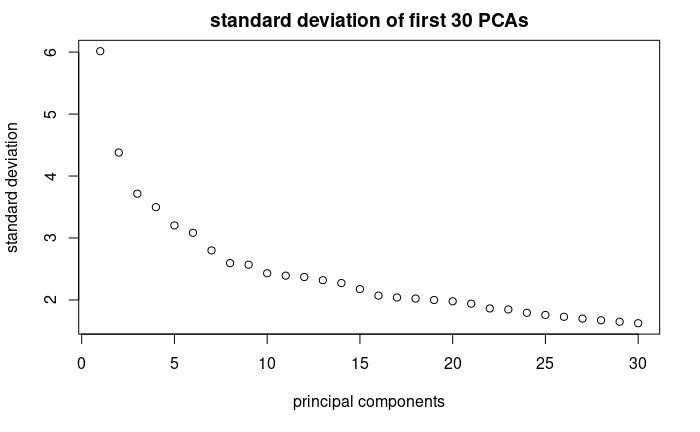
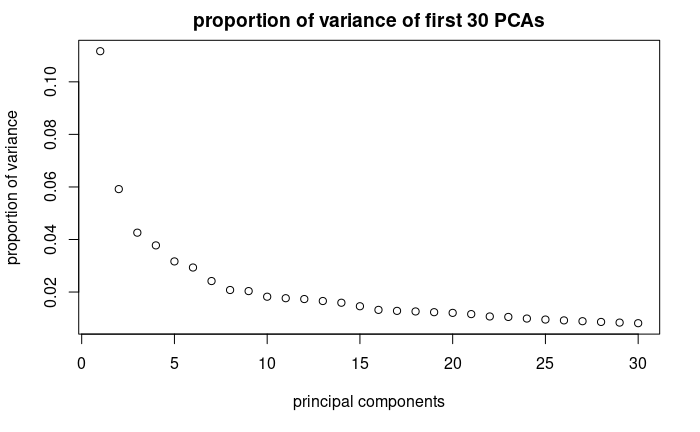
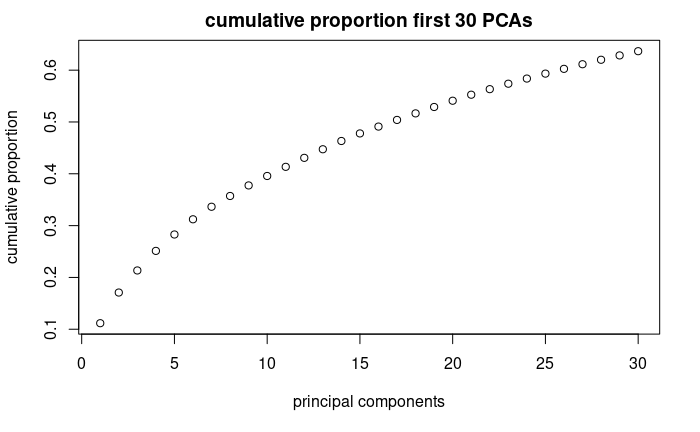
*Group 4*

*Exercise 2 – Data Preprocessing*

*Ines Benomar, Vincenzo Coppola, Karol Szurkowski, Erick Kockar*

This exercise shows how we can use PCA to reduce the dimensions of the dataset with digits into a representation that uses only the most important features and then classify the reduced dataset using the KNN algorithm.

Exercise 2.1: Principal Component Analysis (PCA)

  
Here we can observe how the first principal components accumulate the biggest amount of standard deviation. That simply is a effect of rotating the axis of PC so that it encapsulates the highest possible variance.  
  
  
  
This plot shows how the first components of SVD contribute to the variance of the whole dataset. We can observe that even though the first loading vector (pricinpal component) covers the highest variance - it is barely only 11% of the whole dataset.  
The cumulative variance of the sequential principal components can be observed on the next graph.  
  
  
Most data can't be well-described by a single principal component, using more principal components will reconstruct the more precise data with the growing trend similar to logarithmic function.

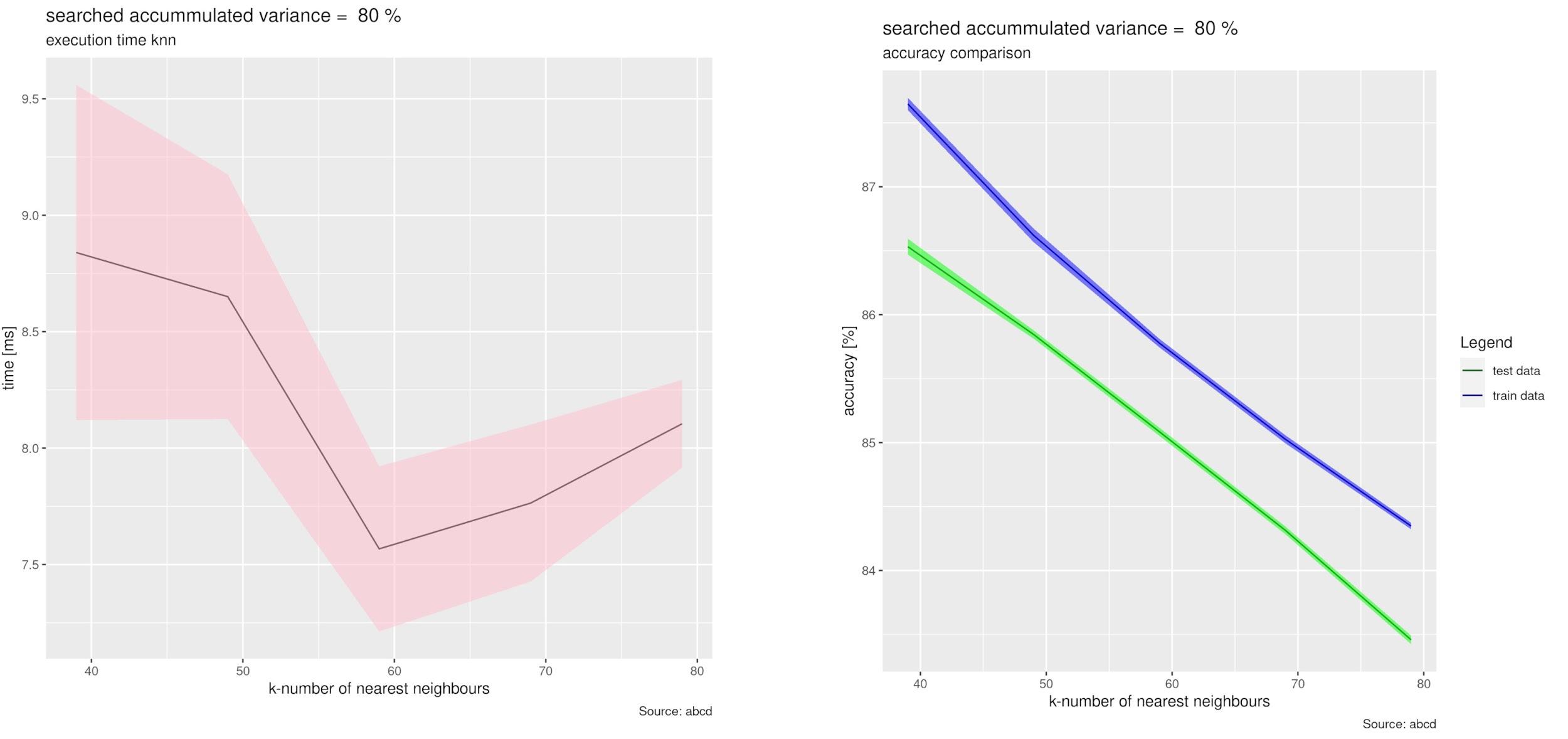
Hence, if we want to collectively explain more of the variability in the original set we need to compute multiple principal components by computing all eigenvectors of and ranking them by their eigenvalues. This can be visualized in the variance plots above.

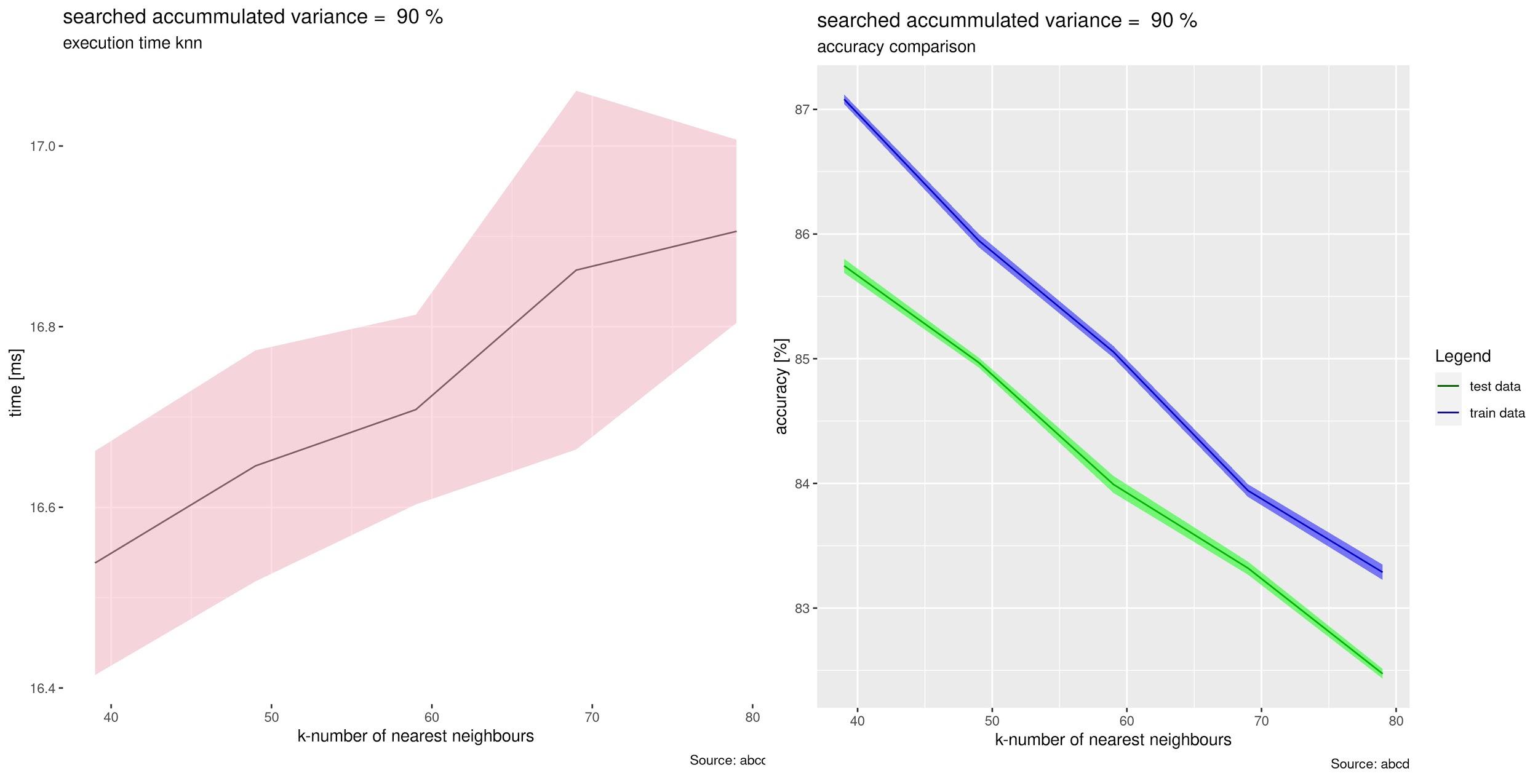
From the 3rd plot, we can see that in the case of the information of the images of digits, 50% of the information i.e. 50% of cumulative proportion of variance can be described using only 17 first principal components.

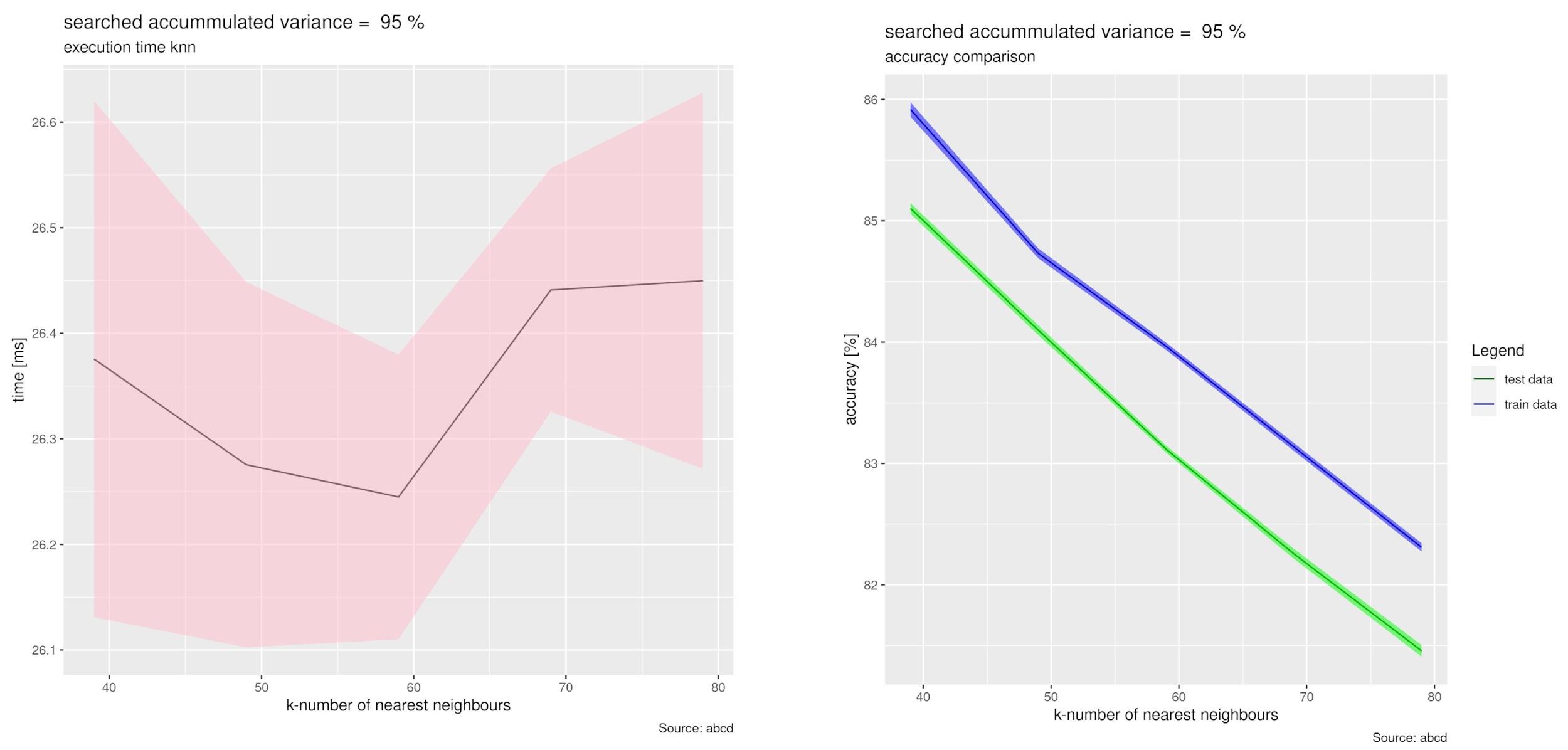
**Exercise 2.1.2 Show the performance of selecting enough principal components to represent 80%, 90%, 95%, 99% of the accumulated variance. For each test vary “k” in kNN, try reasonable values.  
  
 All in Data:**

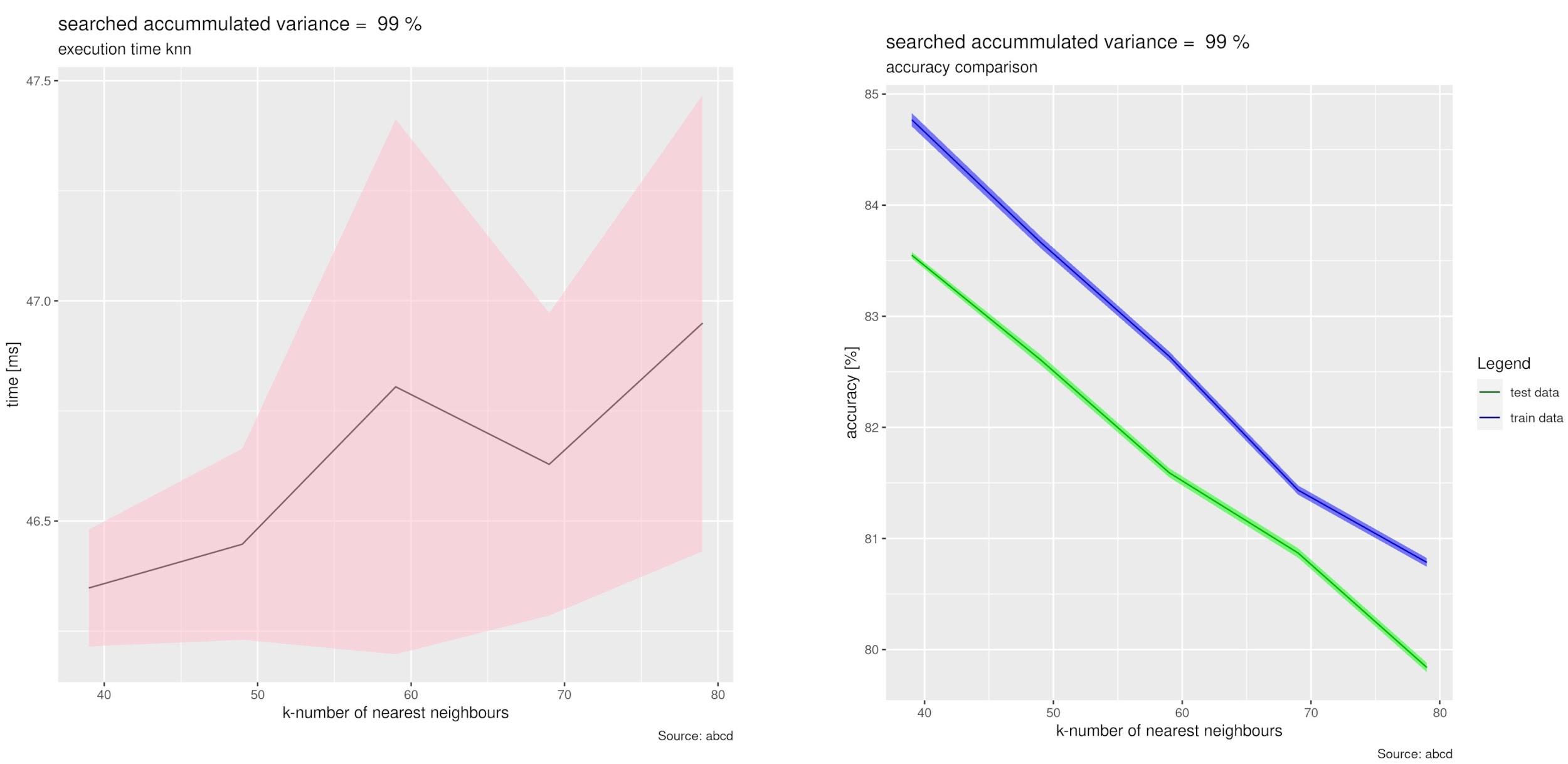
Here we have just chosen enough PCs to cover the amount of accumulated variance.

To get the 80, 90, 95, 99 % of accumulated variance we used 61, 103, 149, 248 first components from the data obtained from 10 people .





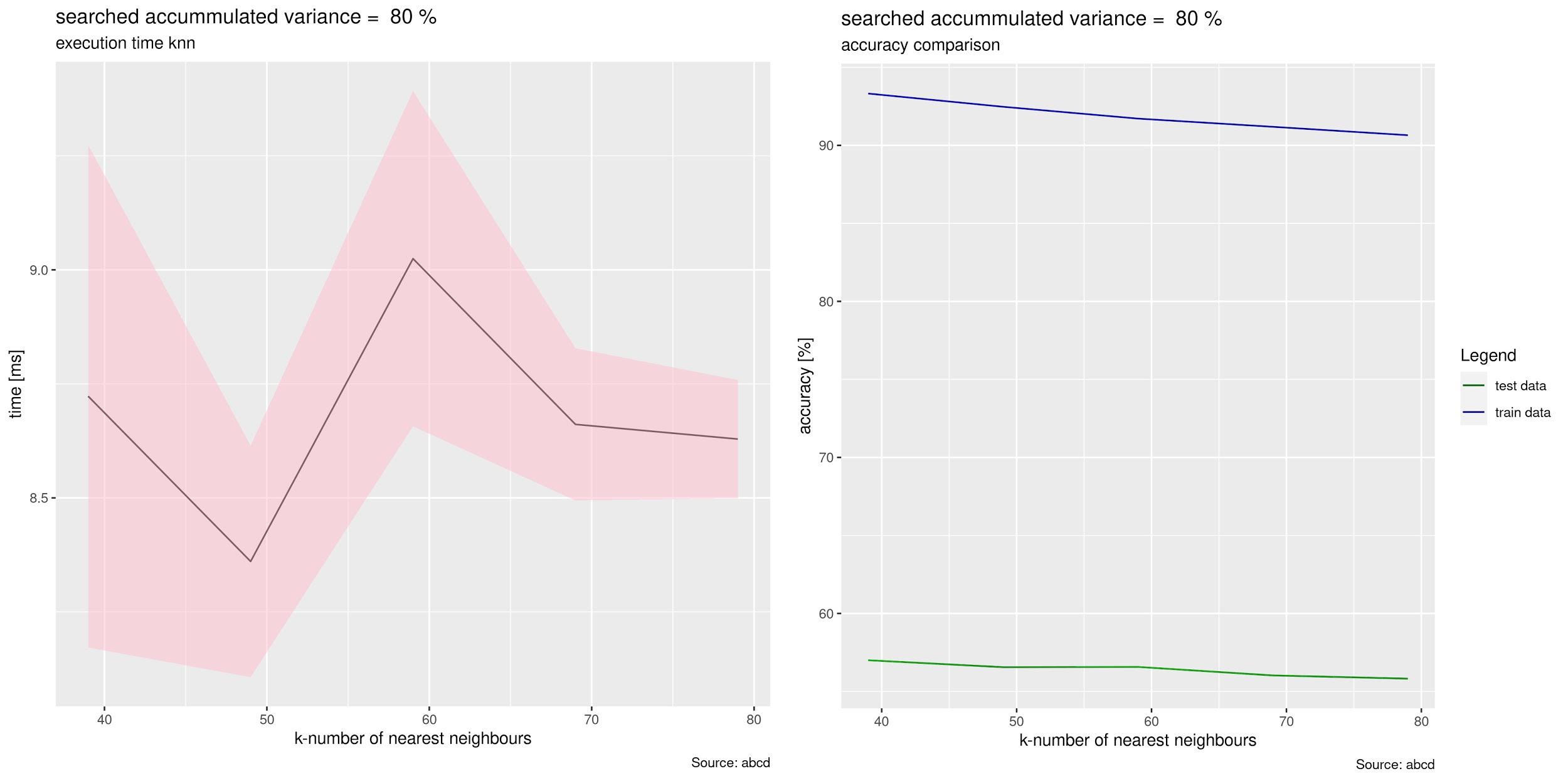


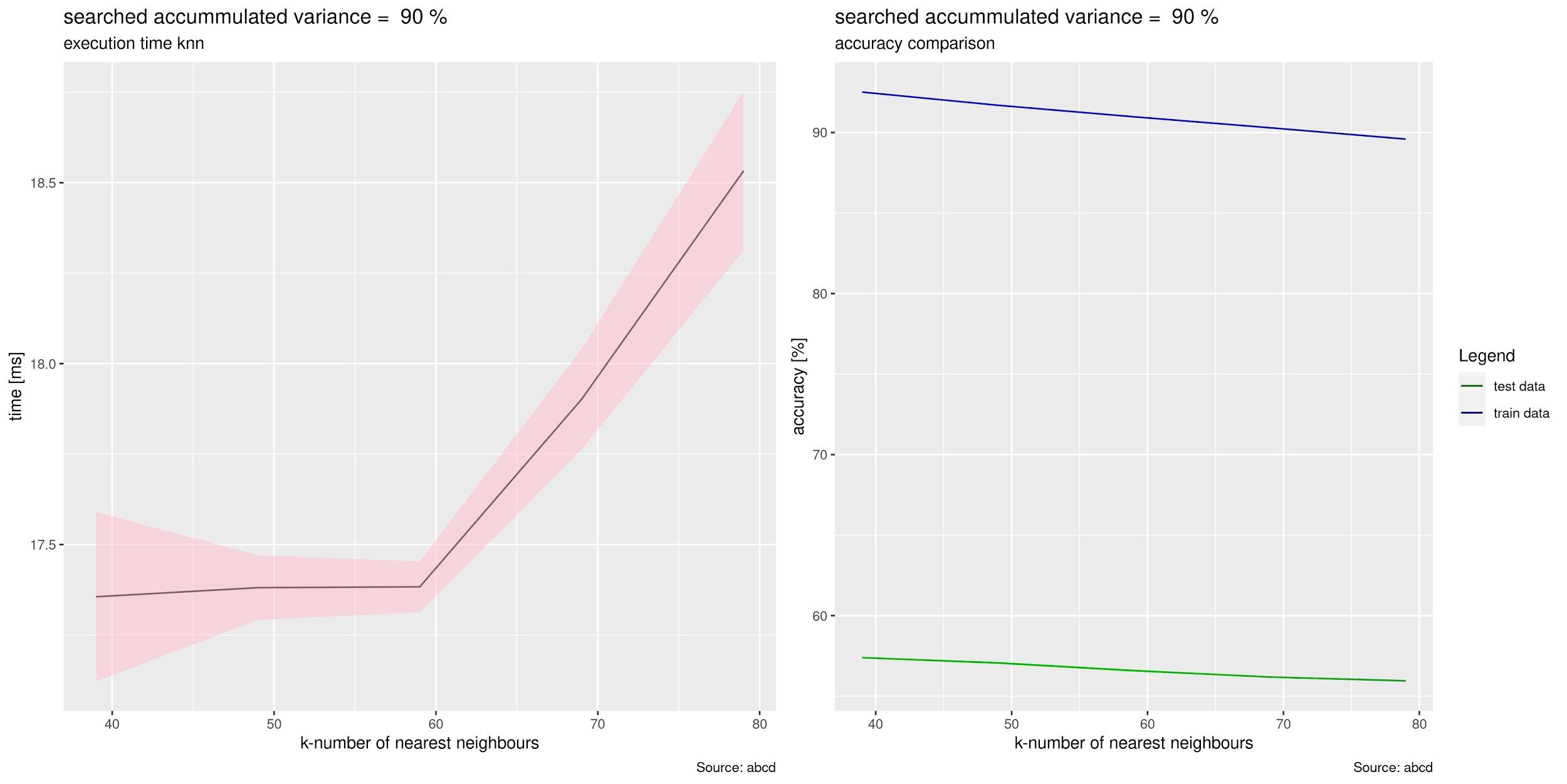


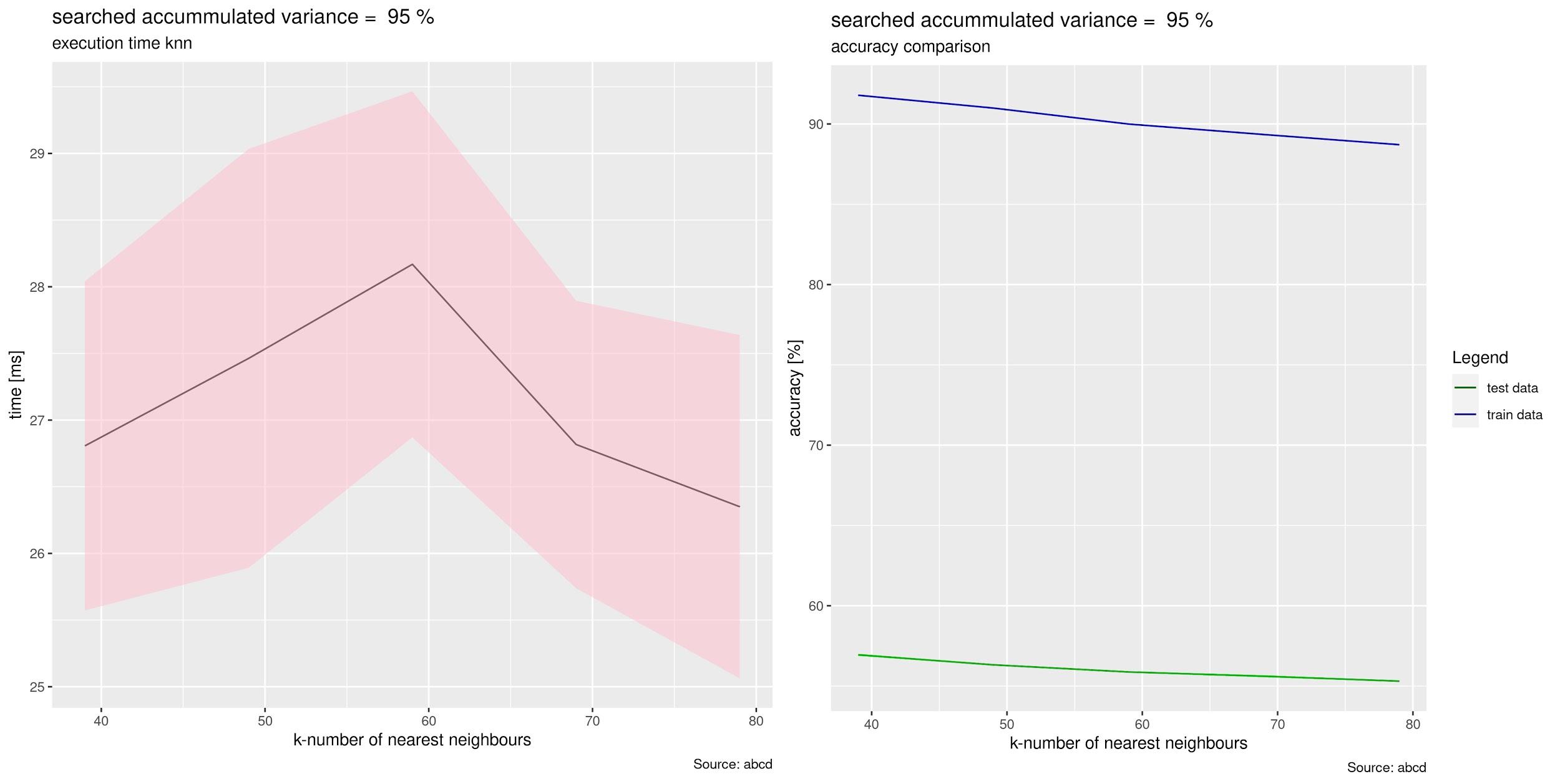
With higher number of used Principal Components we observe significant increase in computational time. Surprisingly our method of evaluation of accuracy shows that with the higher number of PCs, the accuracy of classification decreases.

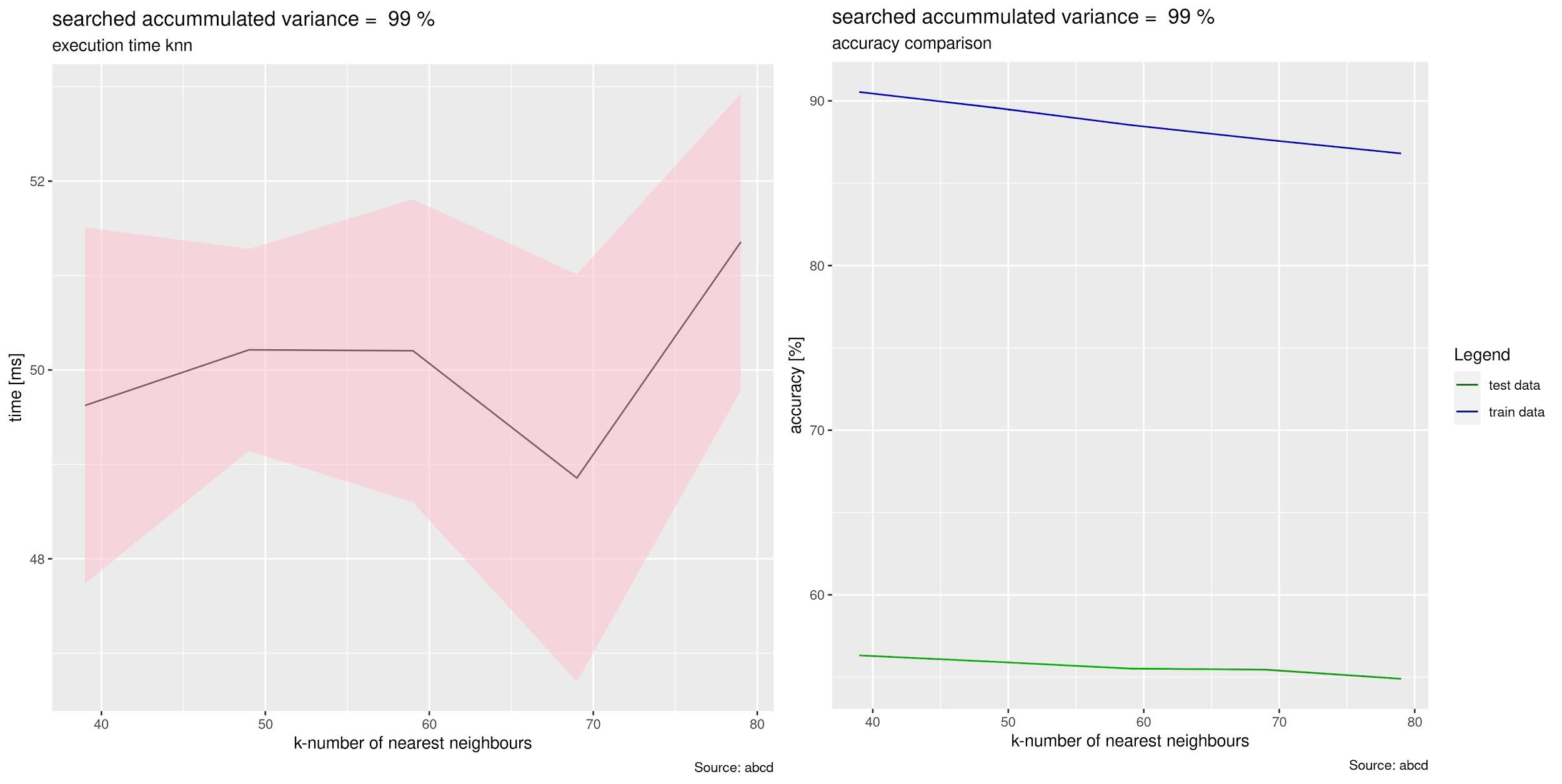
**Disjunct data:**

Following plots present principal component decomposition on a disjunct dataset. We have used data from 5 people as a training set, and remaining 5 as testing set.



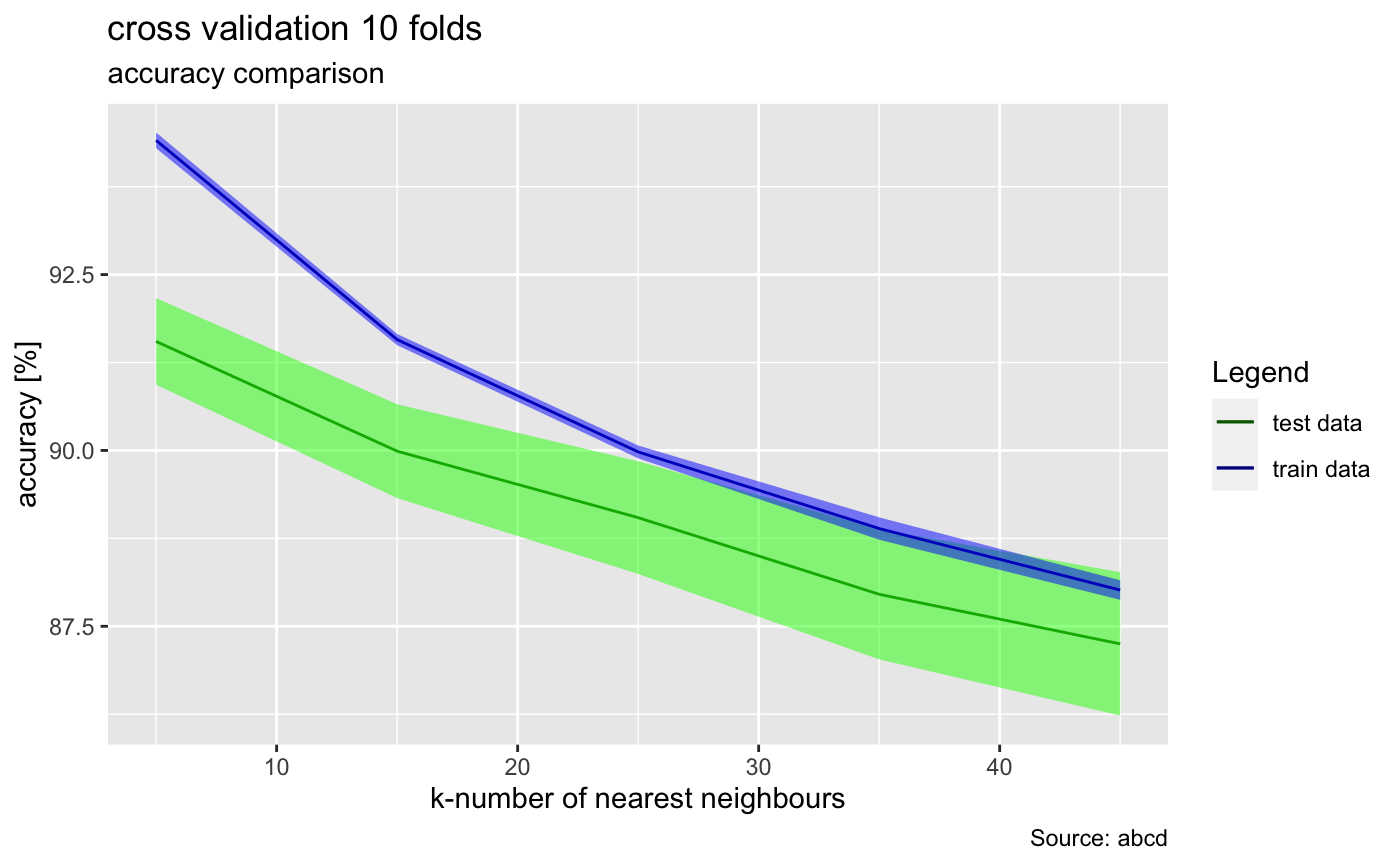


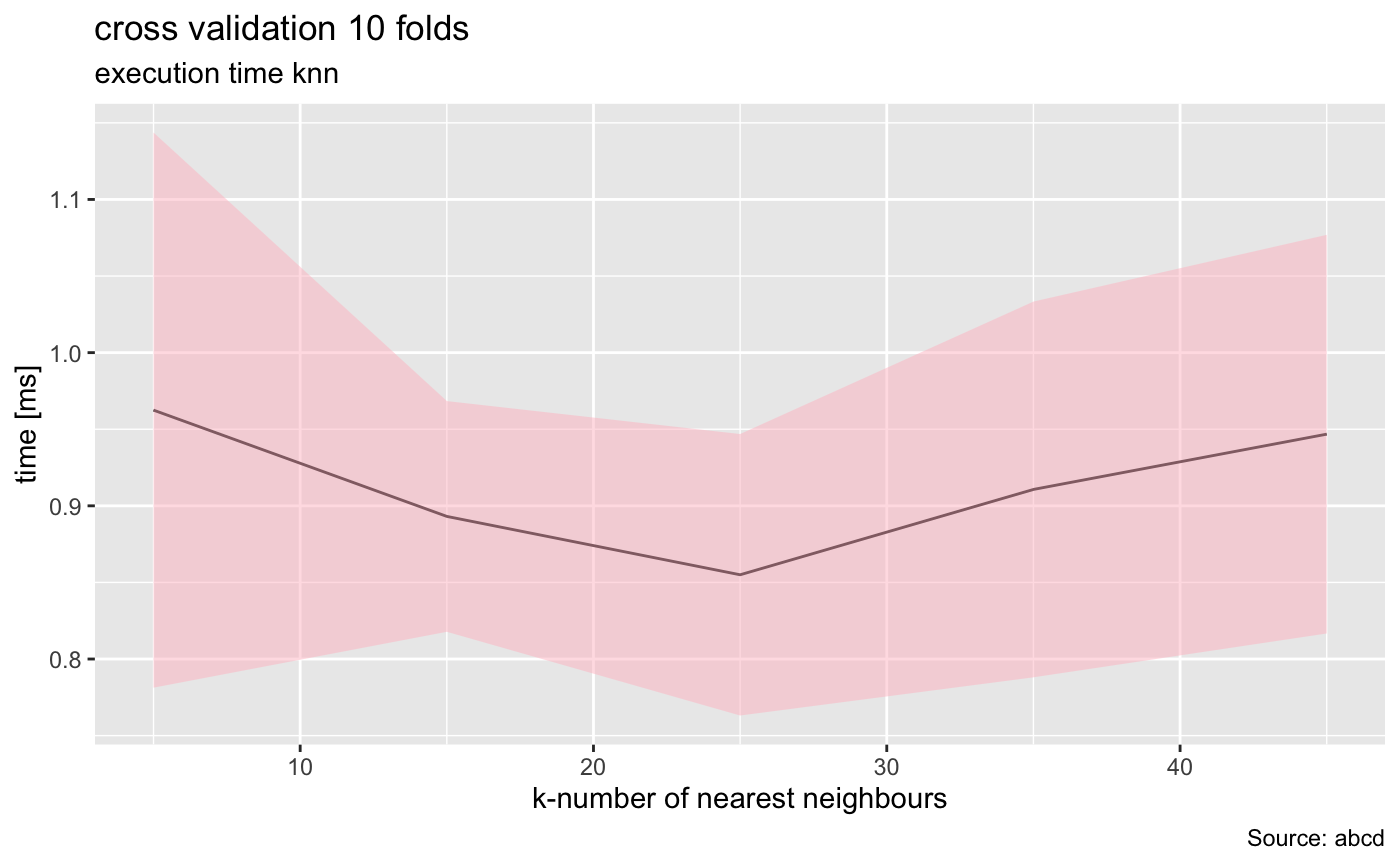




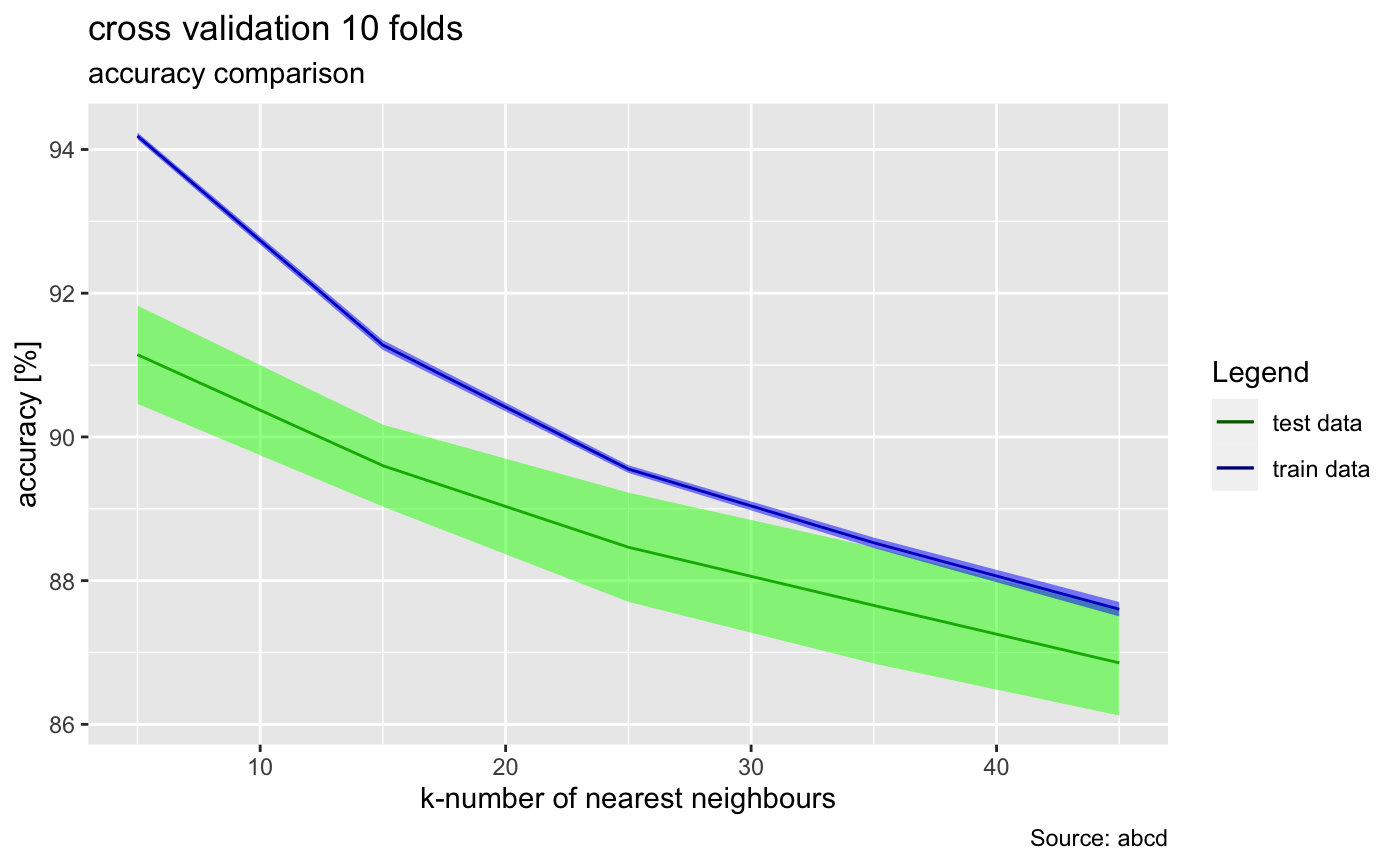
We observe that the accuracy of test data in general doesn’t exceed 60% and the accuracy of training data stays high.   
Surprisingly accuracy of test data doesn’t vary even using the higher number of principal components.

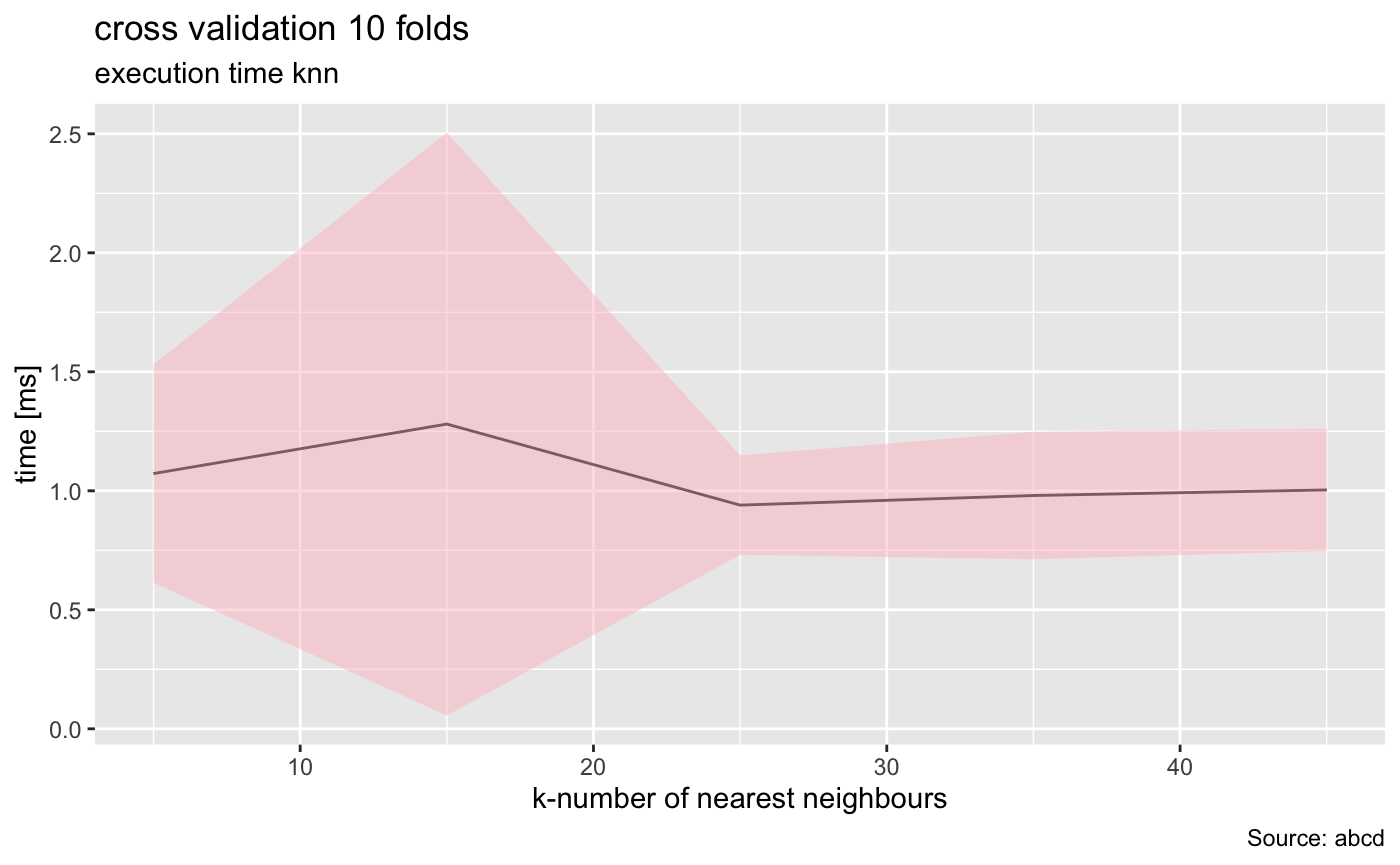
* **Exercise 2.2: Normalization**
* Min-Max Normalization Before PCA





* Min-Max Normalization After PCA



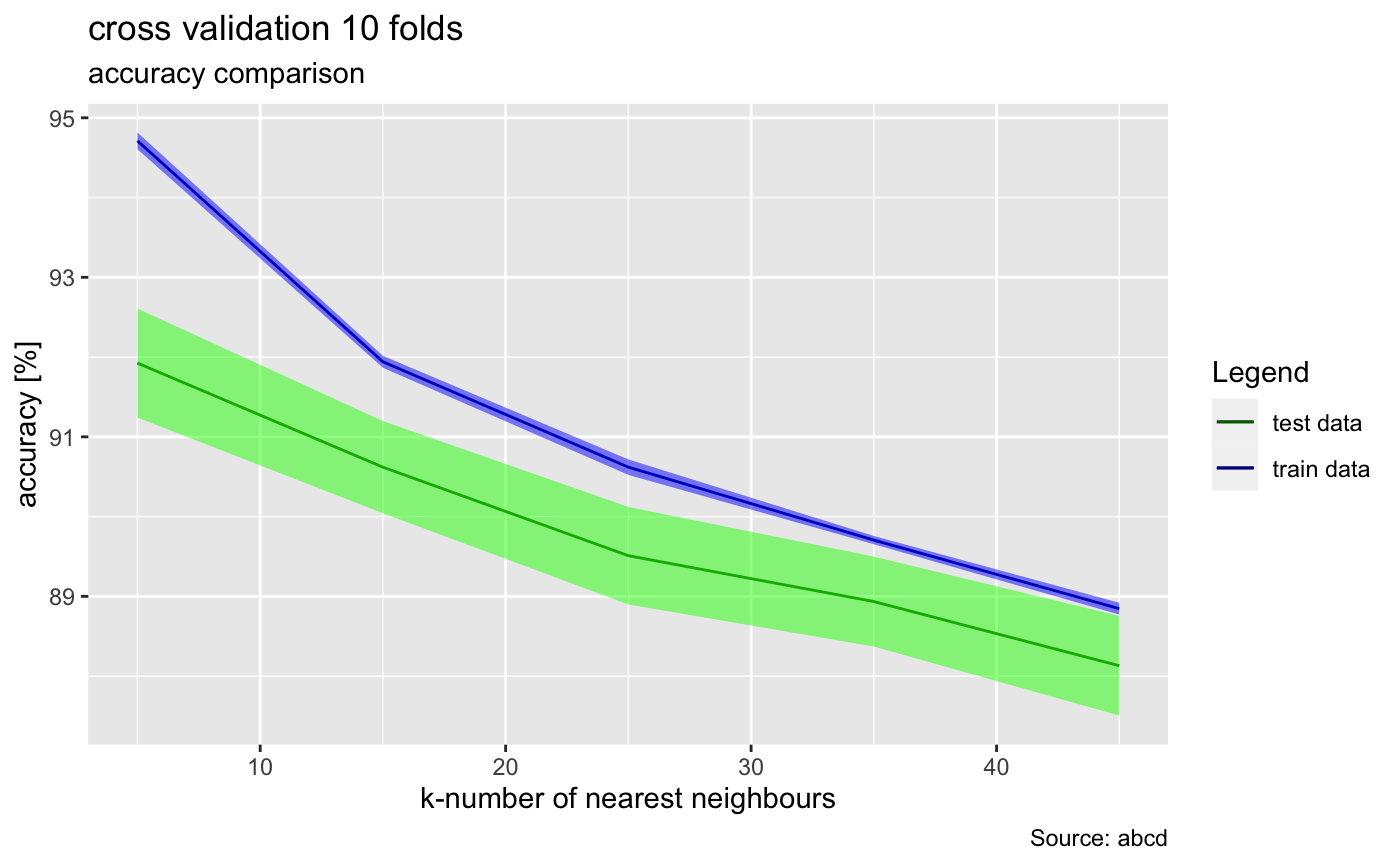
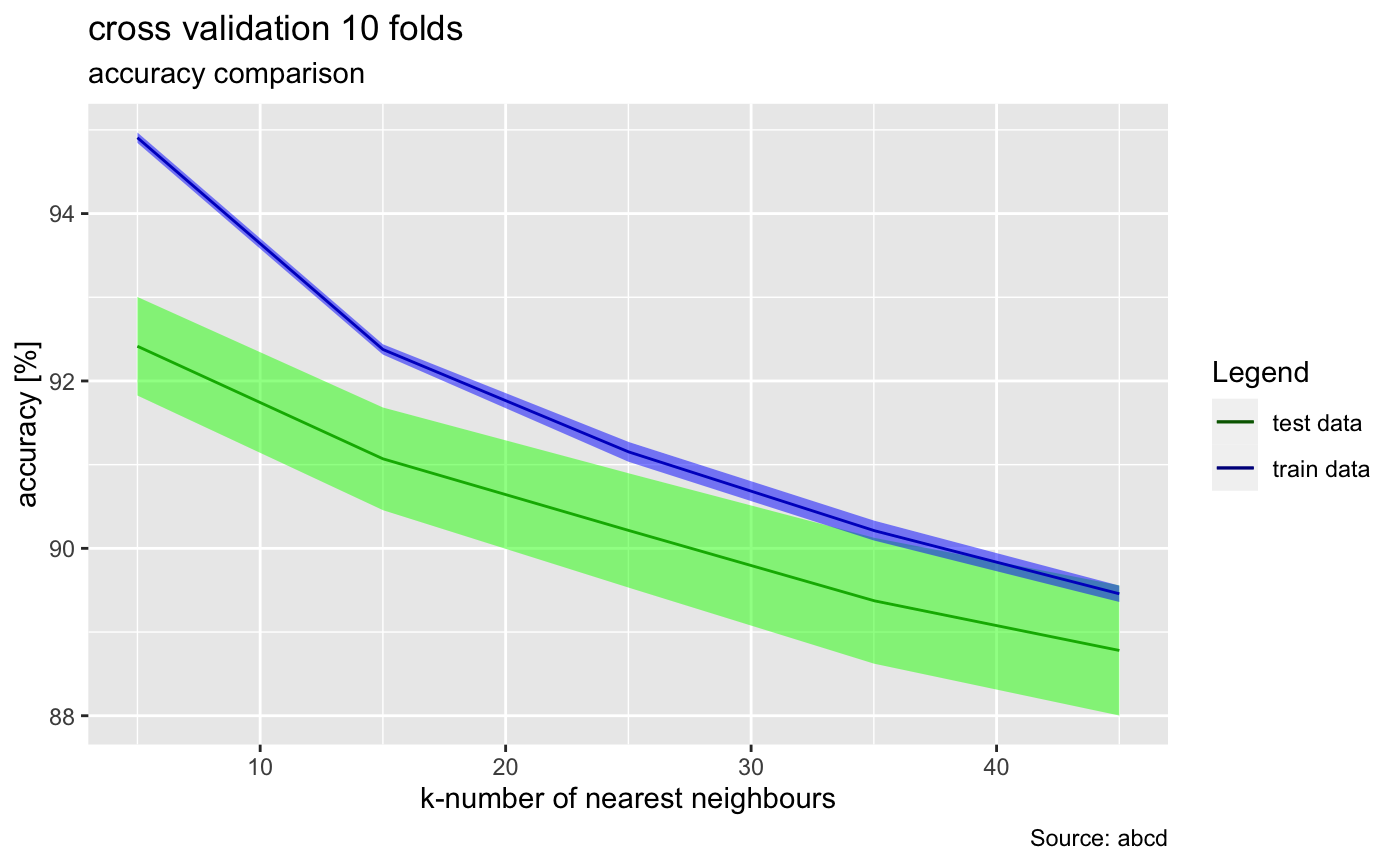


The PCA calculates a new projection of the data set and the new axes are based on the standard deviation of the variables. So a variable with a high standard deviation will have a higher weight for the calculation of axis than a variable with a low standard deviation. When we normalize the data, all variables have the same standard deviation, thus all variables have the same weight and the PCA can calculate the relevant axis.

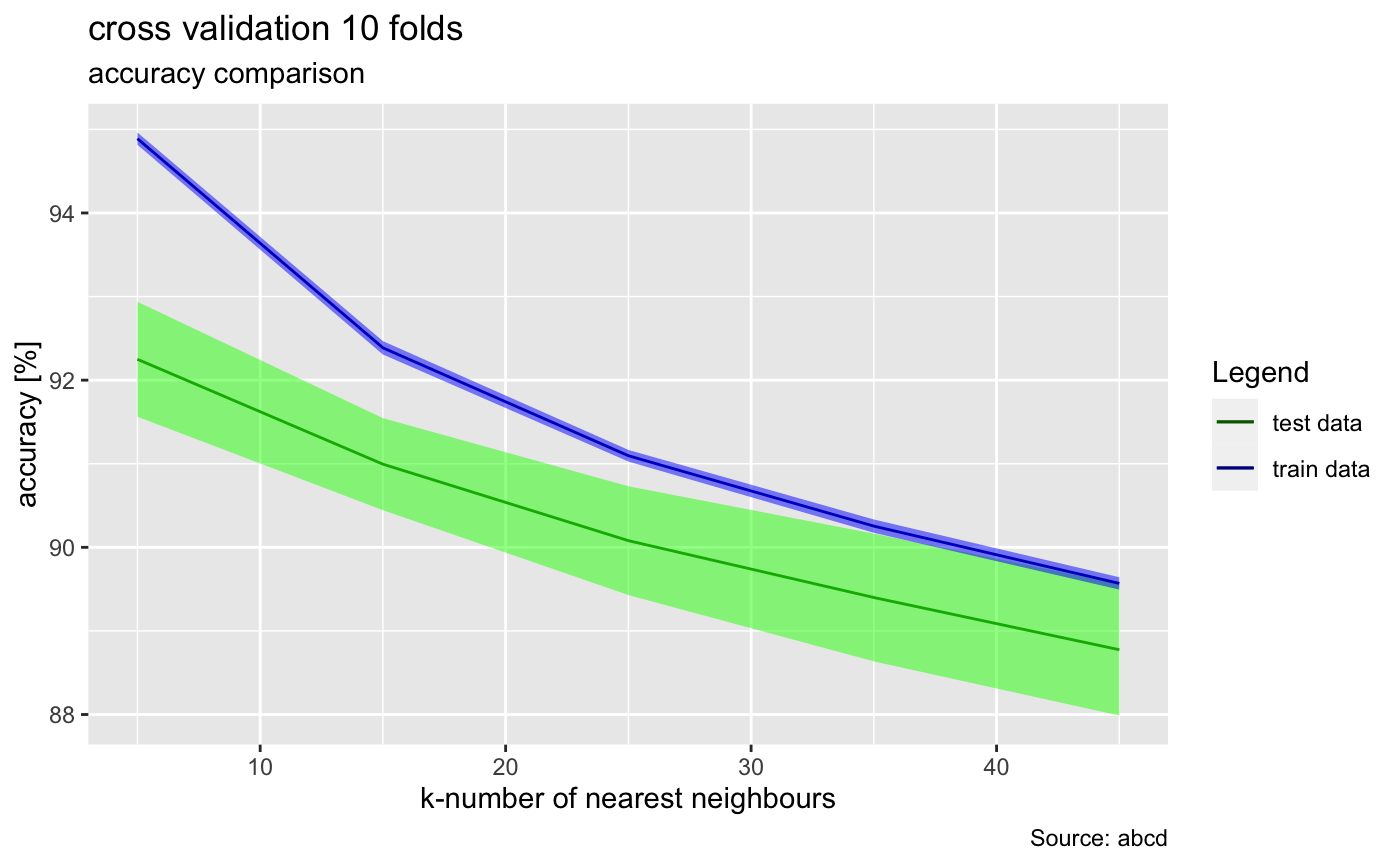
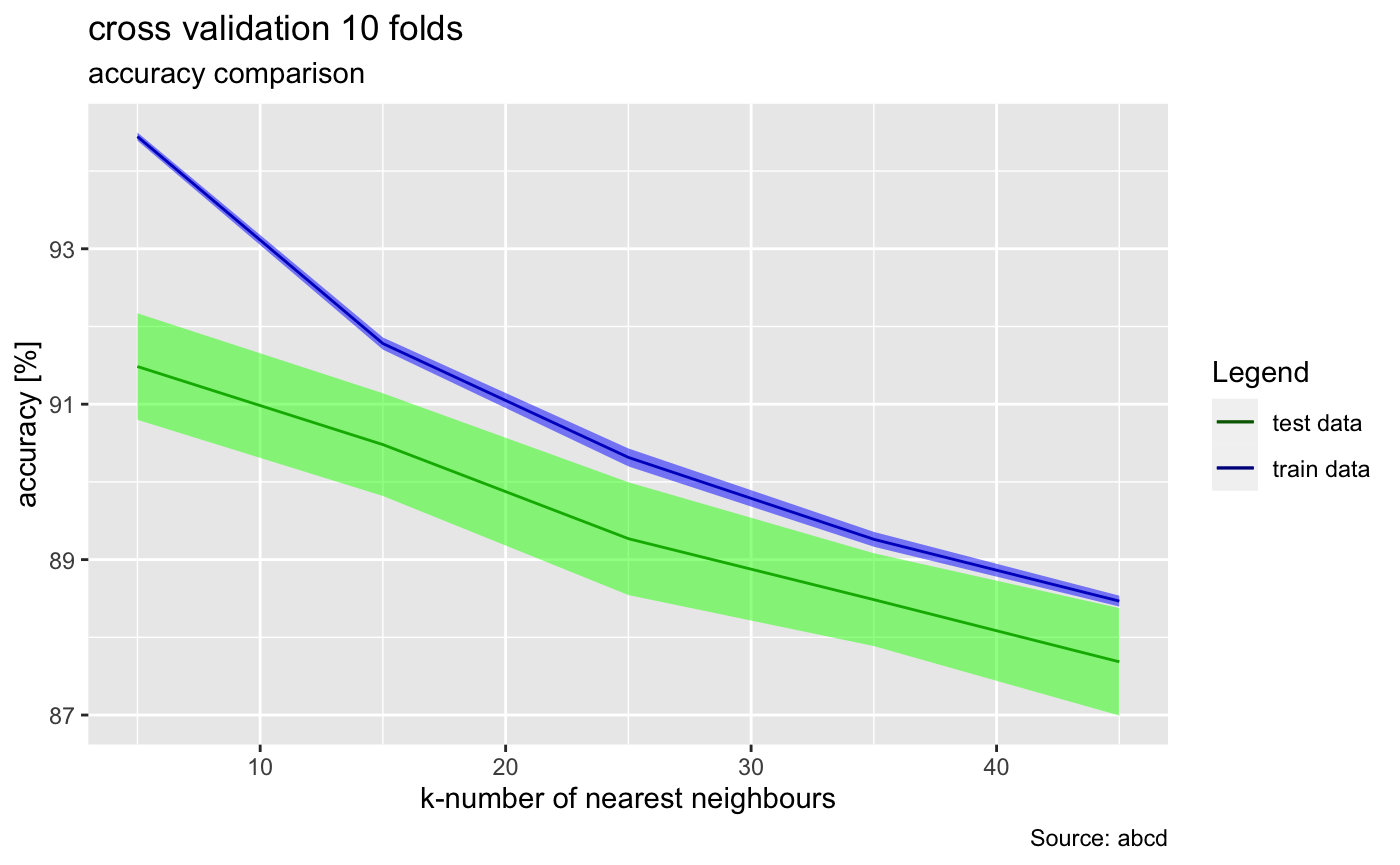
This is especially important when the original predictors have different scales, however given that in our case our loadings have the same scales we don't expect to see that much difference between normalized data and raw data. (with unscaled and scaled predictors)

In order to see this, we can compare the plots generated in 2.1 to the ones in 2.2.

* **Exercise 2.3: Pre-processing**
* **Min-Max Normalization Before PCA with Smoothing**
  + Sigma = 0.5 (left), Sigma = 1.0 (right)

* **Min-Max Normalization After PCA with Smoothing**
  + Sigma = 0.5 (left), Sigma = 1.0 (right)



Some notes for myself (that I still need to organize)

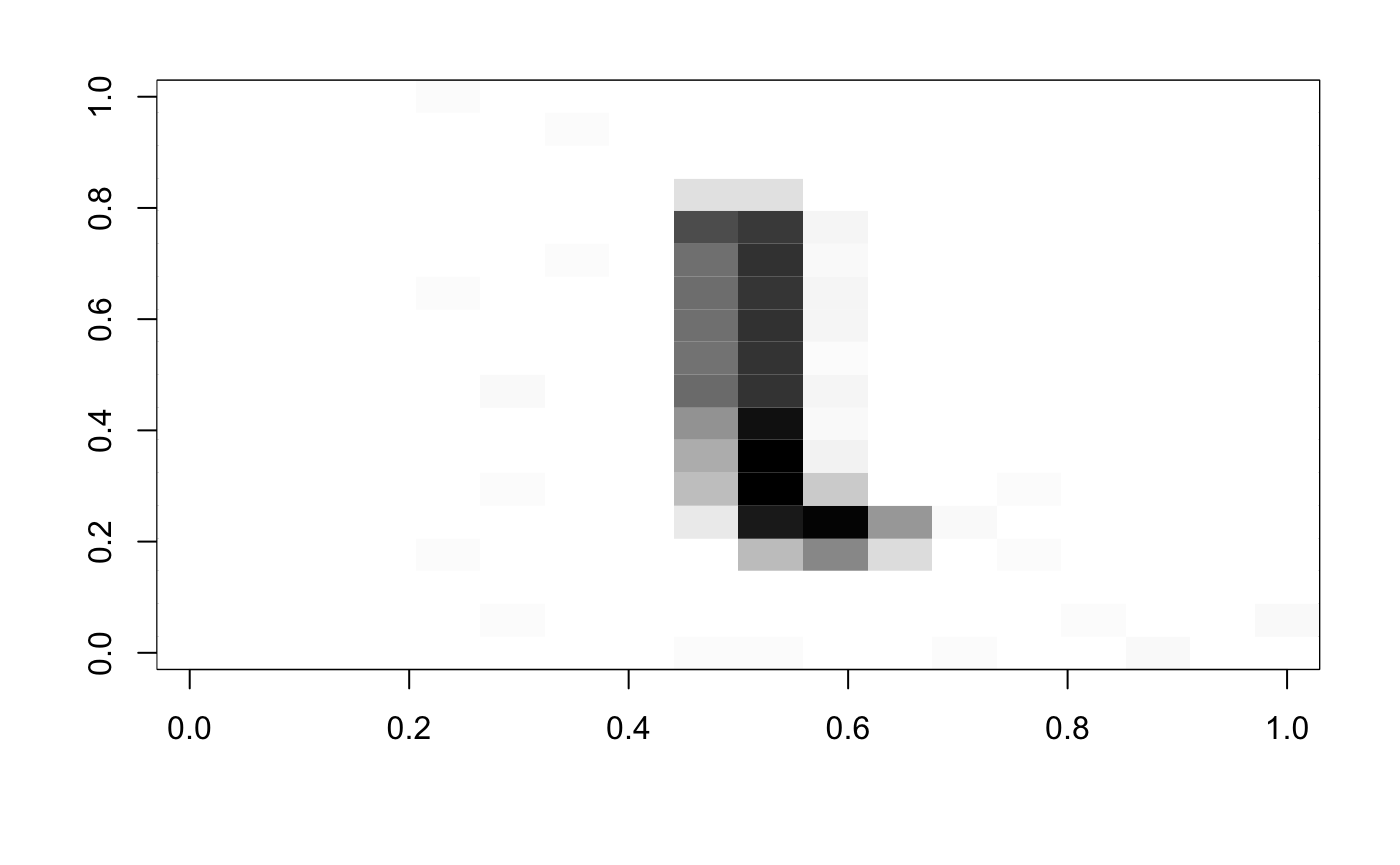
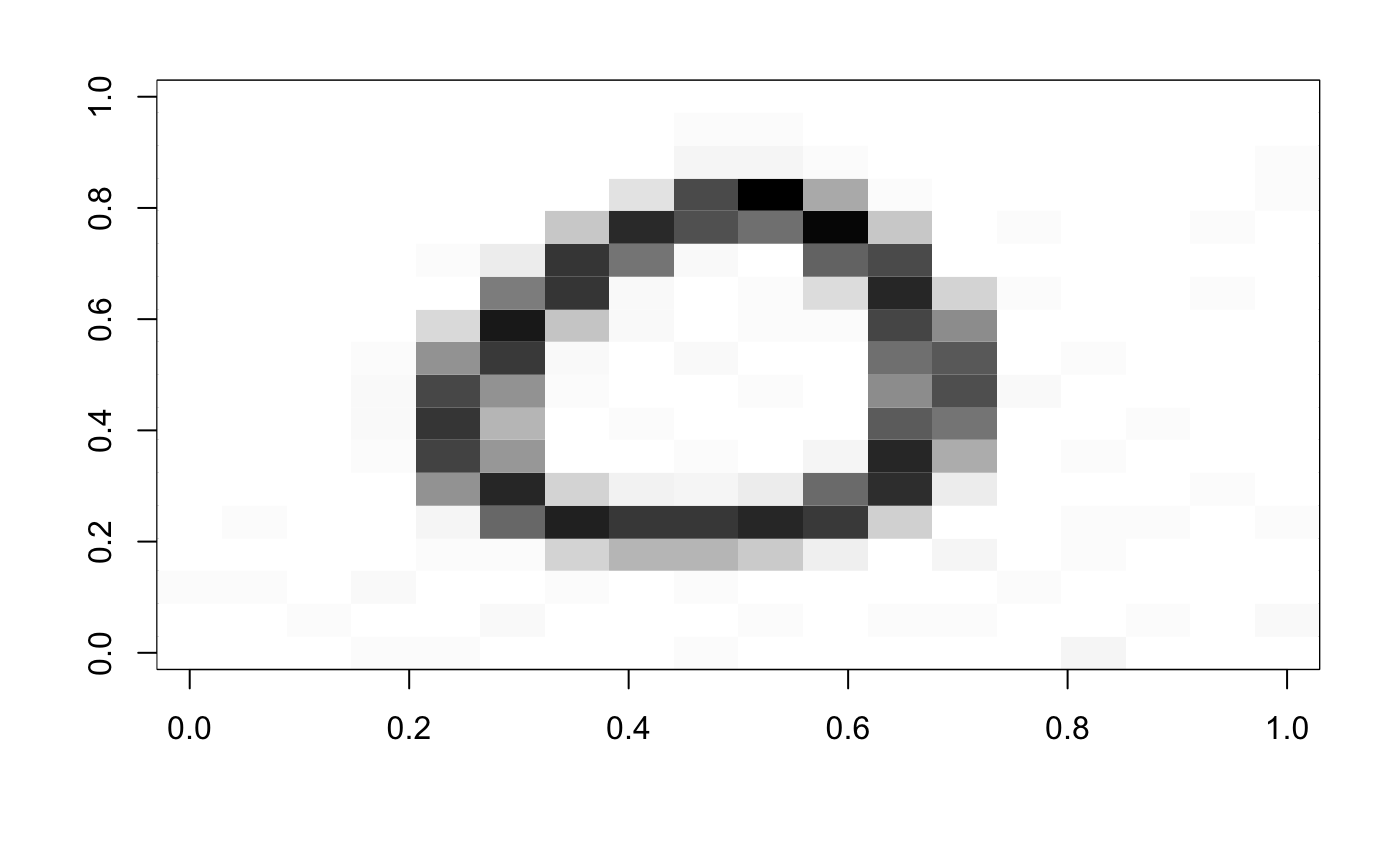
- Sigma is the variance (i.e. standard deviation squared). If you increase standard deviation in normal distribution, the distribution will be more spread out (the more variance allowed around mean), and the peak will be less spiky. Similarly in gaussian smoothing, which is a low pass filter, it makes everything blurry, by de-emphasising sharp gradient changes in the image, thus if you increase the variance / stddev, it will be more blurry. But this is limited by the size of your gaussian kernel

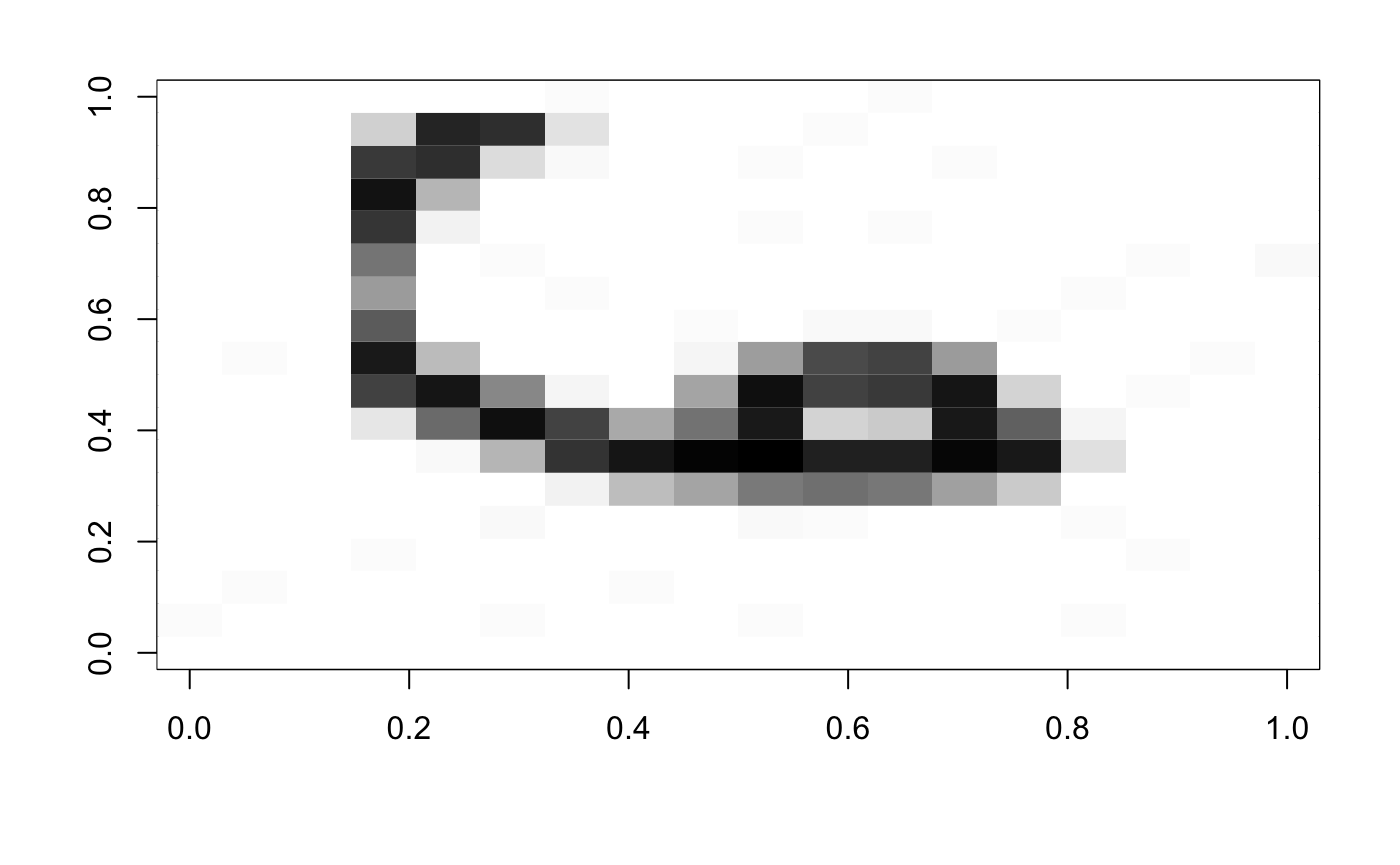
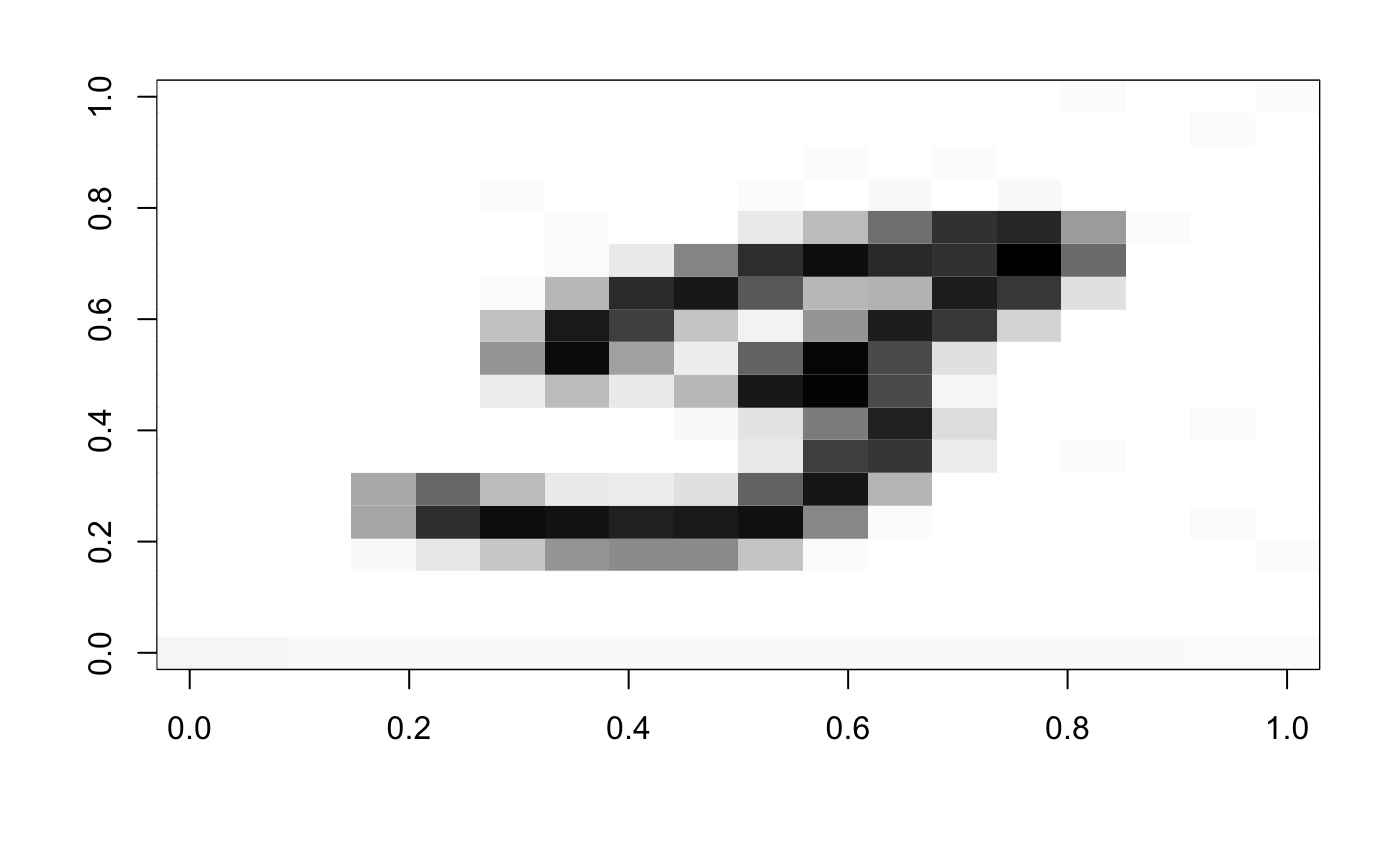
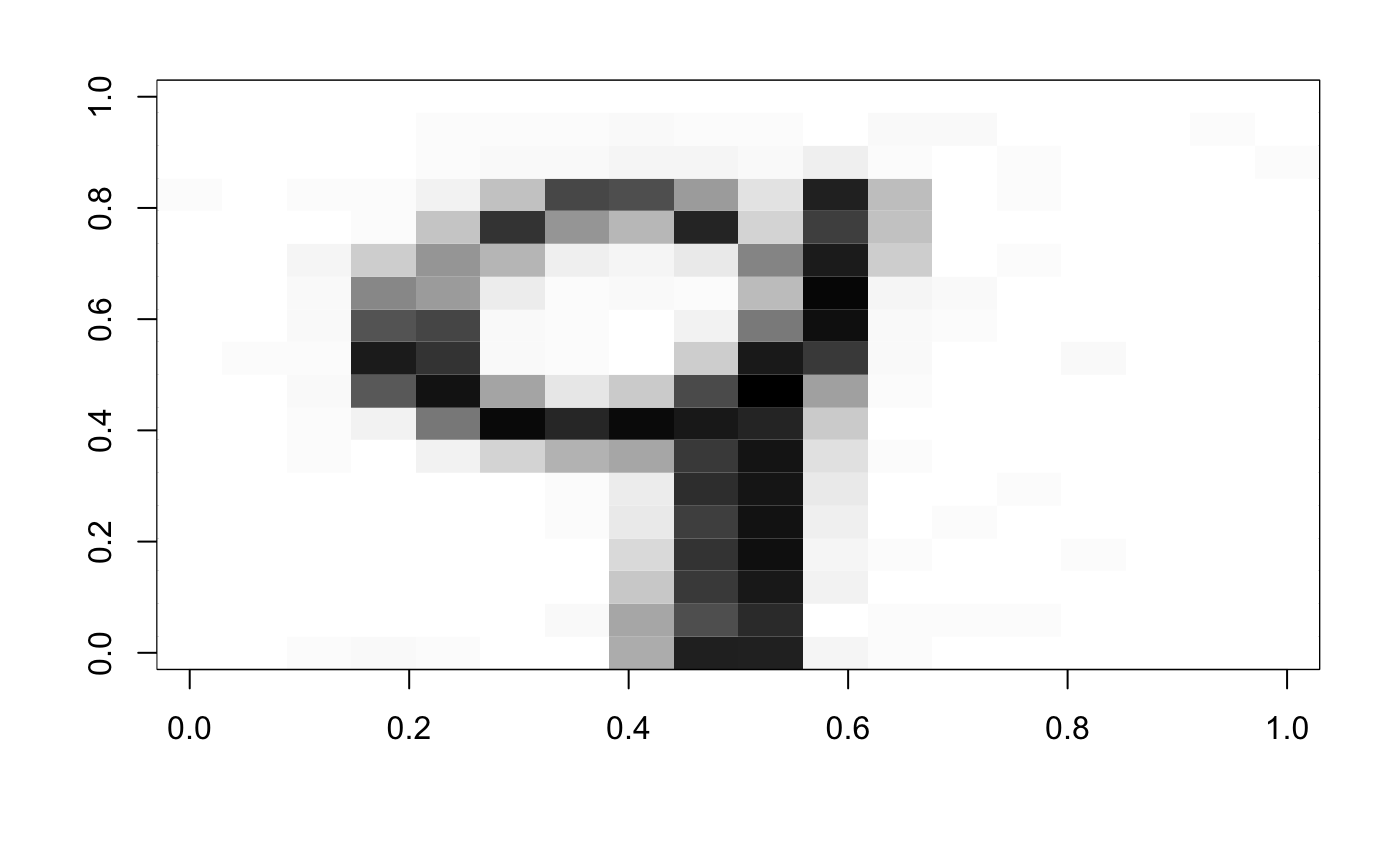
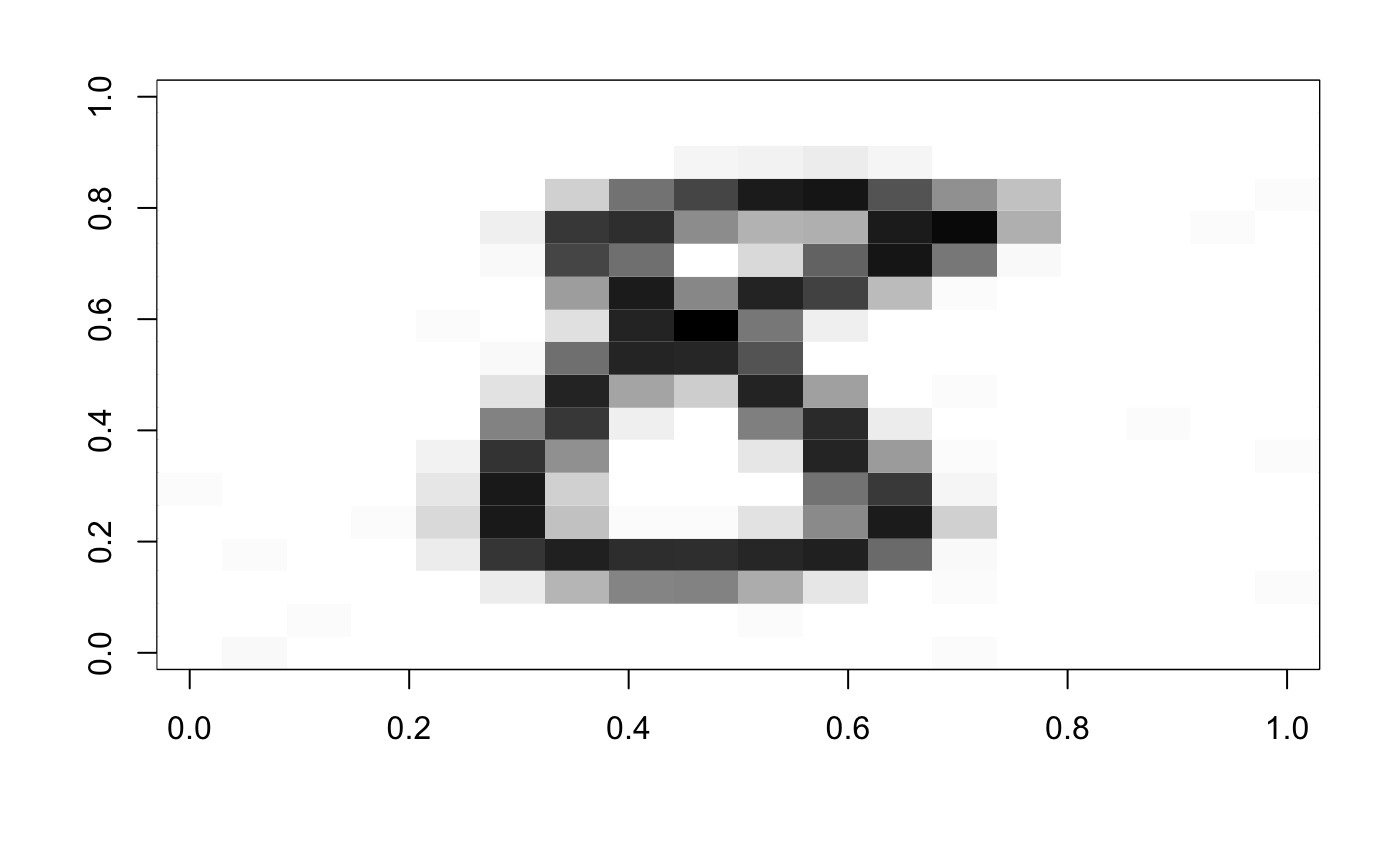
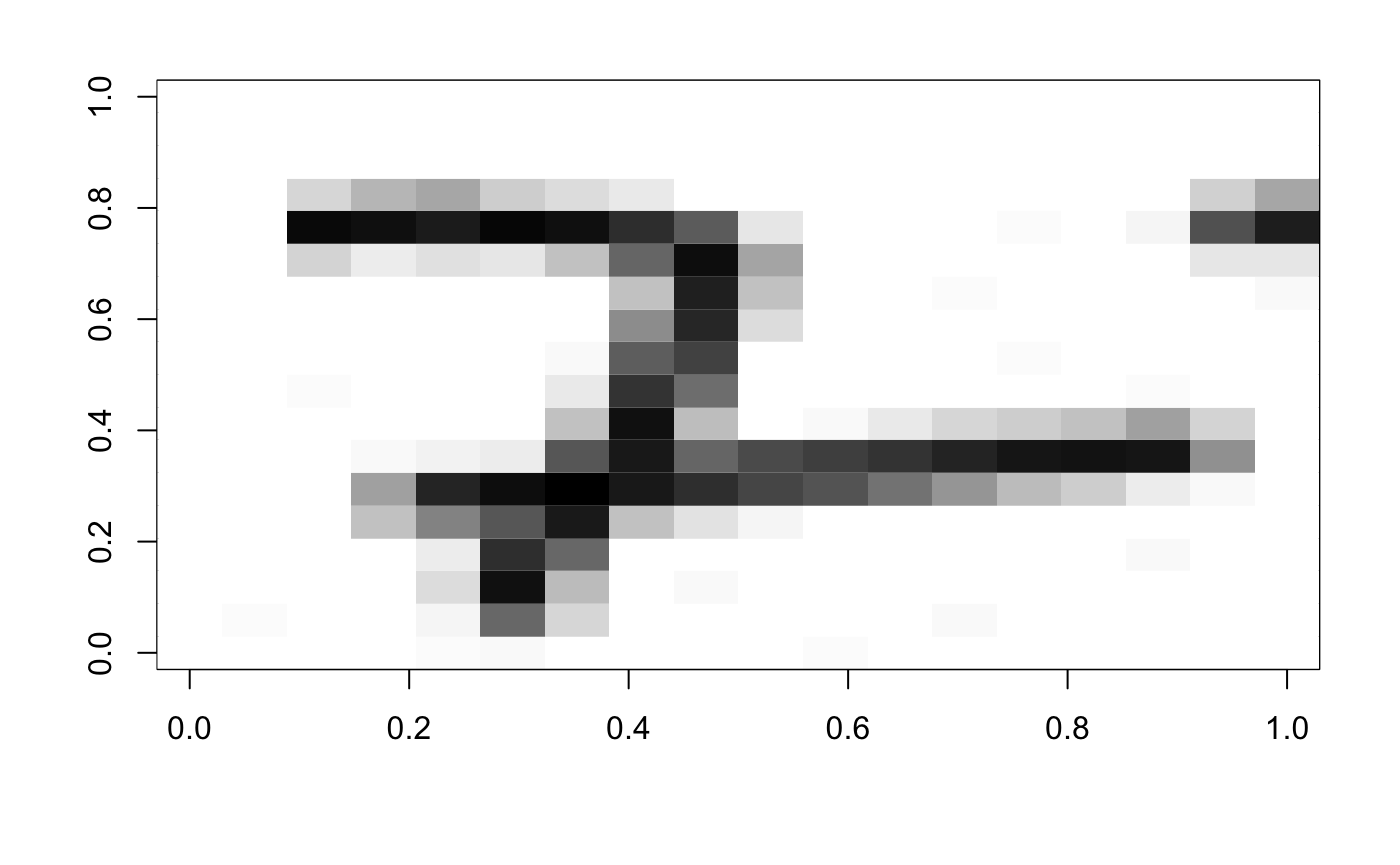
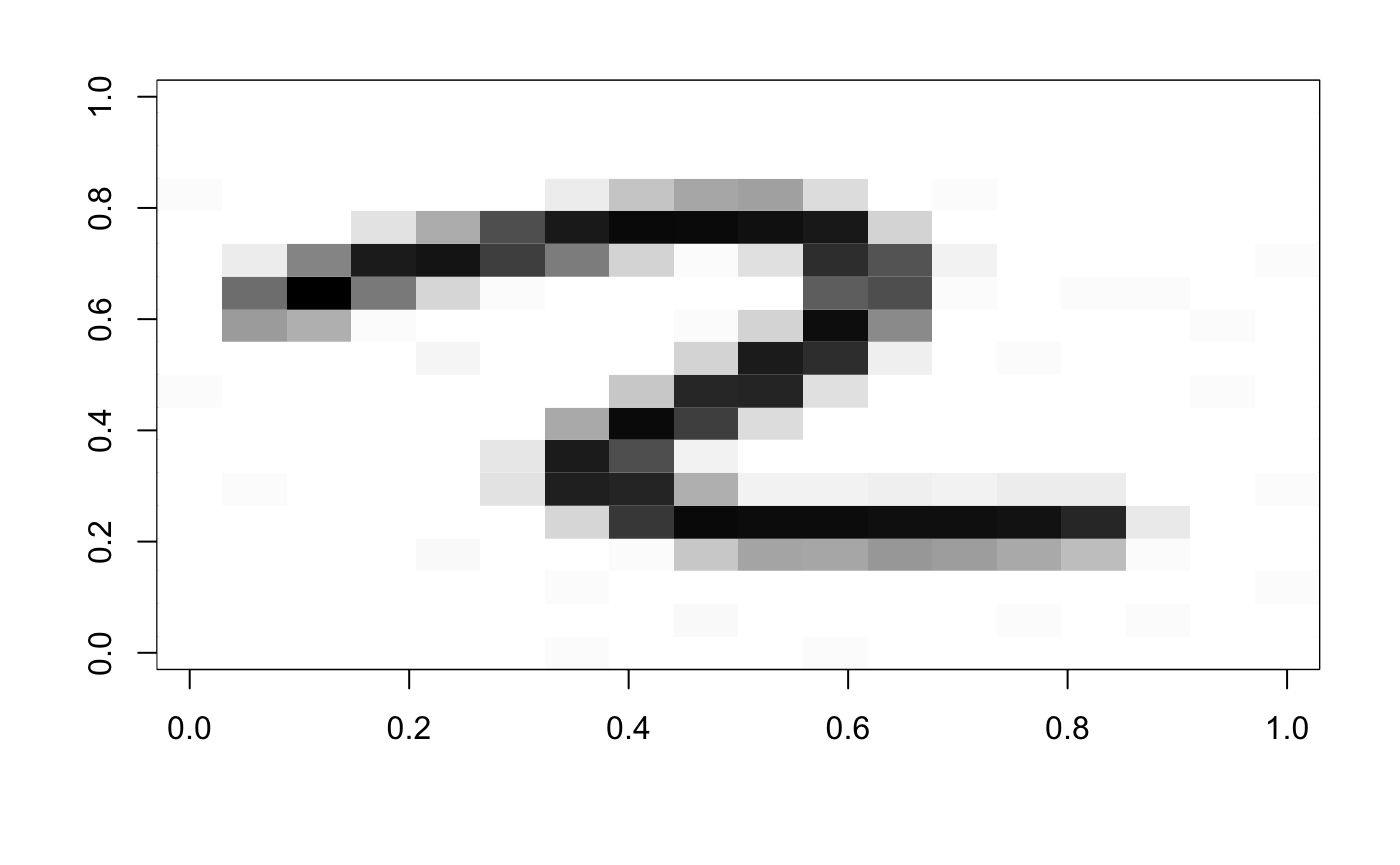
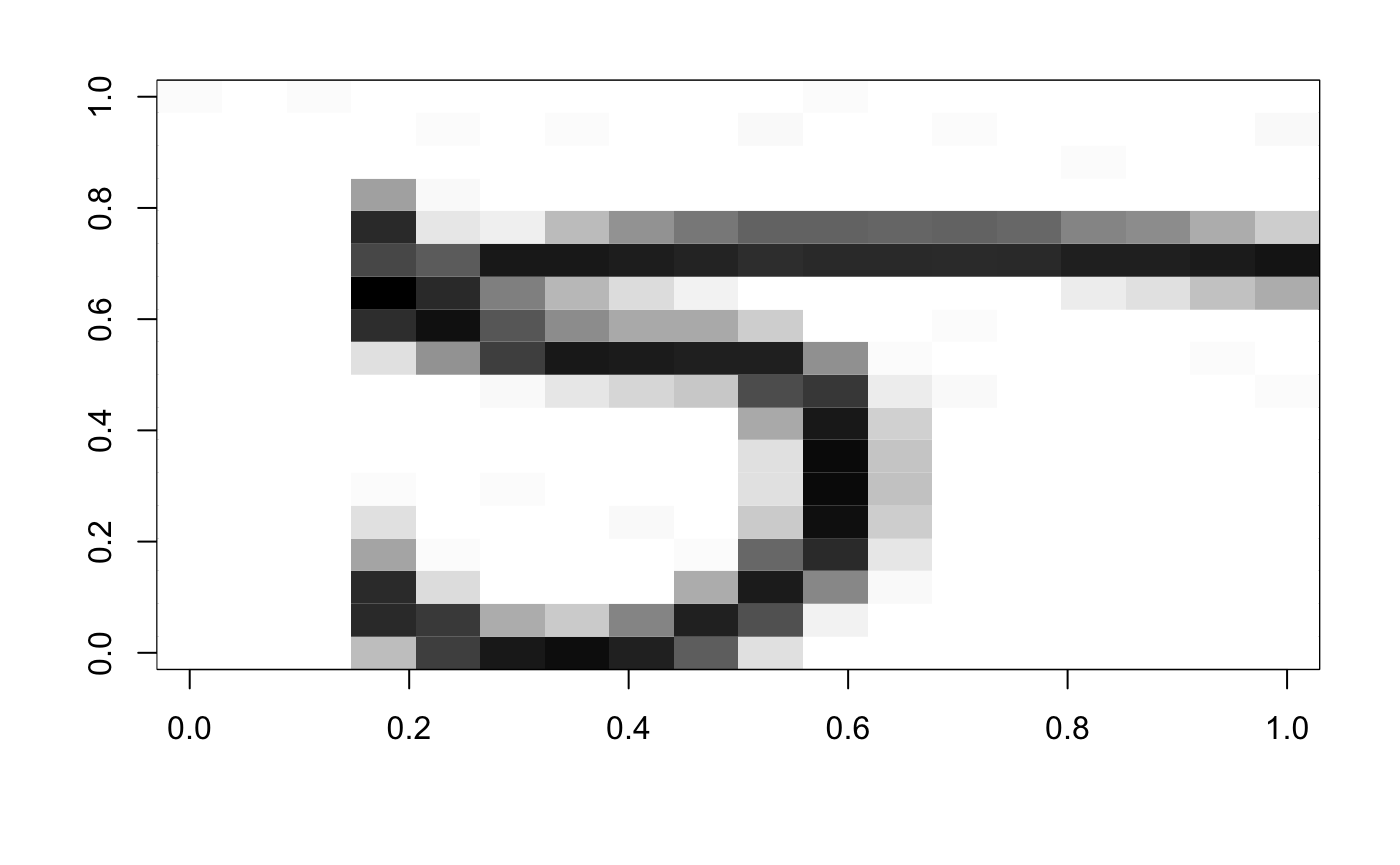
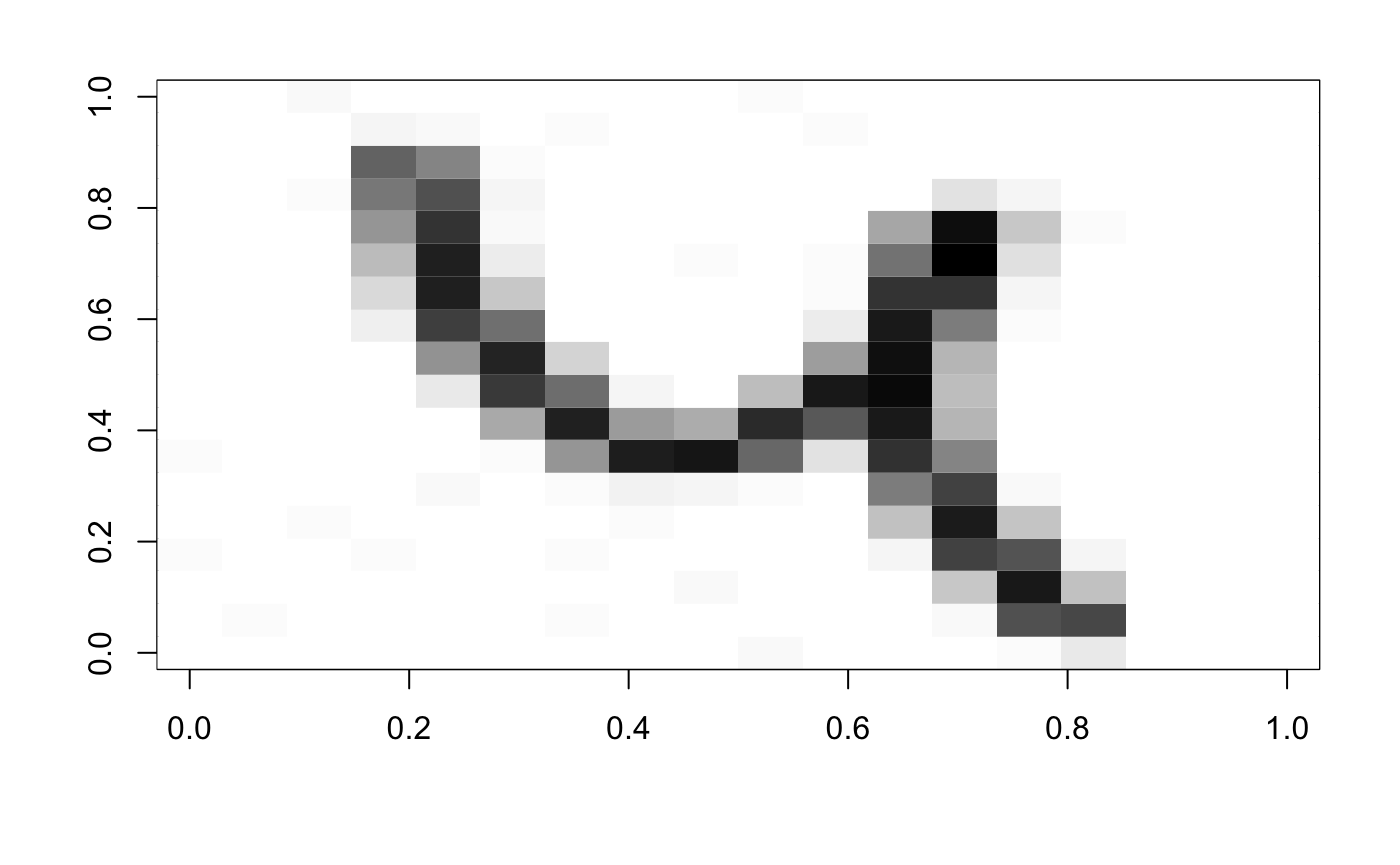
Q: I have got another confusion. Higher frequency-> details (e.g. noise), Lower Frequency-> kind of overview of the image. By increasing sigma, we are allowing some higher frequencies....so we should get more detailed with increasing frequency but the case is opposite, when we increase sigma, the image becomes more blurry.

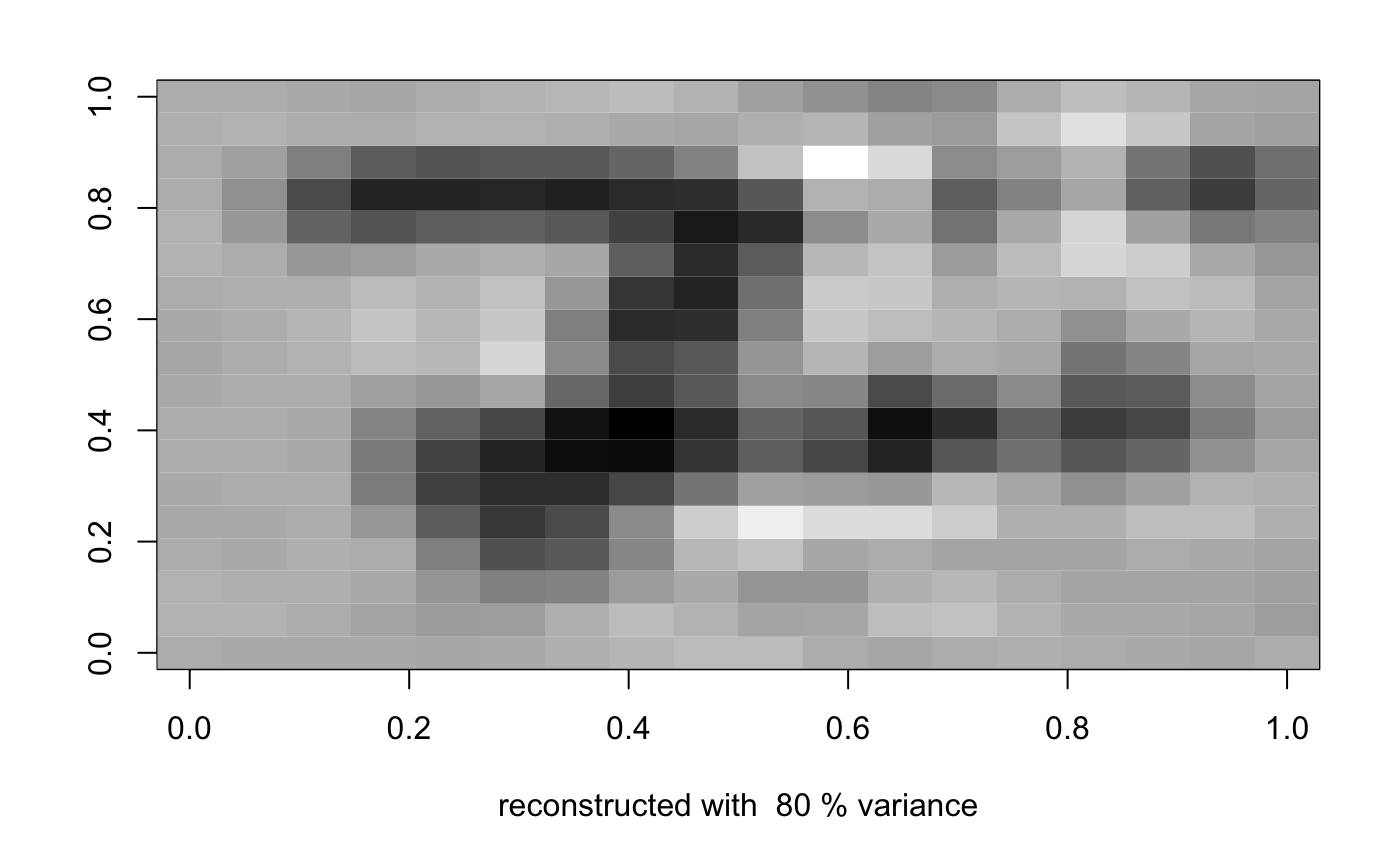
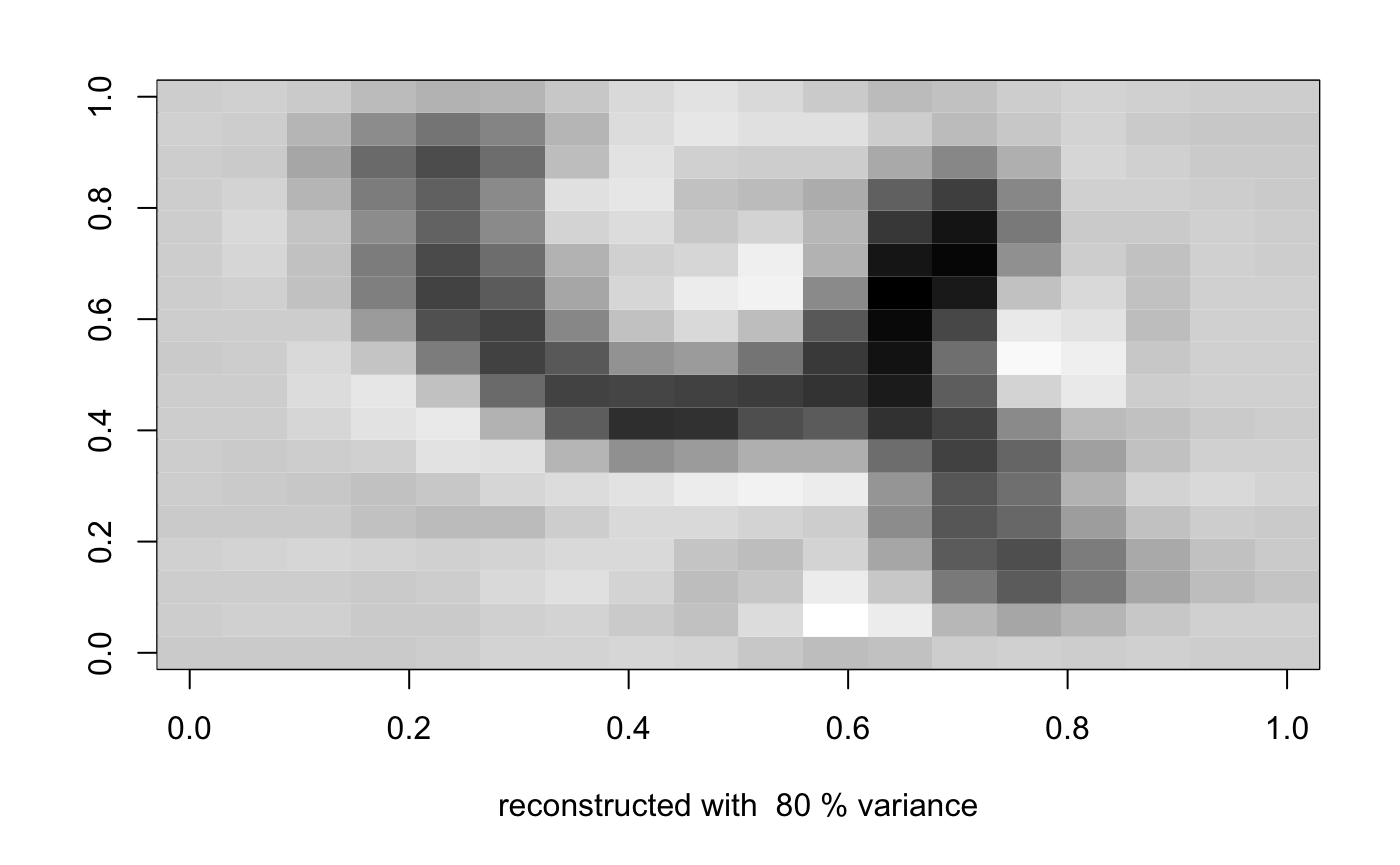
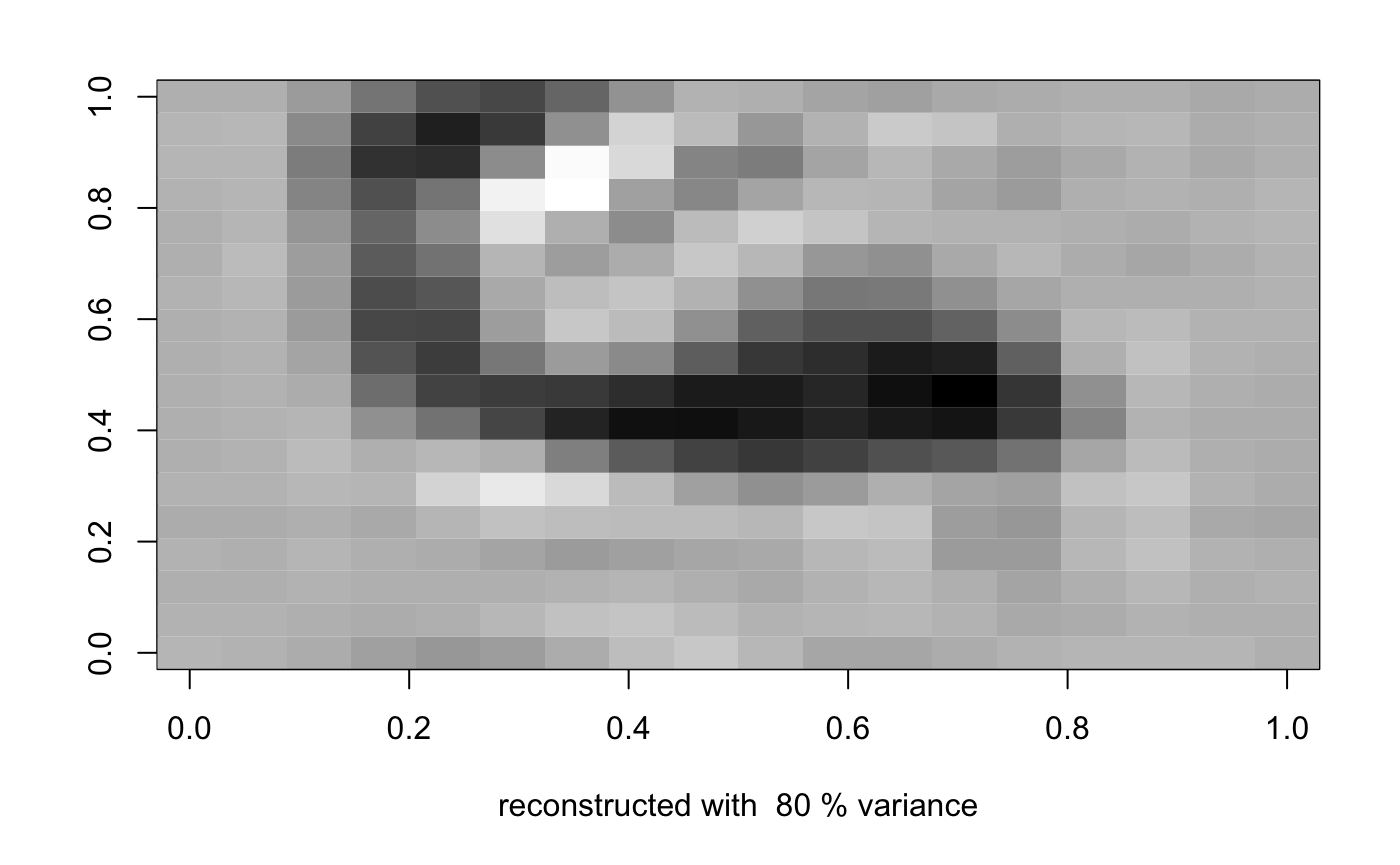
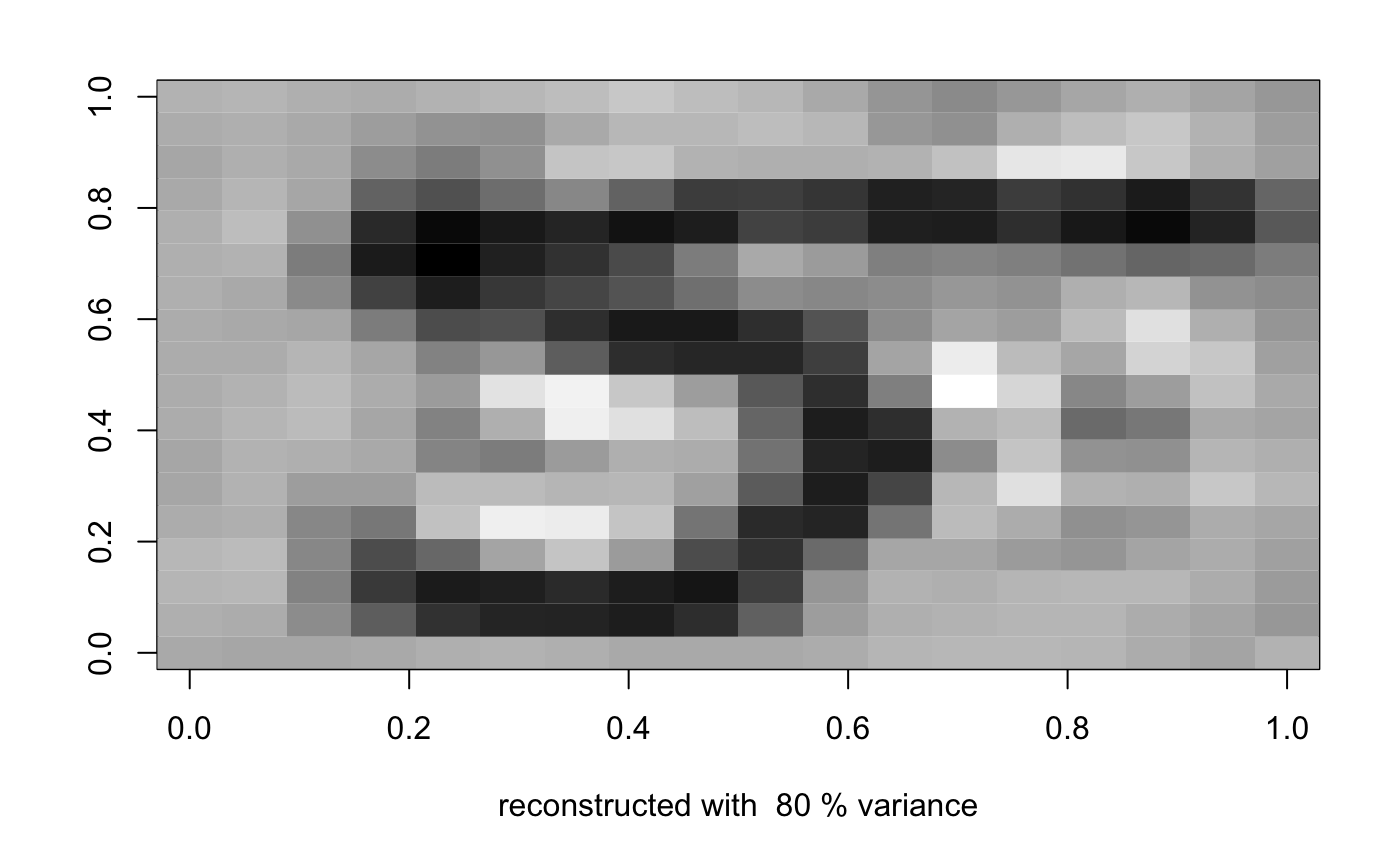
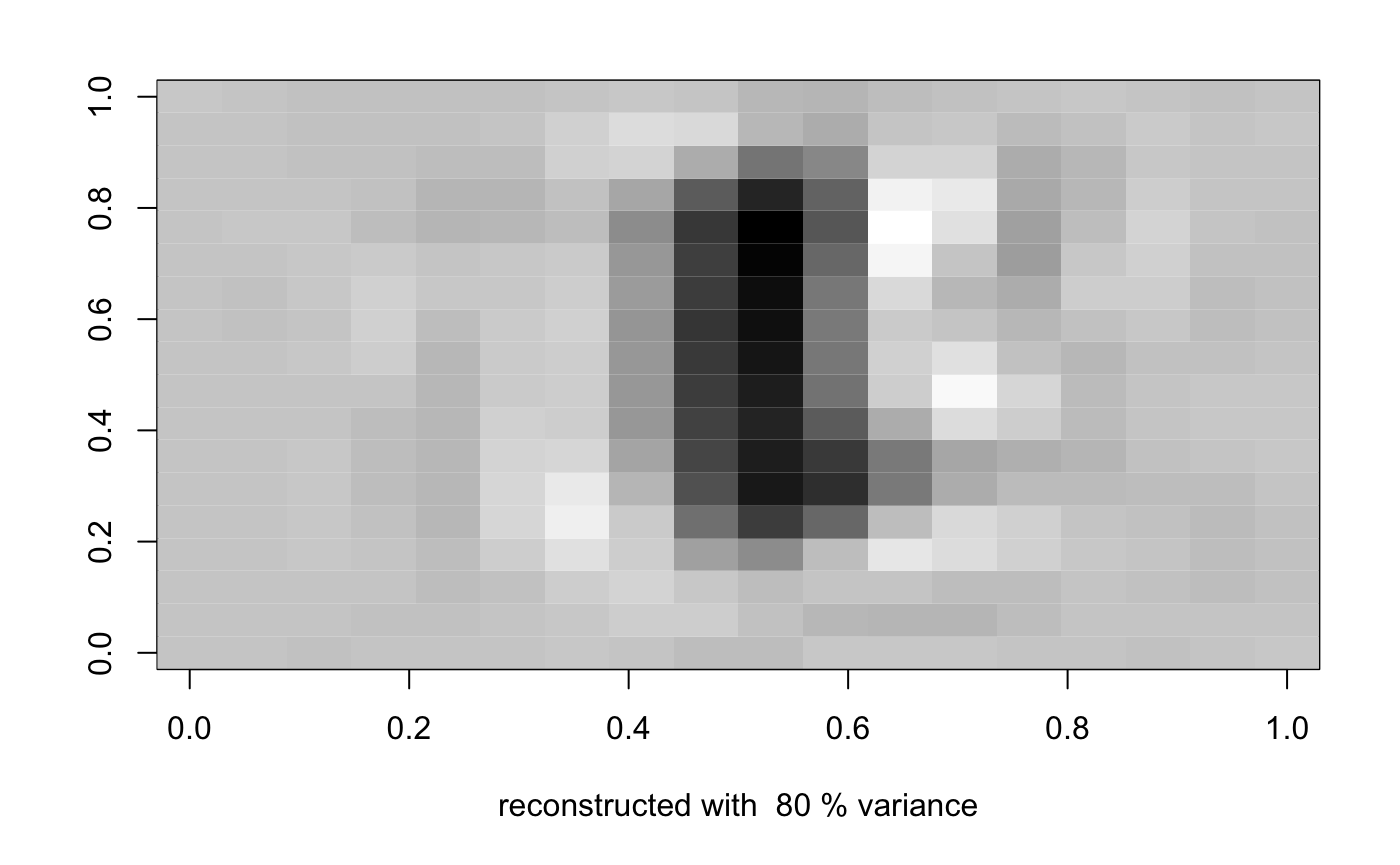
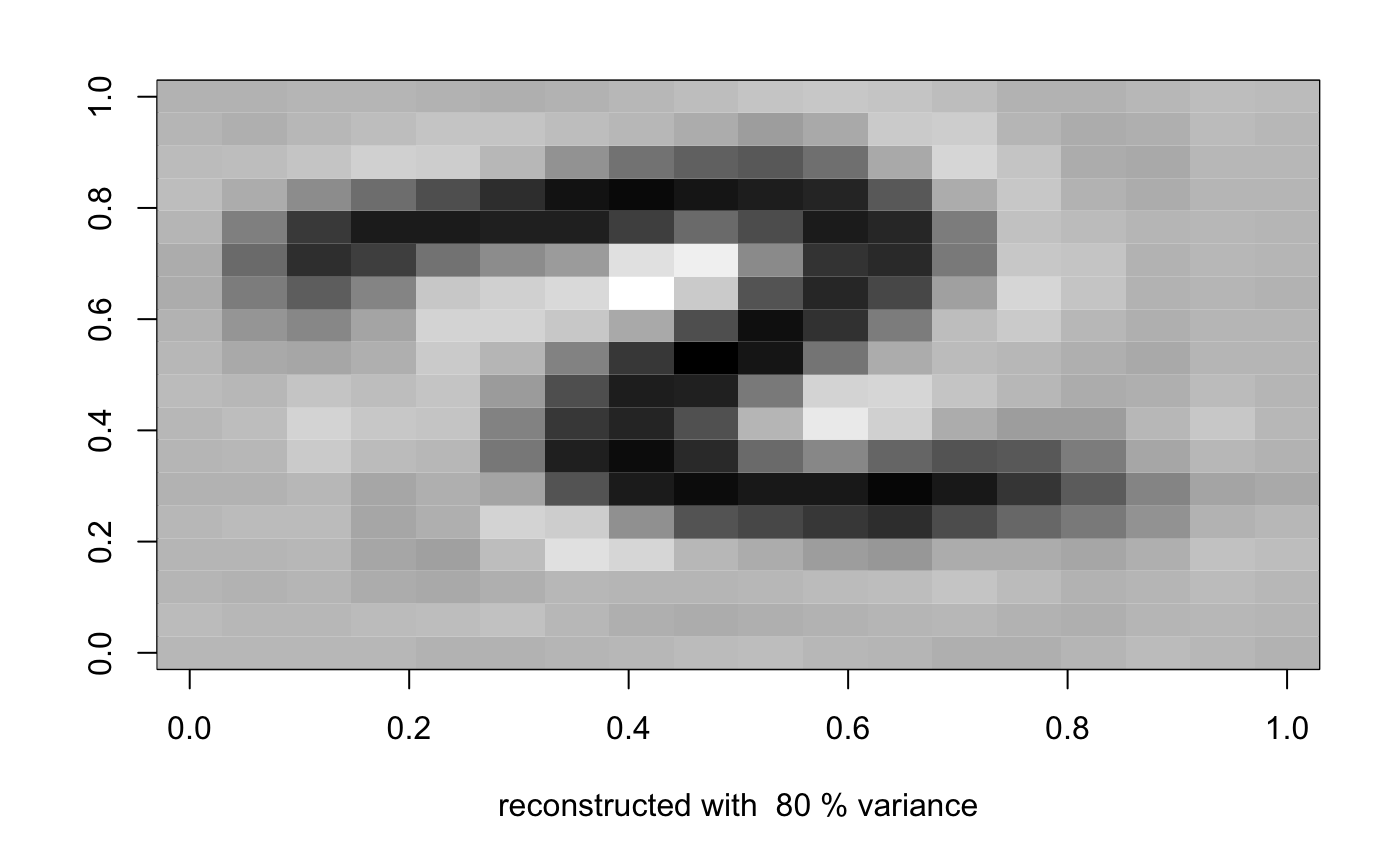
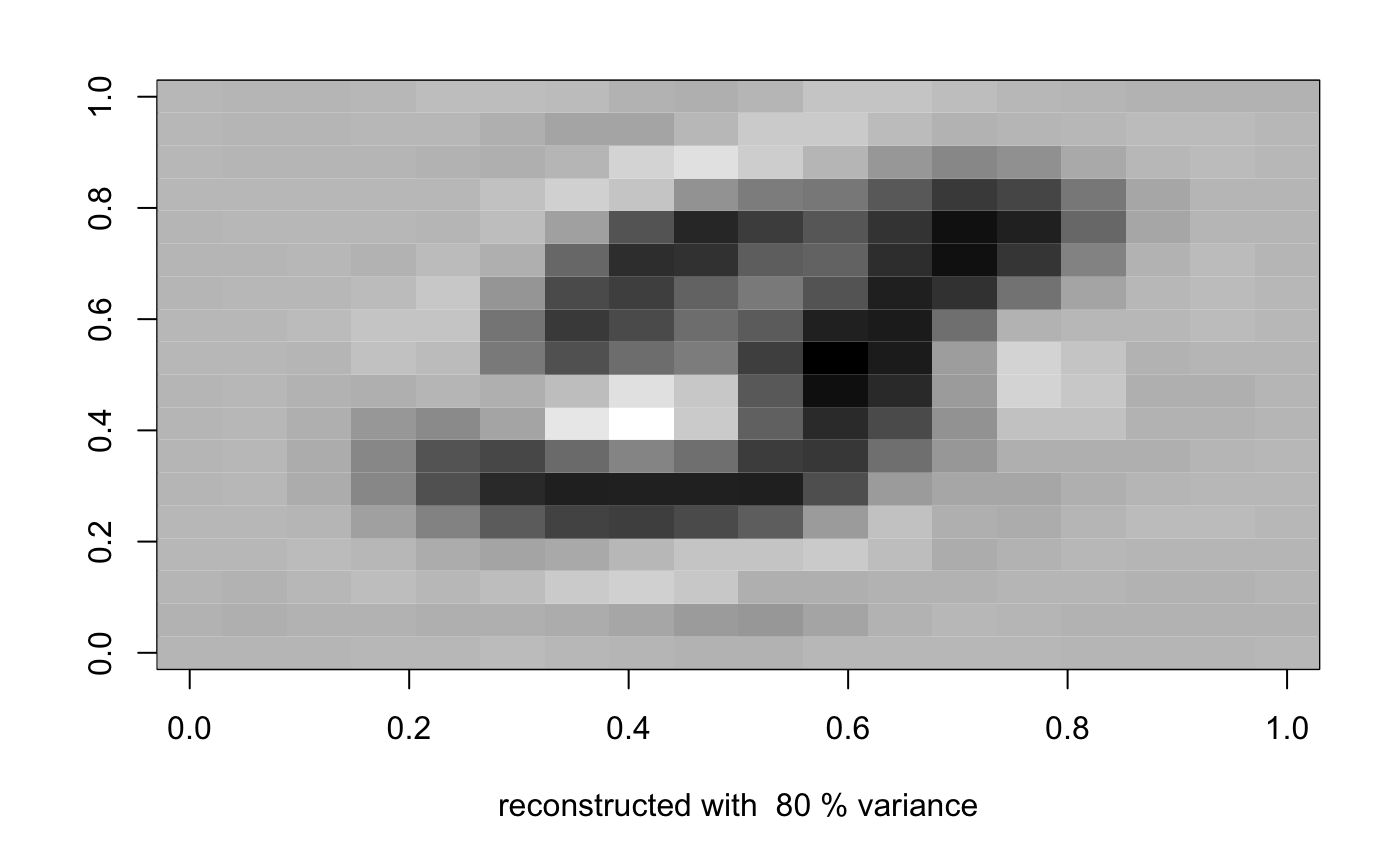
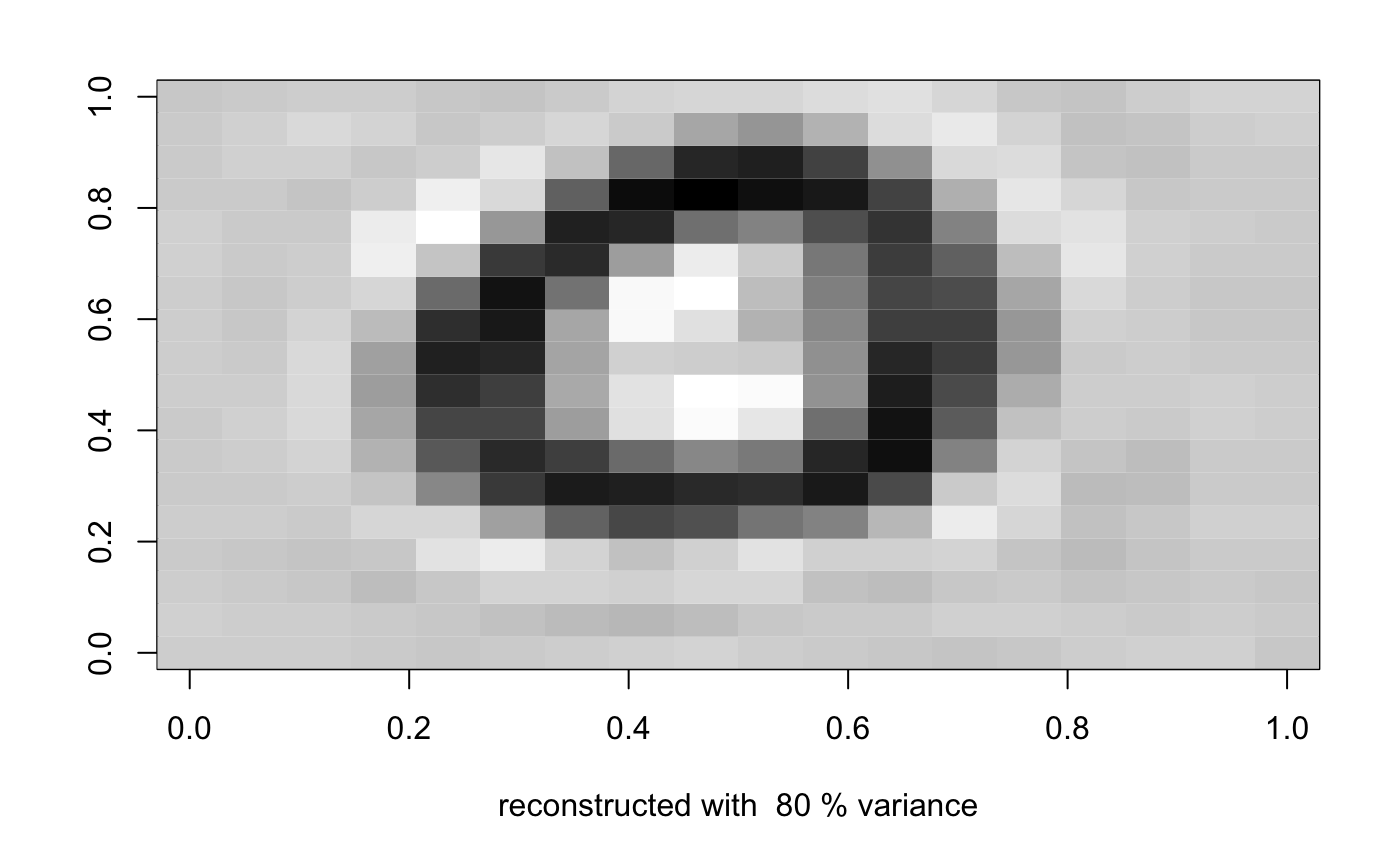
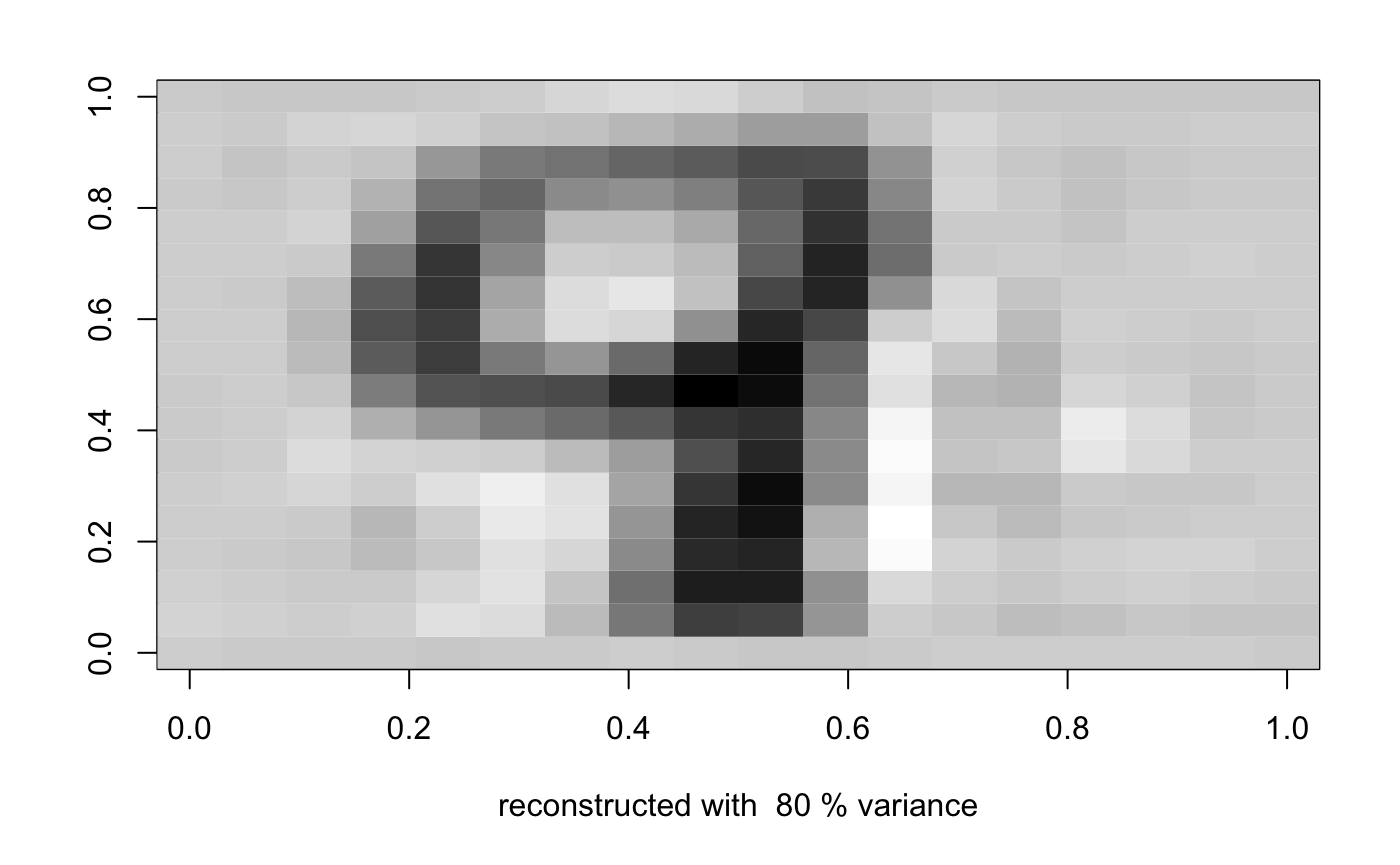
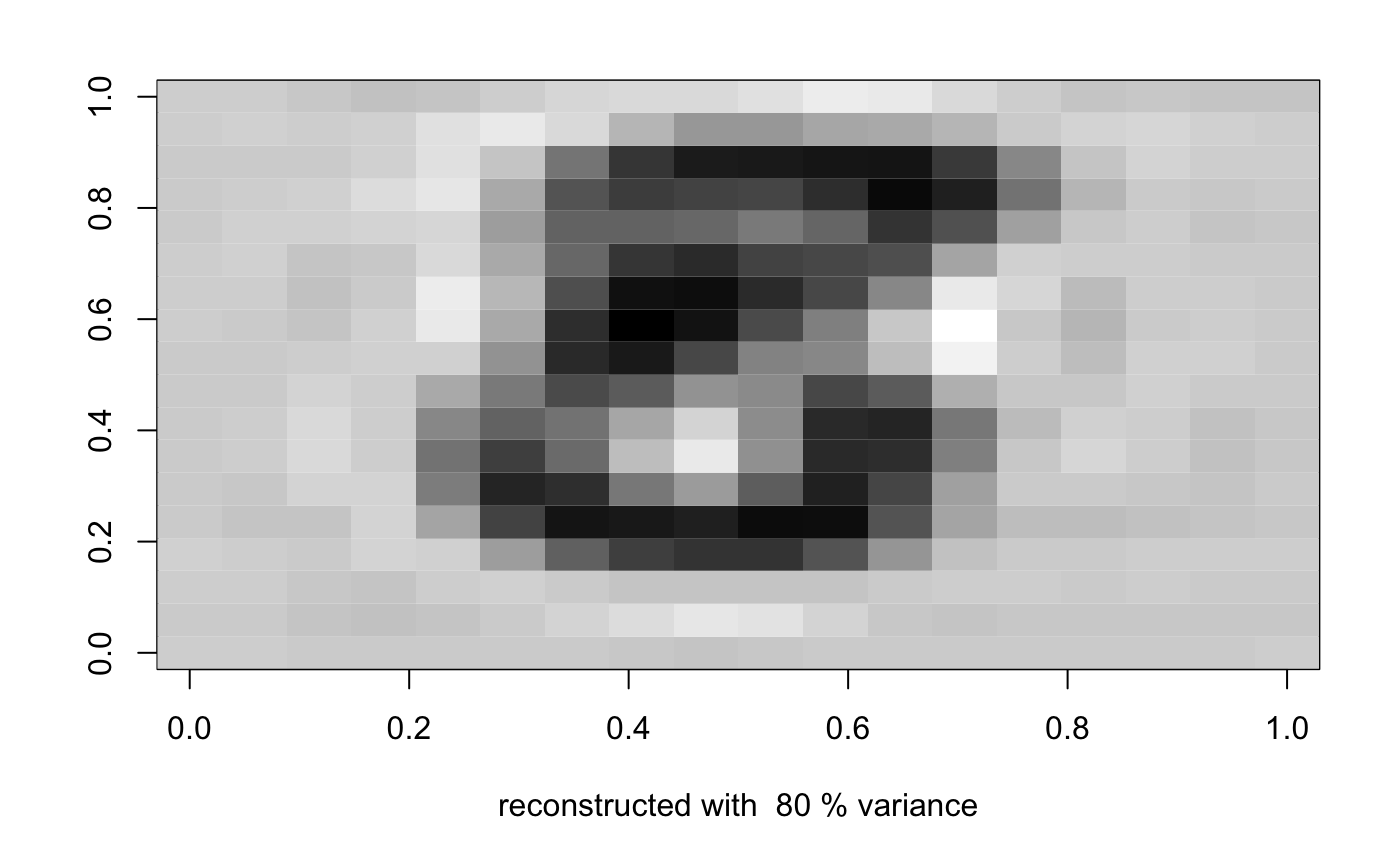
A: The edges in the image show themselves in higher frequencies (rapid changes in the intensity spatially) and also noises (noise in the image means unwanted changes of intensities). The lower frequencies mean there are not a lot of changes in intensity. as you increase sigma you will get a broad overview because the details vanished; we get a blurred image. That kind of rapid change (high frequencies) becomes smoother. As we decrease the sigma we get more details.

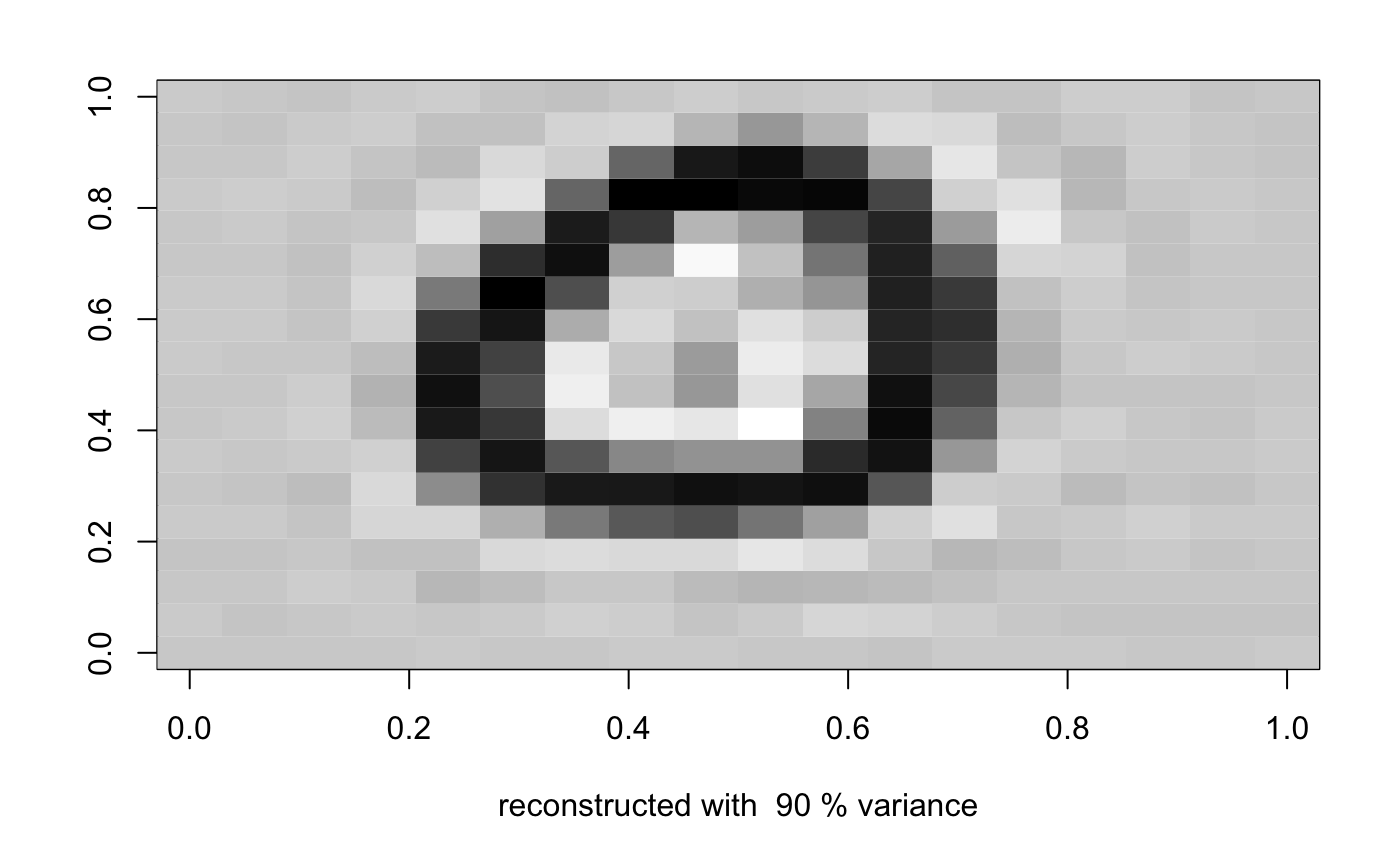
* **Exercise 2.4: Reconstruction using PCA**

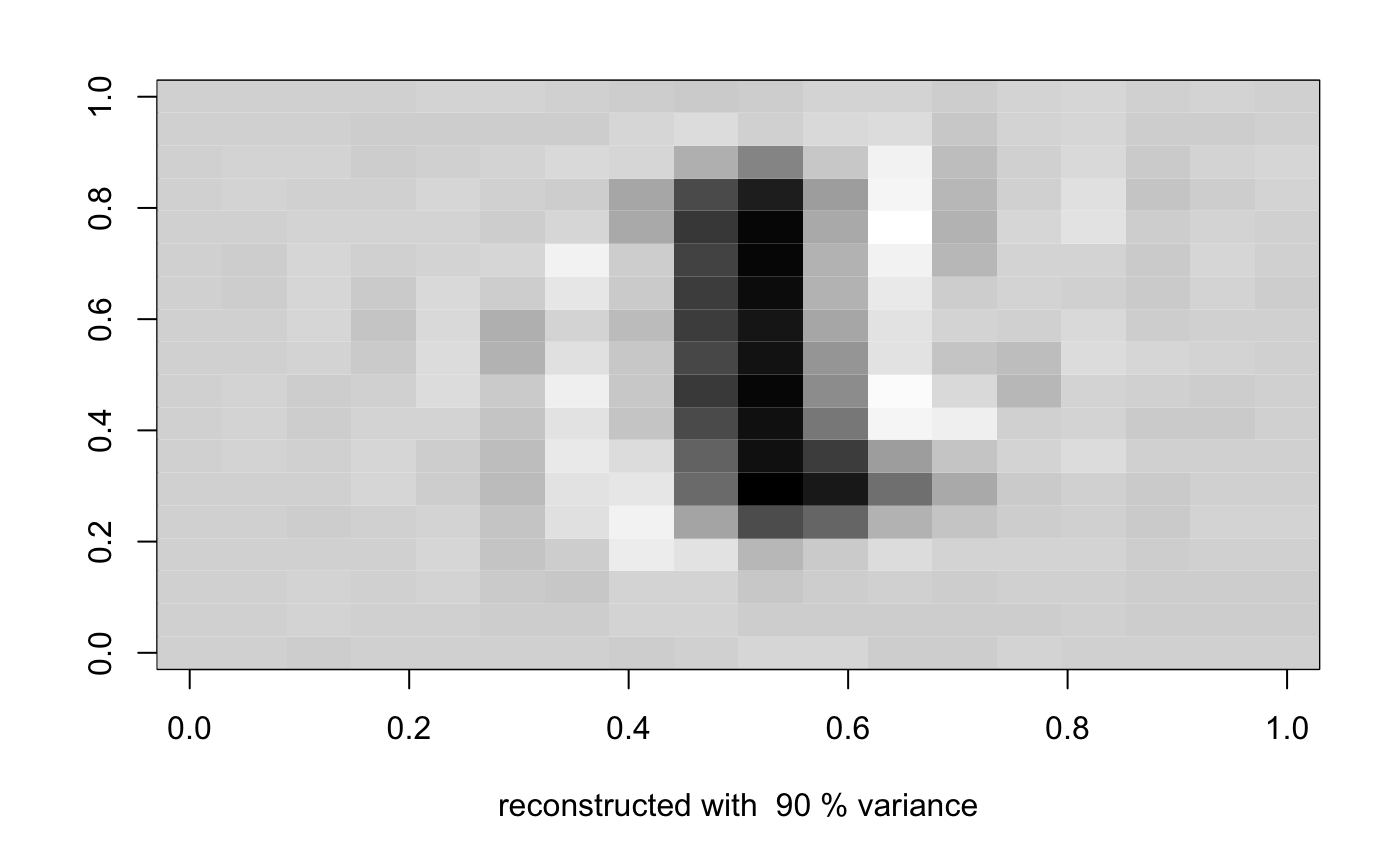
This task is about reconstructing data using PCA. First we can plot an image of a single cipher:

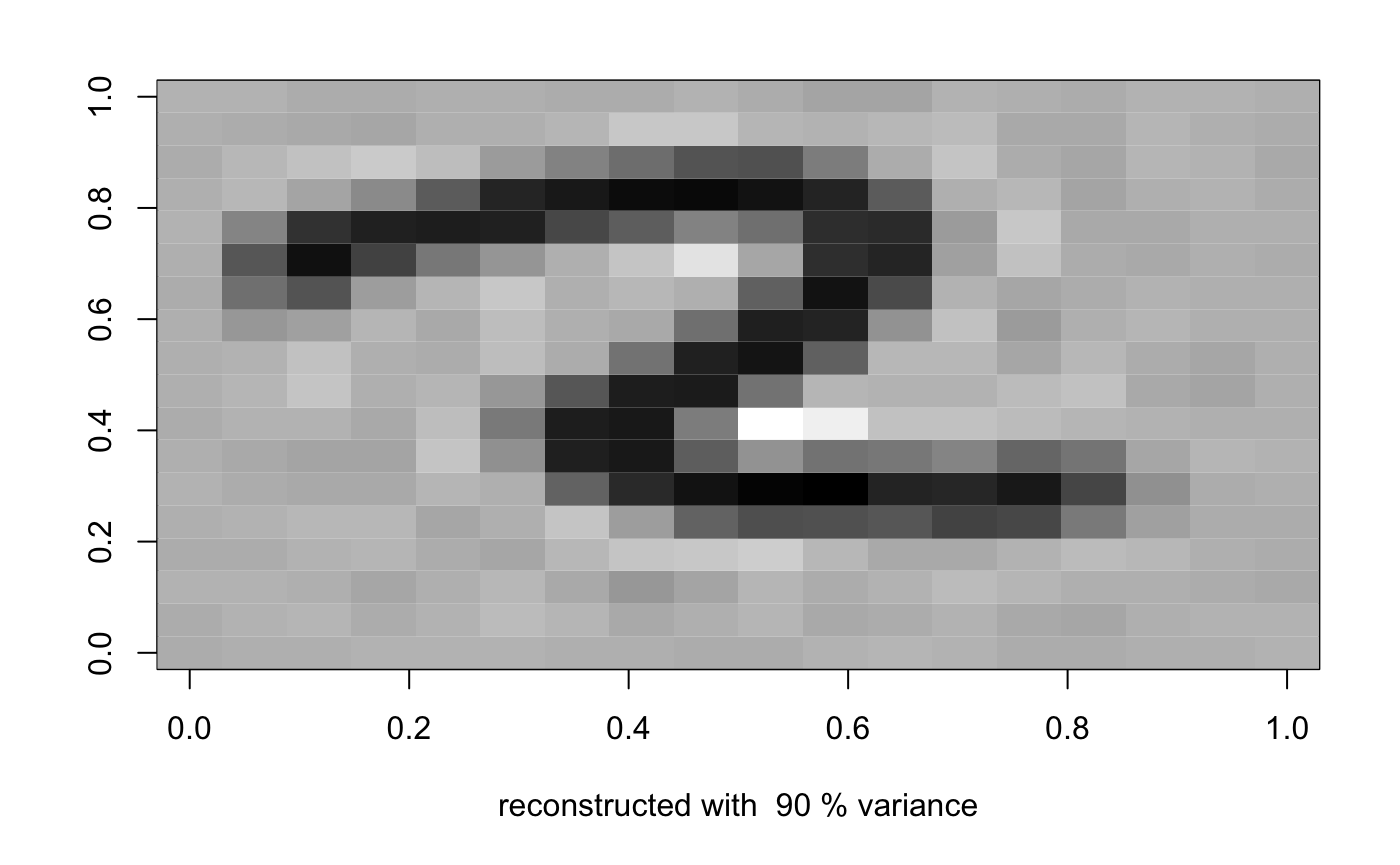
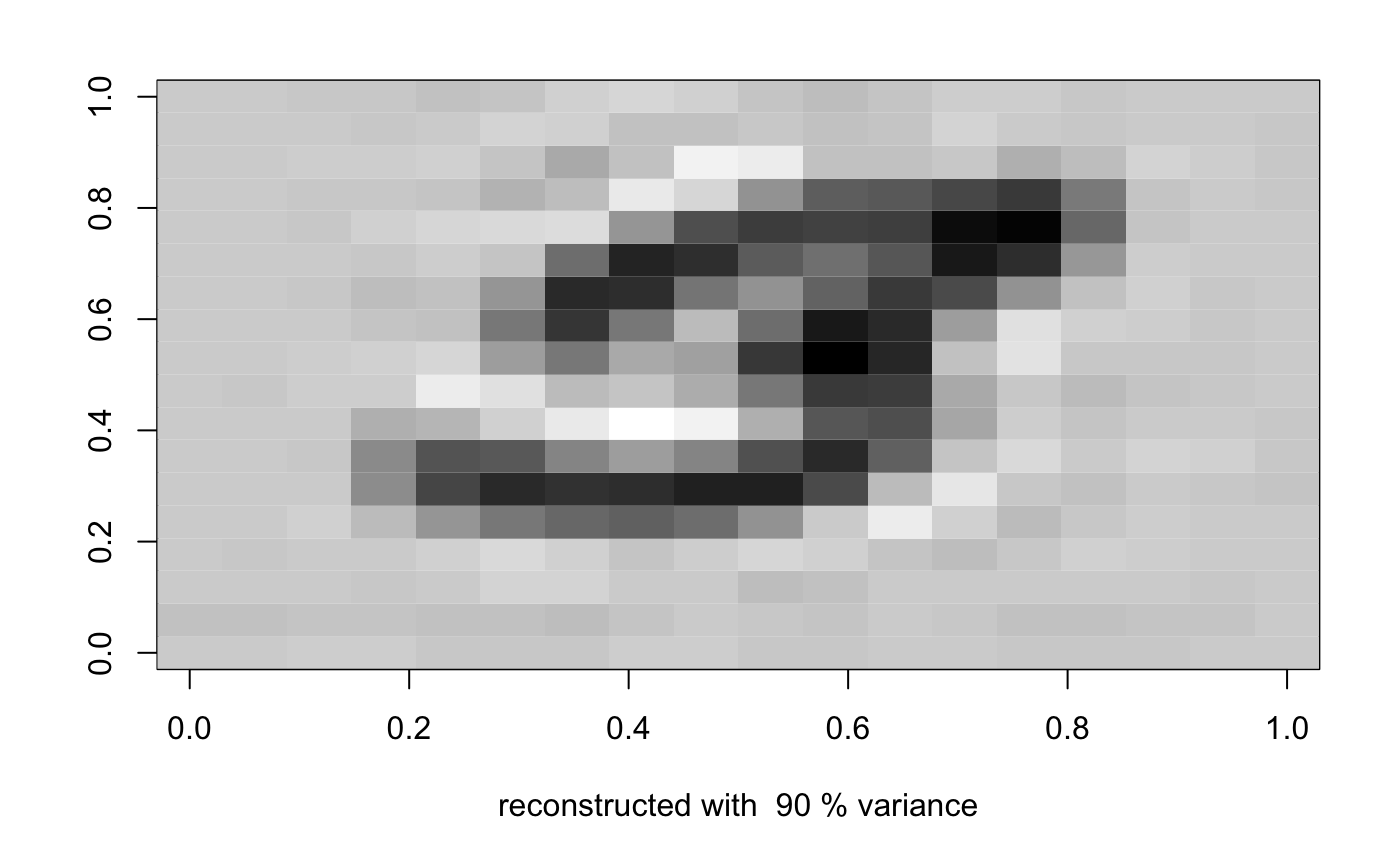


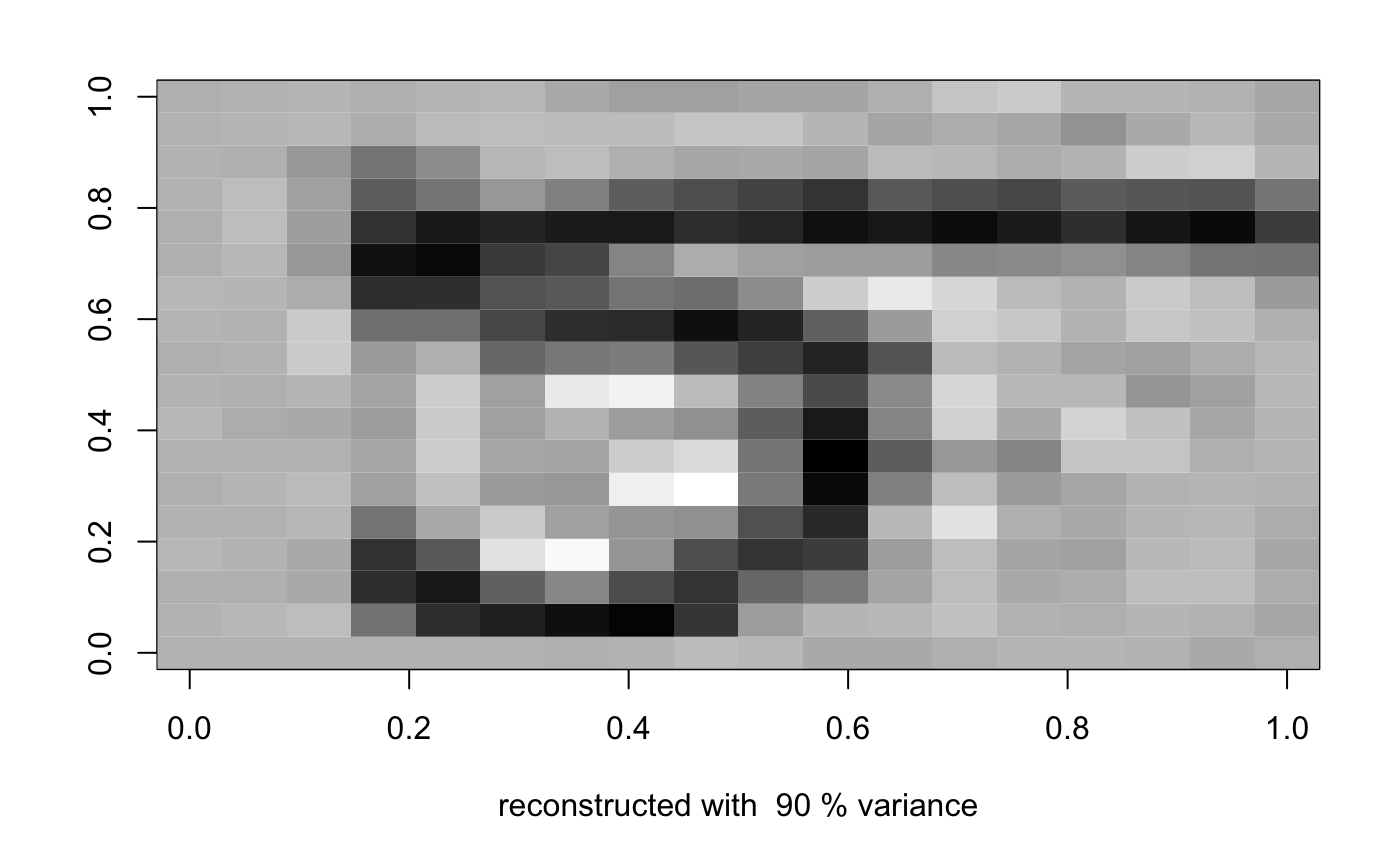


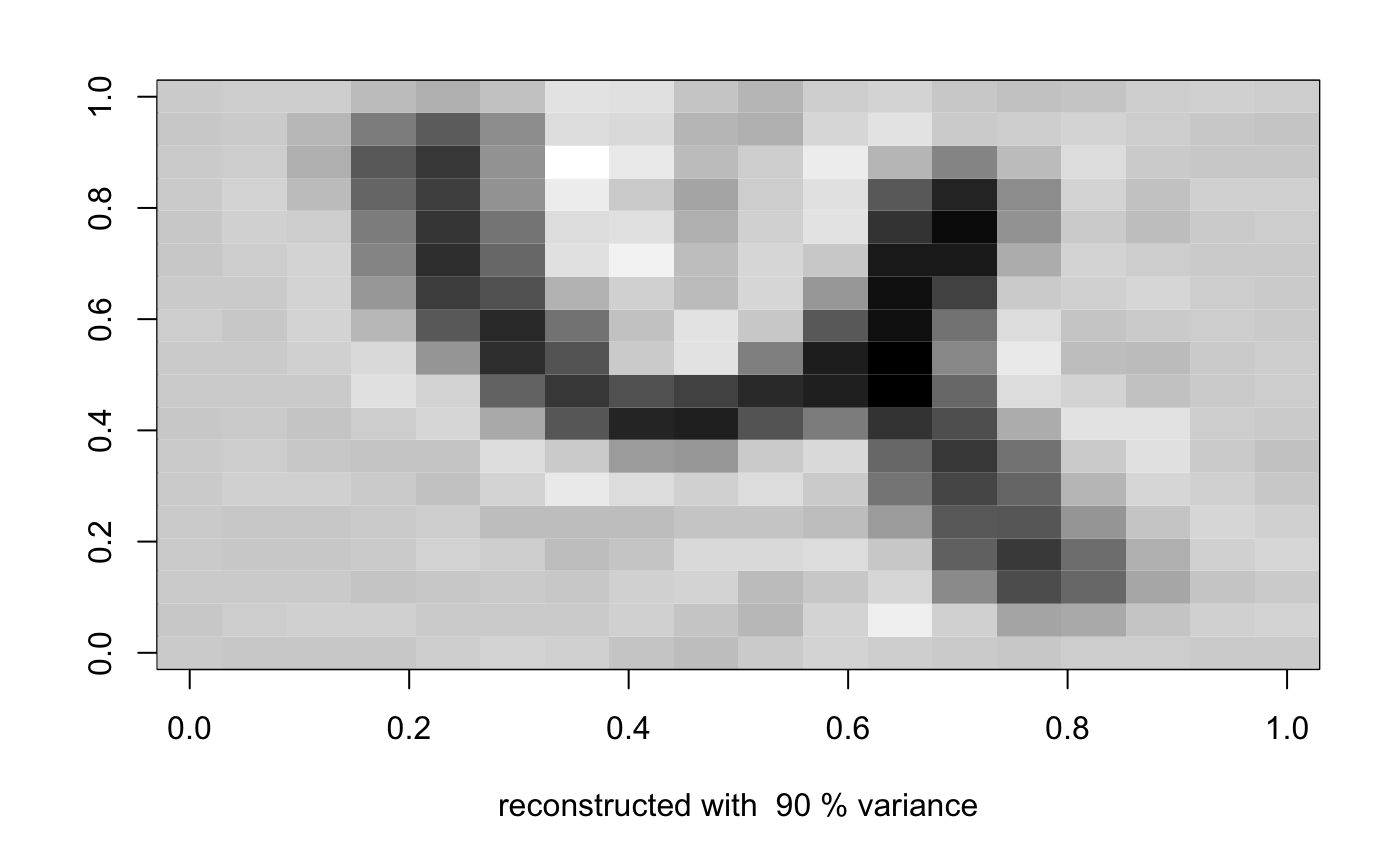
Reconstructed with 80% variance

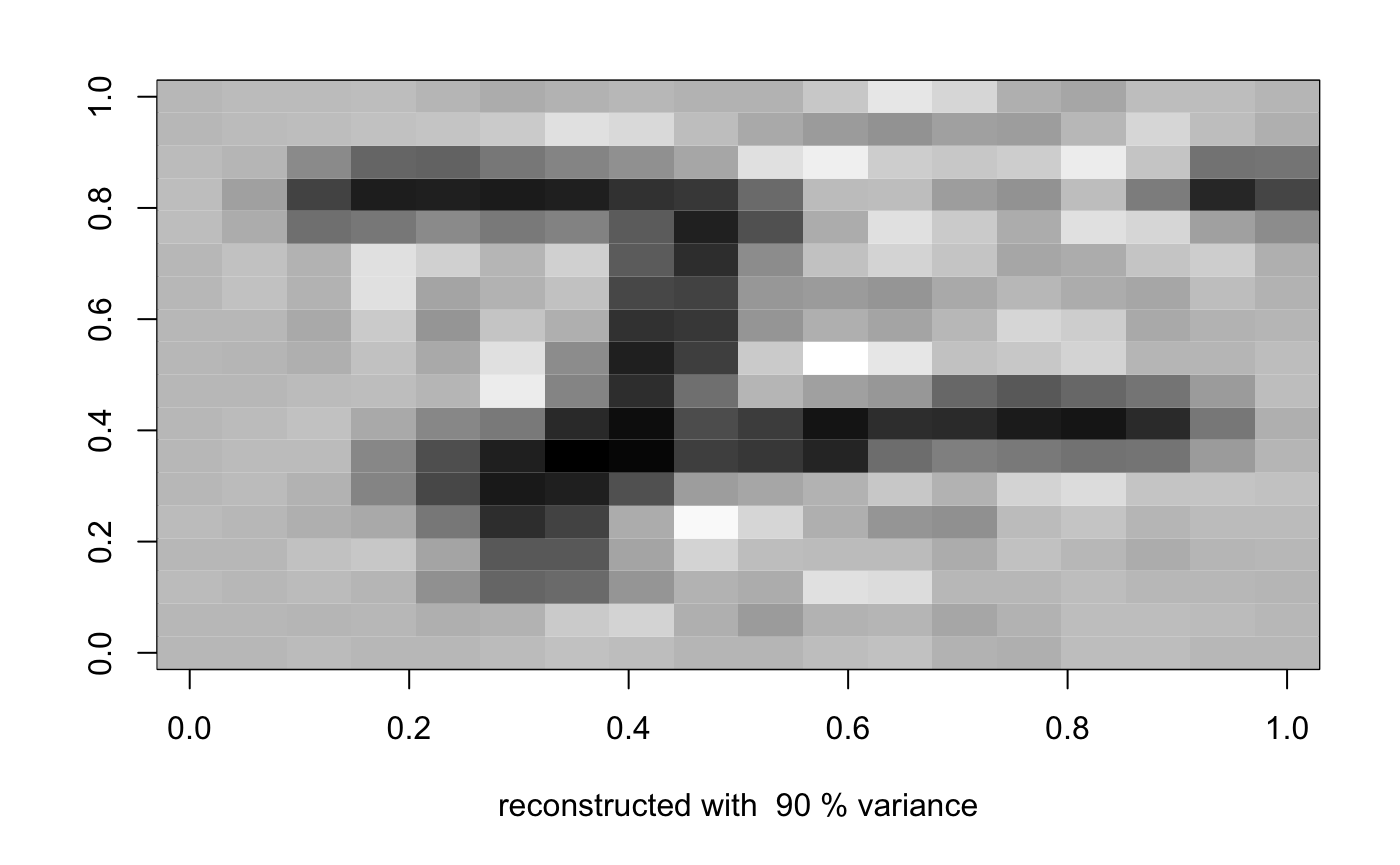
Reconstructed with 90% variance

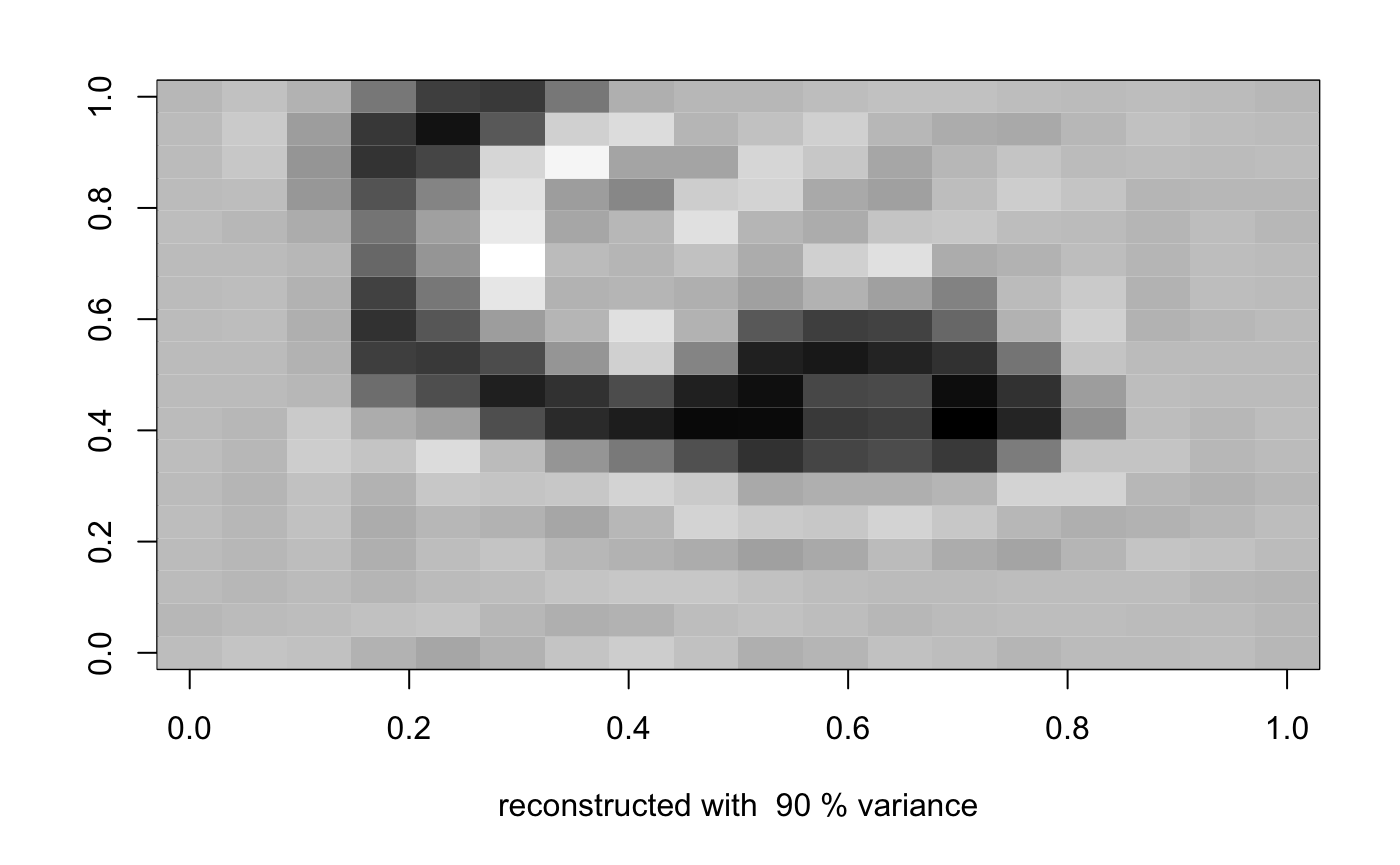


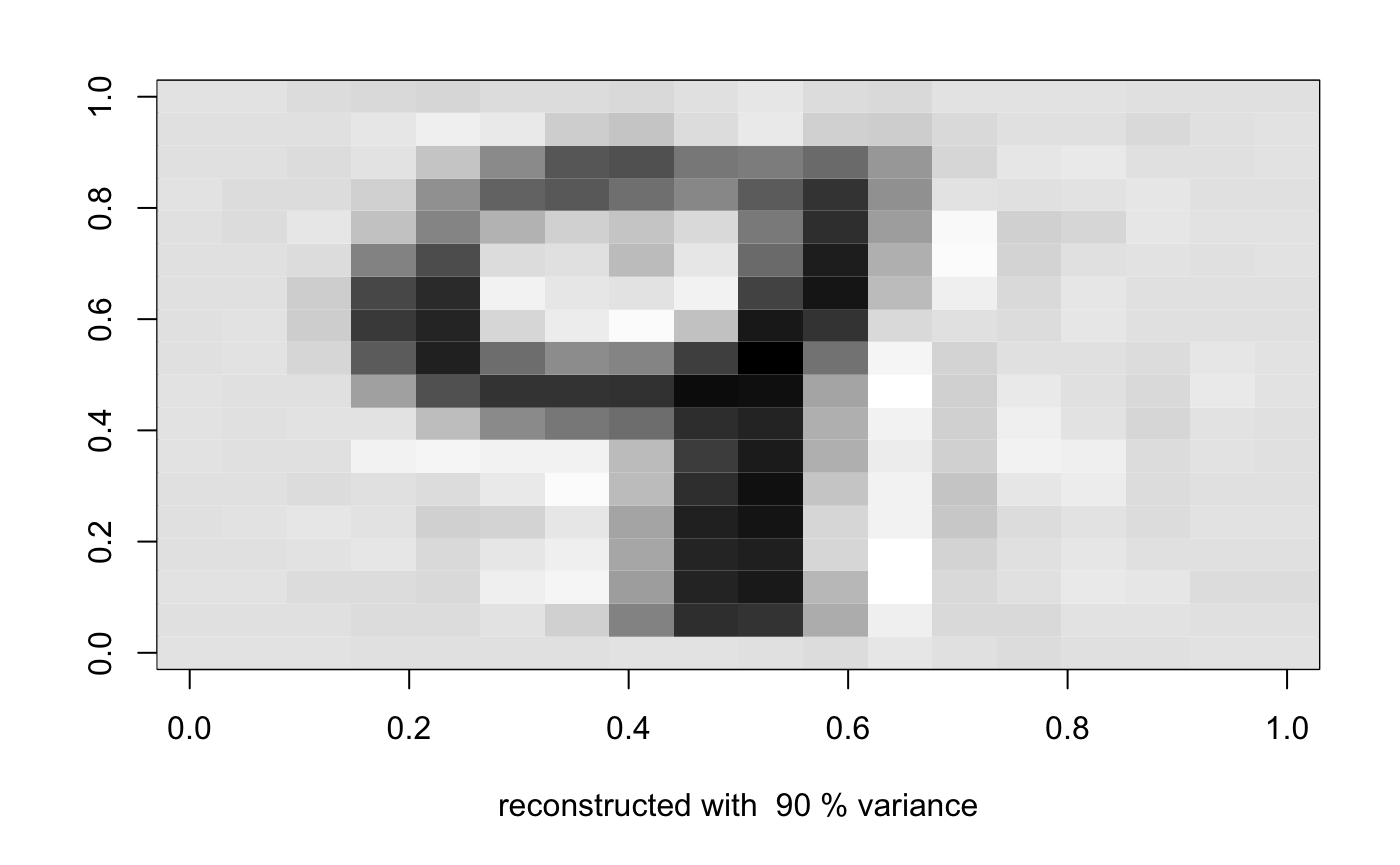
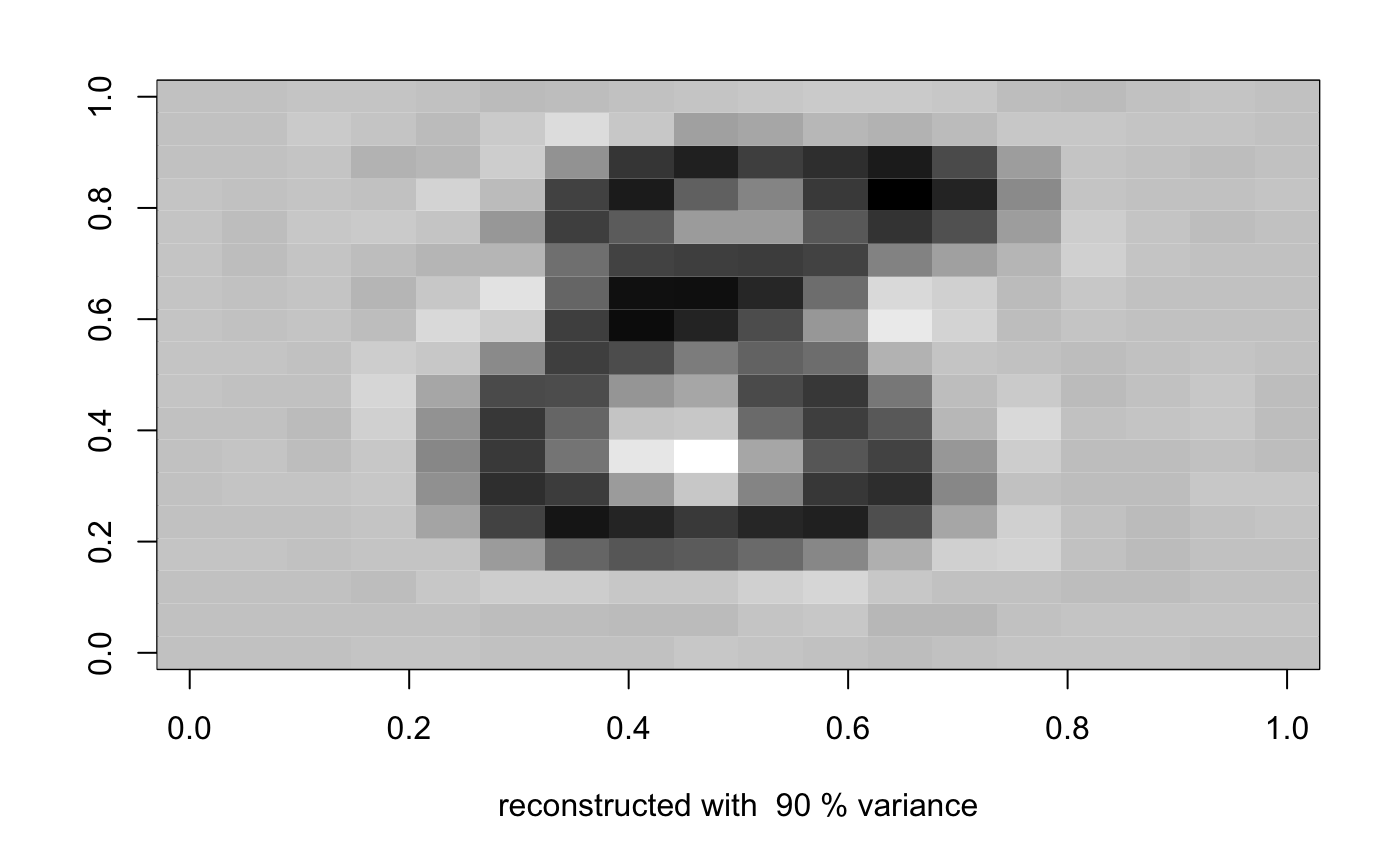




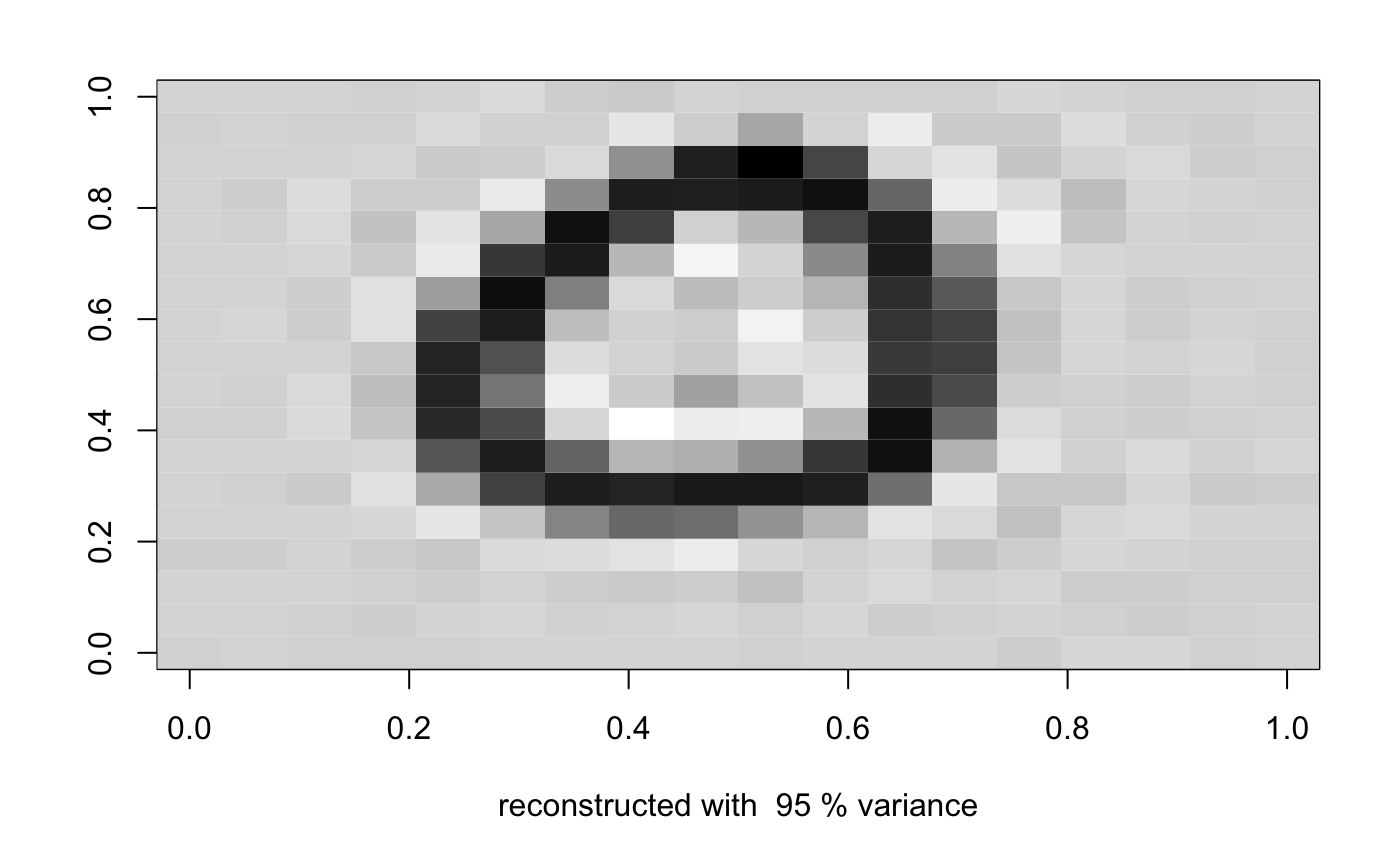
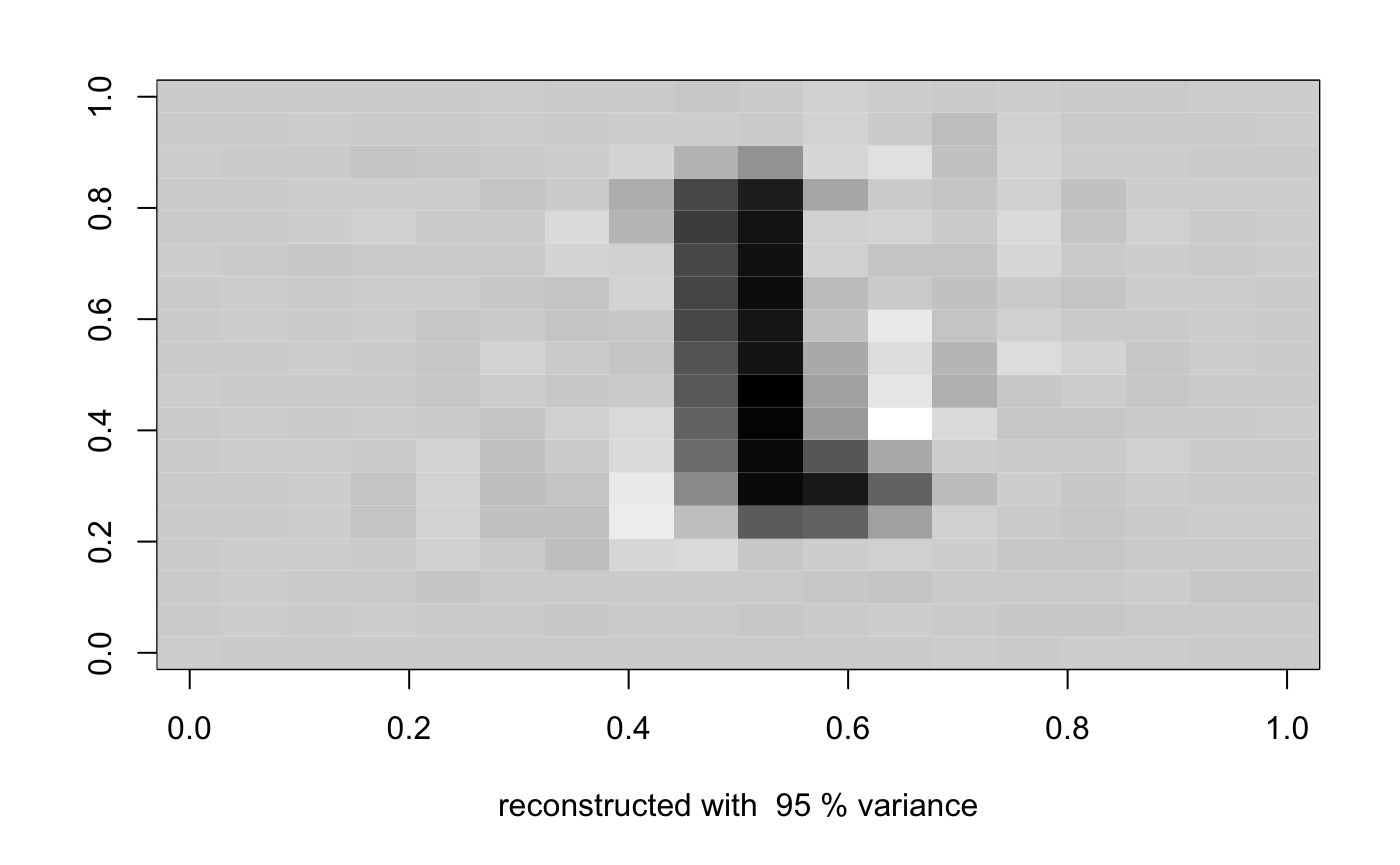


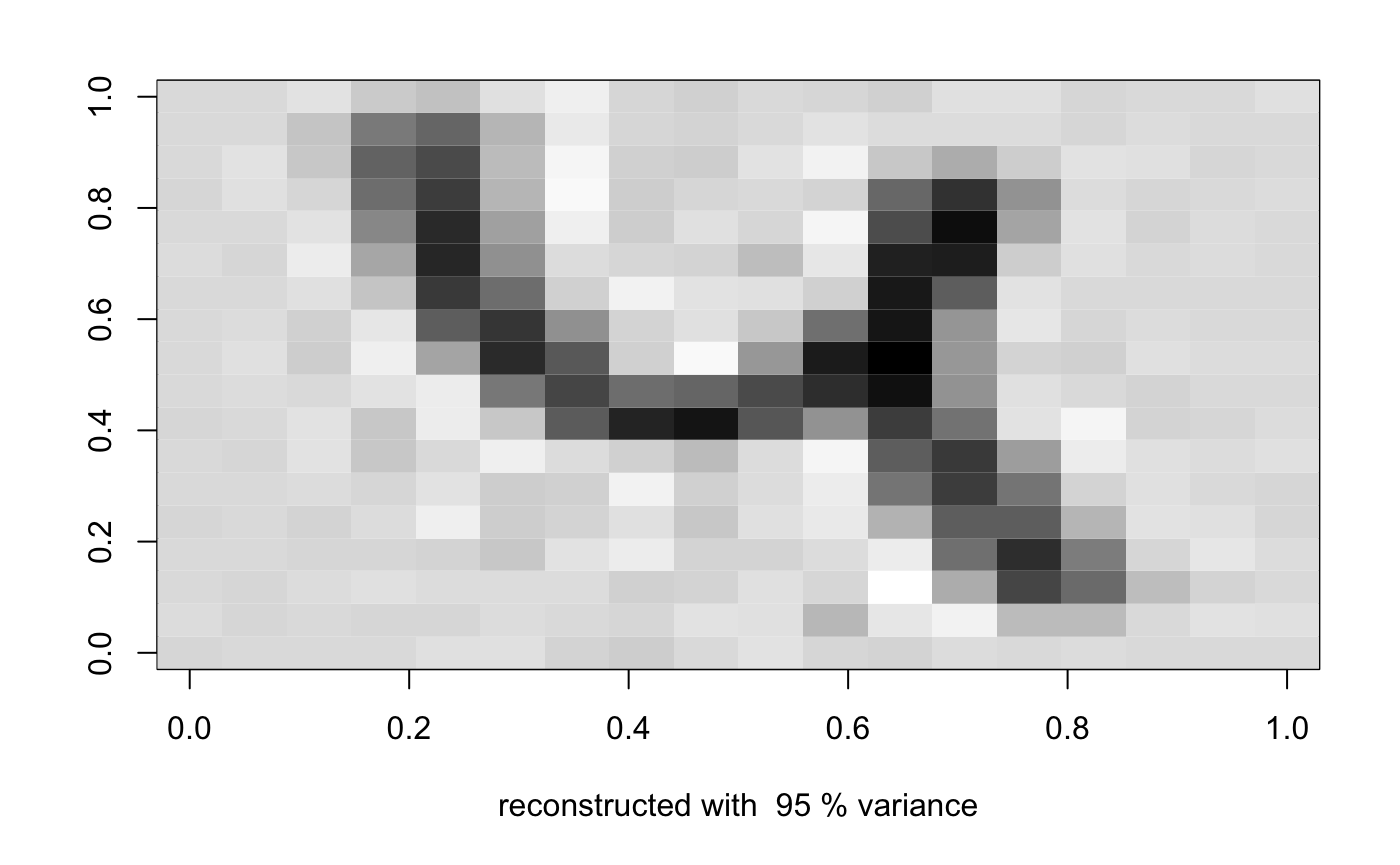
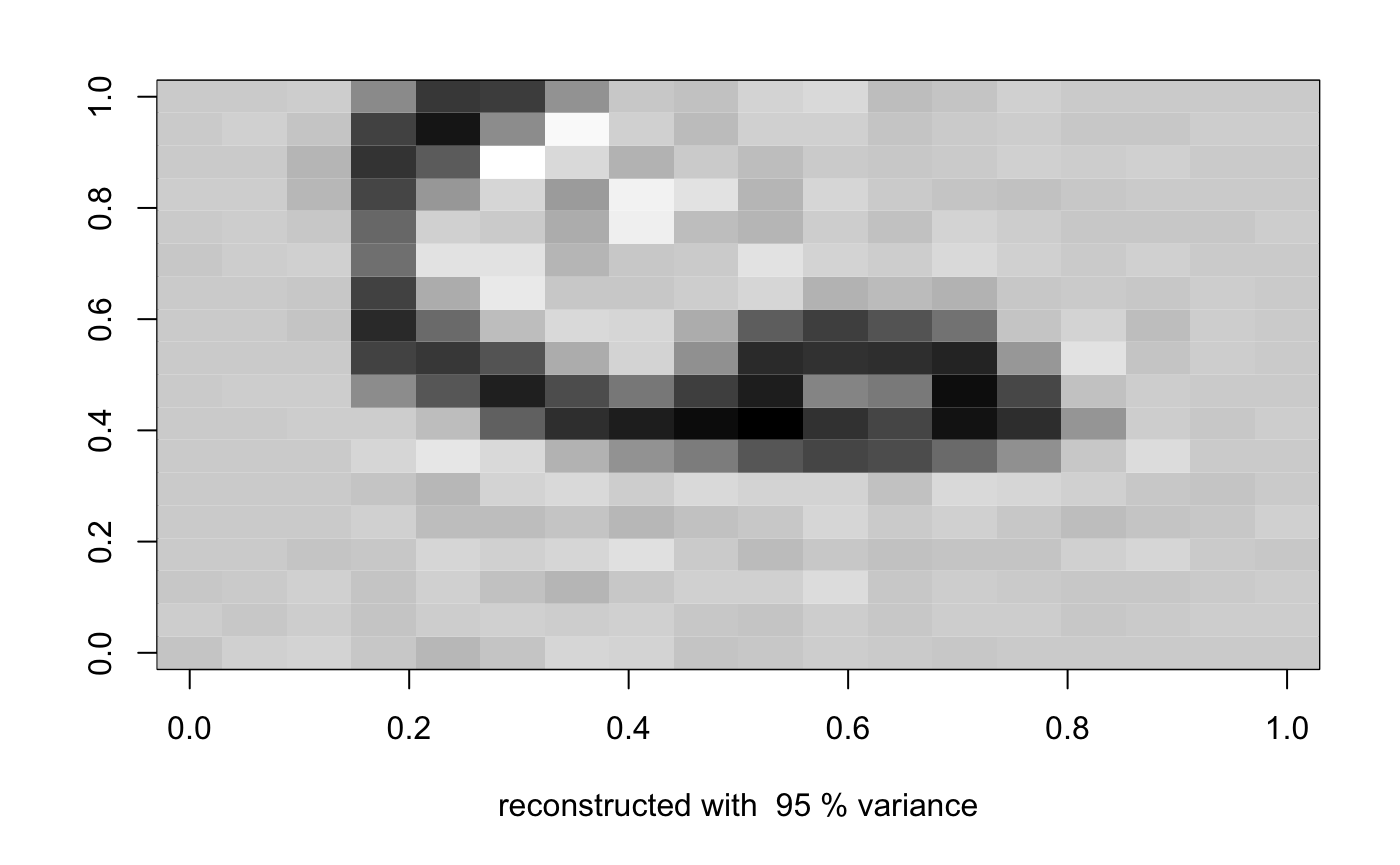
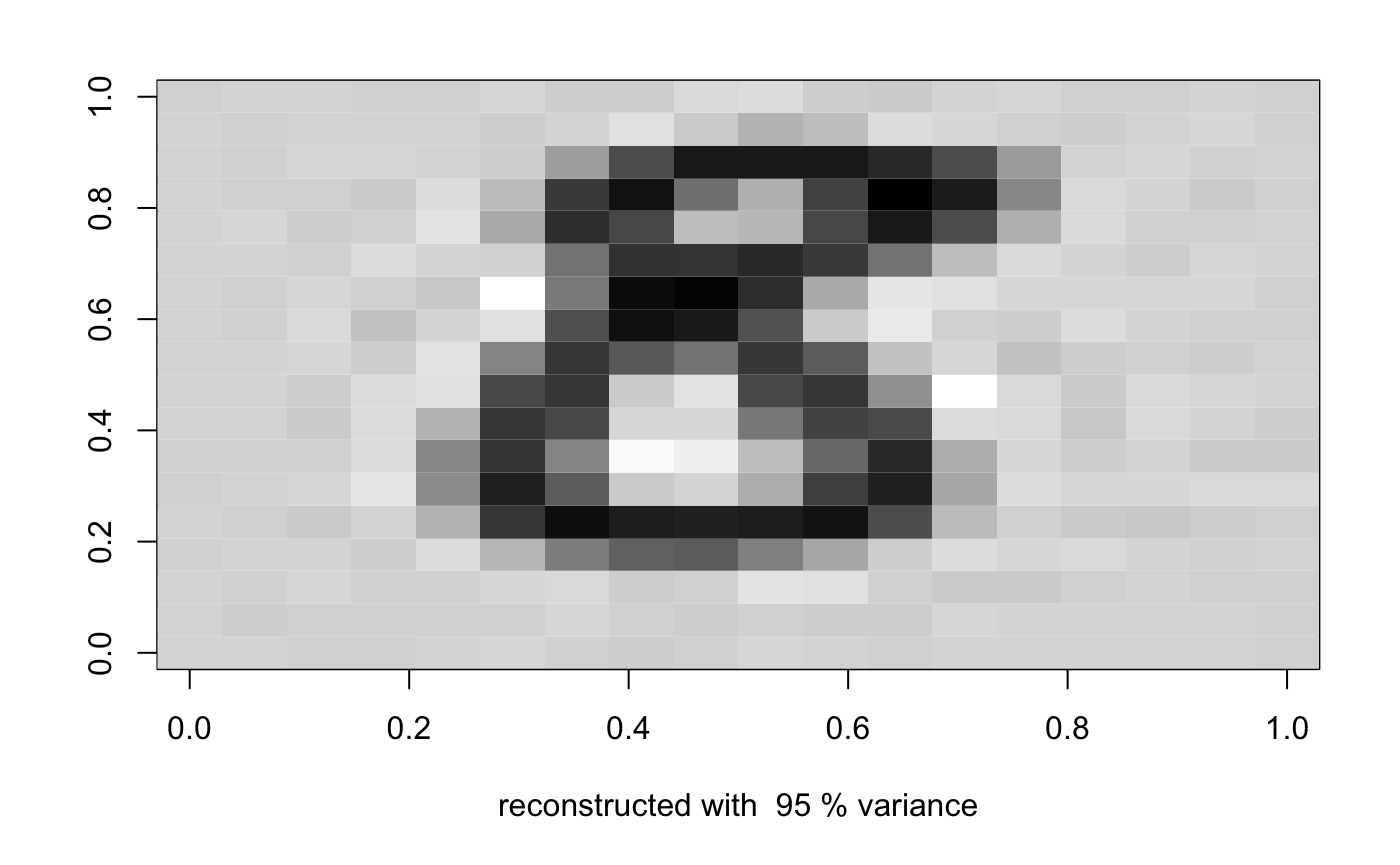
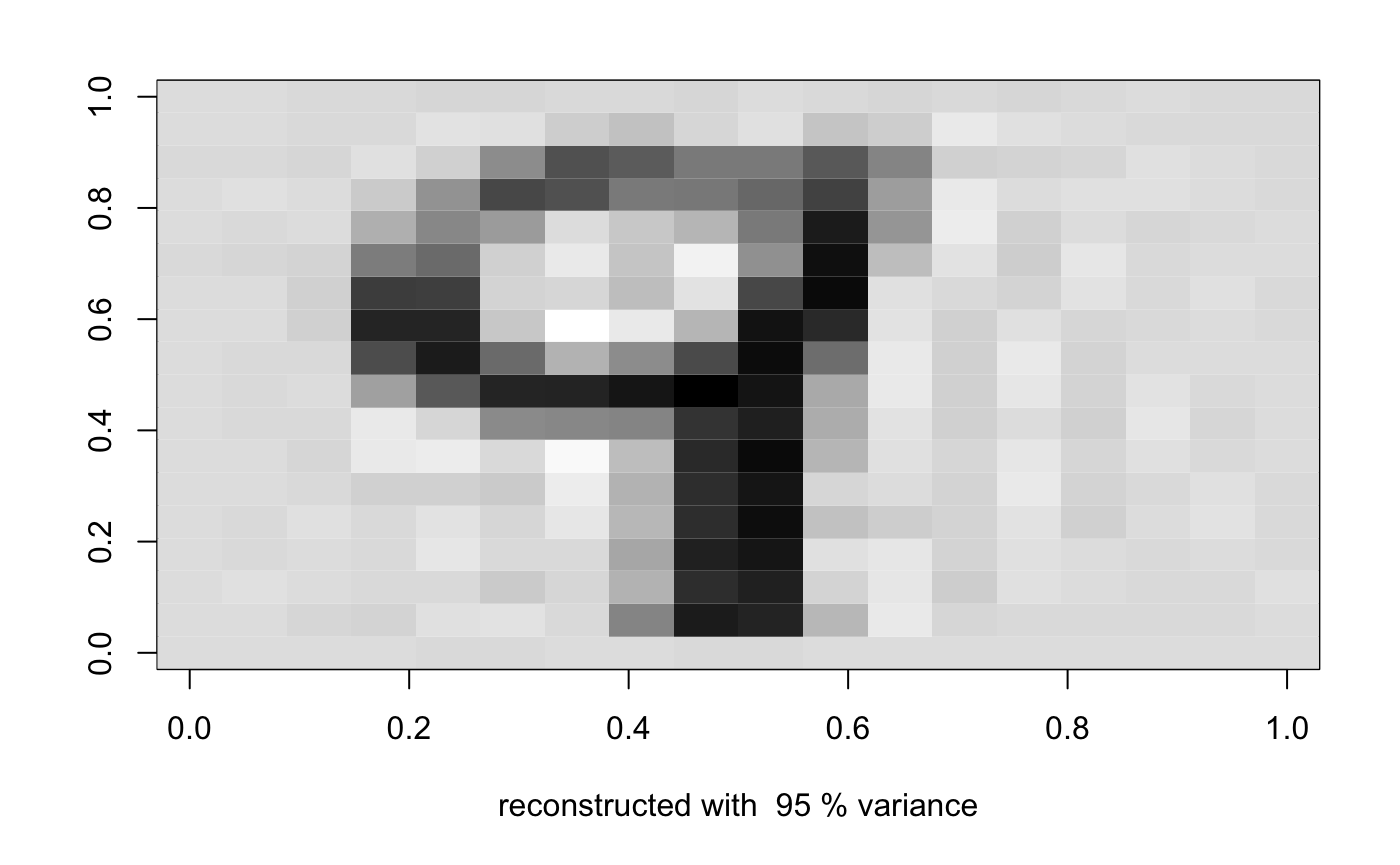
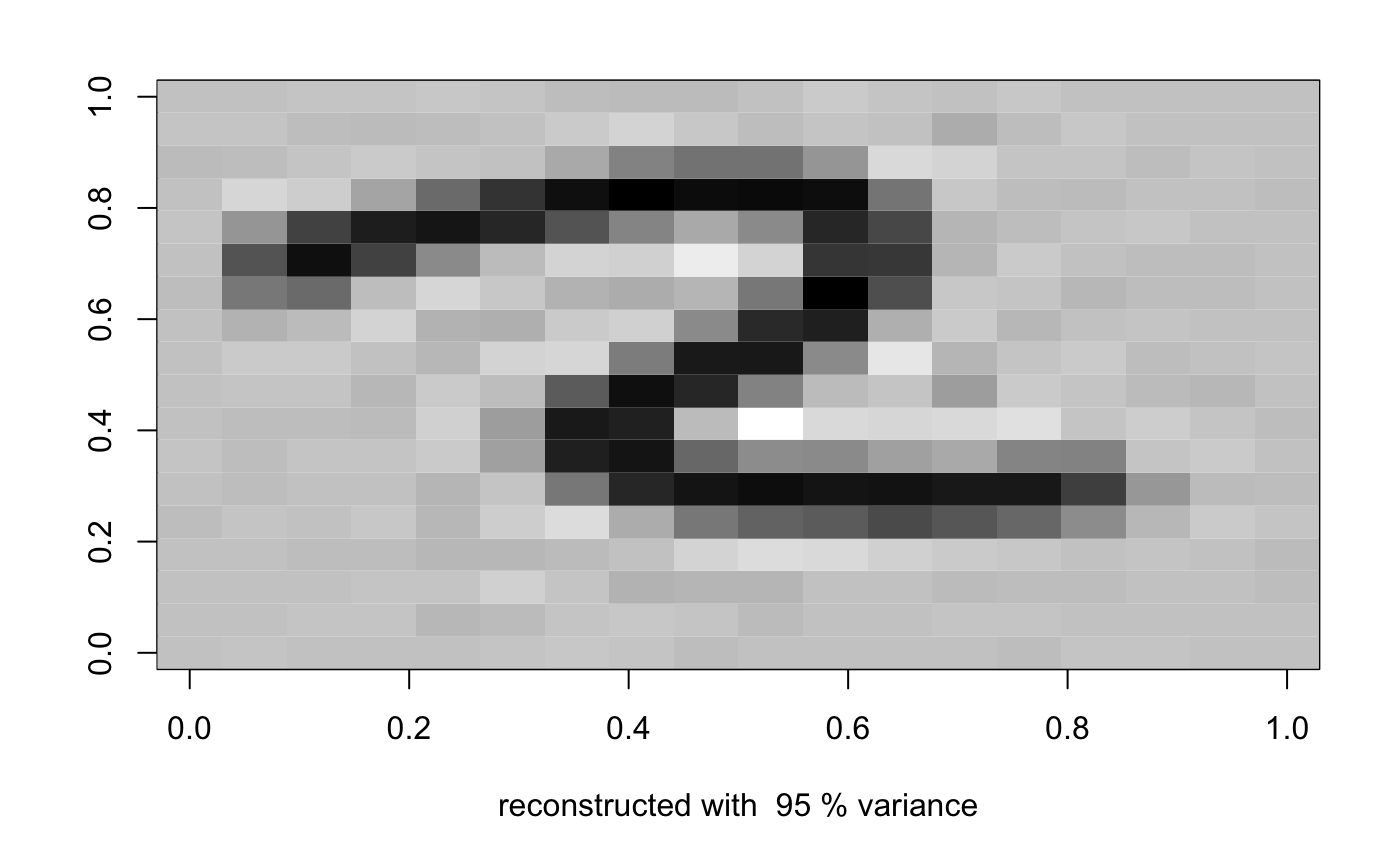
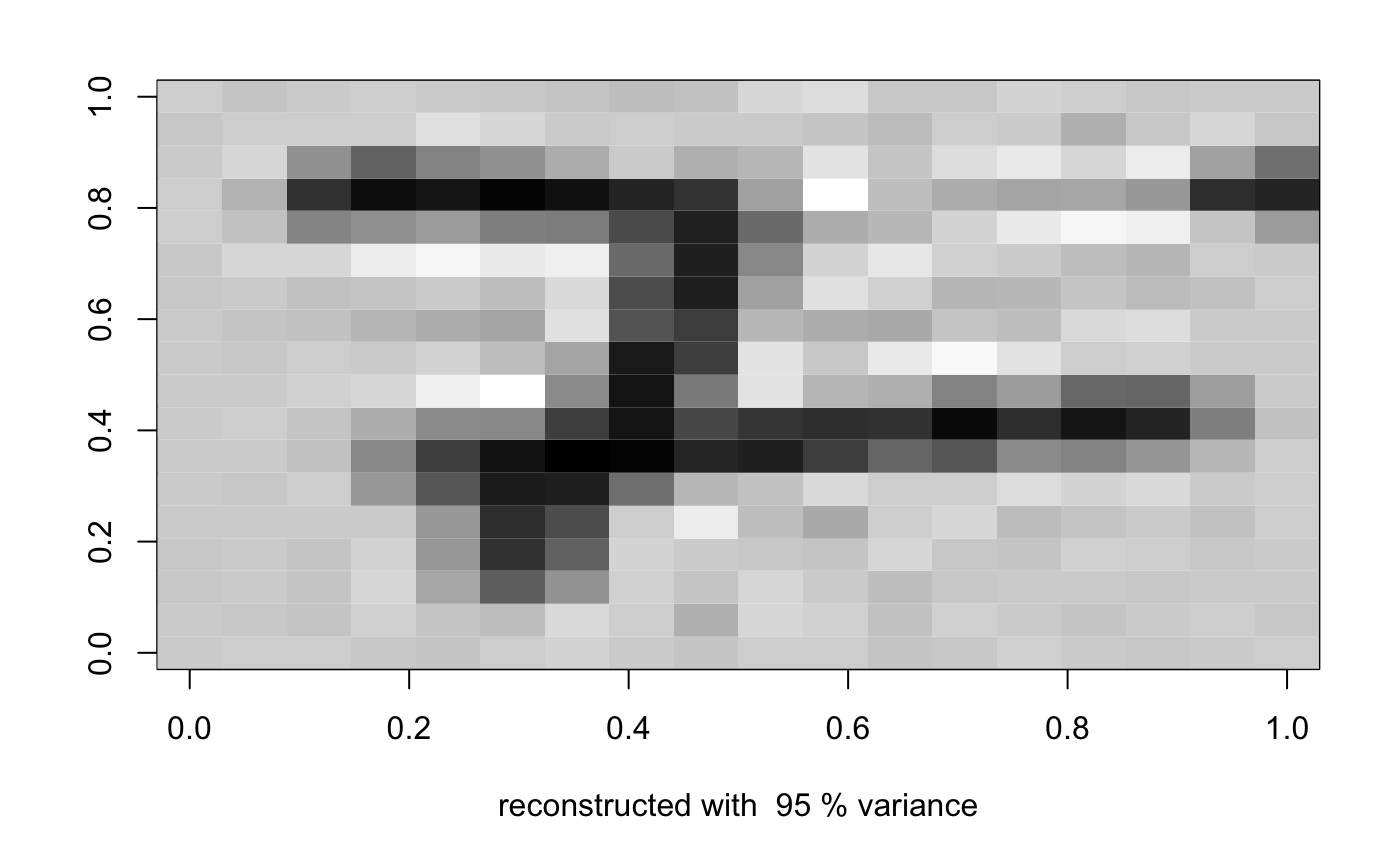
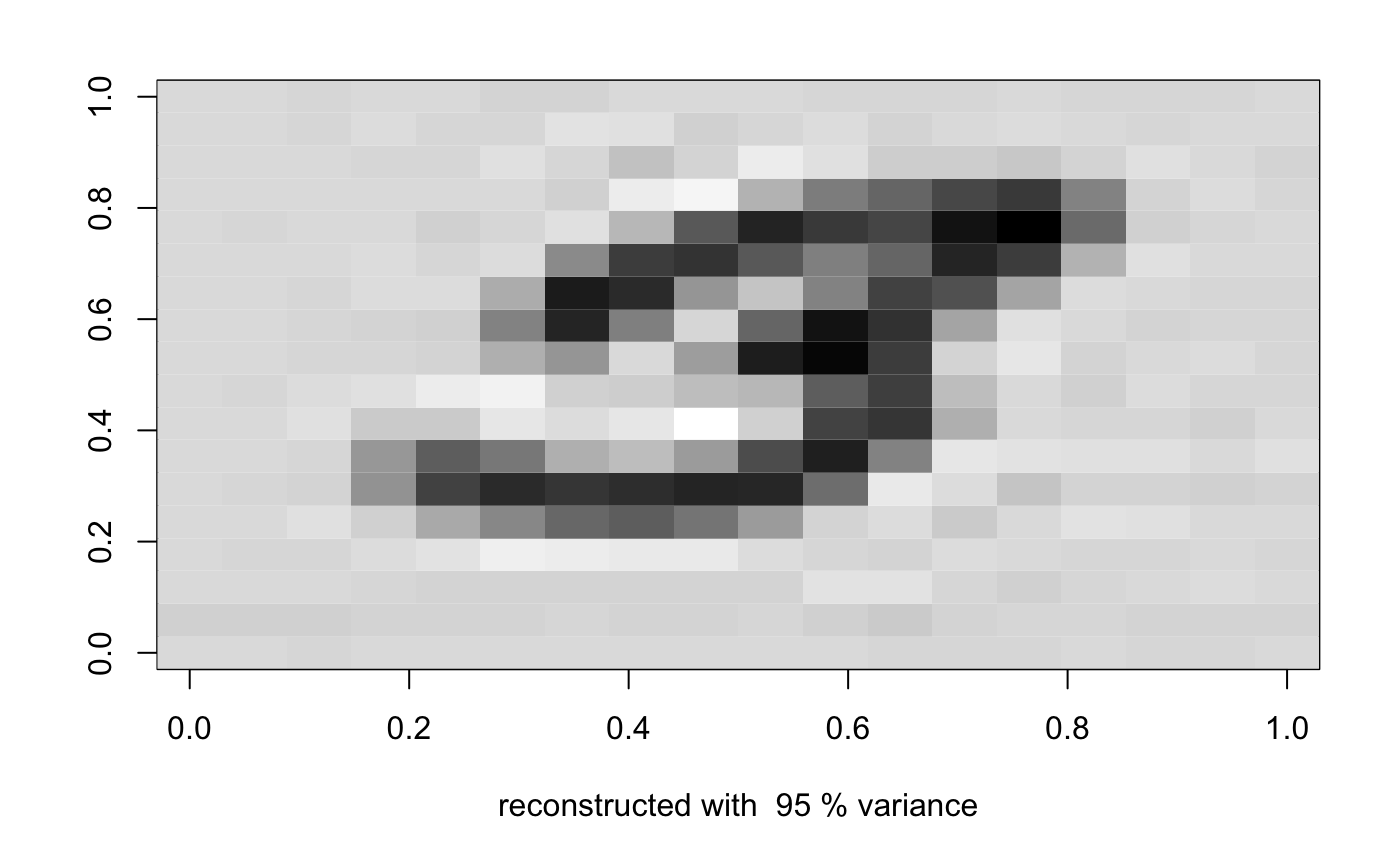
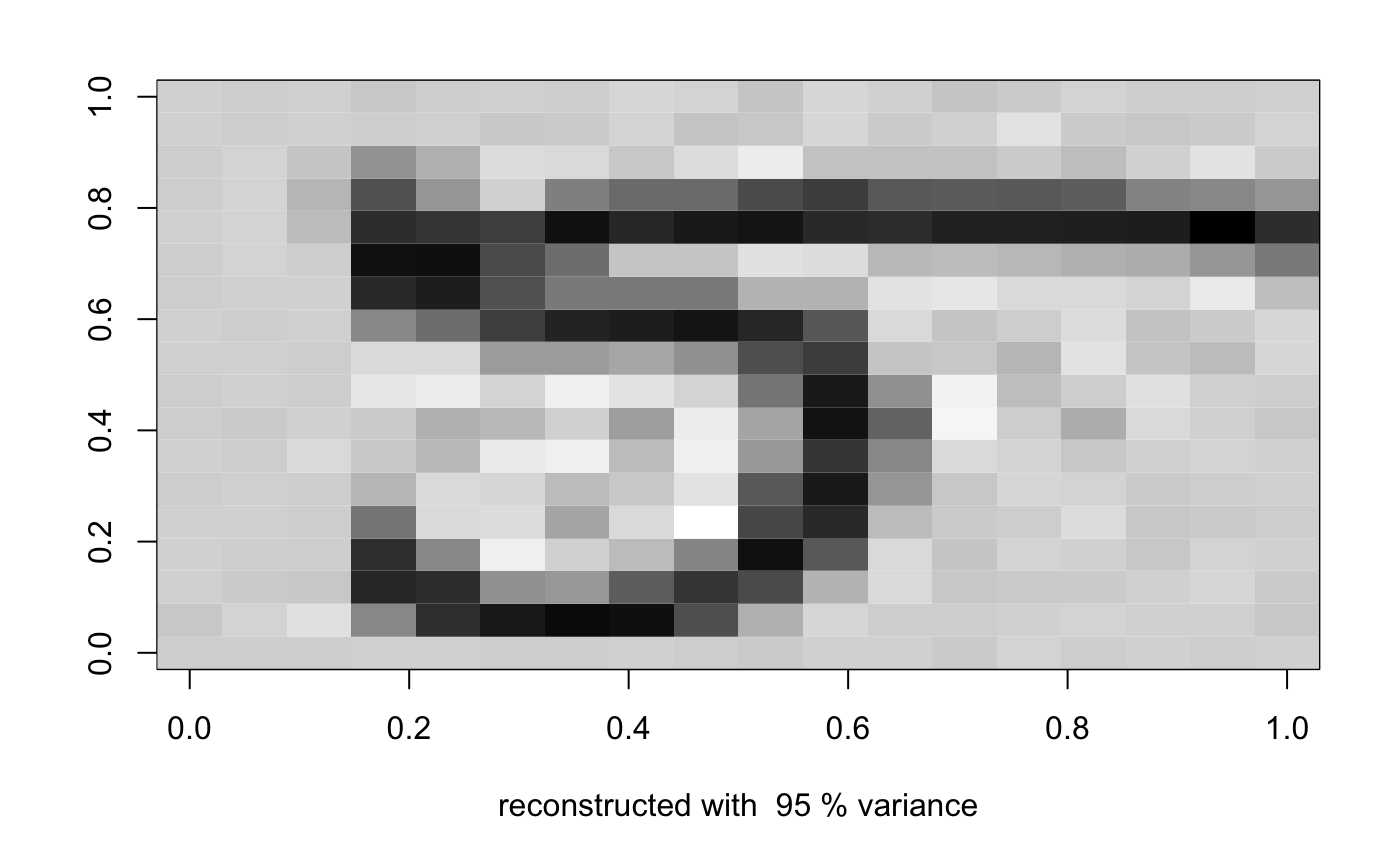






Reconstructed with 95% variance





Reconstructed with 99% variance

