# Workshop 4

The workshop this week is a little more open ended. You will be using Keras to solve a time series prediction problem.

1. Export the “International airline passengers” as a csv file from: <https://datamarket.com/data/set/22u3/international-airline-passengers-monthly-totals-in-thousands-jan-49-dec-60>
2. Load the data into Python and plot it using matplotlib

Think about how you would train a feed-forward network to predict the next value in the series given one or more previous values.

1. Write some code to create a dataset suitable for supervised training
   * You will need an X and Y training and test set (perhaps setup for k-folds)
   * What will the network input be?
   * What will the output be?
   * Do you need to normalise the data?
2. Design a network using **only** the Keras sequential model and any of the layers we have used so far. For example:
   * Dense
   * Conv (you can use 1D convolution)
   * Dropout
   * Hint 1: you will probably want to use “activation=None” for the final layer of your network. This is usually best when you are working with a regression problem rather than a classification problem.
   * Hint 2: for regression problems, the mean square error is usually the most appropriate loss function (loss='mean\_squared\_error').
   * Hint 3: you can also evaluate your network by looking at the MSE. You can use scikit-learn to do this:  
     from sklearn.metrics import mean\_squared\_error  
     score = mean\_squared\_error(Y\_test, model.predict(X\_test))
3. Compile and train your network
4. Plot your result and compare it to the real data (you should always visualise your output if you can).
5. Tune and/or restructure your network to see if you can get better performance