Observations

Note:

```
1. For simplicity sake, the sum of squared error is replaced by variance of error
2. The code is both Python2.7 and Python3 compatible
3. Plots have been included for each question in the zip file and as PDF. The
polynomial considered was degree 7 and N=10
4. Plots are in same order as the questions
5. For question 4, two codes have been attached - one for linear regression and
another for basis function regression
```

For Q1.

1. Noise variance which is being added to labels

Observations:

- As we increased the training samples, the error variance reduced very slightly (reduced by 0.1)
- The error variance has saturated at around 0.198 for 100,000 samples
- As the additive noise variance increased, the error variance also increased.
- Plot wise, small variation in variance didn't shift the graph much (plots for variance 0.05 and 0.1 were pretty close)

For Q2

1. Noise variance which is being added to labels
2. Degree of the polynomial basis function

Observations:

- The best generalization was observed when the degree of the polynomial was 6-8
- Variance increase in noise was resulting in slight bumos near the local maxima and minima of the polynomial curve
- When the degree of the polynomial crosses the number of training samples, the curve overfits - the data points are satisfied but generalization was not achieved. This can be observed immediately after degree 10

For Q3

1. Noise variance which is being added to labels
2. Degree of the polynomial basis function
3. Zero centering data
4. Lagrangian multiplier (λ)

Observations:

- Keeping $\lambda=0$ results same plots as Q2
- Varying λ by a factor of 10 results in improper curve fitting in the beginning
- The above two observations are when degree of the polynomial is less than the number of training training samples
- Zero centering and λ play an important role when degree of polynomial we're trying to fit is more than the training samples
- For instance, when $N = 10$ and degree is 13, zero centred curve fits the data better than the other case
- λ plays an important role when the polynomial we're trying to fit is in the proximity of training samples
- For instance, when $N = 10$ and degree ▯ [10,13], λ can be tweaked around to fit the data appropriately
- For very high degree polynomials, no matter how much we set λ, the tail part of polynomial fails to fit properly

For Q4

1. Noise variance which is being added to labels
2. Variance the labels can follow as its a normal distribution

Observations:

```
Note that in Q4 the best fit will be when
1/β = variance(y^-y)

When the likelihood labels have been calculated by np.random.normal(y^,1/β,N) and
plotted against
the curve and initial labels, they seem to be quite a good estimate.
```

For Q5

1. Noise variance which is being added to labels
2. Degree of the polynomial basis function
3. Zero centering data
4. The variance parameter of weights (**α**)
5. The variance parameter of labels (**σ**)

Observations:

- The observations here are pretty similar to that of Q3 considering $(\sigma/\alpha)^2 = \lambda$
- The flexibility we have here is we can idependently study the effects of α and σ
- For a degree more than the number of training samples, we can fix the variance in labels (**σ**) and experiment with (**α**) for a better fit
- Slowly reducing α in small steps can achieve the best fit
- For instance, consider $N = 10$ and polynomial degree of 11, the best fit occurs when $\sigma = 0.009$ and $\alpha = 0.01$