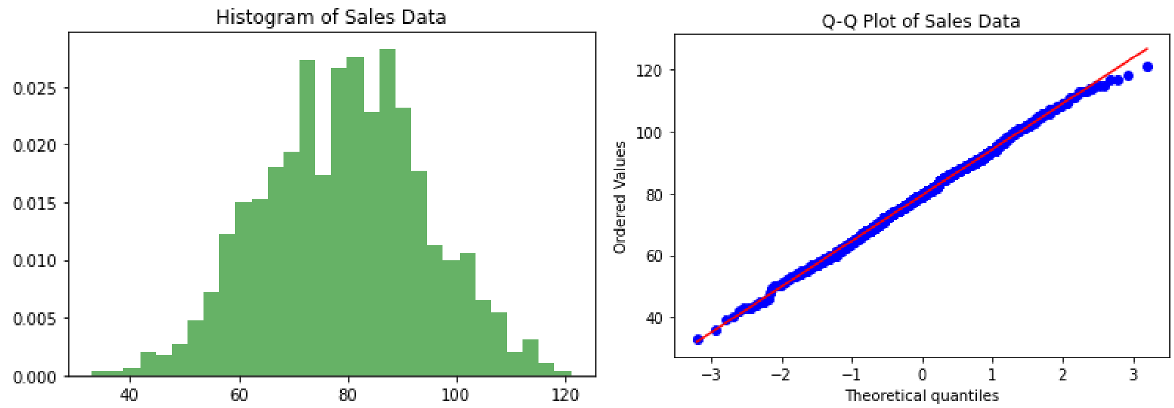
Problem 1

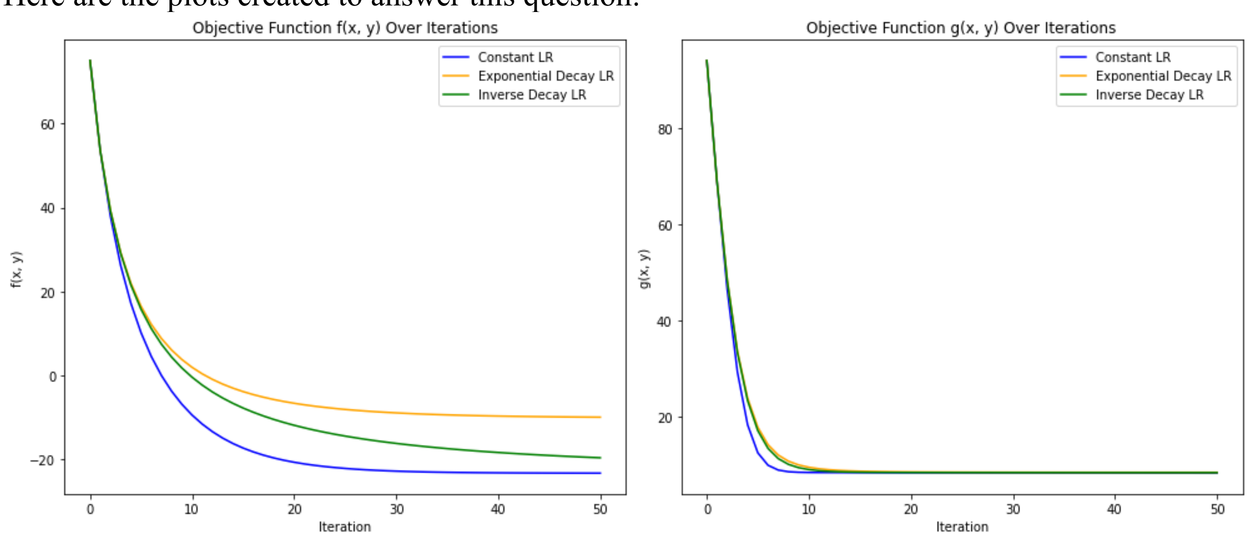
We can see from the histogram and a QQ plot of the sales data below, that the data follows a normal distribution. This is because the histogram shows a bell-shaped curve and the QQ plot shows that the data points fall approximately on the reference line.



After running a gradient descent function, the following parameters were obtained: mean = 79.379 and standard deviation = 14.78016776. These parameters indicate a central tendency of 79.379 and a spread of 14.78016776. The negative log-likelihood function is 4112.224799102469. Therefore, the sales data appears to be normally distributed as indicated by the histogram and the successful fitting of a normal distribution using gradient descent.

Problem 2

Here are the plots created to answer this question:



The performance comparison of different learning rate schedules for gradient descent optimization reveals that both the exponential and inverse decay learning rates outperform the constant learning rate in terms of convergence speed. The exponential decay rate, characterized by a rapid initial decline that stabilizes quickly, offers a quick approach towards the optimum, making it suitable for functions where the initial parameters are far from the optimum. On the other hand, the inverse decay rate reduces the learning rate more gradually, providing a more consistent and moderate convergence, which might be advantageous in navigating complex function landscapes with multiple local minima. Despite their different approaches, all three schedules seem to converge to similar final values, indicating that they are equally capable of finding the function's minimum, with the primary difference being the speed and stability of convergence.

The tuning process has provided us with the best combinations of initial learning rates and decay rates for both exponential and inverse decay learning rate schedules. Here are the results:

For $f(x,y)$:
Exponential Decay Learning Rate: Best initial learning rate $\gamma=0.001$, best decay rate $\delta=0.2$, optimized in 45 iterations.
Inverse Decay Learning Rate: Best initial learning rate $\gamma=0.1$, best decay rate $\delta=0.01$, optimized in 100 iterations.

For $g(x,y)$:
Exponential Decay Learning Rate: Best initial learning rate $\gamma=0.012$, best decay rate $\delta=0.137$, optimized in 31 iterations.
Inverse Decay Learning Rate: Best initial learning rate $\gamma=0.012$, best decay rate $\delta=0.2$, optimized in 178 iterations.

In addition to the results above, it was found that using the constant learning rate, function f was optimized in 86 iterations and function g was optimized in 1000 iterations. From the results, we can see that for function f, the exponential decay learning rate with these parameters yields the fastest optimization (45 iterations). For function g, again, the exponential decay learning rate outperforms the inverse decay in terms of the number of iterations required for optimization (31 iterations).