- Contents:

1. Packages
2. Dataset
   - cat vs non-cat
   - train set of m_train images labelled as cat (1)/non-cat (0)
   - test set of m_test images —— " —— —— " ——
   - image shape: (num_px, num_px, 3) → 3 channels
   - Reshape & standardise images
     - into vector & ~~flatten~~ make values 0 & 1
3. Model architecture
   - 2 models: 2-layer NN, L-layer NN

   3.1. 2-layer NN
   - Input: (64,64,3) image, flattened to a (12288,1) vector
   - $[x_0, x_1, \dots, x_{12287}]^T$ multiplied by $W^{[1]}$ of size $(n^{[1]}, 12288)$
   - Add bias term ($b^{[1]}$) & take its relu to get $[a_0^{[1]}, a_1^{[1]}, \dots, a_{n^{[1]}-1}^{[1]}]^T$
   - Repeat
   - Multiply resulting vector by $W^{[2]}$ and add bias
   - Take sigmoid of result; if >0.5, then cat

   3.2. L-layer NN
   - Same input
   - Same vector multiplied by $W^{[1]}$ & add $b^{[1]}$: result is the linear unit
   - Take ReLU of LU; repeat many times for each $W^{[l]}$, $b^{[l]}$, depending on model architecture
   - Take sigmoid of final LU; if >0.5, then cat

   3.3. General methodology
   - DL methodology:
     1. Initialise params / define hyperparams.
     2. Loop for num_iterations:
        a. For. prop.
        b. Compute CF
        c. Backprop.
        d. Update params. (using parameters & grads from backprop)
     3. Use trained params. to predict label