Assignment A2a: Signal Detection
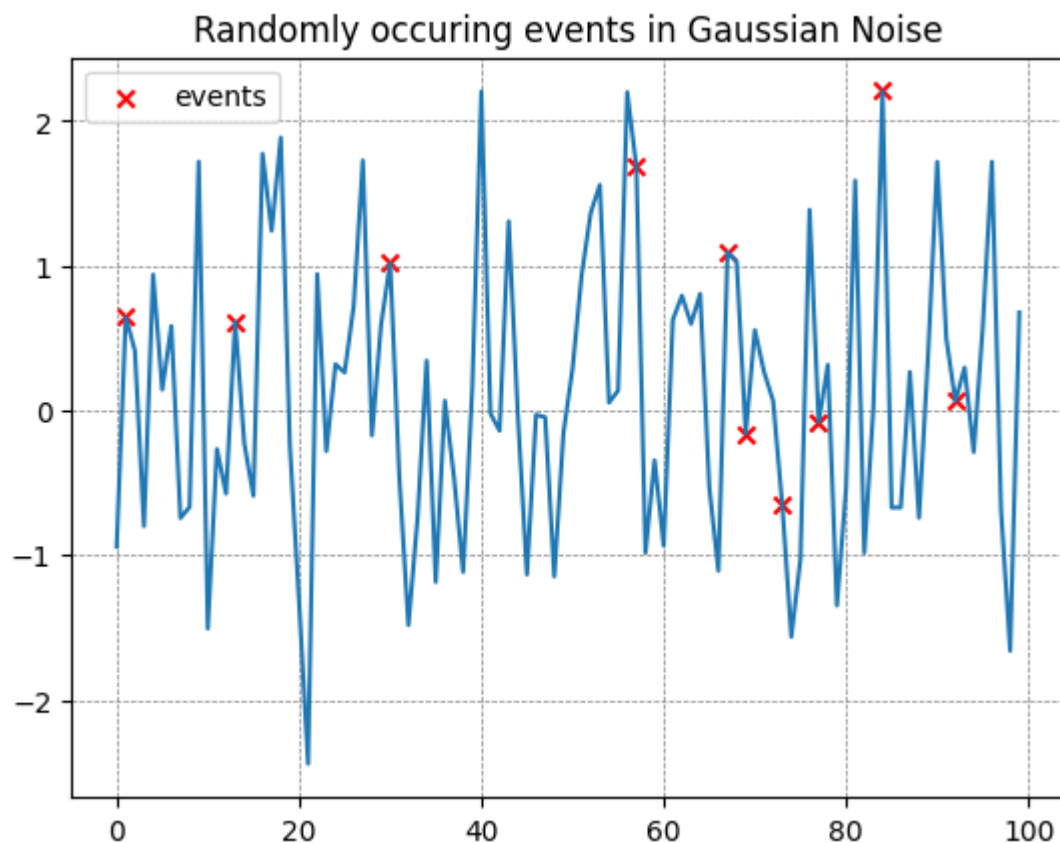
Hunter Welch 2/24/2023

1. Generating signals with events and additive noise

1a. Randomly occurring events in Gaussian noise

Write a function `genwaveform(N=100, α=0.1, A=1, σ=1)` to generate a waveform that is a linear combination of a sparsely occurring events and additive Gaussian noise.

```
In [8]:   from noisy_gaussian import plot_genwaveform

          plot_genwaveform()
```
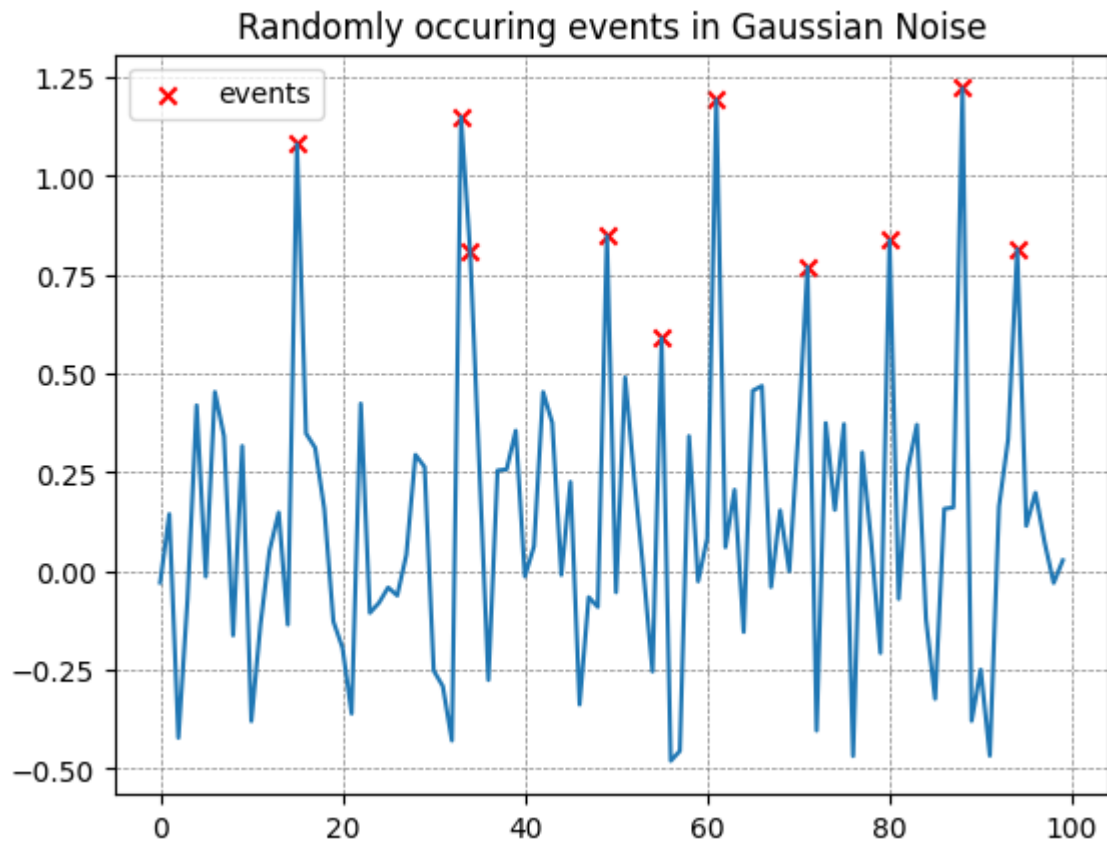

1b. Signals in uniform noise

Modify the `genwaveform` function so that it accepts an argument `noisetype` to specify the type of noise. Here we will use `Gaussian` and `uniform`.

```python
from noisy_gaussian import plot_genwaveform

plot_genwaveform(noisetype="uniform")
```


Randomly occuring events in Gaussian Noise

1c. Notation

To write an equation to express the signal $x[t]$ as a sum of N events of amplitude A that occur at times τ_i, it is important to use the discrete delta-function to denote when time τ is an event. The function of $x[t]$ would look like:

$$x[t] = \sum_{i=1}^{N} A * \delta[t - \tau]$$

The equation to denote the noise distributed according to a Normal is:

$$\epsilon[t] = p(\mu, \sigma^2)$$

1d. Conditional probability

The expression for the probability distribution of the waveform at time t given that there is a signal is the sum of the signal and the noise:

$$y[t] \sim N(A, \sigma)$$

Meaning that the probability distribution obeys the normal distribution according to the mean and standard deviation.

$$p(y[t]) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp - \frac{(x - A)^2}{2\sigma^2}$$

2. Signal detection

2a. Effect of parameters on detection probability

Explain what effect the parameters and type of noise have on detection probability. For what values does the probability reduce to pure chance? Or become certain (i.e. approach 1)? Explain your reasoning and illustrate with plots.

In [2]:
```python
from noisy_gaussian import plot_comparison

plot_comparison()
```
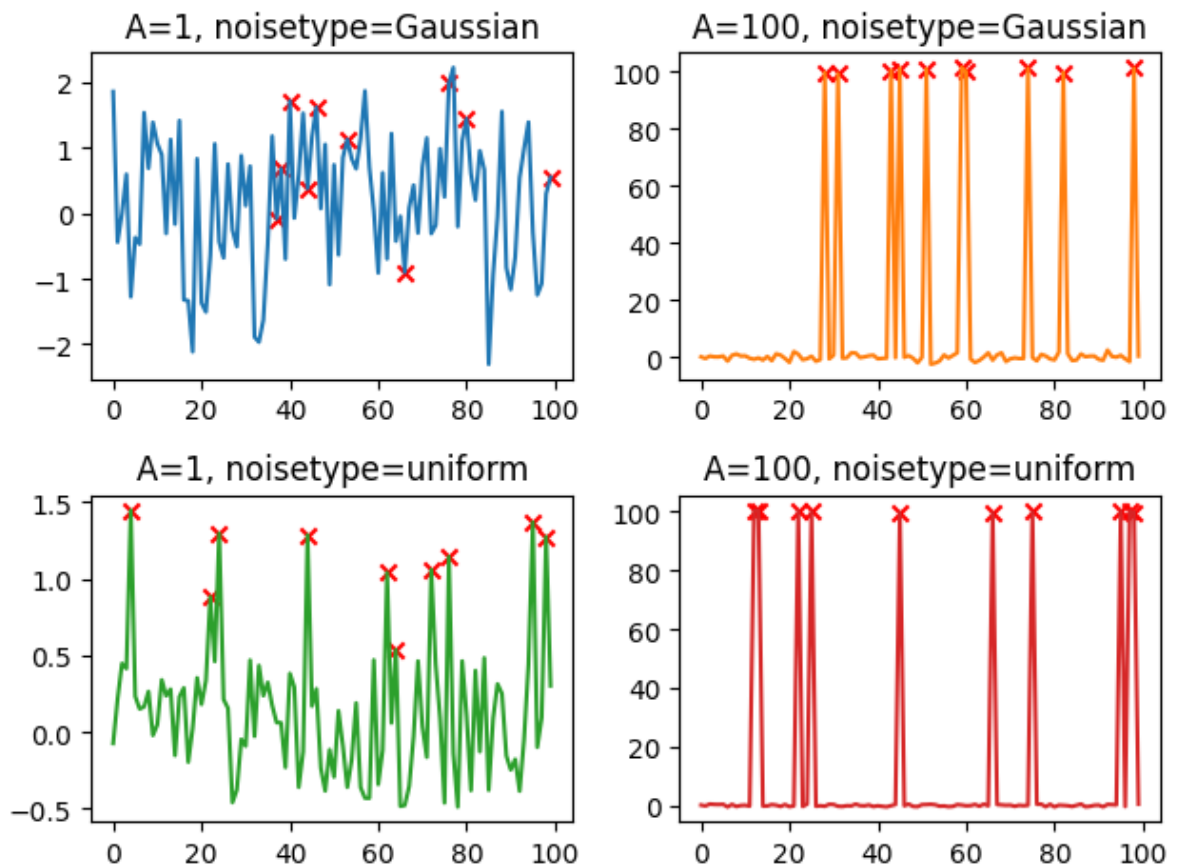
It is clear by the graphs that the noisetype does not play that big of a factor on detection probability, but uniform is slightly more clear.

The larger impact is having a higher amplitude makes it extremely clear in distinguishing events from noise. Having a higher A value will result in a higher SNR.
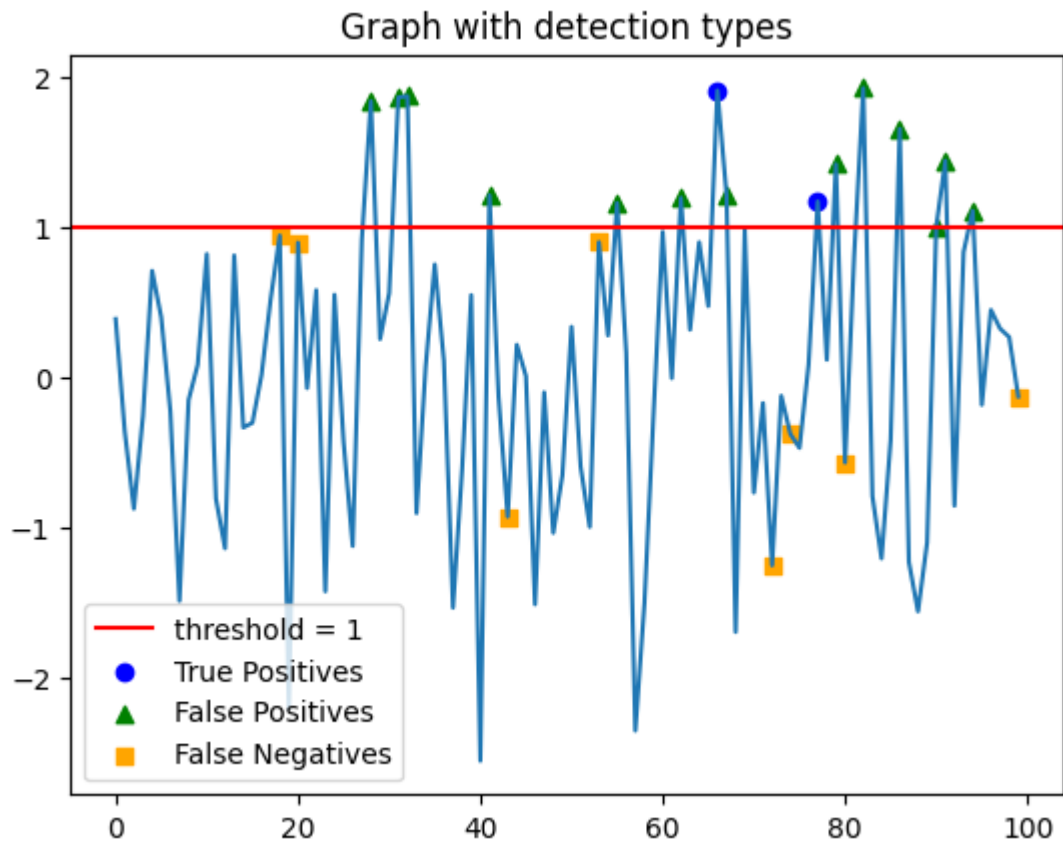
2b. Types of detections and detection errors

```python
from noisy_gaussian import plot_detection_types, detectioncounts, genwaveform

waveform, event_indices = genwaveform()

detectioncounts(event_indices, waveform, 1)

plot_detection_types(event_indices, waveform, 1)
```


2c. Detection probabilities

The probability of a false positive in terms of our parameters is the probability that a point is not on the signal indices array and over the threshold, given that it is not on the signal indices array. In other words, it is the probability the detection occurs given there is no signal. By utilizing the rules of probability and CDF we can rewrite this equation as

$$FRP = p(s > \theta)$$

$$FPR = 1 - p(signal \leq \theta)$$

$$p(signal \leq \theta) = \frac{1}{2}(1 + erf(\frac{\theta}{\sqrt{2} * \sigma}))$$

So

$$FPR = 1 - \frac{1}{2}(1 + erf(\frac{\theta}{\sqrt{2} * \sigma}))$$

Using the same logic we can find the rate of false negatives, knowing that the opposite logic holds true. The probability is that a point is under the threshold and on the signal indices array, given that it is on the signal indices array. In simplified terms of the CDF this can be written as:

$$FNR = \frac{1}{2}(1 + erf(\frac{\theta - A}{\sqrt{2} * \sigma}))$$

In [1]:
```python
from noisy_gaussian import falsepos, falseneg

print(falsepos(theta=2, alpha=0.25, A=2, sigma=1, noisetype="Gaussian"))
print(falseneg(theta=2, alpha=0.25, A=2, sigma=1, noisetype="Gaussian"))
```

```
0.02275013194817921
0.5
```

3. ROC cures

3a. Threshold considerations

There is not an optimal value for the threshold because it is just a separator to determine if something is being classified as one thing or another. By shifting the threshold in either direction, there will always incur a tradeoff. If you optimize for a minimum number of false positives, you will have a lot more false negatives. By using non-linear thresholds you can get even more exact but at the price of potentially overtuning the data.

The value that minimizes the total error probability would be where there is the lowest sum of false positives and false negatives errors. This is different from minimizing the toal number of errors, since it is specific to the dataset it is dealing with, rather than the entire problem

It is also important to note that in some cases false negatives and false positives have different effects. If a person is diangosed with cancer for example, a false positive could mean that they undergo unnecessary treatment, but a false negative could mean that they don't get treated and possibly lose their life. Therefore it is not possible to determine an optimal value for a threshold.

3b. ROC plot

In [1]:
```python
from noisy_gaussian import plotROC

plotROC()
```


ROC Curve