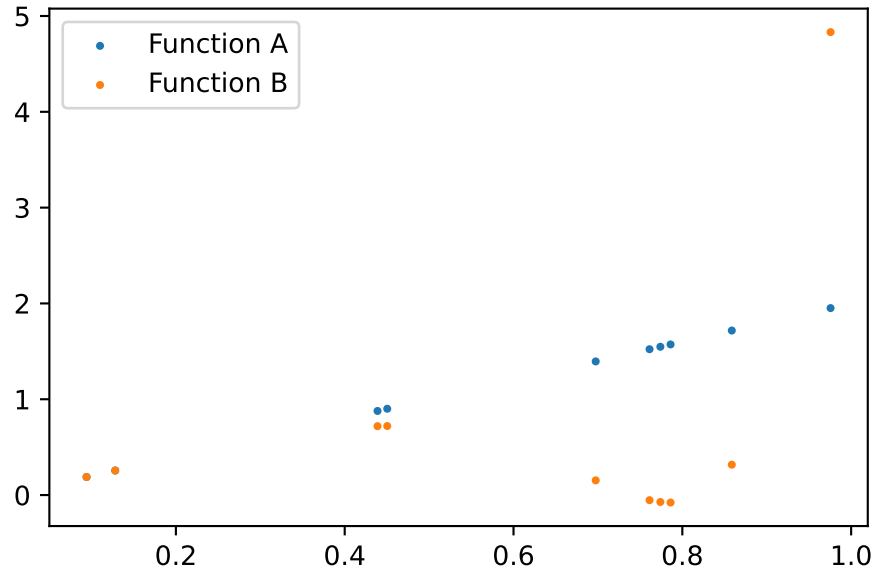
Generate the training set

We will generate data from two *hidden* functions:

$$f_A(x) = 2x,$$

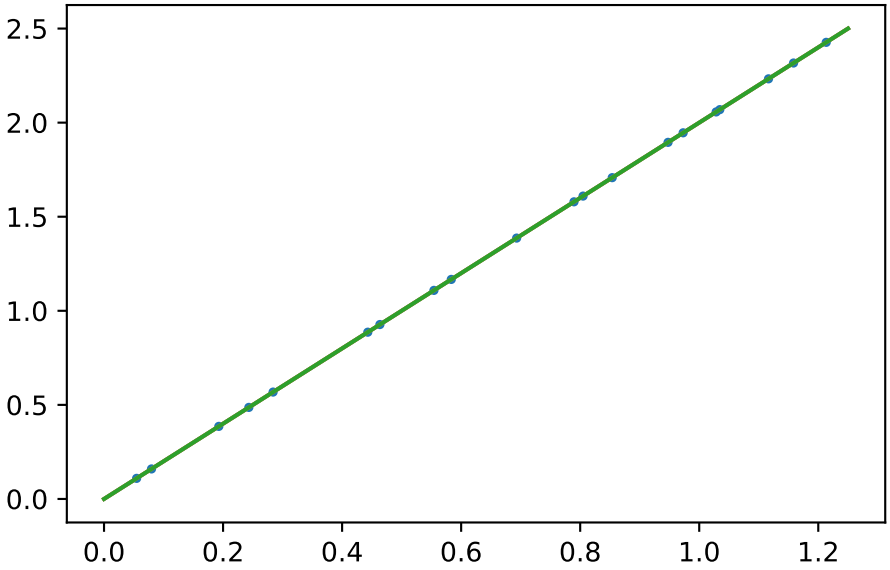
and

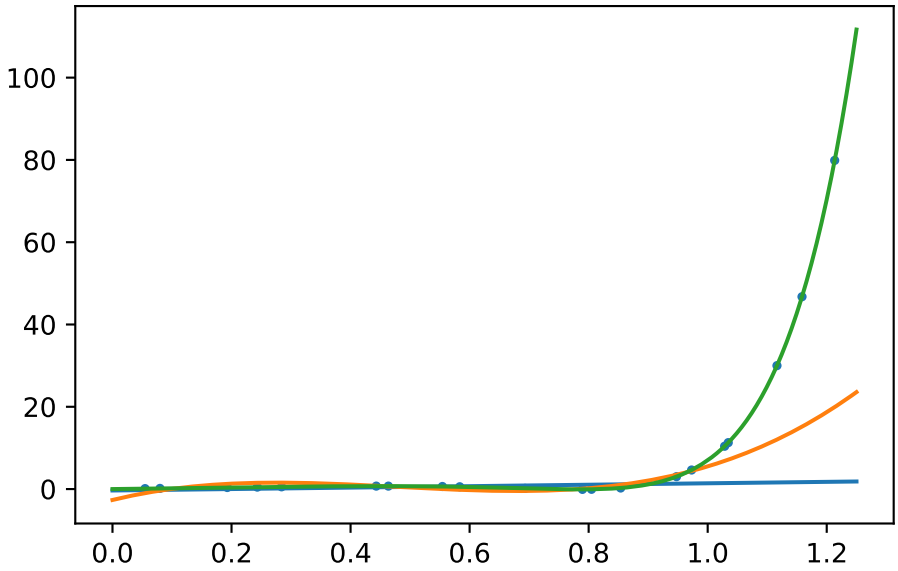
$$f_B(x) = 2x - 10x^5 + 15x^{10}.$$



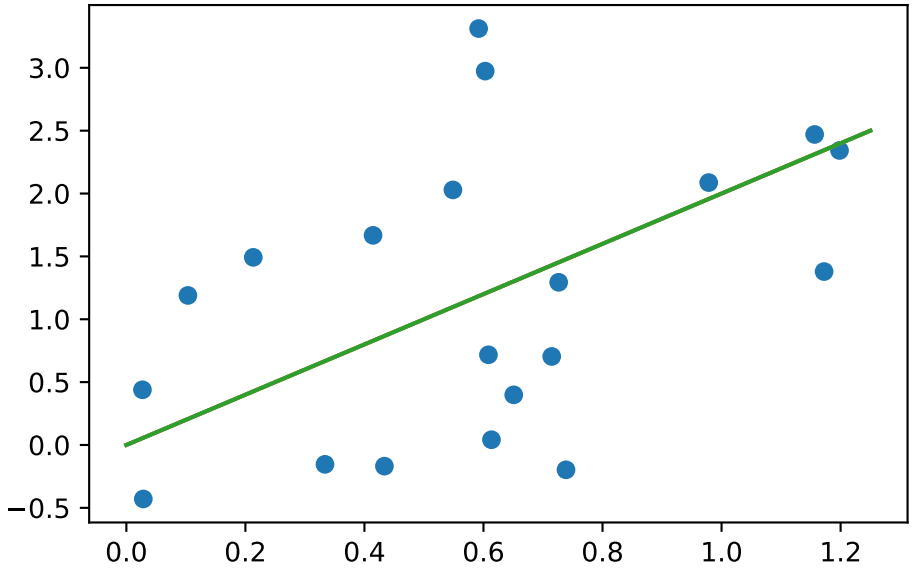

```
/tmp/ipykernel_57472/736400789.py:3: RankWarning: Polyfit may be poorly conditioned
  p10_a = np.polyfit(x_train, y_train_a, 10)
/tmp/ipykernel_57472/736400789.py:11: RankWarning: Polyfit may be poorly conditioned
  p10_b = np.polyfit(x_train, y_train_b, 10)
```

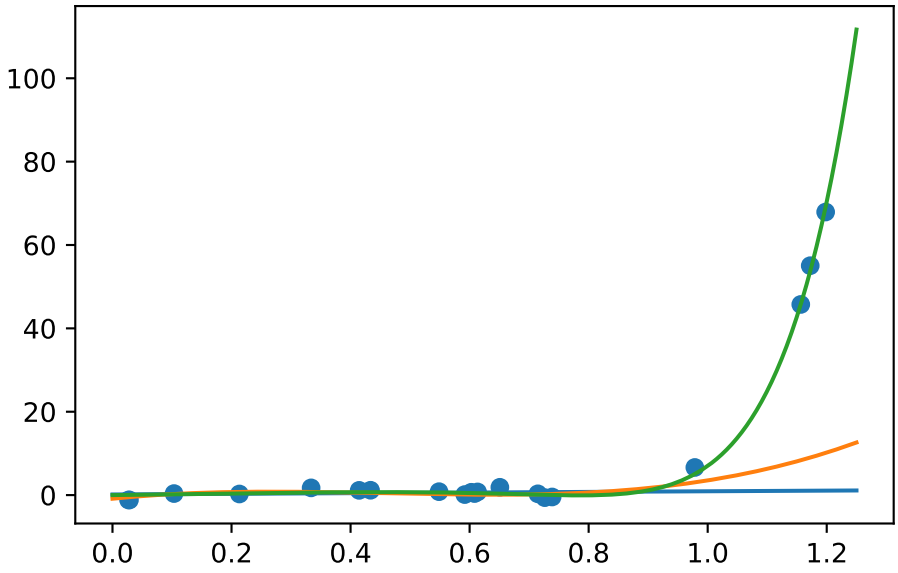
Generate the test set ($\sigma = 0, p = 10, p_t = 20$ case)

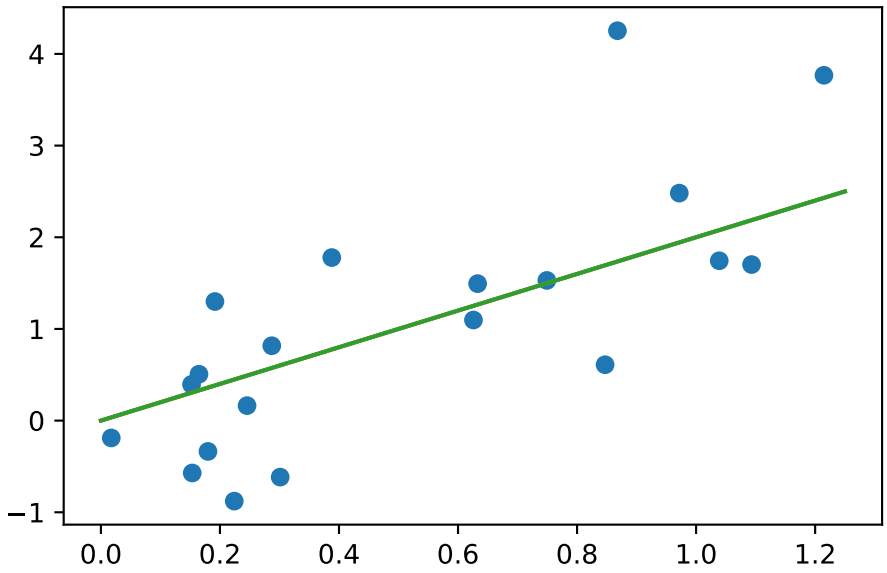


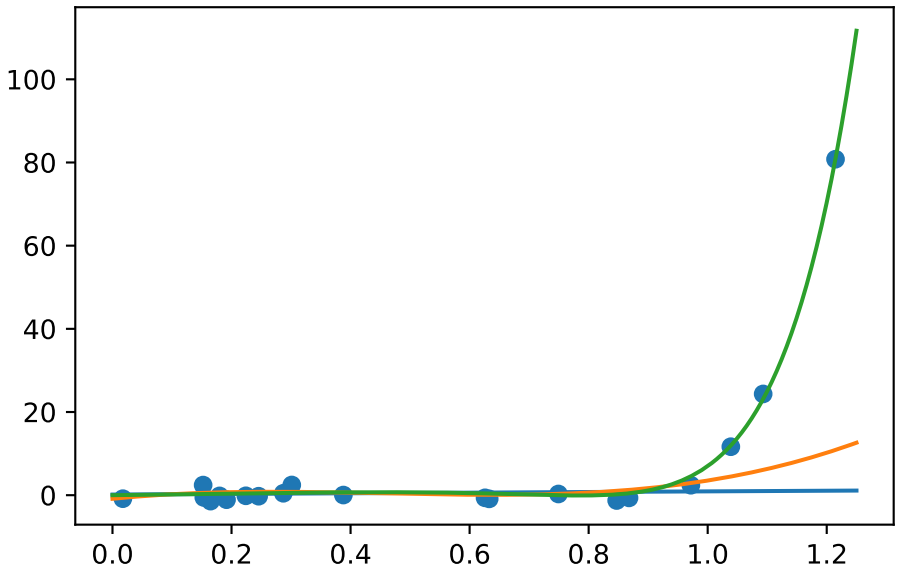


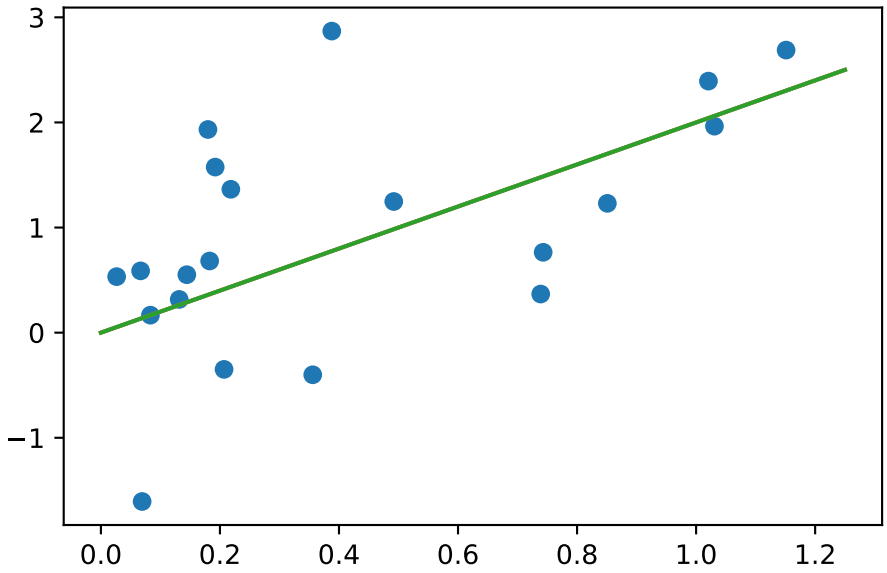
Generate the training and test set ($\sigma = 1, p = 100, p_t = 20$ case)

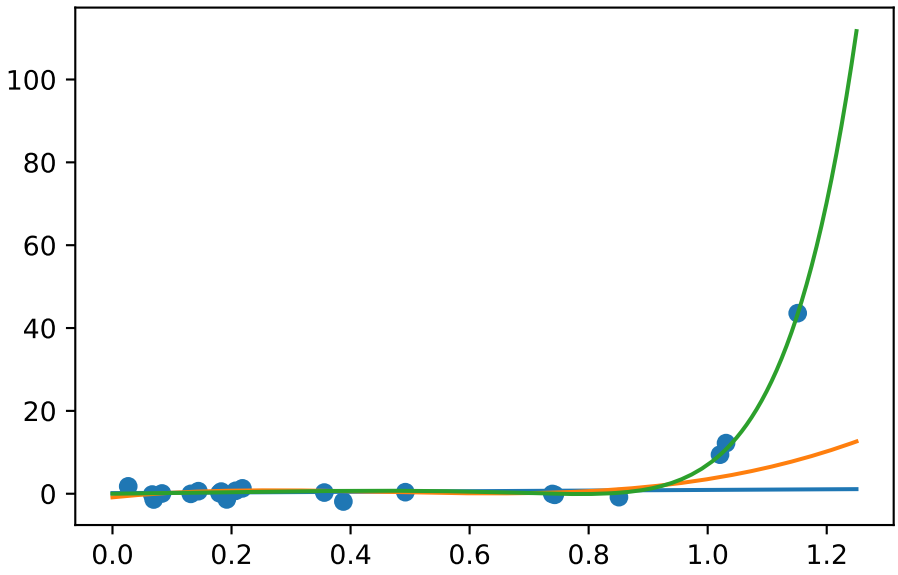


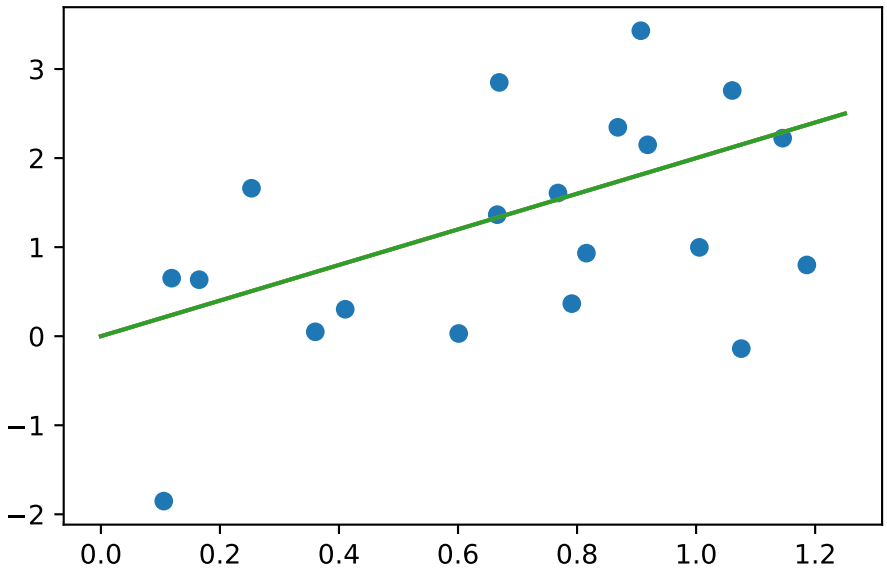


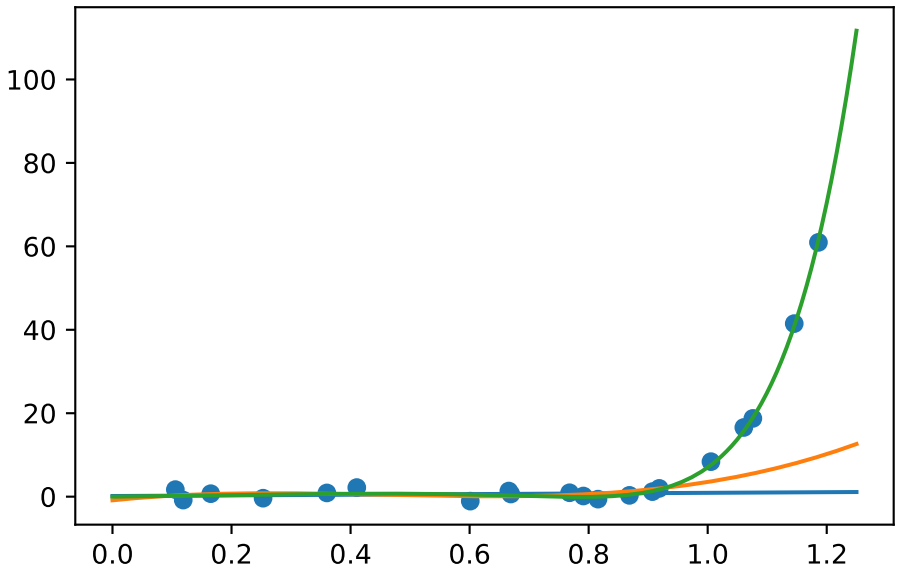

4
3
2
1
0
−1
0.0
0.2
0.4
0.6
0.8
1.0
1.2

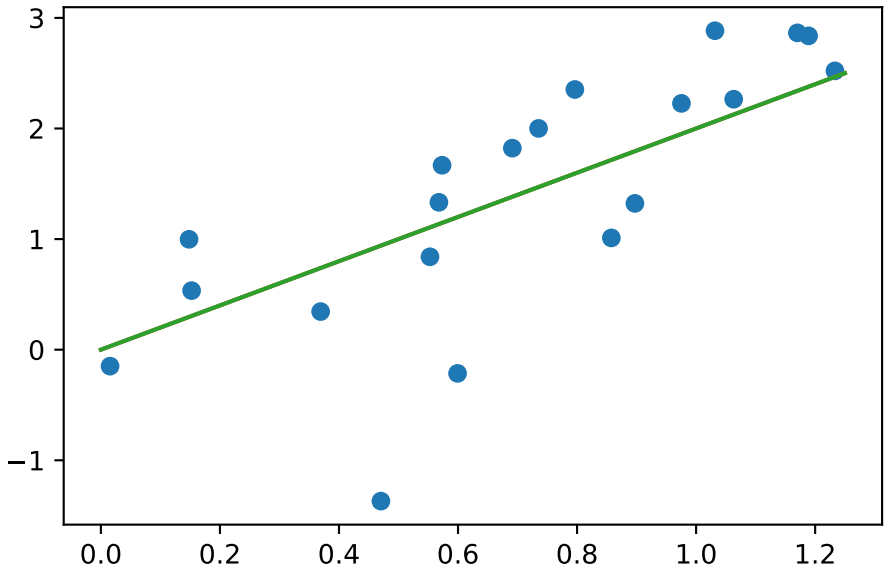

100
80
60
40
20
0
0.0
0.2
0.4
0.6
0.8
1.0
1.2

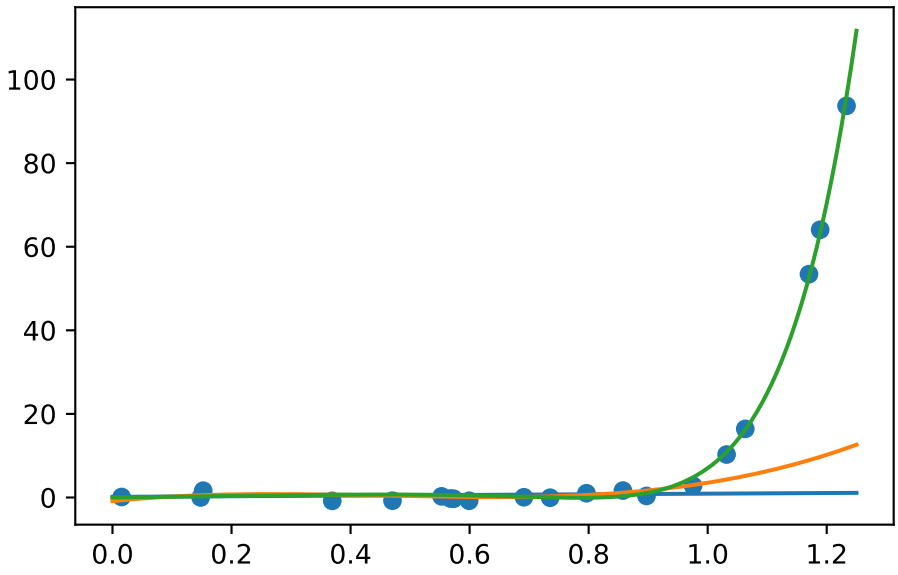

3
2
1
0
−1
0.0
0.2
0.4
0.6
0.8
1.0
1.2


100
80
60
40
20
0
0.0
0.2
0.4
0.6
0.8
1.0
1.2

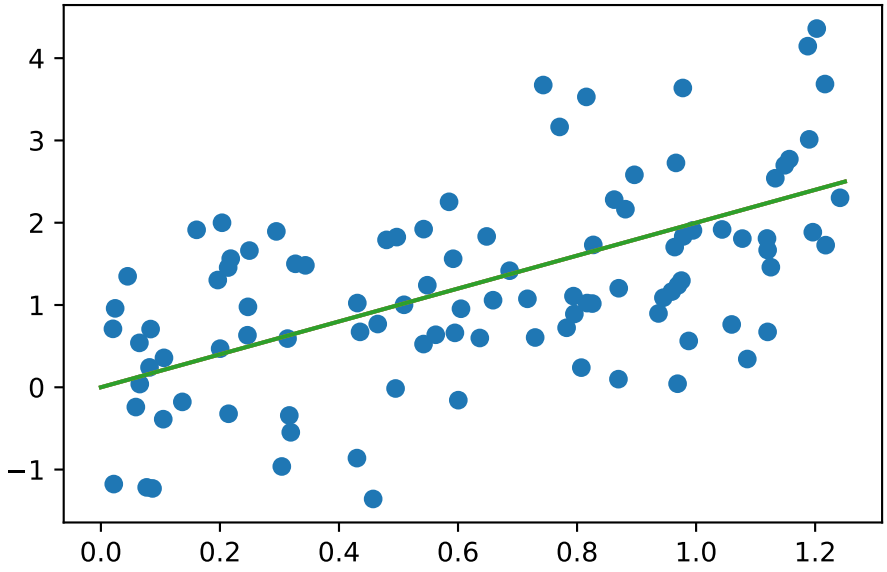

3
2
1
0
−1
−2
0.0
0.2
0.4
0.6
0.8
1.0
1.2

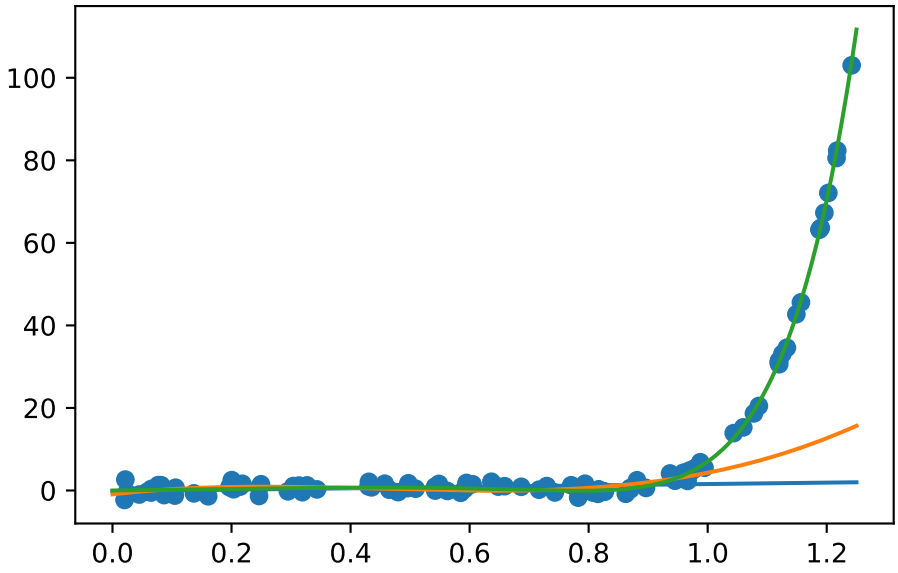

100
80
60
40
20
0
0.0
0.2
0.4
0.6
0.8
1.0
1.2

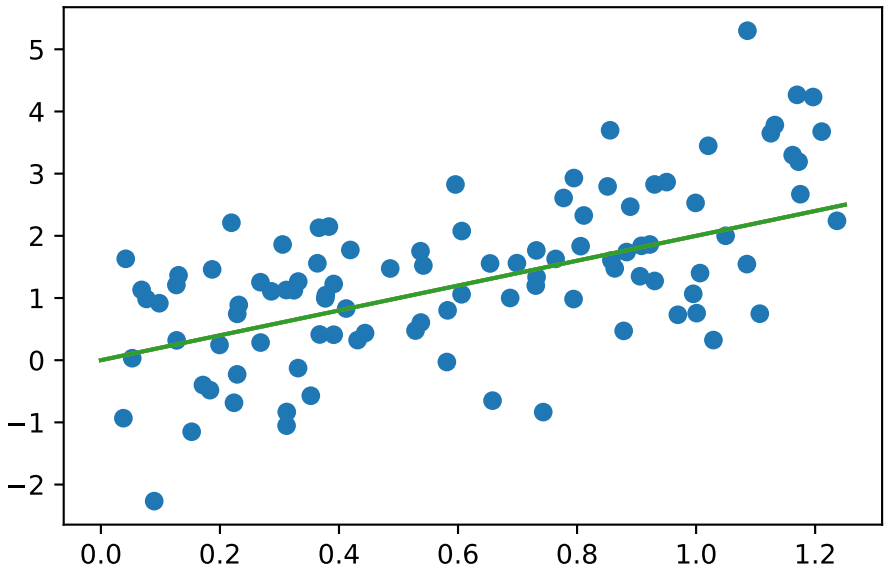

3
2
1
0
−1
0.0
0.2
0.4
0.6
0.8
1.0
1.2

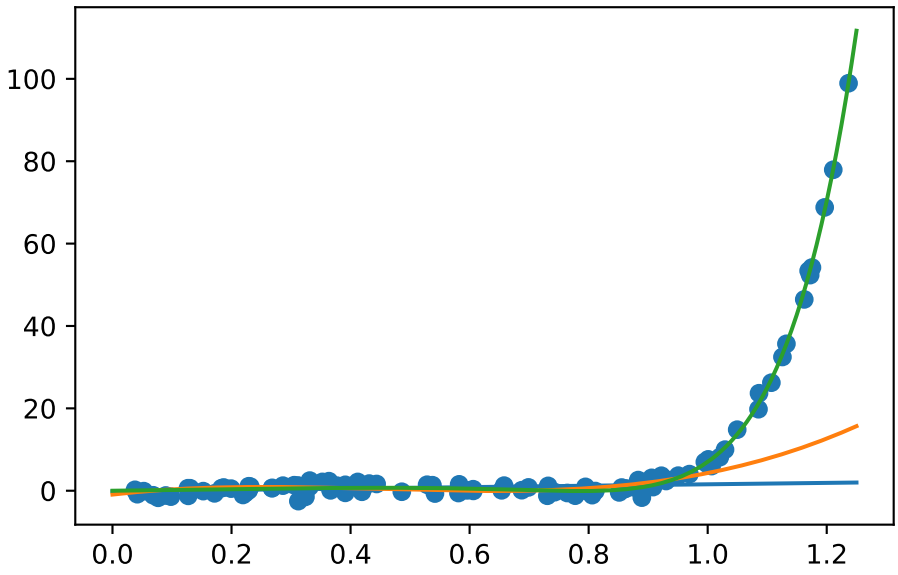

100
80
60
40
20
0
0.0
0.2
0.4
0.6
0.8
1.0
1.2

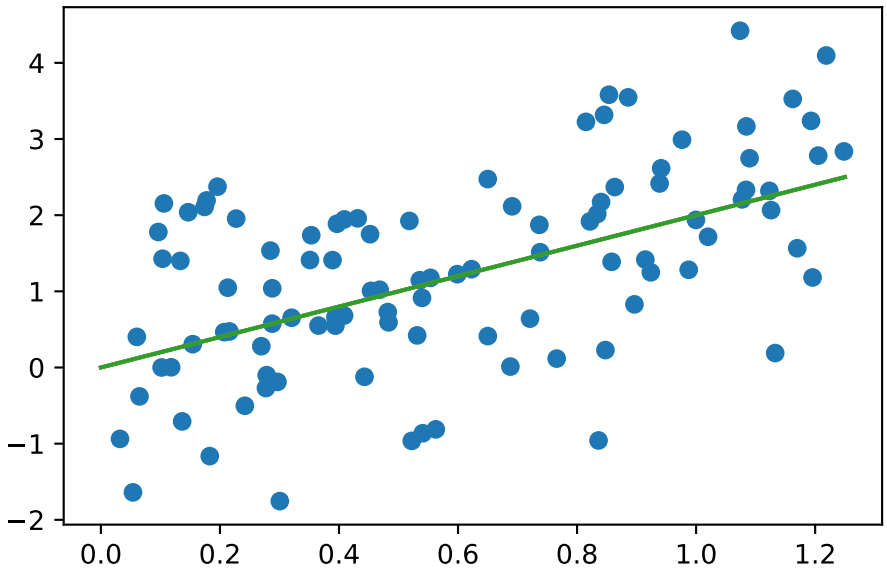
Generate the training and test set ($\sigma = 1, p = 10000, p_t = 100$ case)

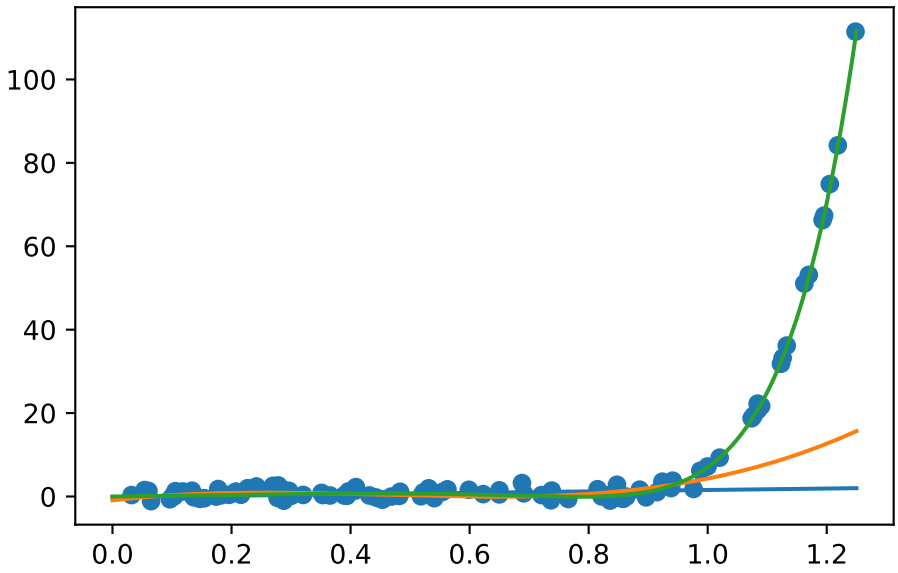


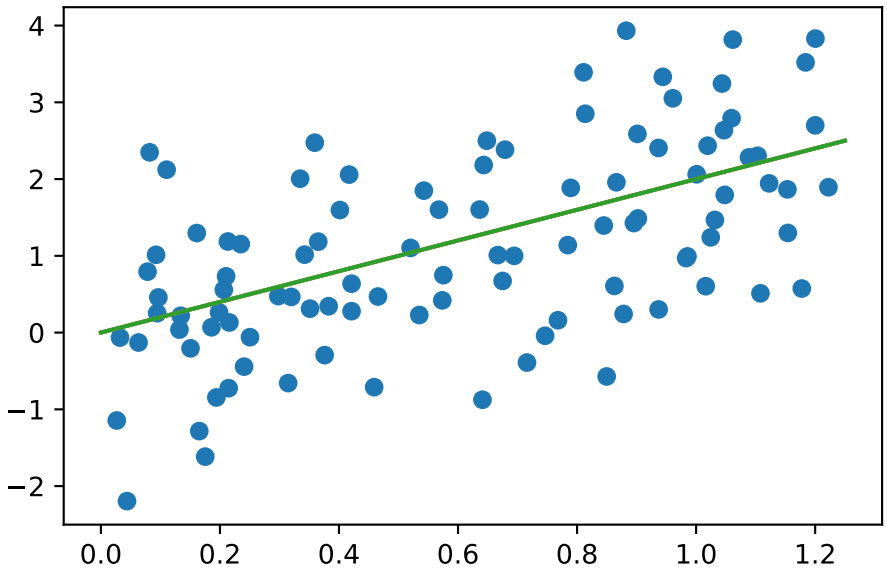


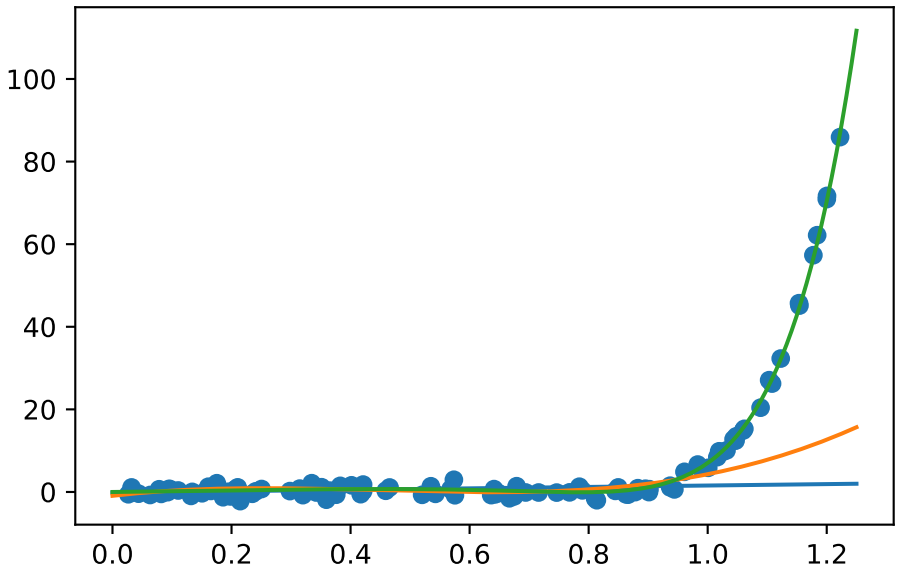

5
4
3
2
1
0
−1
−2
0.0
0.2
0.4
0.6
0.8
1.0
1.2

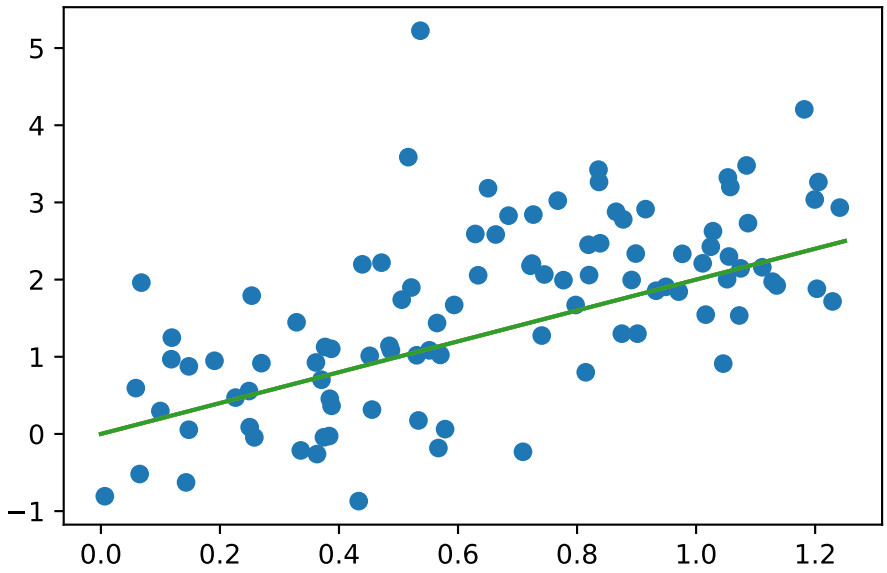

100
80
60
40
20
0
0.0
0.2
0.4
0.6
0.8
1.0
1.2

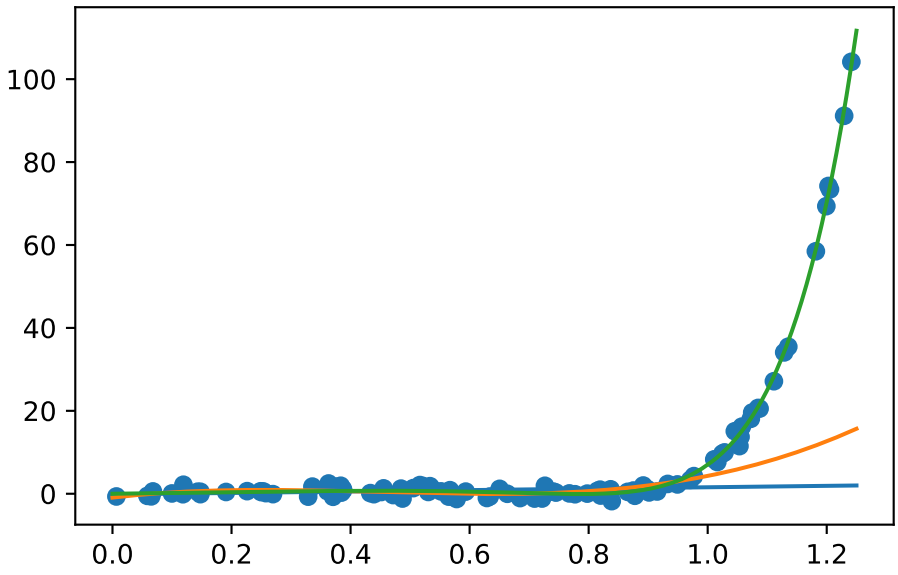

4
3
2
1
0
−1
−2
0.0
0.2
0.4
0.6
0.8
1.0
1.2


100
80
60
40
20
0
0.0
0.2
0.4
0.6
0.8
1.0
1.2


4
3
2
1
0
−1
−2
0.0
0.2
0.4
0.6
0.8
1.0
1.2


100
80
60
40
20
0
0.0
0.2
0.4
0.6
0.8
1.0
1.2


5
4
3
2
1
0
−1
0.0
0.2
0.4
0.6
0.8
1.0
1.2


100
80
60
40
20
0
0.0
0.2
0.4
0.6
0.8
1.0
1.2