Me consultan sobre los parámetros de esta función:

f ARIMA(p,d,q)

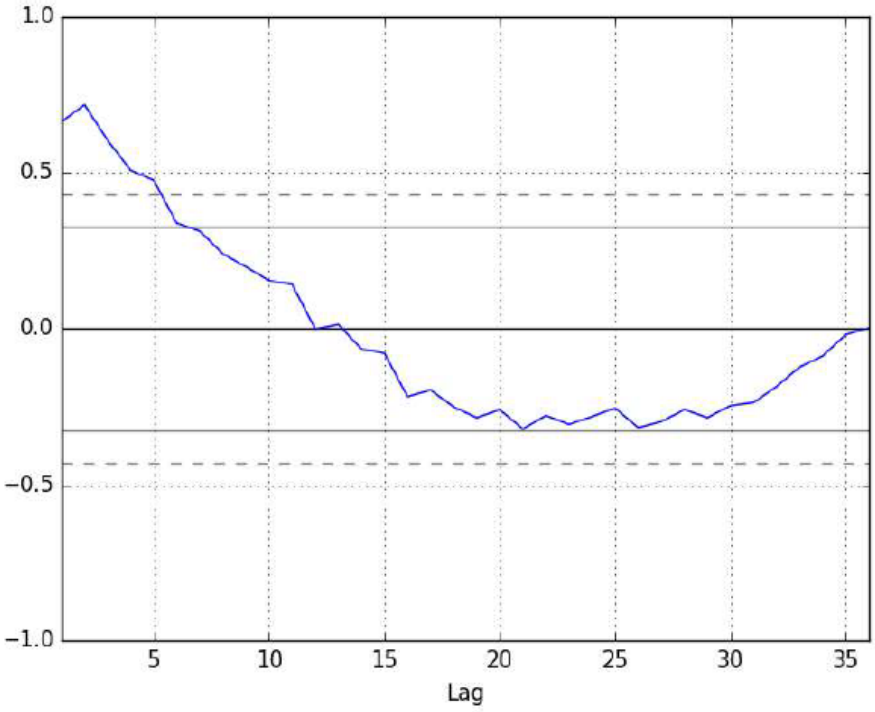
```
# fit model
model = ARIMA(series, order=(5,1,0))
```

Order:

Autoregresión (p):

El eje y indica la correlación, recordemos que se buscaría valores cercanos a 1, y sobre las líneas punteadas y sólida (95% y 99% intervalos de confianza).

Por ejemplo en este es 5 porque hasta el lag de 5 (en x) está positivo, y sobre las barras, claro que se podría probar con un valor menor, no mayor.



Orden de la diferenciación (d):

Precisamente usar **Augmented Dickey-Fuller test** para determinar si require 0 (no diferenciación), 1 , 2 ó n diferenciaciones.

Moving average (q)

- q: The size of the moving average window, also called the order of moving average.

A linear regression model is constructed including the specified number and type of terms, and the data is prepared by a degree of differencing in order to make it stationary, i.e. to remove trend and seasonal structures that negatively affect the regression model. A value of 0 can be used for a parameter, which indicates to not use that element of the model. This way, the ARIMA model can be configured to perform the function of an ARMA model, and even a simple AR, I, or MA model.

Ampliado en el otro pdf

22.5 Autocorrelation Plots

We can plot the correlation coefficient for each lag variable. This can very quickly give an idea of which lag variables may be good candidates for use in a predictive model and how the relationship between the observation and its historic values changes over time. We could manually calculate the correlation values for each lag variable and plot the result. Thankfully, Pandas provides a built-in plot called the `autocorrelation_plot()` function[3].

The plot provides the lag number along the x-axis and the correlation coefficient value between -1 and 1 on the y-axis. The plot also includes solid and dashed lines that indicate the 95% and 99% confidence interval for the correlation values. Correlation values above these lines are more significant than those below the line, providing a threshold or cutoff for selecting more relevant lag values.

```
# autocorrelation plot of time series
from pandas import Series
from matplotlib import pyplot
from pandas.tools.plotting import autocorrelation_plot
series = Series.from_csv('daily-minimum-temperatures.csv', header=0)
autocorrelation_plot(series)
pyplot.show()
```

Listing 22.4: Create an autocorrelation plot of the Minimum Daily Temperatures dataset with pandas.

[3] http://pandas.pydata.org/pandas-docs/stable/visualization.html#autocorrelation-plot

Running the example shows the swing in positive and negative correlation as the temperature values change across summer and winter seasons each previous year.

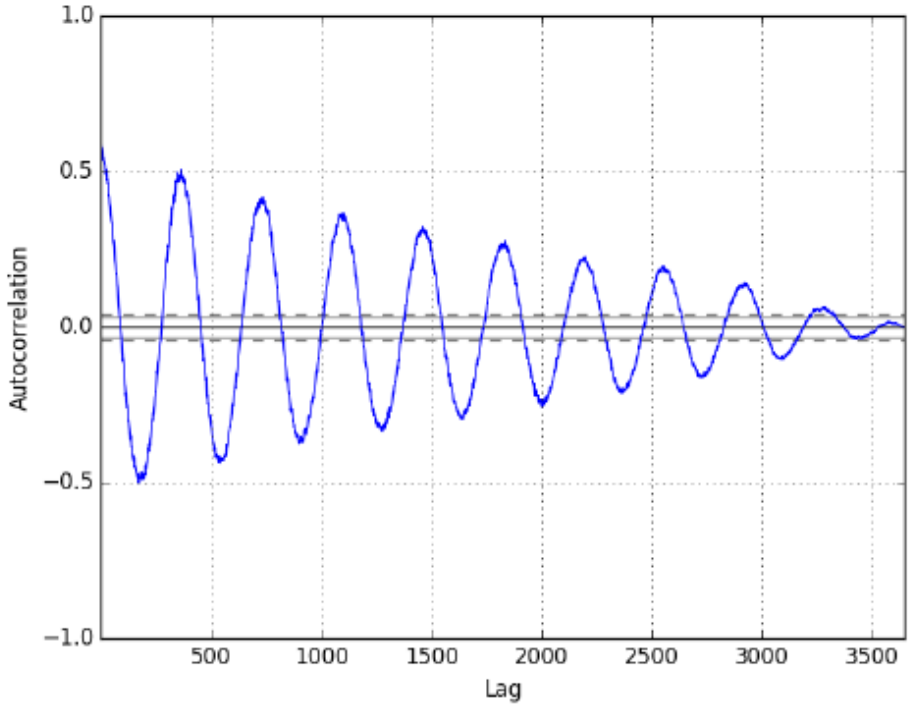


Figure 22.2: Autocorrelation plot of the Minimum Daily Temperatures dataset with pandas.