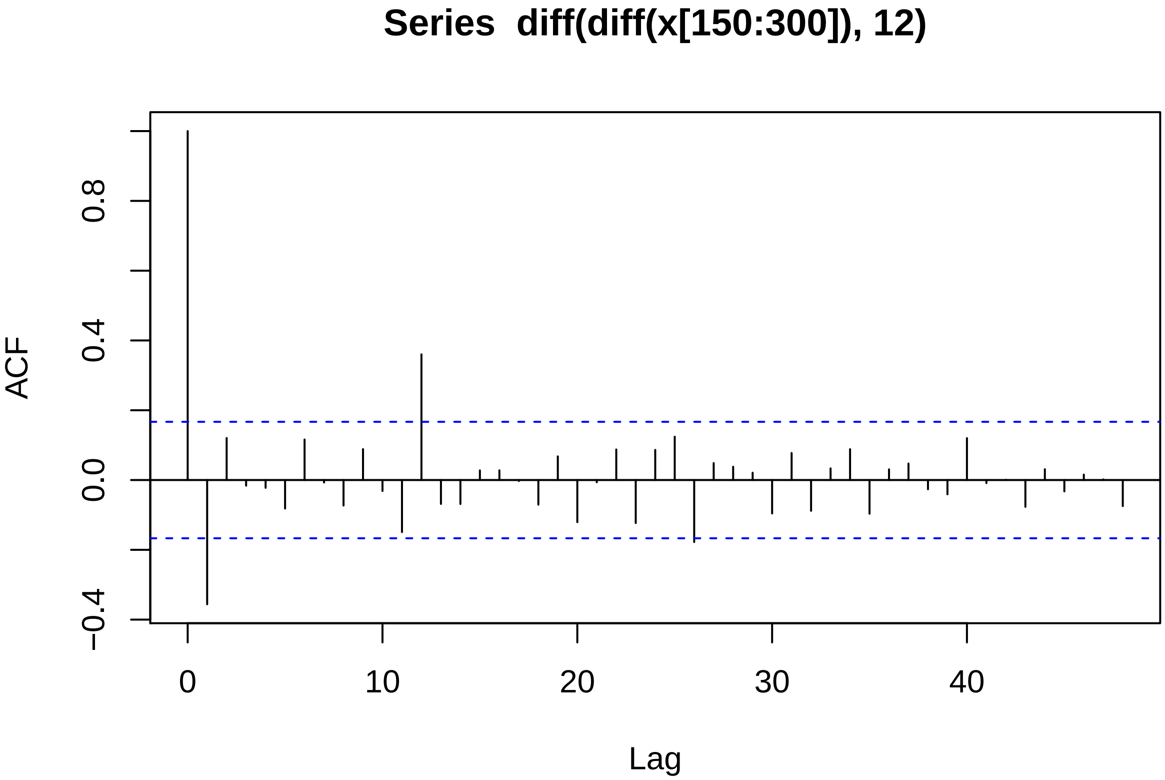


We can see a clearly slow decaying pattern, showing strong trend.

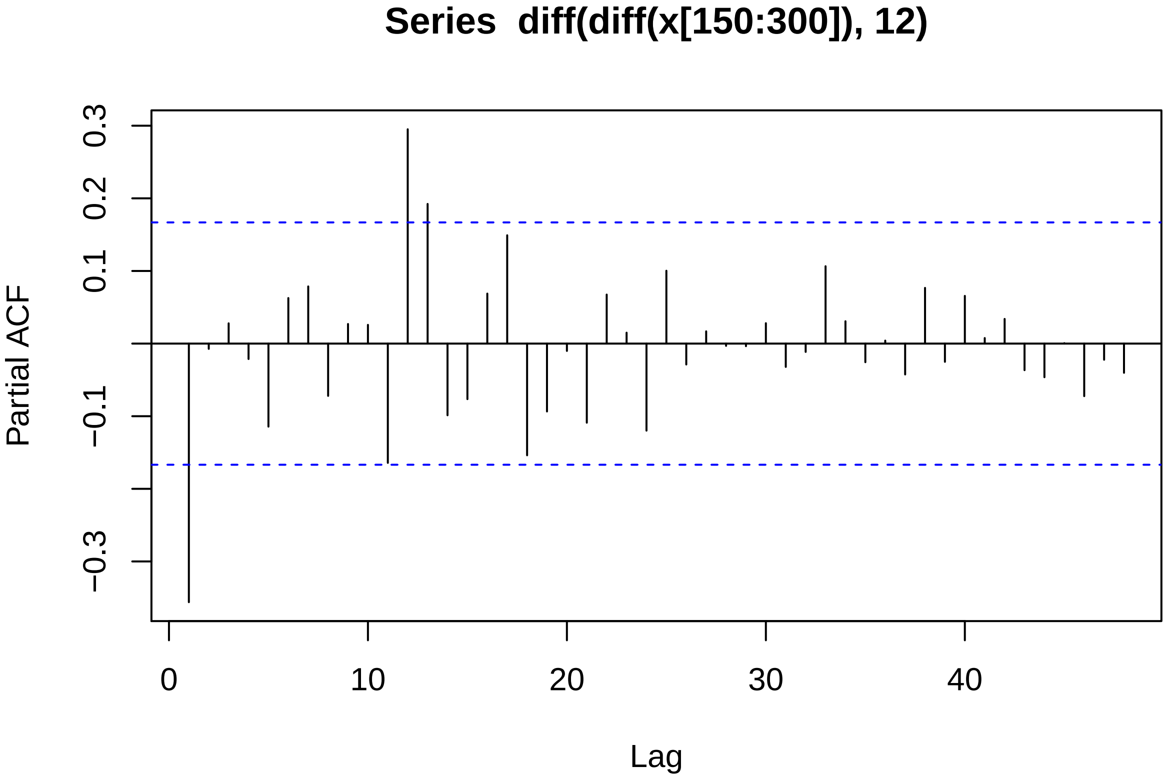
We take first difference and plot a correlogram for it:



And we can notice that there's a strong peak up every 12 periods (months). Therefore we take seasonal difference and plot a correlogram for it:



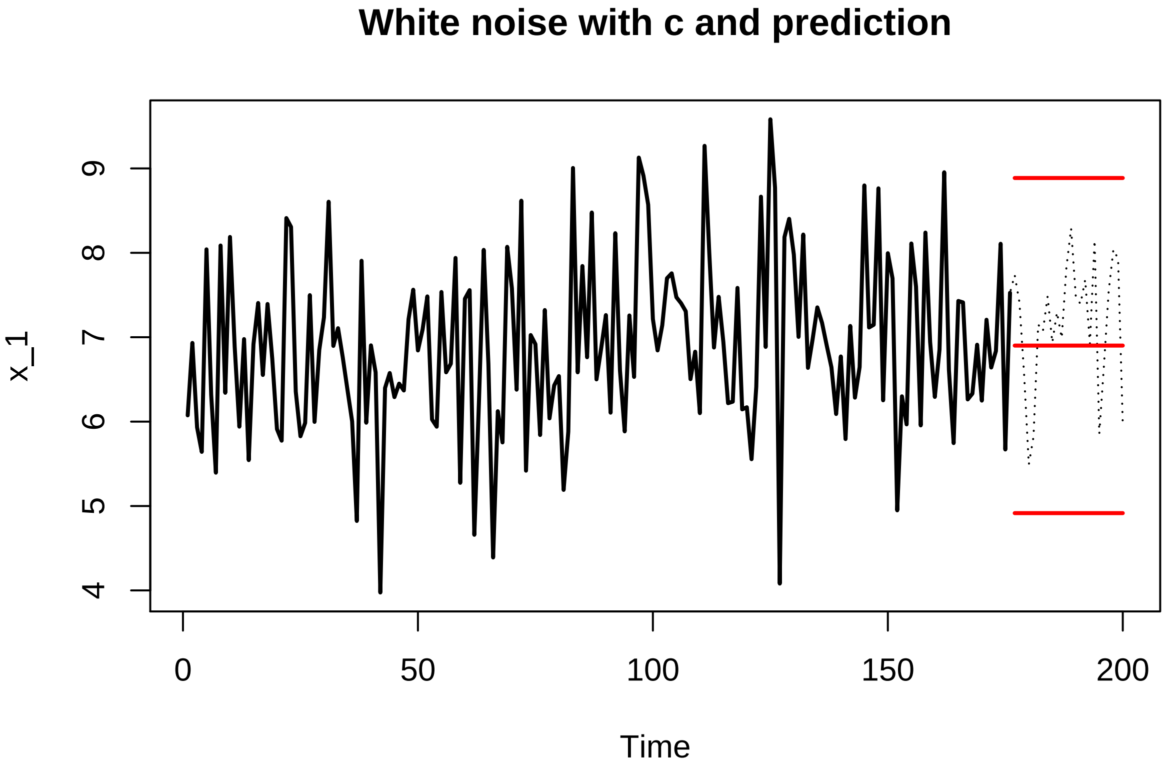
We have 2 peaks, the first two peaks corresponding with variance and period 1.



The PACF plot corroborates with the same observations highlighted earlier.

Simulation and forecasting of white noise

We simulate a white noise process generating 200 normally distributed values. A constant is summed to the values (in our case 7). We only take the first 176 observations, then the next 24 observations are predicted.

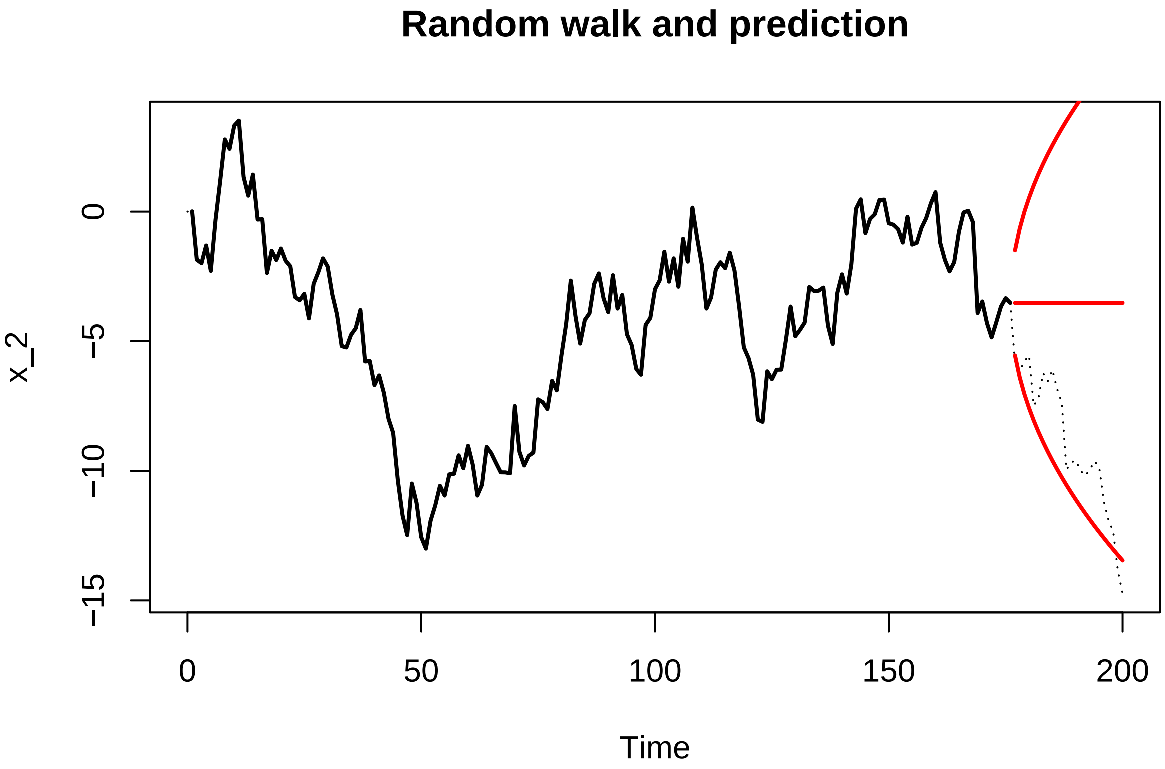


Looking at this we can see that the most solid prediction we can make is a constant (especifically the constant we summed to the generated randomly distributed values, 7).

Our confidence interval is a 95%-CI, it represents $1.959964\sigma^2$.

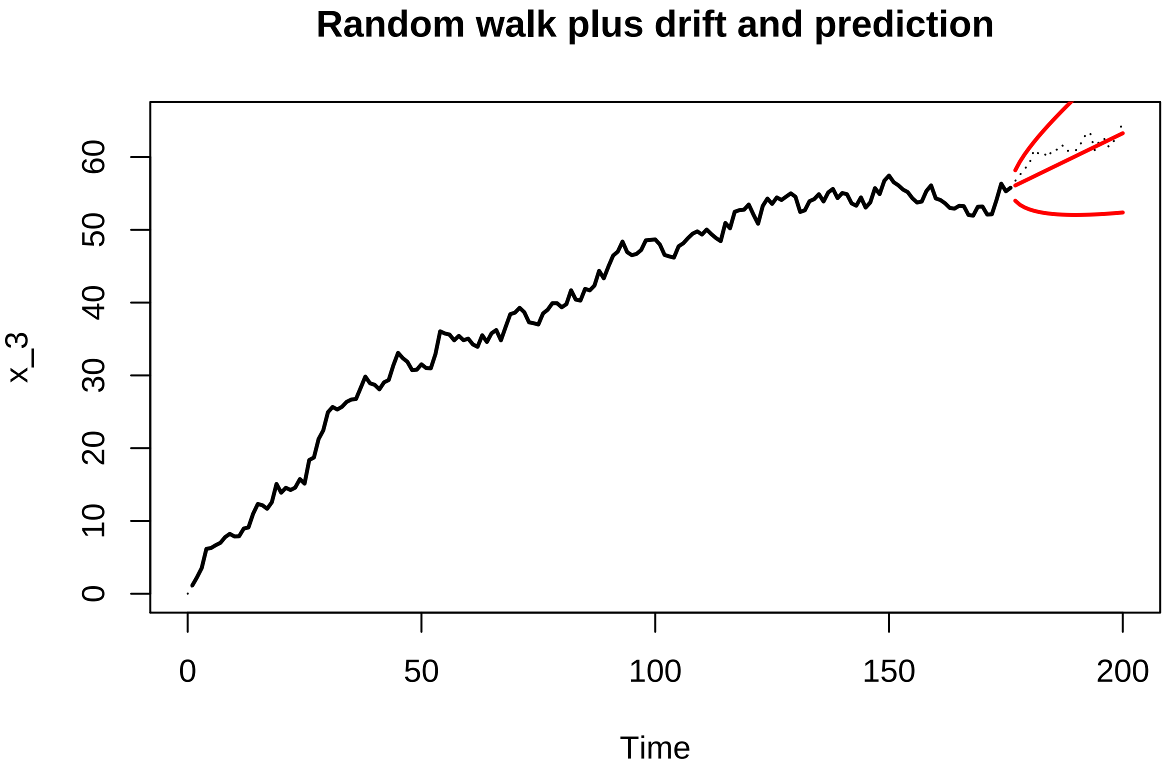
Simulation and forecasting of random walk

We simulate 200 observations, out of which we use a window of 176 and predict the remaining 24. From our prediction range, we pick a straight line (even though when iterating this several times, lower and upper 95% confidence bounds can be quite good at forecasting), however, we have infinite variance for the prediction error if the amount of points to forecast increases.



Simulation and forecasting of random walk plus drift

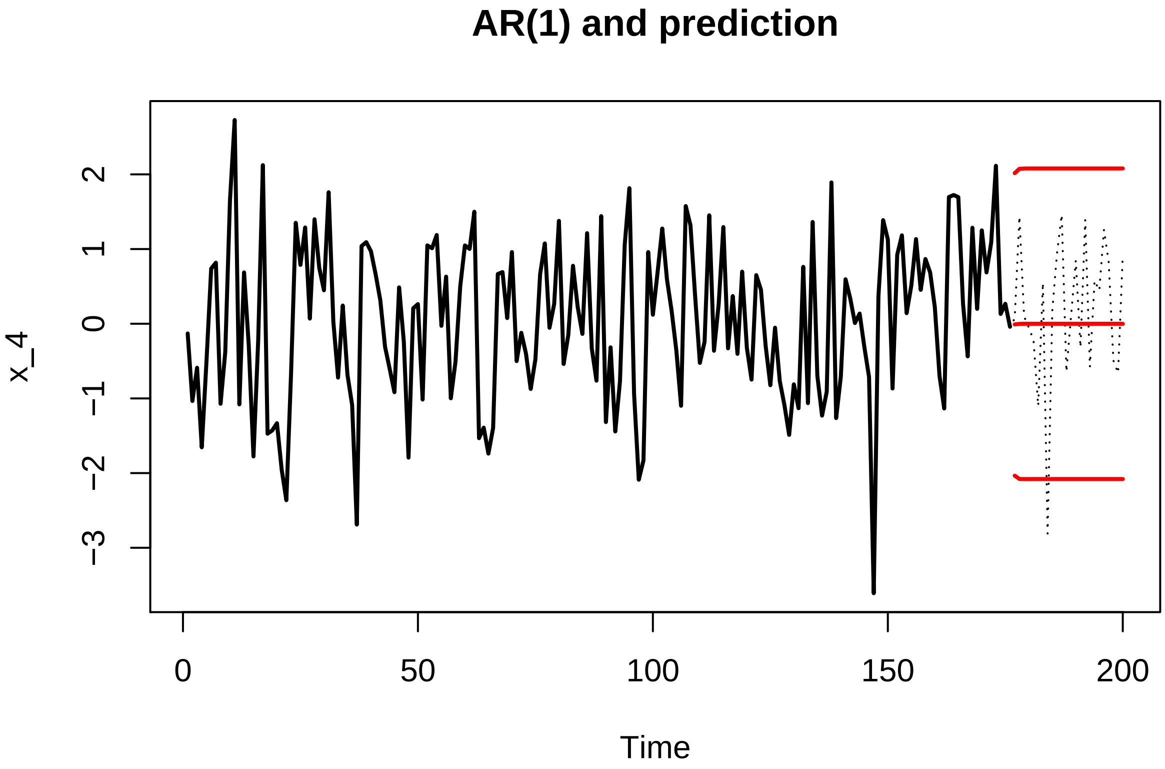
We simulate a random walk with drift, where we have 200 observations, with a window of 176 observations. As before, we forecast 24 observations. The parameters are essentially the same as teh previous random walk, however, we have a $\mu = 0.4$ as a parameter and drift. Here we perform the straight line, average prediction, variance is increased gradually as the walk is projected in the future.



Simulation and forecasting of an AR(1)

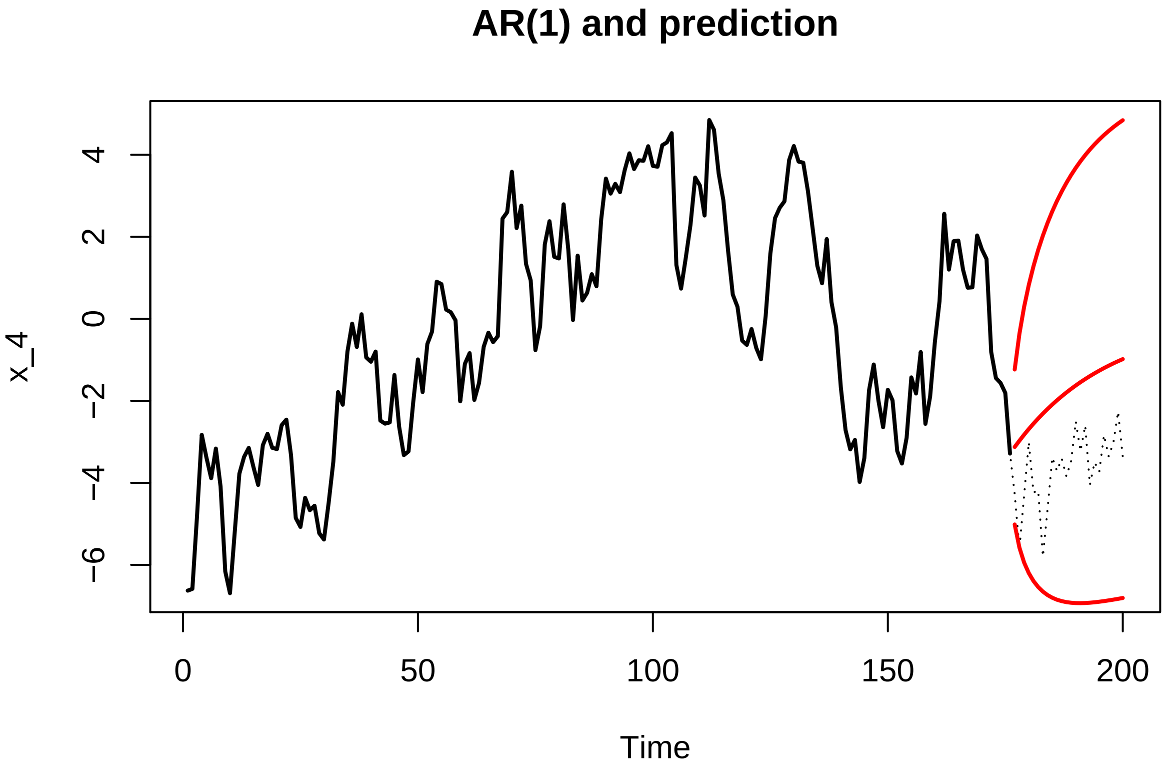
When running the forecast for an AR(1) process we get interesting results:

With a very small coefficient



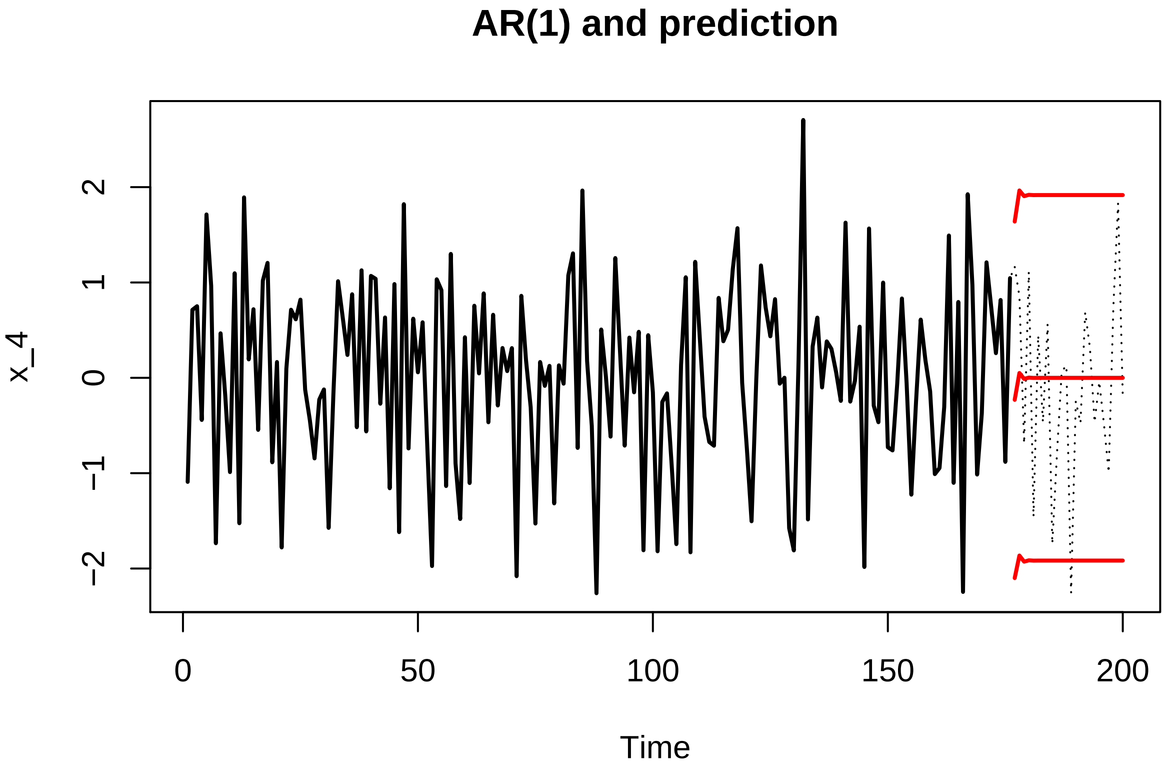
The prediction is pretty much zero, along with the confidence bounds, which are essentially the prediction plus or minus 2. Both are also straight lines.

With a larger coefficient



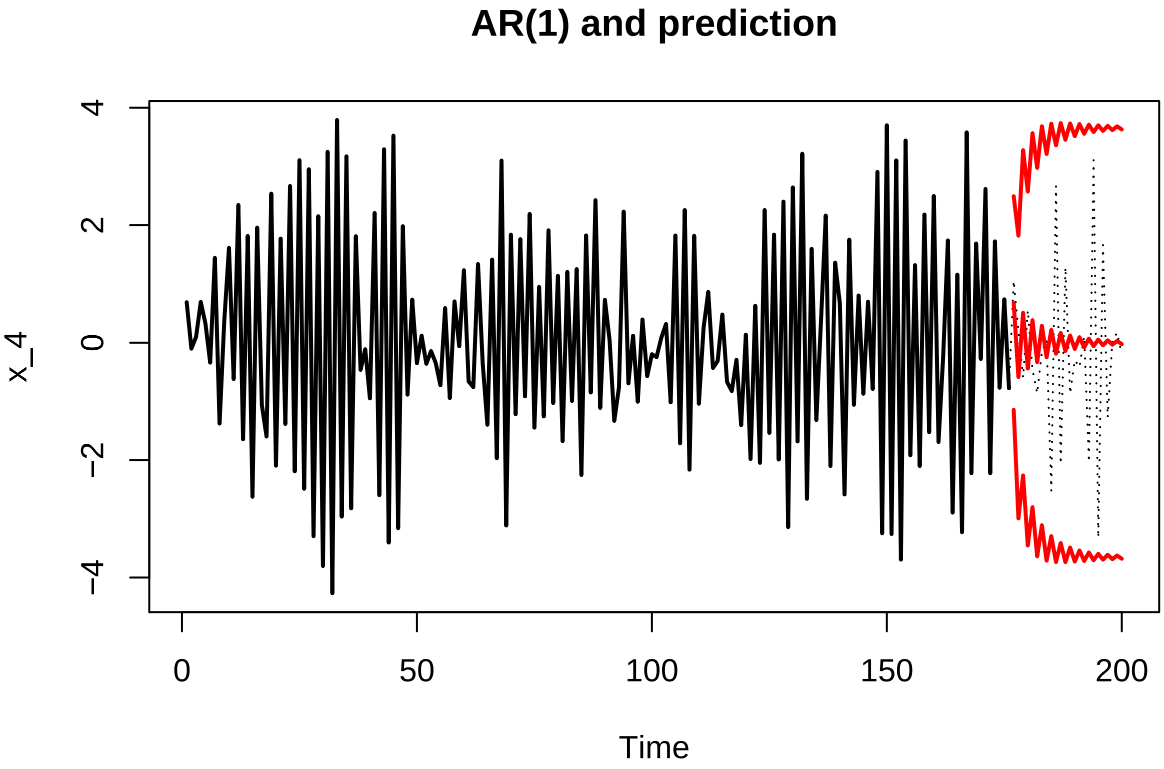
The larger the coefficient is, the least the forecast will resemble a straight line, but rather will be curves that eventually converge in a straight-ish line. This is quite variable though, and we can get lines that never quite converge if the AR(1) process exhibits a significant trend.

With negative coefficient closer to zero



With a negative coefficient close to zero, we get pretty much a straight line with no initial curvature. This applies to the confidence bounds as well.

With negative coefficient further away from zero



Interestingly, we can see that the forecast in this case starts oscillating until it eventually converges to a value, for both the confidence intervals and the ordinary prediction.

Performance of k-step ahead simulation of an AR(1)

Utilizing $k = 1$ often yields a type I error of <0.05, therefore fitting a 5% tolerance.

```
#> [1] 0.048
```

For any $k > 1$ we can certainly get values under the threshold of 5%, however, it is less likely.

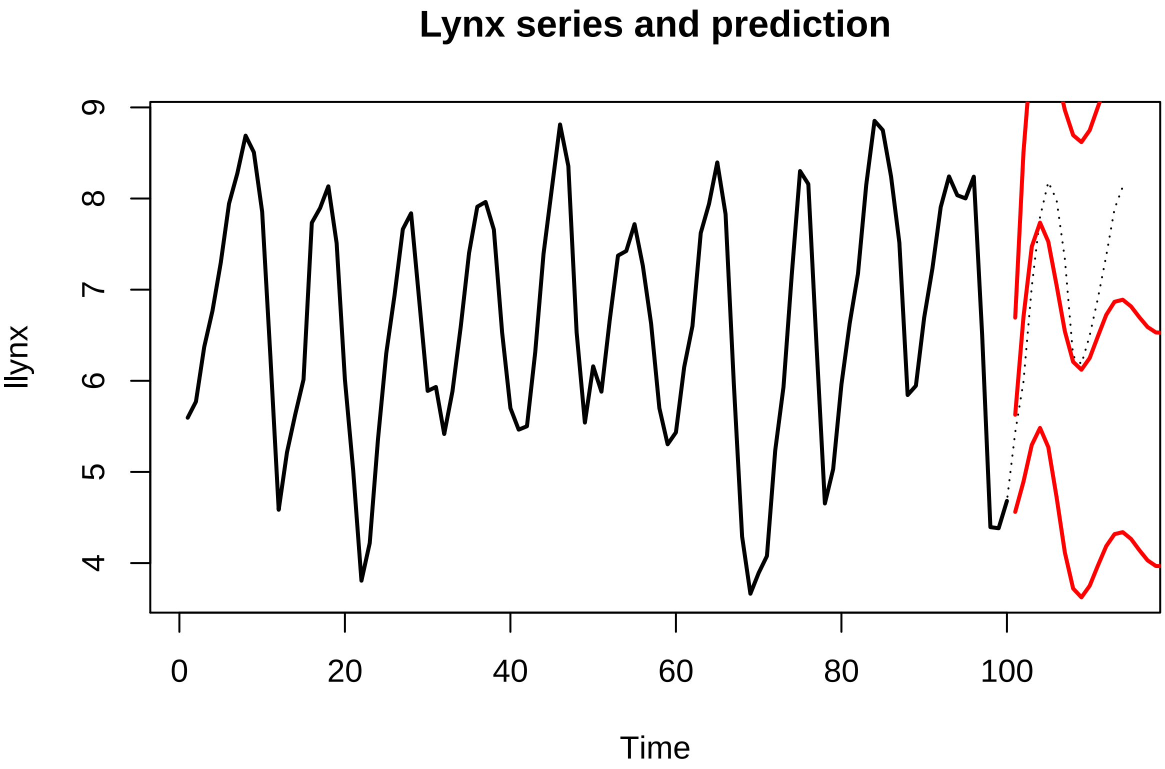
Here we try $k = 5$:

```
#> [1] 0.048
```

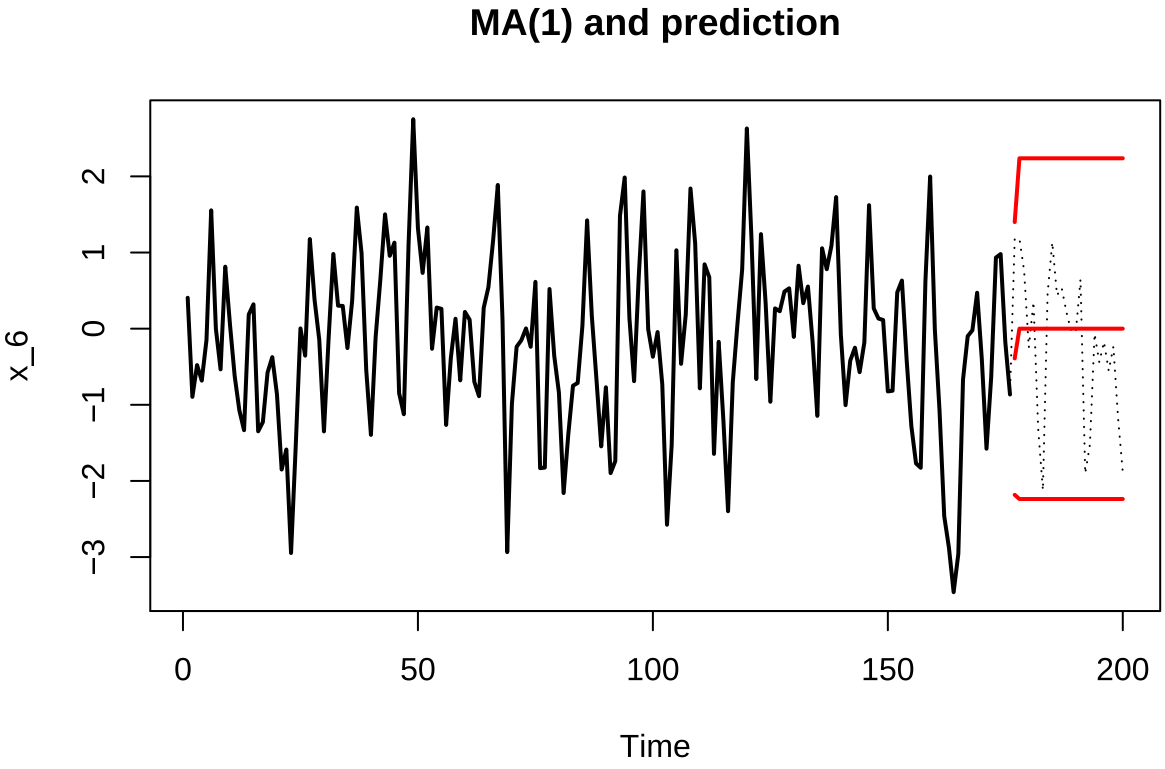
Either way, given the randomness of the process, whether we got a better or worse type I error % entirely depends on randomness. So picking $k = 1$ seems more plausible, as it seems more likely to yield a better type I error %.

Forecasting of Lynx series

Our forecast sort of recreates the observations with decaying weights. Similar to earlier, we have a converging oscillation which eventually returns our series' average (0). Both confidence bounds exhibit the same behavior and the convergence results in a value, (~ 4, ~ -4) for each bound.

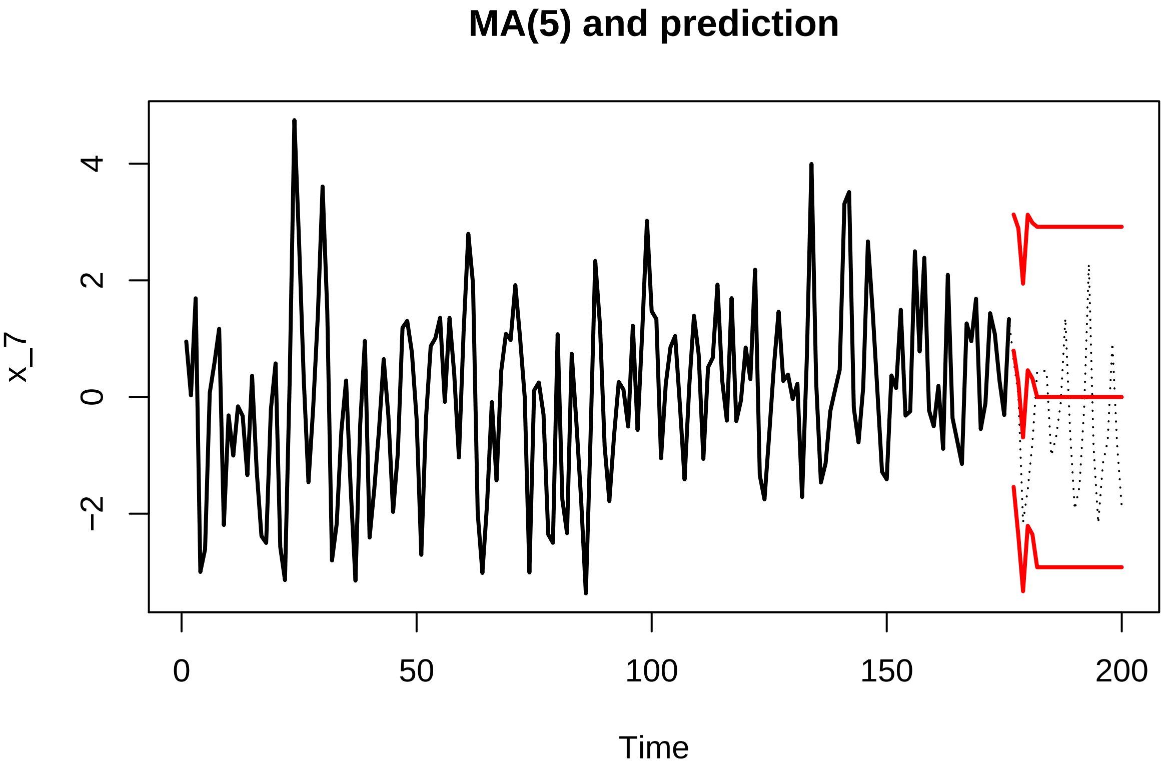


Forecasting of an MA(1)



When forecasting we obtain values in straight lines, however, the first observation of both the bounds and the average are different from the rest of the values in the forecasted lines.

Forecasting of an MA(5)

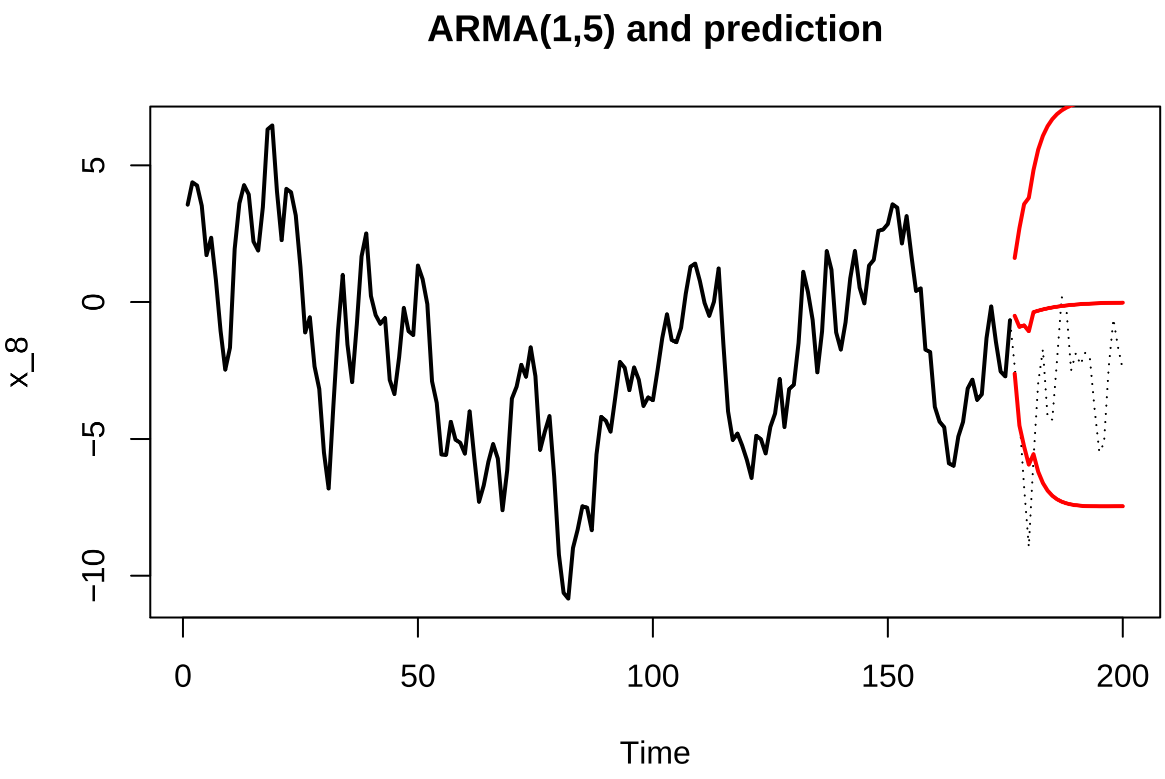


MA(5) exhibits the same behavior as MA(1), where the amount of coefficients in our MA process also, in turn, is the amount of values that differ from the average (in the average prediction) and the bounds.

Forecasting of an ARMA(1,5)

Considering that an ARMA(1,5) model is essentially a mix of AR(1) and MA(5) models, we apply the same observations we noticed before.

Considering the size of our AR(1) coefficient $\phi = 0.85$, we eventually obtain a straight line, however it takes a significantly higher amount of observations to eventually converge to a value and obtain a straight line. We also notice that given the amount of θ coefficients for the MA(5) model, we have 5 little bumps at the start of the average and bound forecast lines. These 5 bumps are different values for the forecast.



Check and discuss through simulation the performance of one-step ahead and k-step ahead forecast of AR(p) and MA(q) processes

one-step ahead and k-step ahead forecast of AR(p)

We run a simulation utilizing an AR(3) process, and the algorithm goes as follows:

- We first create the process 10 times and obtain the confidence and type I error per iteration
- After running this 10 times we obtain the mean type I error and the mean confidence for the 10 different processes (although the processes do have the same coefficient, we reduce randomness by repeating it several times)
- We then construct the following table with the results per k

Table 1: type I error and confidence per k for AR(3)

k	typeIerror	confidence
1	0.0536	0.9464
3	0.0577	0.9423
10	0.0891	0.9109

We use 3 different Ks, therefore: $k \in \{1, 3, 10\}$.

From the results we can clearly see how increasing k clearly tends to increase significantly the probability of type I error, and therefore the confidence

one-step ahead and k-step ahead forecast of MA(q)

We repeat the exact same process as before for an MA(3) process to watch the performance impact of increasing k.

Table 2: type I error and confidence per k for MA(3)

k	typeIerror	confidence
1	0.0539	0.9461
3	0.0685	0.9315
10	0.0621	0.9379
20	0.0570	0.9430

For the MA(3) process used as example we can see that the same conclusion can't be made for it vs an AR(3) process. A higher k does not necessarily mean a significant impact in performance. So we don't necessarily need to select a lower k, and we can actually go higher than this without any significant increase in type I error.