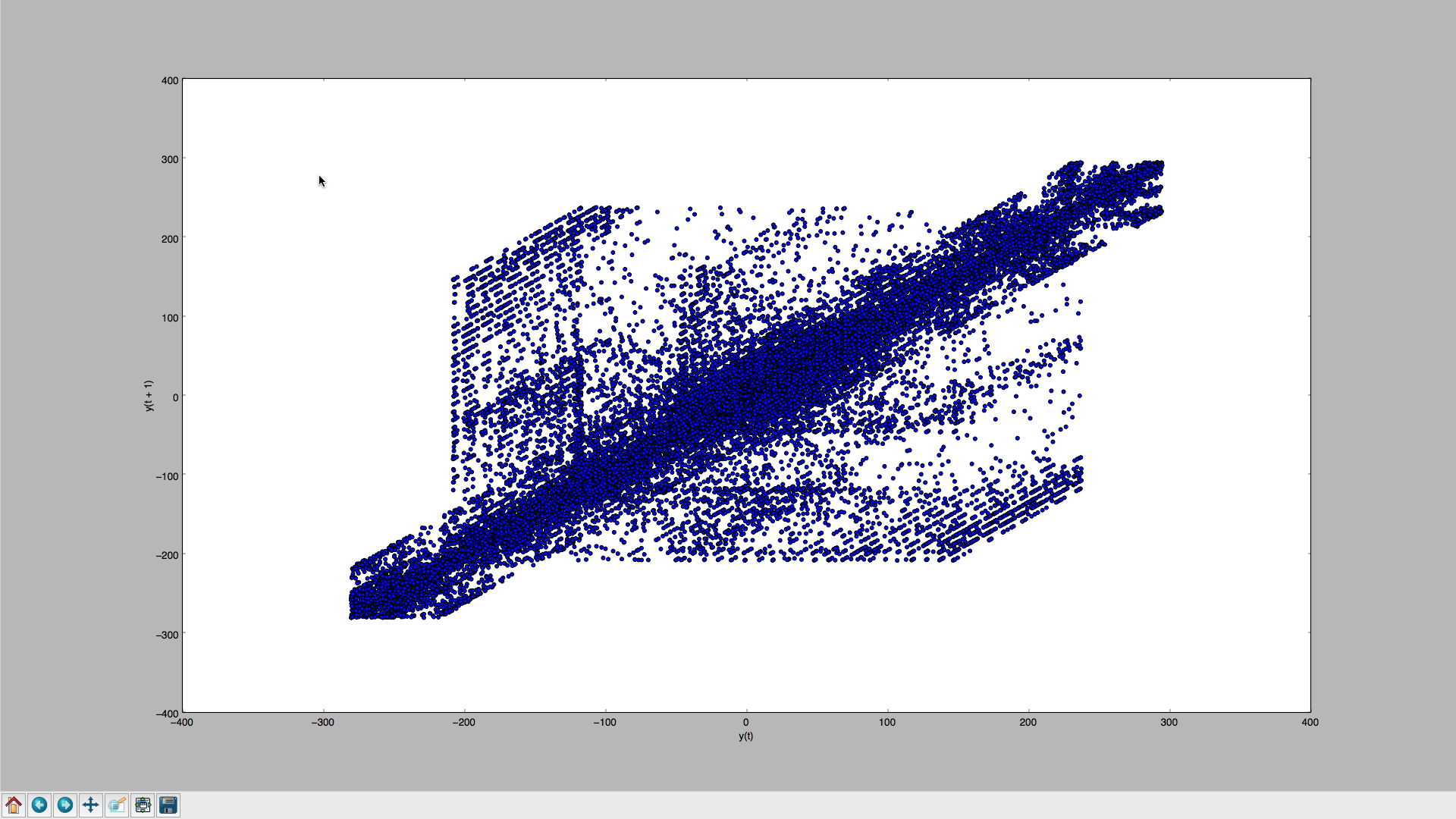
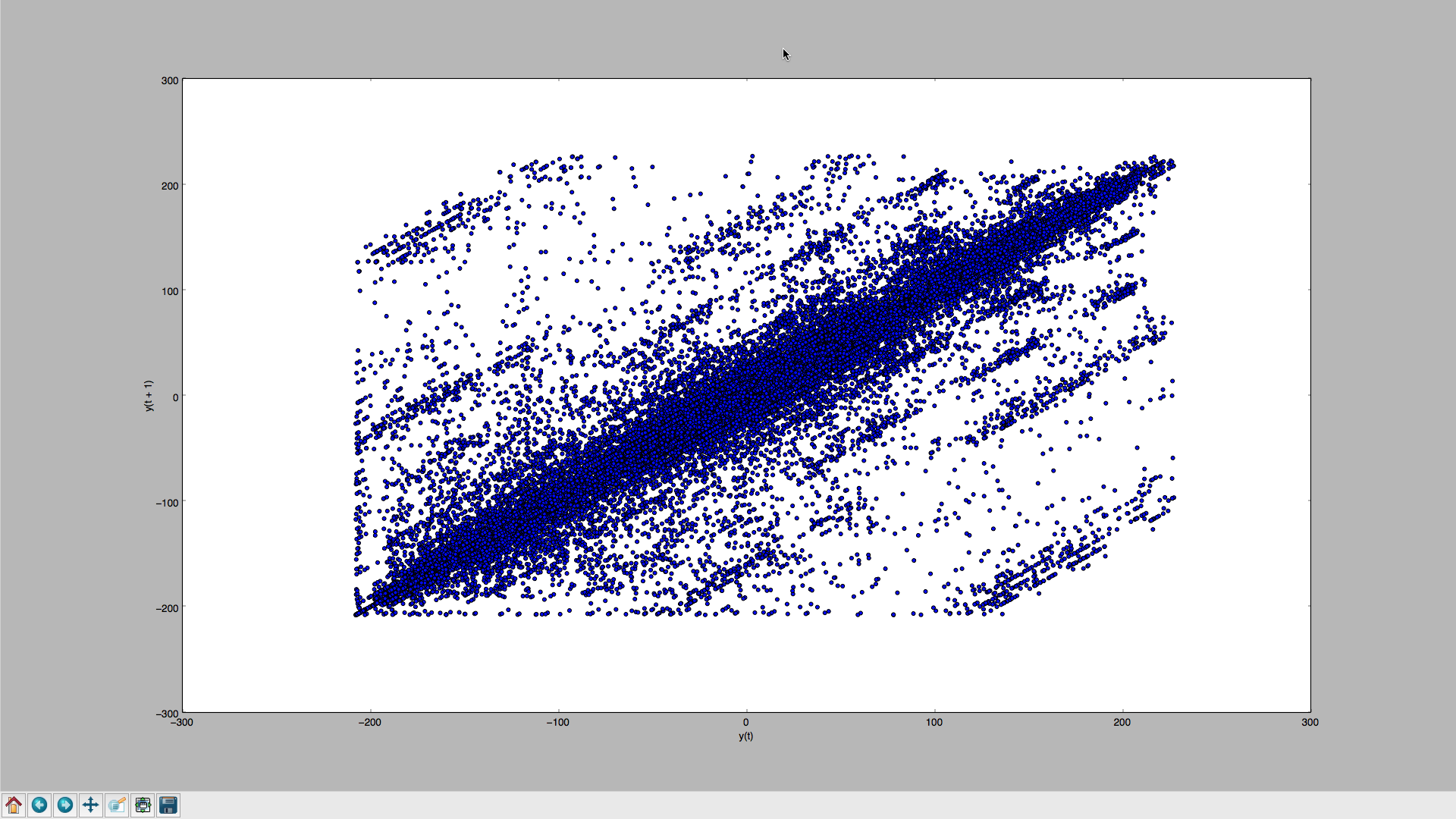
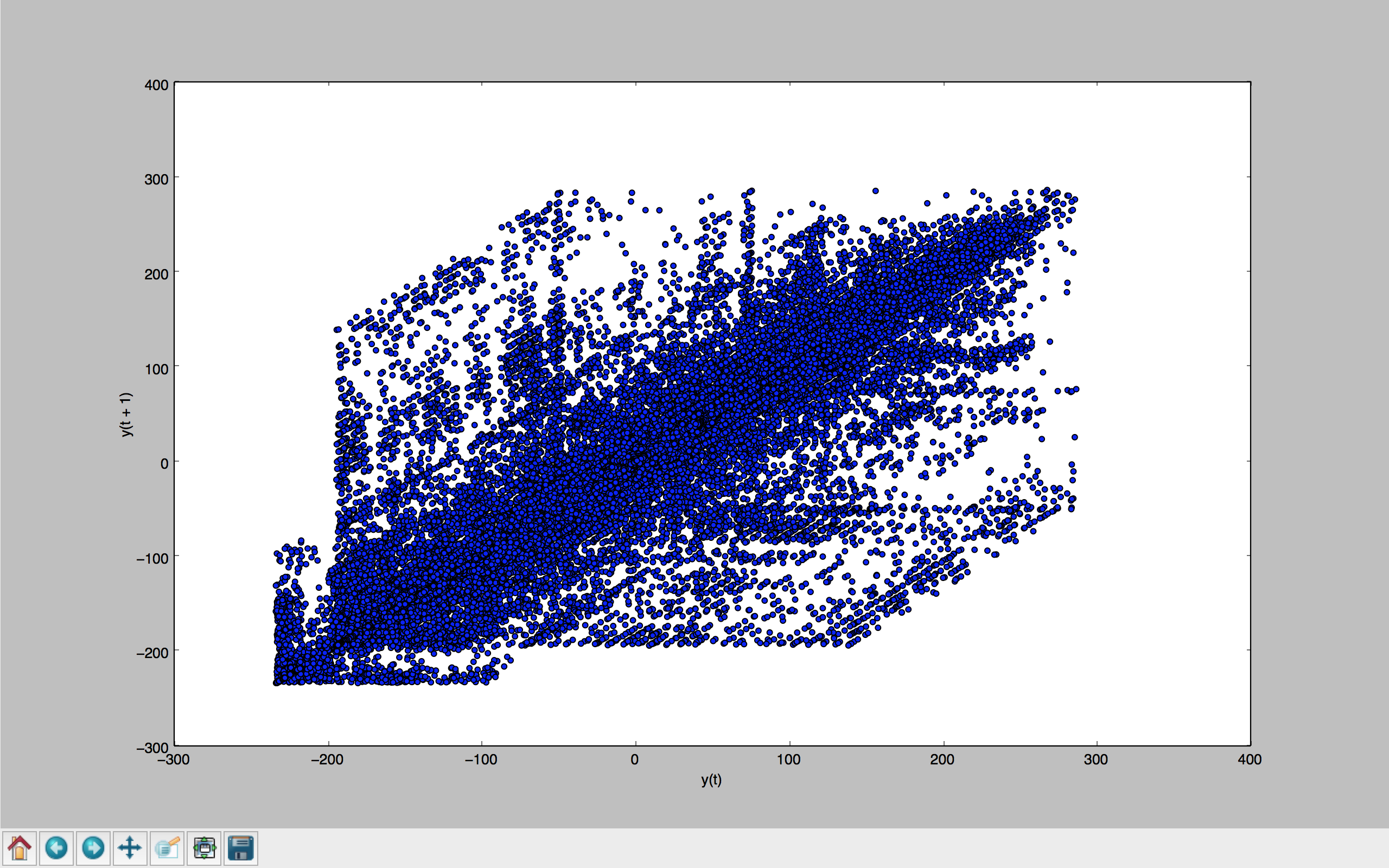
**Auto regression Model for Time Series Data:**

* Auto regression is a time series model that uses observations from the previous time steps as *input* to the regression equation to predict the value of the next time step.
* AR processes encapsulate a Markov like process where the future depends on the past
* Auto regression is like a linear regression model
  + Linear regression equation is given by

yhat = b0 + b1\*X1

* + This same technique is used in predicting time series where the input variables are extracted from the observations from the previous time steps in the series
  + Why Auto Regression:
    - Since the model uses data from the same input variable at previous time steps, it is referred to as auto regression
* So, for our weather prediction, before we can apply auto regression we need to study the correlation between the observation at the previous time step(t-1) which is the specific lagged variable and the next time step(t) which is the output variable
* The stronger the correlation between the output variable and the lagged variable, the more weight that auto regression can impose on the variable when modelling
* Interestingly, if all lag variables show low or no correlation with the output variable, then it suggests that the time series problem may not be predictable. This can be very useful when getting started on a new dataset.
* In our problem, we have carried our experiments on three states namely Seattle, Dallas and Cincinnati and based on the weather statistics by the US weather department, they are arranged in the following order in terms of their increasing unpredictability
  + Seattle
  + Dallas
  + Cincinnati
* Check for auto correlation
  + We create a lag plot using the pandas python library which plots the temperature (t) on the X-Axis against the temperature on the next hour (t+1) on the Y – Axis
  + The correlation plot indicates that a thick line of observations along the diagonal line which infers that there exists a strong correlation.
  + However, on comparing the plots for Seattle, Dallas and Cincinnati; it indicates that the concentration and dispersal of the points increases indicating the decrease in correlation



* AR Modelling
  + We split the dataset into 66% and 34% for testing and training, which is basically the hourly data for the year 2014 and 2015 for training and predicting the temperature for the year 2016
  + First we build the AR model by calling the AR() function of the statsmodel library and by calling the fit() function to train in our dataset.
  + The output lag value is 44 (which I am not sure why, but that’s the best value the AR model has chosen)
  + This implies the prediction is obtained by forming a regression equation with the values in the past 44 time steps with the saved co-efficients that were calculated
  + The MSE values for Seattle, Dallas and Cincinnati are 1.879, 2.394 and 4.247
  + This complements to our plot in auto correlation check step, where the weight of the correlation decreased for the three cities and consequently the MSE is also seen increasing for the AR model for time series.
  + Since we are building the model under the assumption that *the value of next instance solely depends on the current observation*, this model is more precisely called *AR(1) formulation*