statistics:
statistics.correlation(x, y, /, *, method='linear')

This function calculates the Pearson correlation coefficient of two datasets, given as a sequence of numbers. The design of this function is better with the introduction of x and y as positional only parameters since in the case of calculating the Pearson correlation coefficient, the order in which each dataset is inputted doesn’t matter. Also, because the function can take in several different types of data types, including arrays and tuples, you cannot specify exactly what the two initial arguments are in terms of their datatype.

hashlib:
hashlib.file_digest(fileobj, digest, /)

This function takes a file object (built in python file object, socket makefile, etc.) read in binary mode and digests it into a hashmap given the type of hashing method with the digest argument. This function is better specifically because of the removal of the use of “digest” as a keyword argument. The word digest is hard to understand as a reader, as it suggests many different possible uses, which is true given that it can be fed in as a string or hash constructor. However, the removal of fileobj as a keyword operator may be worse since fileobj still has a lot of meaning for a reader.

zlib:
zlib.compress(data, /, level=- 1, wbits=MAX_WBITS)

This function compresses the given “data” parameter and returns an object containing the compressed data. This function is better because of the removal of the data keyword, as the data keyword is very ambiguous in terms of what exactly can be fed into the compress function.