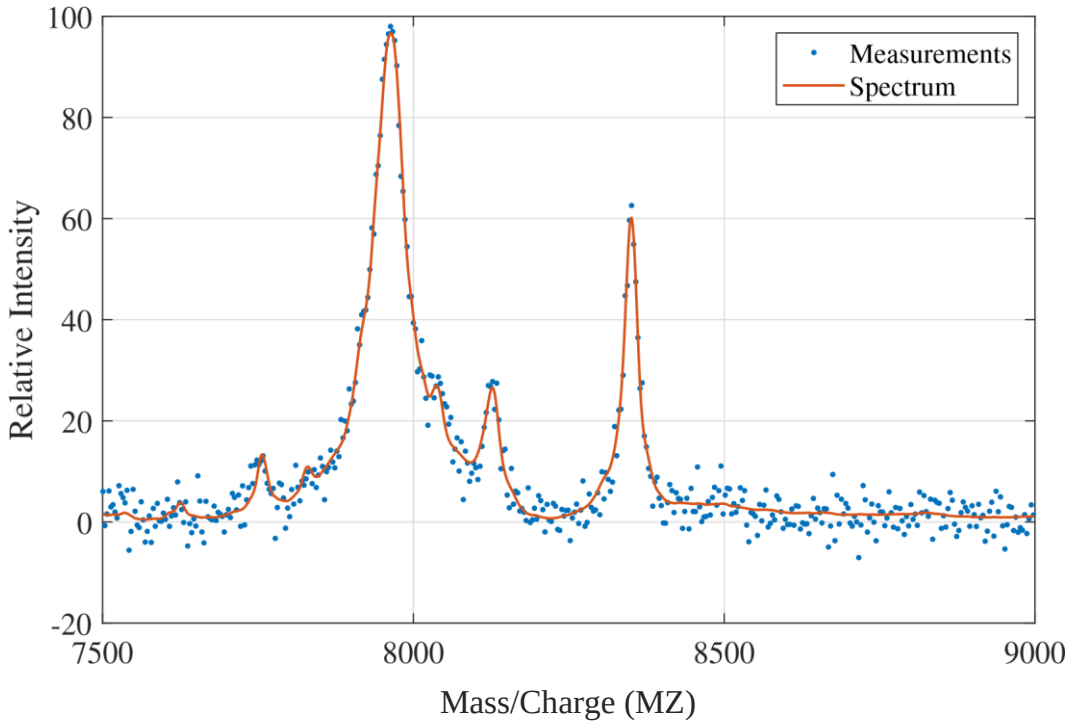

Figure 1: Noisy measurements from a (simulated) mass spectrometry reading. The "true" (unknown) measurements are shown in orange, and the noisy measurements are shown in blue.

Question 3 (14 marks)

Data Smoothing

Data "smoothing" is a very common problem in data science and statistics. We are often interested in examining the unknown relationship between a dependent variable (y) and an independent variable (x), under the assumption that the dependent variable has been imperfectly measured and has been contaminated by measurement noise. The model of reality that we use is

$$y = f(x) + \varepsilon$$

where $f(x)$ is some unknown, "true", potentially non-linear function of x, and $\varepsilon \sim N(0,\sigma^2)$ is a random disturbance or error. This is called the problem of function estimation, and the process of estimating $f(x)$ from the noisy measurements y is sometimes called "smoothing the data" (even if the resulting curve is not "smooth" in a traditional sense, it is less rough than the original data).

In this question you will use the k-nearest neighbours machine learning technique to smooth data. This technique is used frequently in practice (think for example the 14-day rolling averages used to estimate coronavirus infection numbers). This question will explore its effectiveness as a smoothing tool.

Mass Spectrometry Data Smoothing

The file ms.train.2020.csv contains n = 443 measurements from a mass spectrometer. Mass spectrometry is a chemical analysis tool that provides a measure of the physical composition of a material. The outputs of a mass spectrometry reading are the intensities of various ions, indexed by their massto-charge ratio. The resulting spectrum usually consists of a number of relatively sharp peaks that indicate a concentration of particular ions, along with an overall background level. A standard problem is that the measurement process is generally affected by noise – that is, the sensor readings are imprecise and corrupted by measurement noise. Therefore, smoothing, or removing the noise is crucial as it allows us to get a more accurate idea of the true spectrum, as well as determine the relative quantity of the ions more accurately. However, we would *ideally* like for our smoothing procedure to not damage the important information contained in the spectrum (i.e., the heights of the peaks).

The file ms.train.csv contains measurements of our mass spectrometry reading; ms.train$MZ are the mass-to-charge ratios of various ions, and ms.train$intensity are the measured (noisy) intensities of these ions in our material. The file ms.test.2020.csv contains n = 886 different values of MZ along with the "true" intensity values, stored in ms.test.2020$intensity. These true values have been found by using several advanced statistical techniques to smooth the data, and are being used here to see how close your estimated spectrum is to the truth. For reference, the samples ms.train$intensity and the value of the true spectrum ms.test$intensity are plotted in Figure 1 against their respective MZ values. To answer this question you must:

- Provide an R script containing all the code you used to answer the questions. Please use comments to ensure that the code used to identify each question is **clearly identifiable**. Call this file fn.sn.Q3.R, where "fn.sn" is your first name followed by your family name.
- Provide appropriate written answers to the questions, along with any graphs, in a non-handwritten report document.

To answer this question, you must use the kknn and boot packages that we used in Studios 9 and 10.

Questions

1. Use the k-nearest neighbours method (k-NN) to estimate the underlying spectrum from the training data. Use the kknn package we examined in Studio 9 to provide predictions for the MZ values in ms.test, using ms.train as the training data. You should use the kernel = "optimal" option when calling the kknn() function. This means that the predictions are formed by a weighted average of the k points nearest to the point we are trying to predict, the weights being determined by how far away the neighbours are from the point we are trying to predict.

   (a) For each value of k = 1,...,25, use k-NN to estimate the values of the spectrum for the MZ values in ms.test$MZ. Then, compute the mean-squared error between your estimates of the spectrum, and the true values in ms.test$intensity. Produce a plot of these errors against the various values of k. **[1 mark]**

   (b) Produce four graphs, each one showing: (i) the training data points (ms.train$intensity), (ii) the true spectrum (ms.test$intensity) and (iii) the estimated spectrum (predicted intensity values for the MZ values in ms.test.csv) produced by the k-NN method for four different values of k; do this for k = 2, k = 5, k = 10 and k = 25. Make sure the graphs have clearly labelled axis' and a clear legend. Use a different colour for your estimated curve. **[3 marks]**

(c) Discuss, qualitatively, and quantitatively (in terms of mean-squared error on the true spectrum) the four different estimates of the spectrum. **[2 marks]**

2. Use the cross-validation functionality in the kknn package to select an estimate of the best value of *k* (make sure you still use the optimal kernel). What value of *k* does the method select? How does it compare to the (in practice, unknown) value of *k* that would minimise the actual mean-squared error (as computed in Question 3.1a)? **[1 mark]**

3. Using the estimates of the curve produced in the previous question, see if you can provide an estimate of the variance of the sensor/measurement noise that has corrupted our intensity measurements. **[1 mark]**

4. Do any of the estimated spectra achieve our aim of providing a smooth, low-noise estimate of background level as well as accurate estimation of the peaks? Explain why you think the *k*-NN method is able to achieve, or not achieve, this aim. **[2 marks]** .

5. An important task when processing mass spectrometry signals is to locate the peaks, as this gives information on which elements are present. From the smoothed signal produced using the value of *k* found in Question 3.2, which value of MZ corresponds to the maximum estimated abundance? **[1 mark]**

6. Using the bootstrap procedure (use at least 5,000 bootstrap replications), write code to find a confidence interval for the *k*-nearest neighbours estimate of relative abundance at a specific MZ value. Use this code to obtain a 95% confidence interval for the estimate of relative abundance at the MZ value you determined previously in Question 3.5 (i.e., the value corresponding to the highest relative intensity). Compute confidence intervals using the *k* determined in Question 3.2, as well as *k* = 3 neighbour and *k* = 20 neighbours. Report these confidence intervals. Explain why you think these confidence intervals vary in size for different values of *k*. **[3 marks]**