* Adding “Colorless green ideas sleep furiously.” Worked as intended
  + The sentence popped to the top of the rankings of both the PCA and ICA dimensionally reduced matrices
* How to calculate number of bins
  + Sqrt(# of elements)
  + Freedman-Diaconis (in test notebook)
  + Hideaki Shimazaki method (gives edges not # of bins)
    - Method based off this paper: <https://s-shinomoto.com/papersPDF/NeCo07.pdf>
  + Another popular one seems to be Sturges Rule however, for large n (n > 200) is it not very accurate
    - <https://robjhyndman.com/papers/sturges.pdf>
  + Scott’s method
    - Seems to return a value for # of bins between Shimazaki and Freedman-Diaconis
* Further Research
  + Mahalanobis distance instead of surprisal values
  + Principal component weighted surprisal
    - Seems to do very little if anything to the results
  + Entropy?
  + Other ways to determine PDF
* Can’t tell if its with highly homogenous data or with a smallish data set but at these extremes, ICA does not seem to work as well as PCA
* Obviously, results of inputting the weird sentence rely heavily on how homogenous the set of sentences is
  + If quite homogenous then the sentence will pop up to the top, if quite heterogenous then the sentence will drift towards the top but may not be the top result
* Great explanation of ICA
  + “ICA essentially reverses the central limit theorem: if averages (or generally speaking: linear combinations) of data becomes more and more Gaussian, then one way to uncover the original signals from a linear combination of unknown sources is to look for those that are most non-Gaussian. “
* Polynomial KPCA seems to work somewhat well, but not incredibly well
* Gaussian RBF KPCA for some reason completely flipped the results where the weirder sentences were at the bottom
* Jabberwocky
* Formalize a rubric to grade the dim reduction methods
  + Stability
  + How good it is
  + Timing
* Statistics for ranked lists
  + How to vizualise and grade
* Add into presentation that KDE oversmoothed
* Things that didn’t
* Maybe try to use different languages for sentences
* RBO for comparing sentences
  + Maybe can arbitrarily rank top 10% of sentences or bottom 10% of sentences
* Training took 1:28

Rubric for Grading:

Tests for each dataset of sentences:

1. Raking of weird sentence “Colorless green ideas sleep furiously”
2. I go through each of the methods and look at the top 20, compile every sentence in all of the top 20s and rank them according to how weird I feel them to be and then score each method
   1. Can maybe compare the top 10% of sentences
3. Maybe test with different number of components for each of the methods and see how well it identifies the weird sentences
   1. Can maybe find minimum number of components that still ranks the “Colorless green ideas sleep furiously” at the top
4. Maybe do the same thing as (2) but with the last 20 or so of each of the methods and score based off how normal the sentences are