

Assignment08_20133096_HyunjaeLee

May 10, 2019

1 20133096 Hyunjae Lee

[Polynomial fitting]

Solve a least square problem to find an optimal polynomial curve for a given set of two dimensional points.

Demonstrate the effect of the degree of polynomial in fitting a given set of points.

- choose a polynomial curve and generate points along the curve with random noise
- plot the generated noisy points along with its original polynomial without noise
- plot the approximating polynomial curve obtained by solving a least square problem
- plot the approximating polynomial curve with varying polynomial degree

1.1 1. Setup

x : x-coordinate data

y1 : (noisy) y-coordinate data

y2 : (clean) y-coordinate data

num : the number of data

std : standard deviation

```
In [52]: import numpy as np
         import matplotlib.pyplot as plt

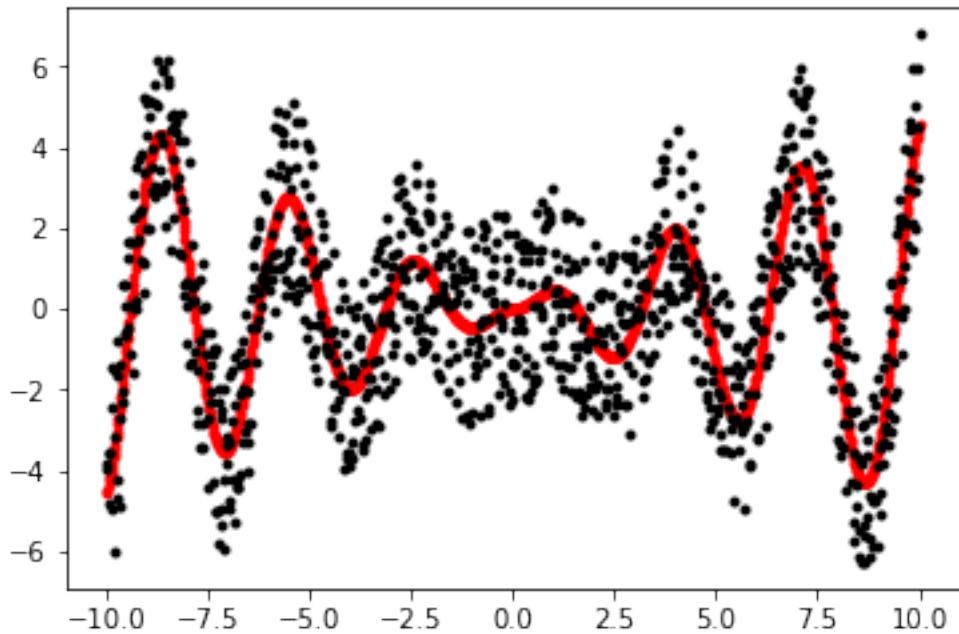
         def function(x):
             f = np.sin(x) * np.cos(x) * np.abs(x)
             return f
```

```
In [58]: num = 1000
         std = 5
         n = np.random.rand(num)
         de_mean = n - np.mean(n)

         x = np.linspace(-10, 10, num)
         y1 = function(x)
         y2 = y1 + de_mean * std

         plt.plot(x, y1, 'r.', x, y2, 'k.')
```

```
Out[58]: [
```



1.2 2. Define functions

res_x, res_y : error plot
matrixA : A matrix ($p = 0, 1, \dots, 19$)
getResidual : the sum of residual depends on p

```
In [69]: res_x = range(0,20)
         res_y = np.zeros((20))

         def matrixA(p,x):
             res = np.zeros((p+1,num))

             for i in range(p):
                 res[i] = x**(p-i)

             res[p] = 1

             return np.matrix(np.transpose(res))

         def getResidual(f_hat,y):
             return sum((f_hat - y)**2)
```

1.3 3. Execute Approxcimation

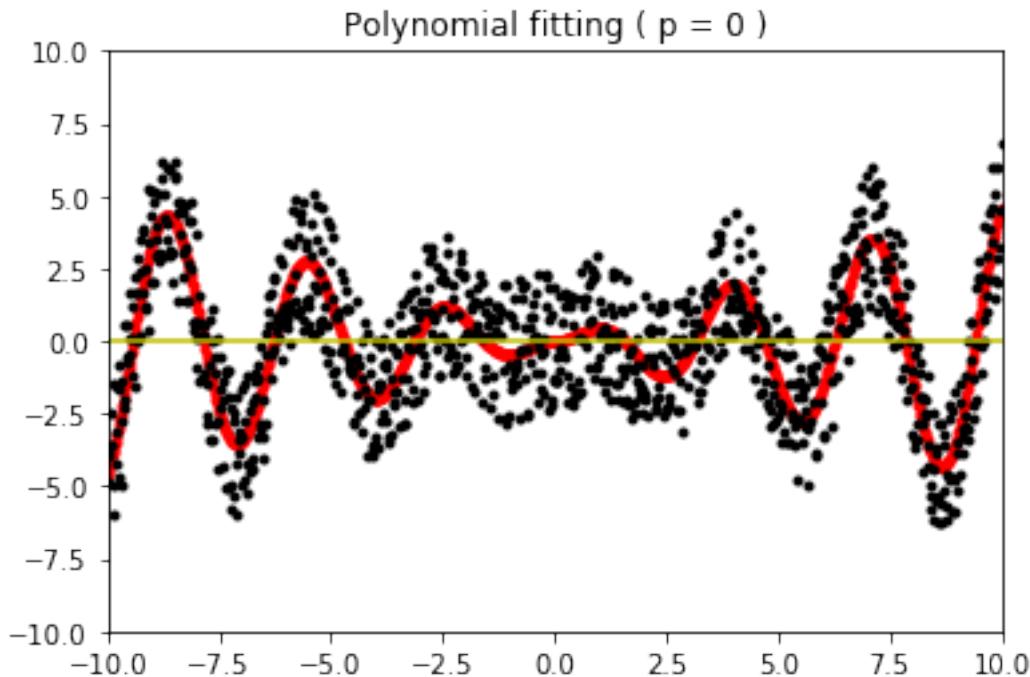
```
In [74]: for i in range(20):
    A = matrixA(i,x)
    B = np.matrix(y1)
    theta = (A.T * A).I * A.T * B.T
    theta = np.asarray(theta)

    print(theta.shape)
    print(theta)

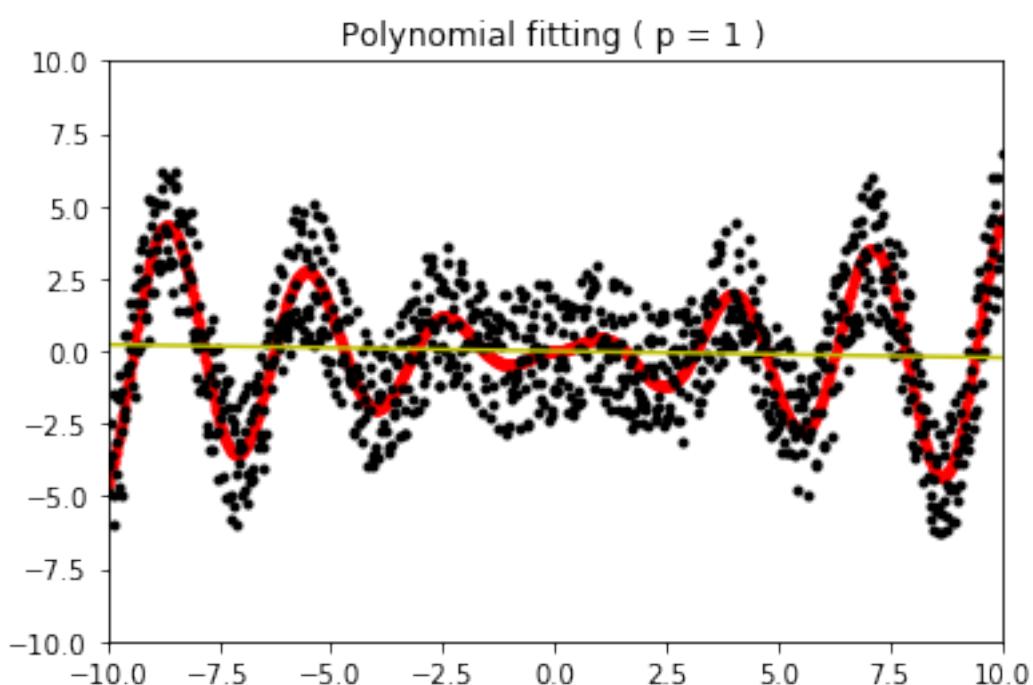
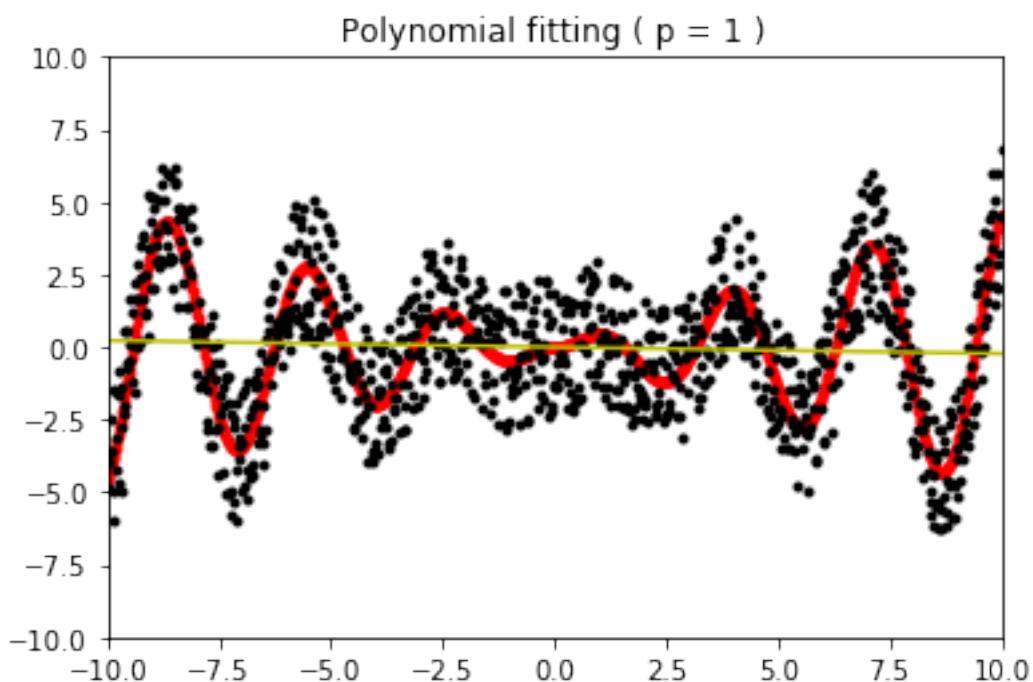
    approx = 0

    for j in range(len(theta)):
        approx += theta[j][0] * x**(i-j)
        res_y[i] = getResidual(approx,y1)
        plt.plot(x, y1, 'r.', x, y2, 'k.', x, approx, 'y')
        plt.axis([-10,10,-10,10])
    plt.title("Polynomial fitting ( p = "+str(i)+" )")
    plt.show()

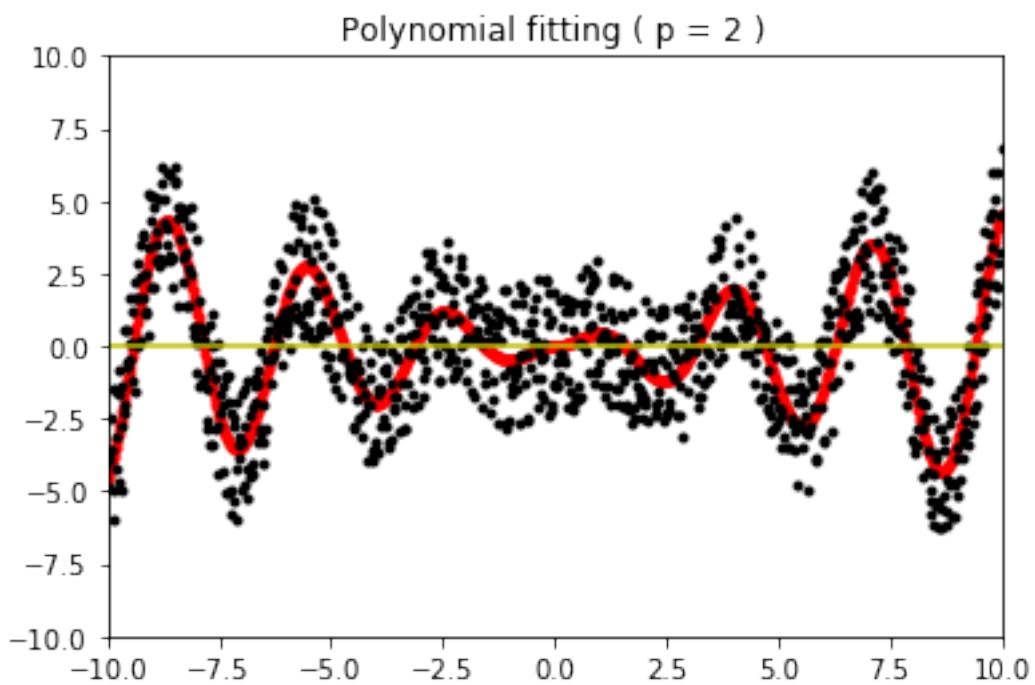
(1, 1)
[[3.06178694e-16]]
```

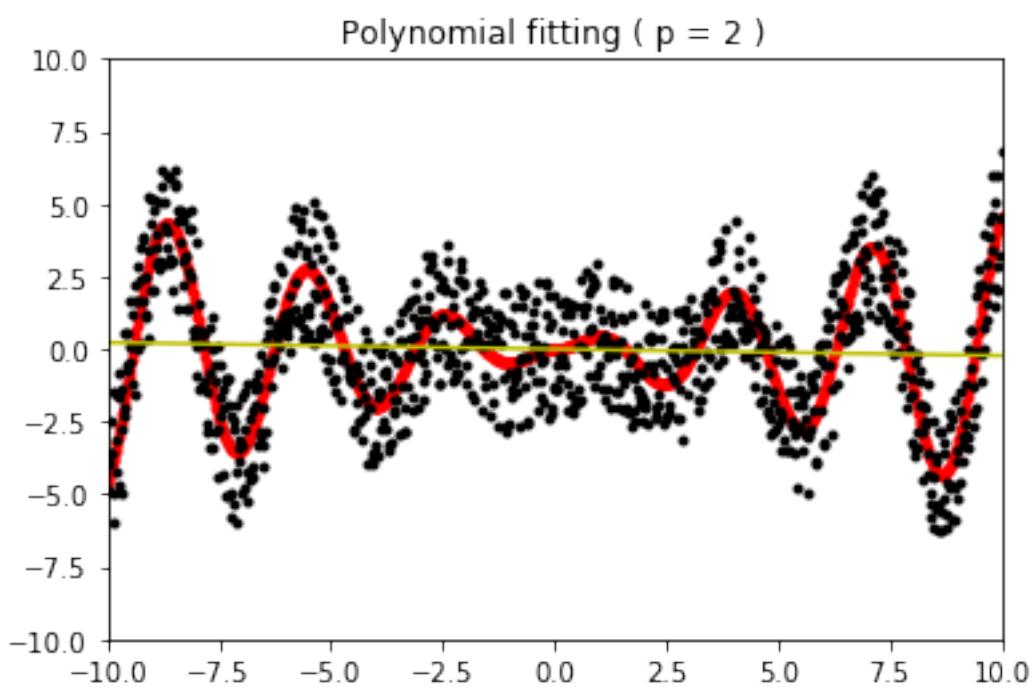
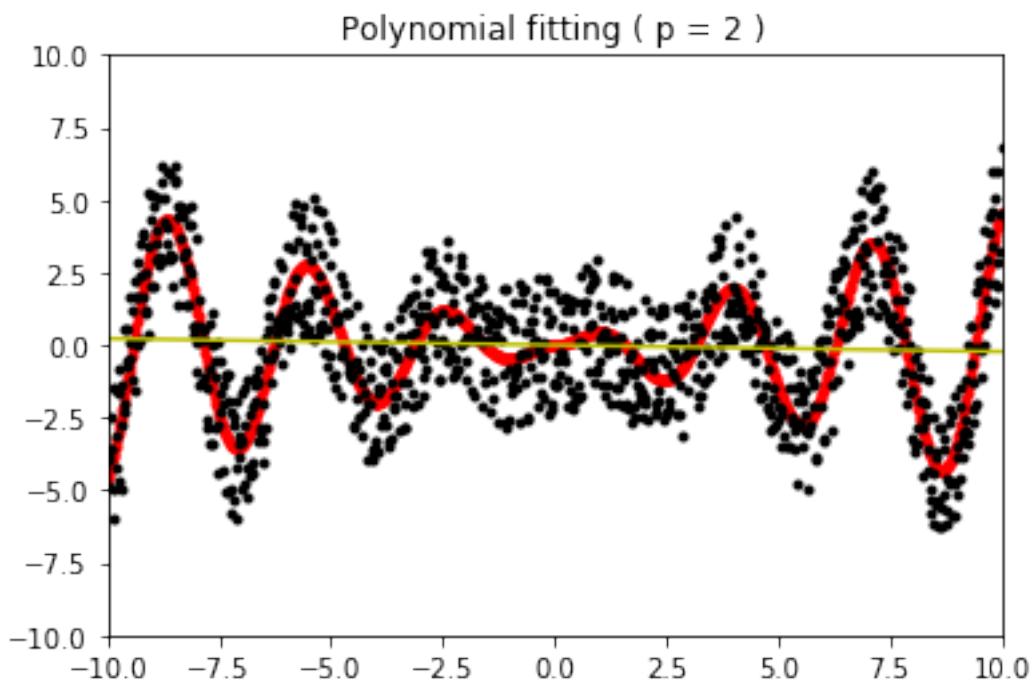


(2, 1)
[-2.25375165e-02]
[2.35922393e-16]]



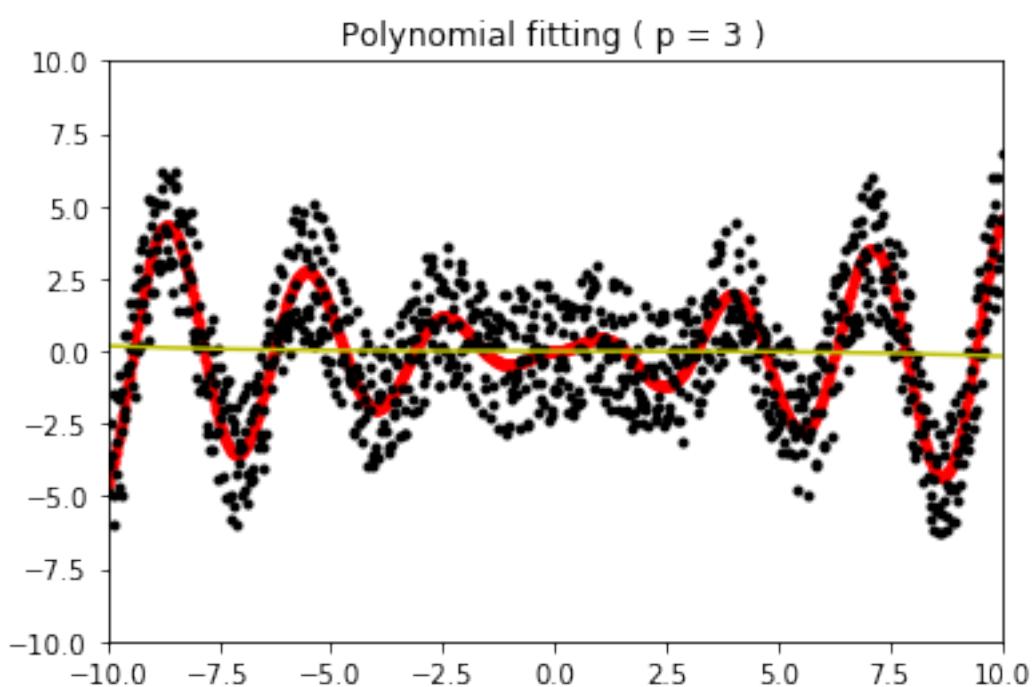
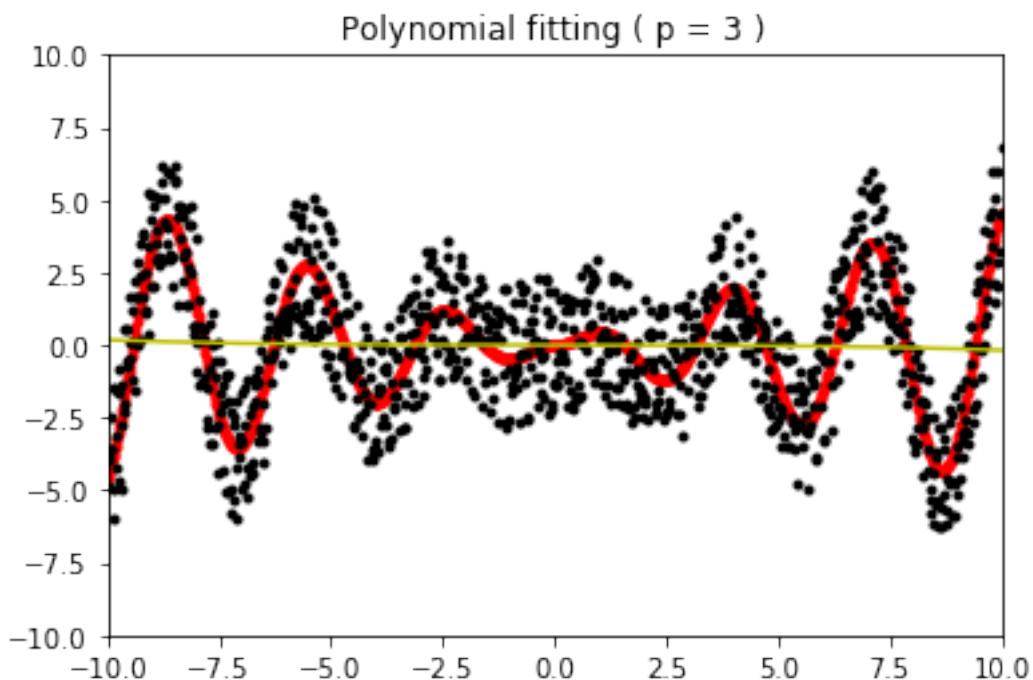
```
(3, 1)
[[-3.46944695e-18]
 [-2.25375165e-02]
 [ 3.11382864e-16]]
```

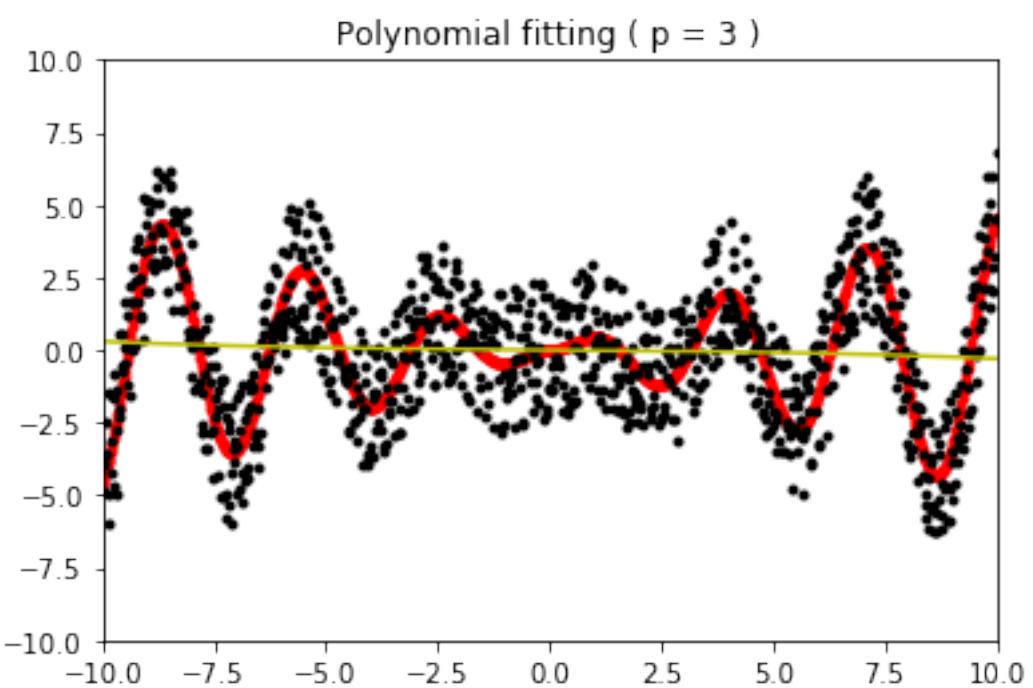
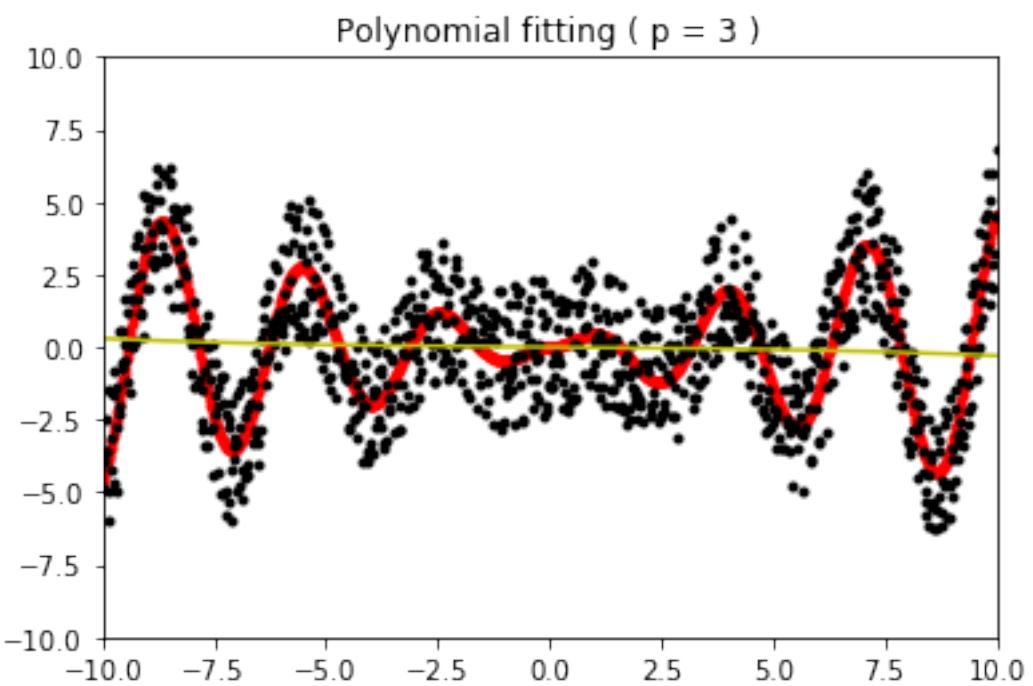




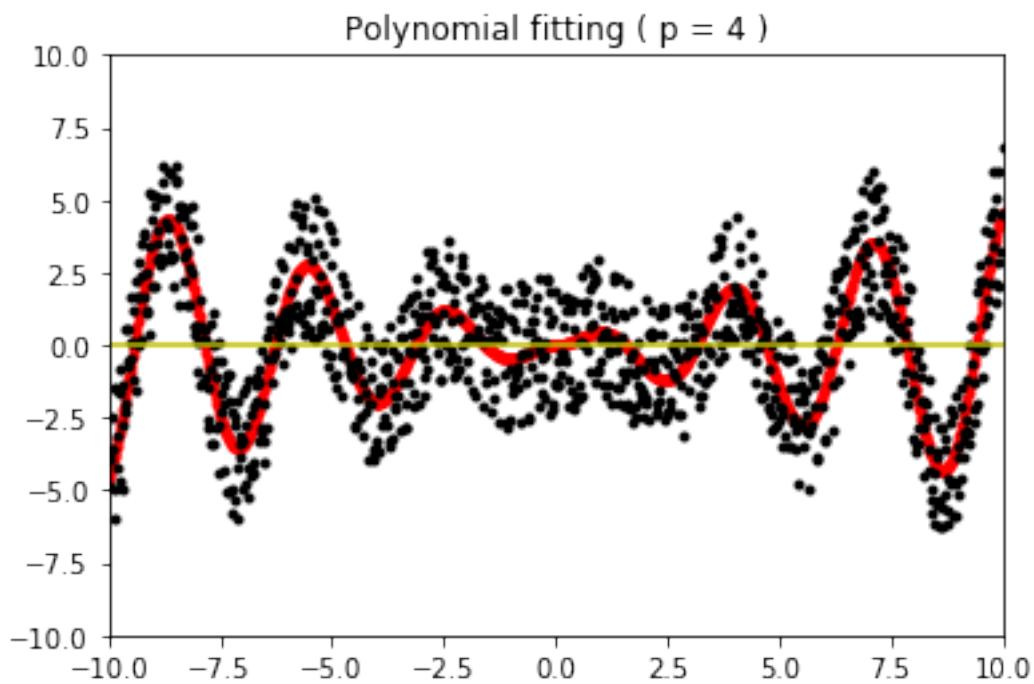
(4, 1)
[[-1.77482196e-04]]

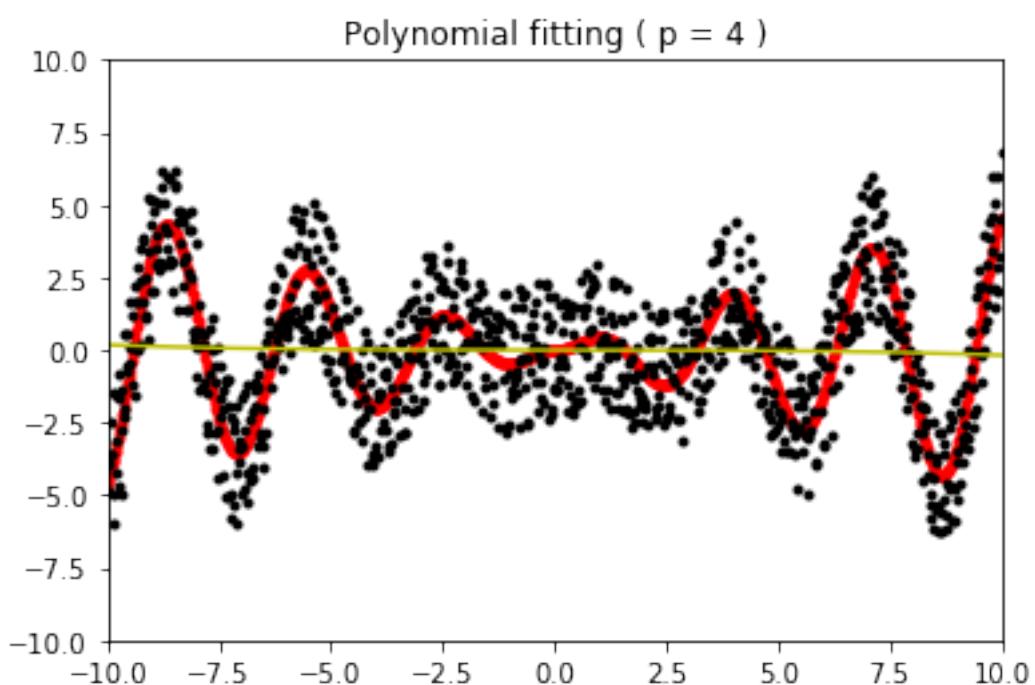
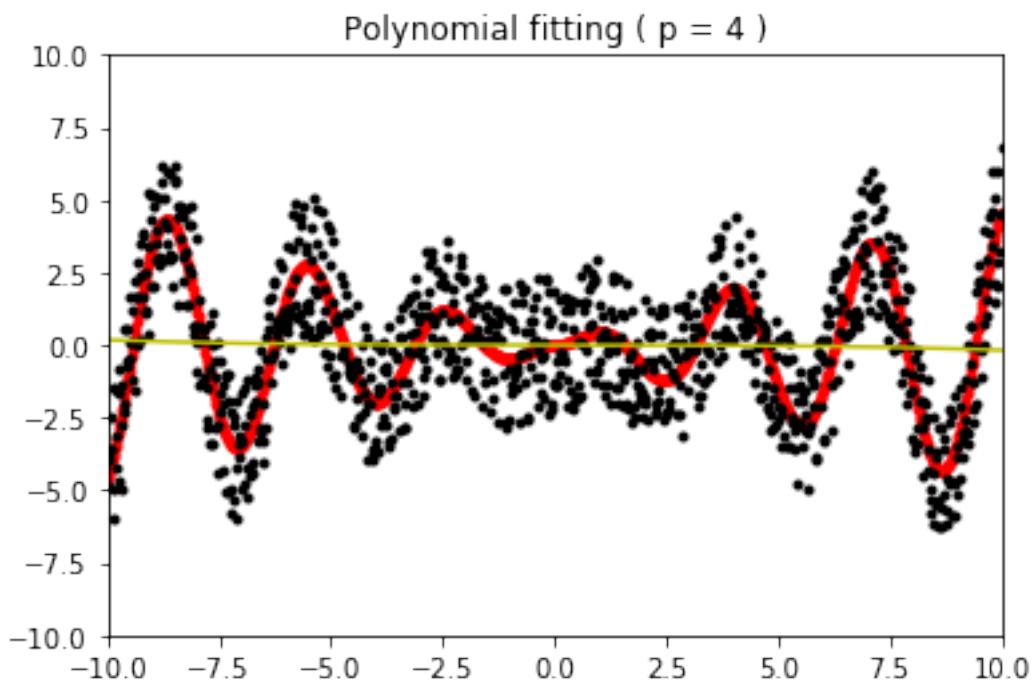
```
[ -8.13151629e-19]  
[ -1.18672797e-02]  
[ 2.82759927e-16]]
```

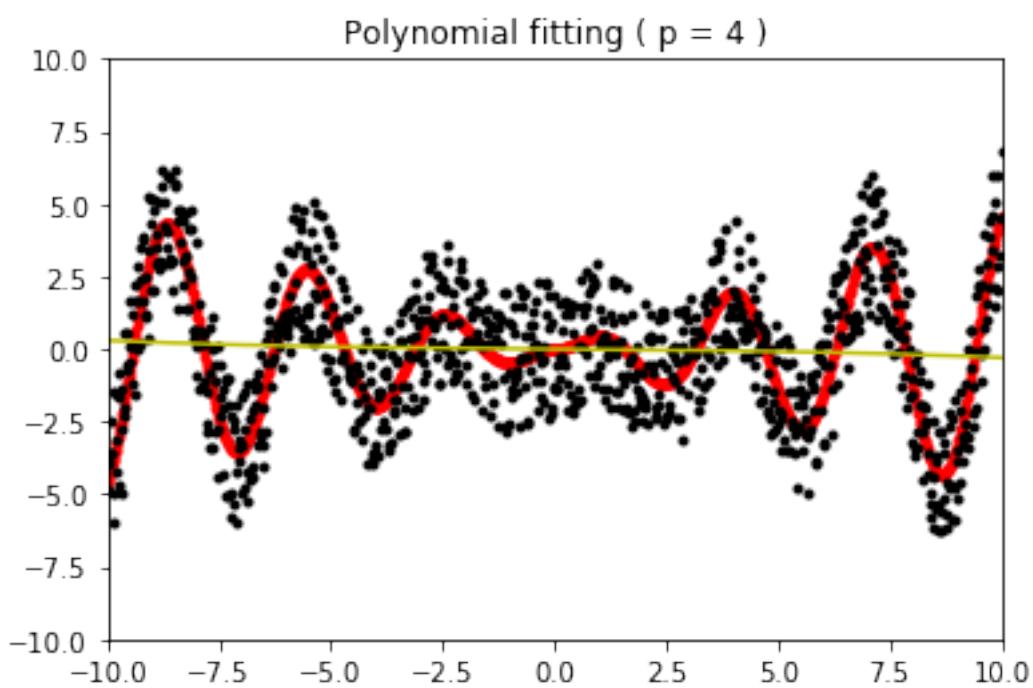
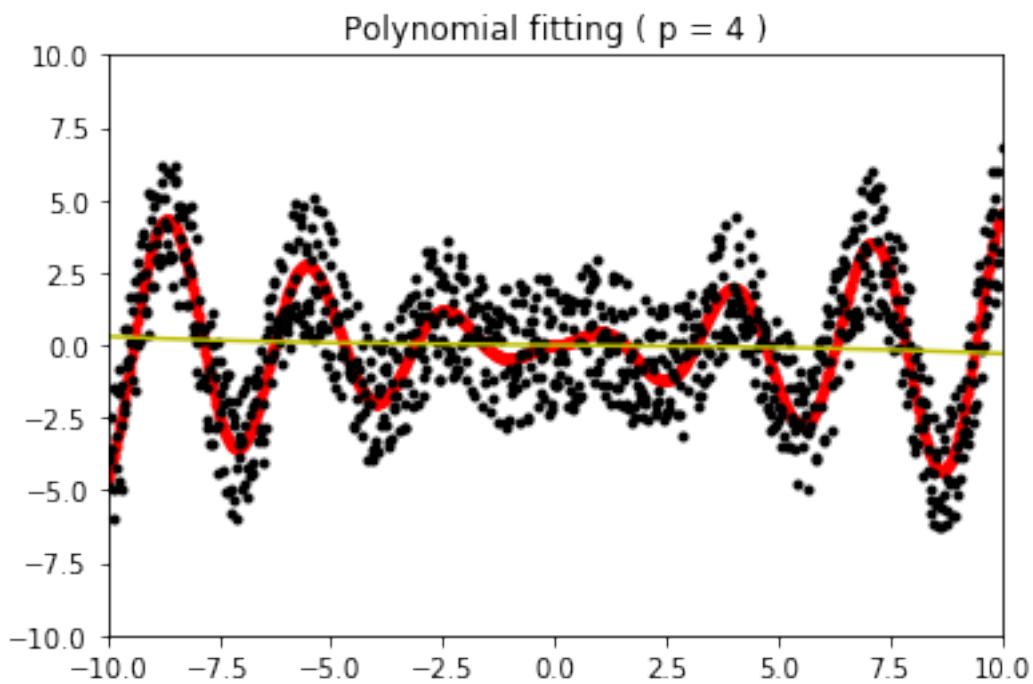




```
(5, 1)
[[-5.04831637e-19]
 [-1.77482196e-04]
 [ 3.70797143e-17]
 [-1.18672797e-02]
 [-1.19695920e-16]]
```

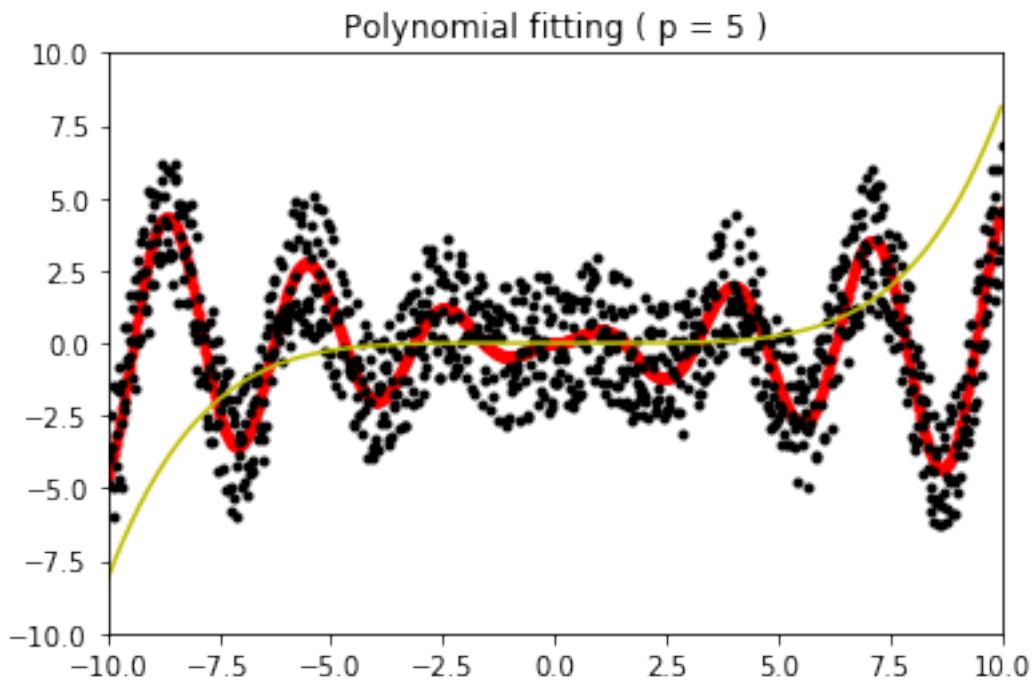


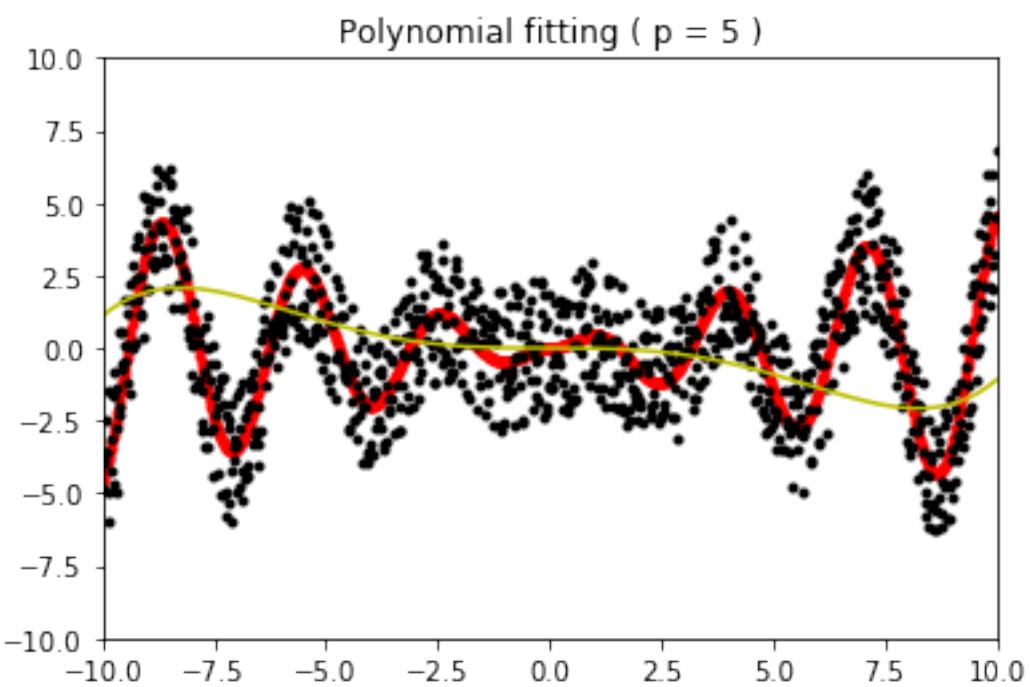
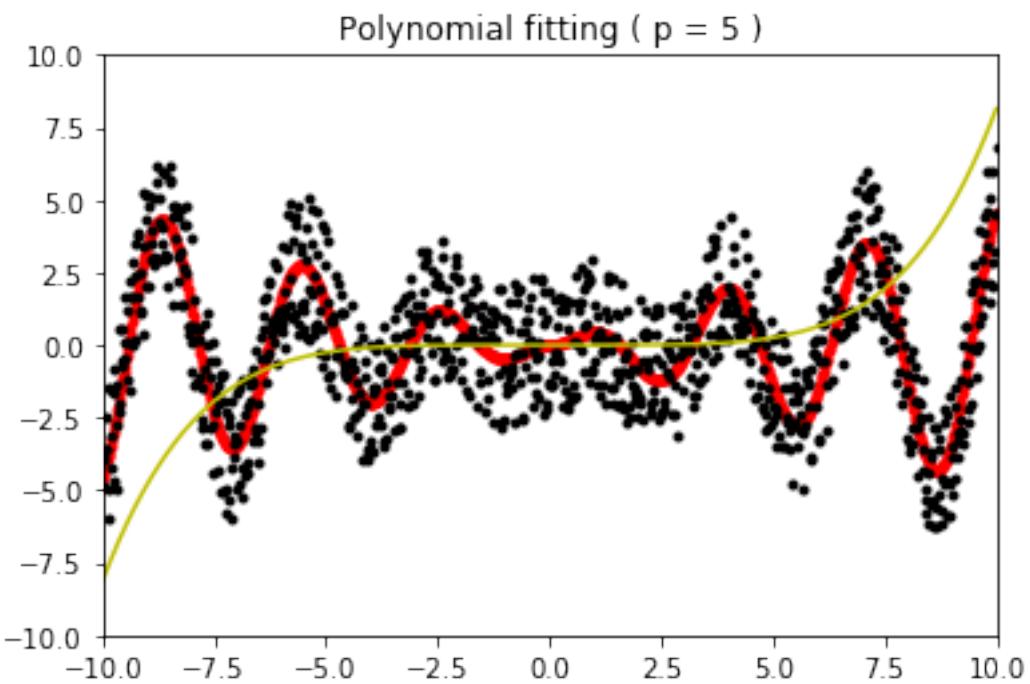


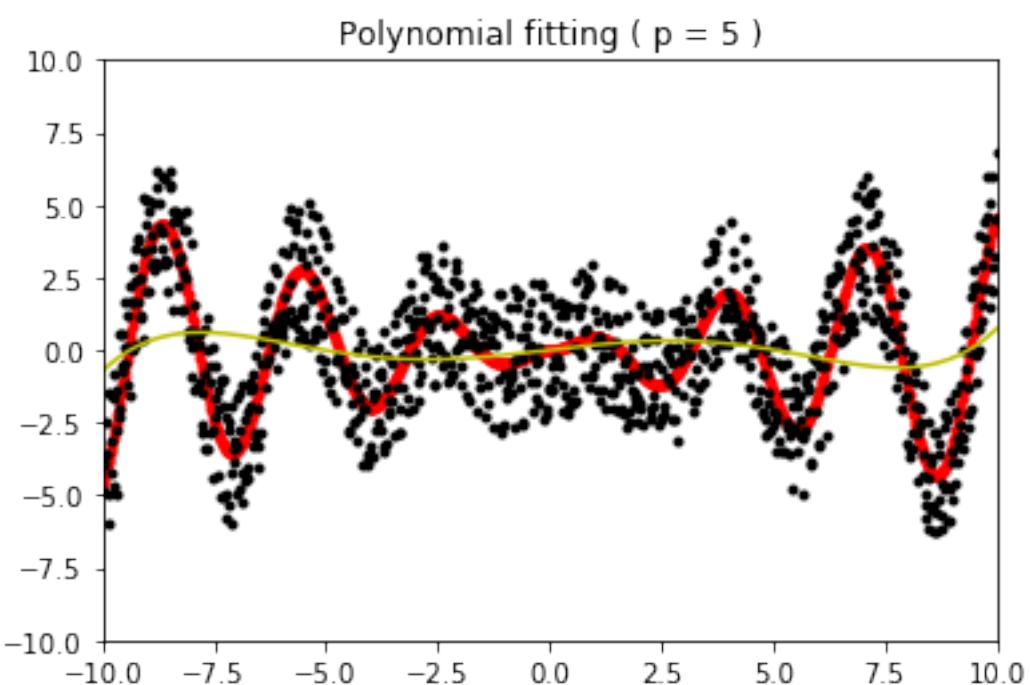
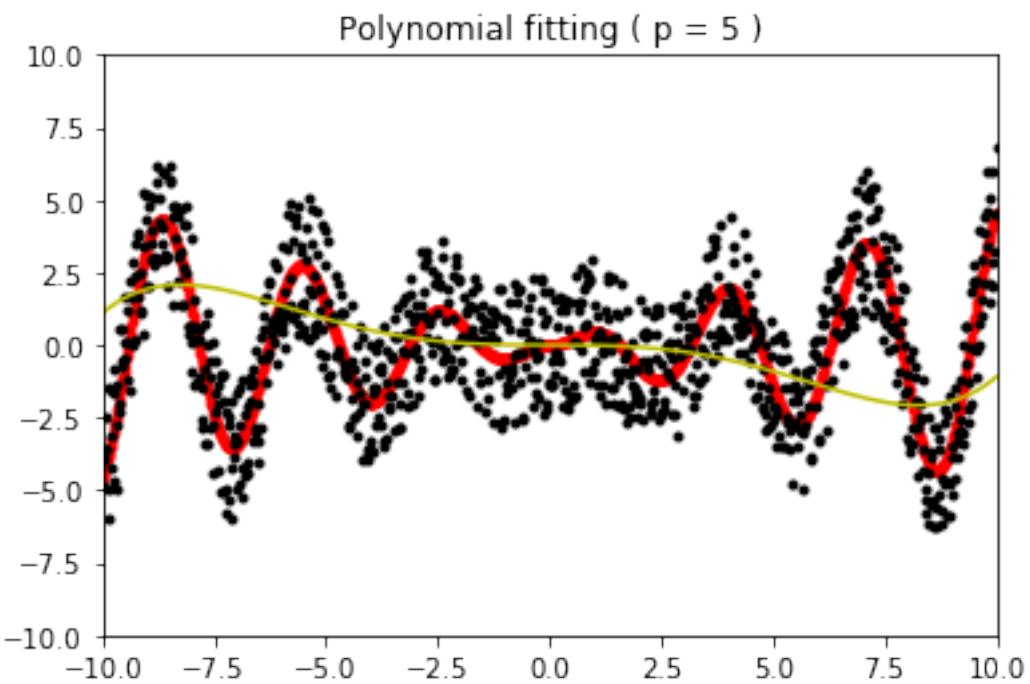


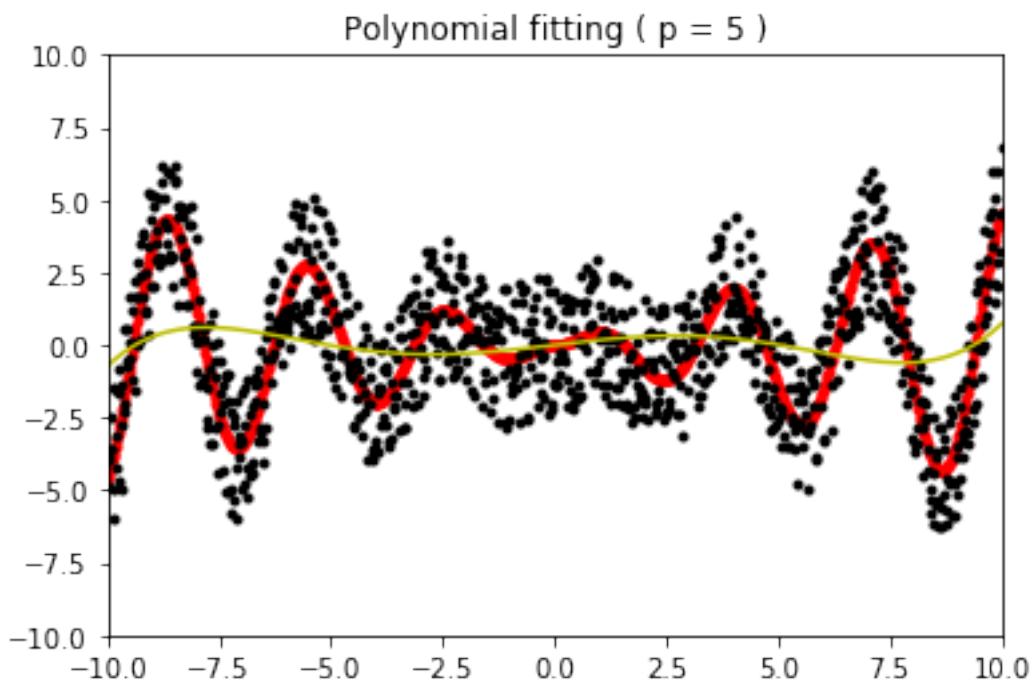
(6, 1)
[[8.13824012e-05]

```
[-1.82620303e-17]
[-9.23801994e-03]
[ 1.62153277e-15]
[ 1.82674371e-01]
[-1.71538131e-14]]
```

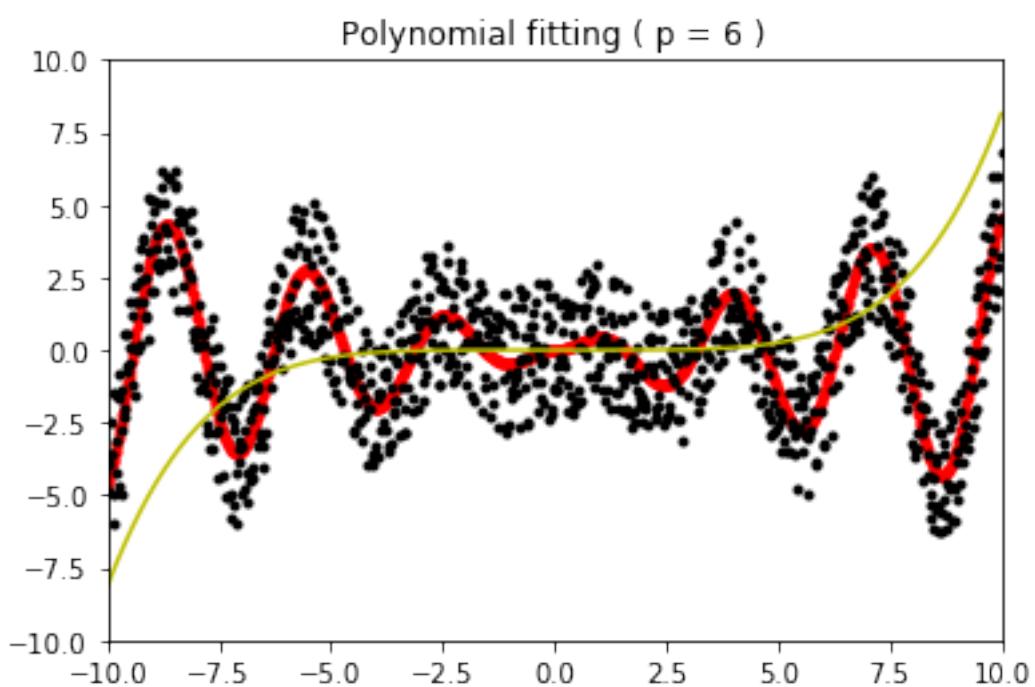
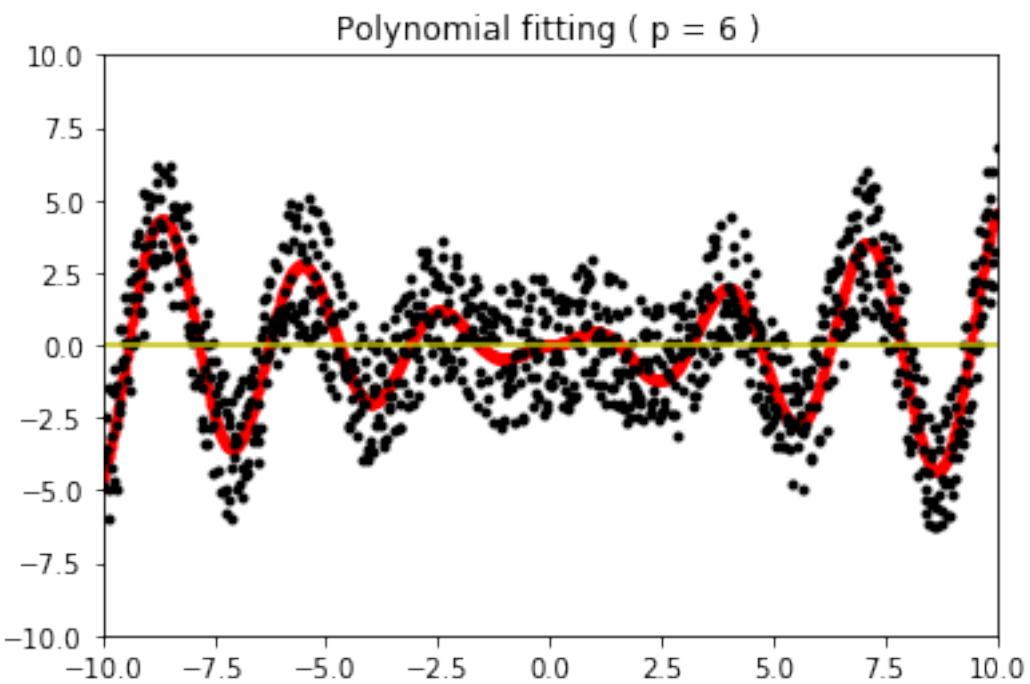


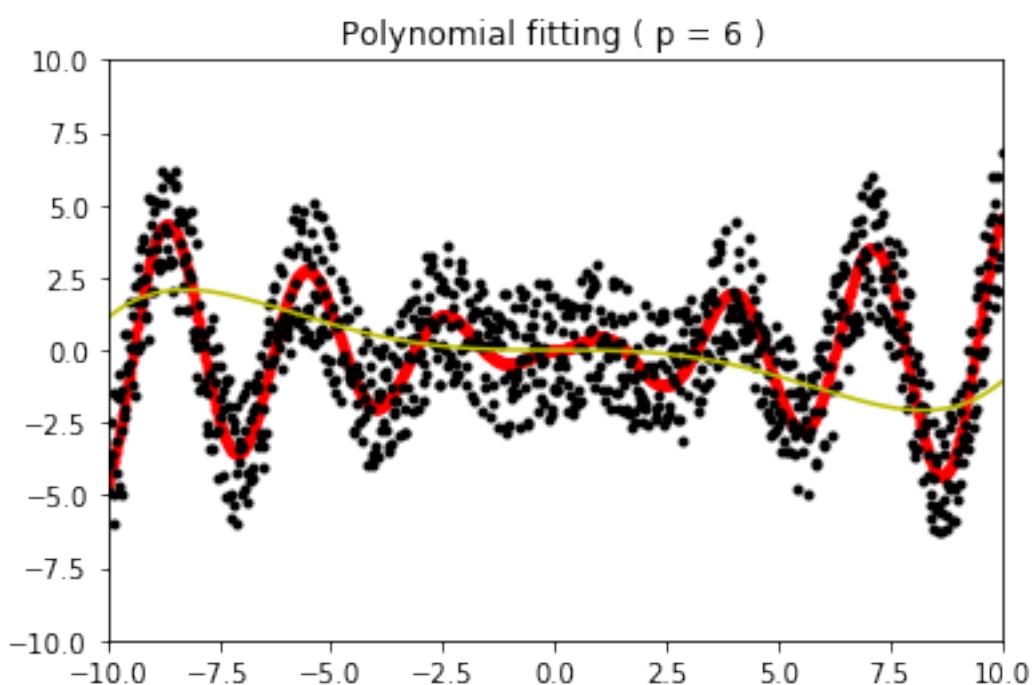
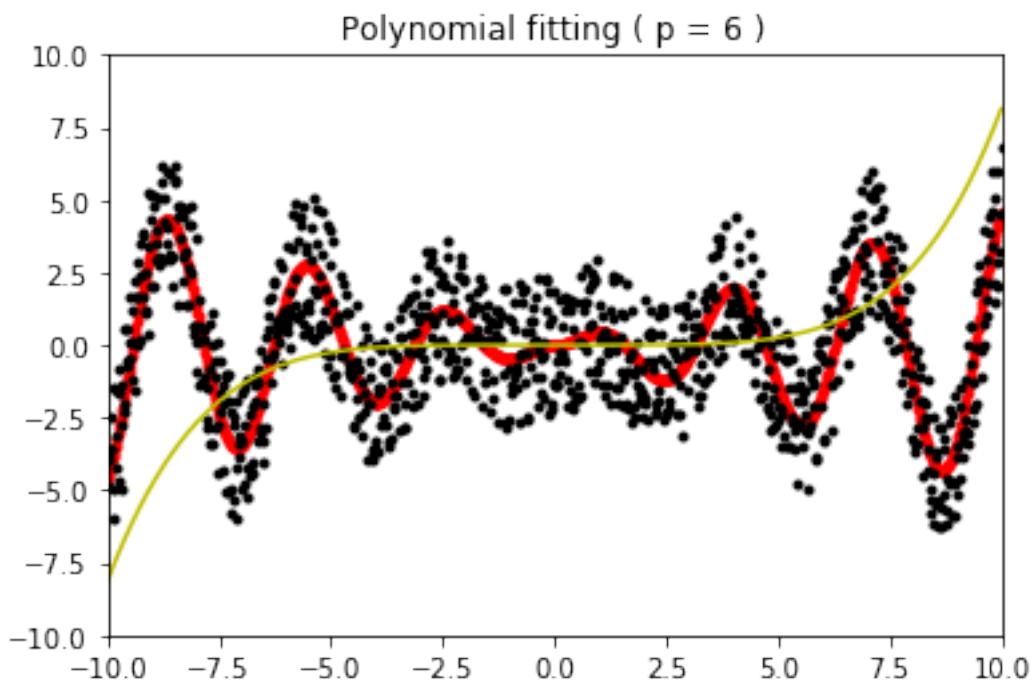


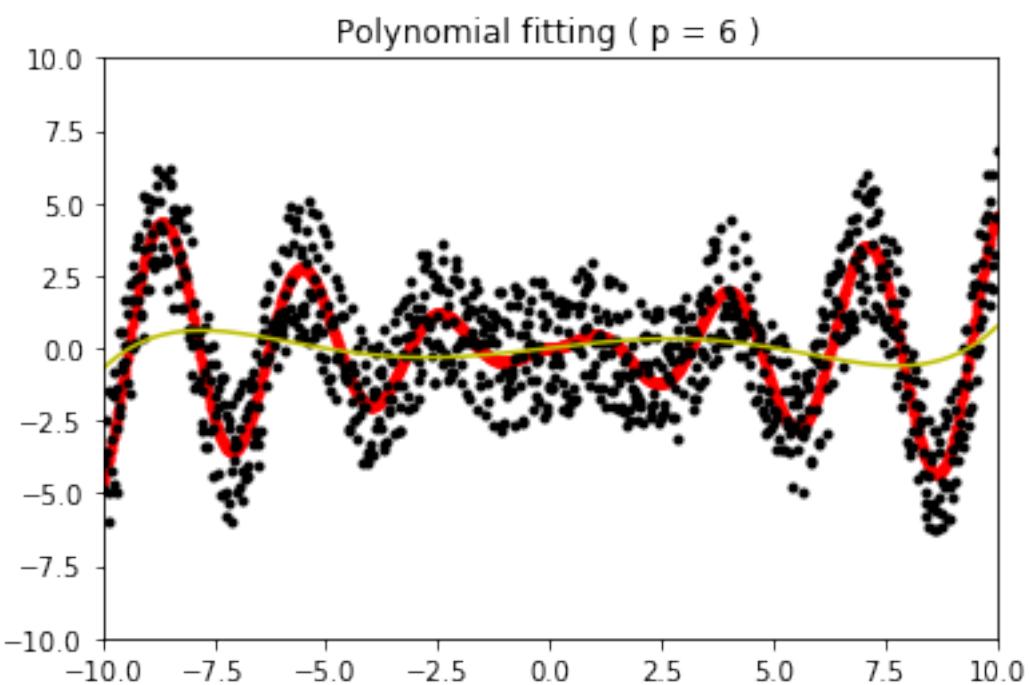
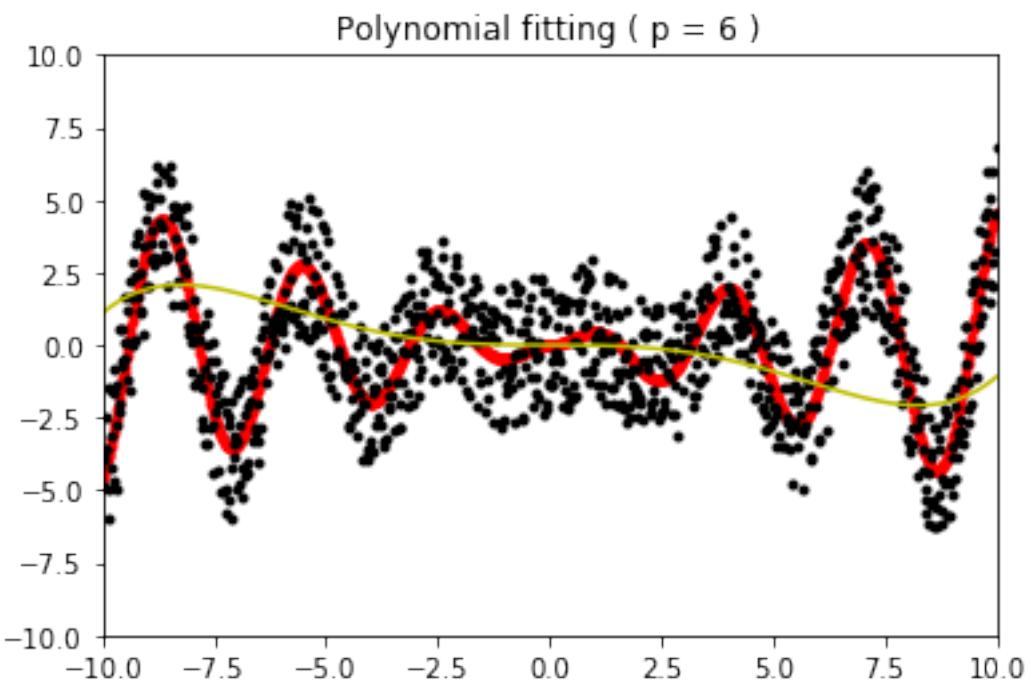


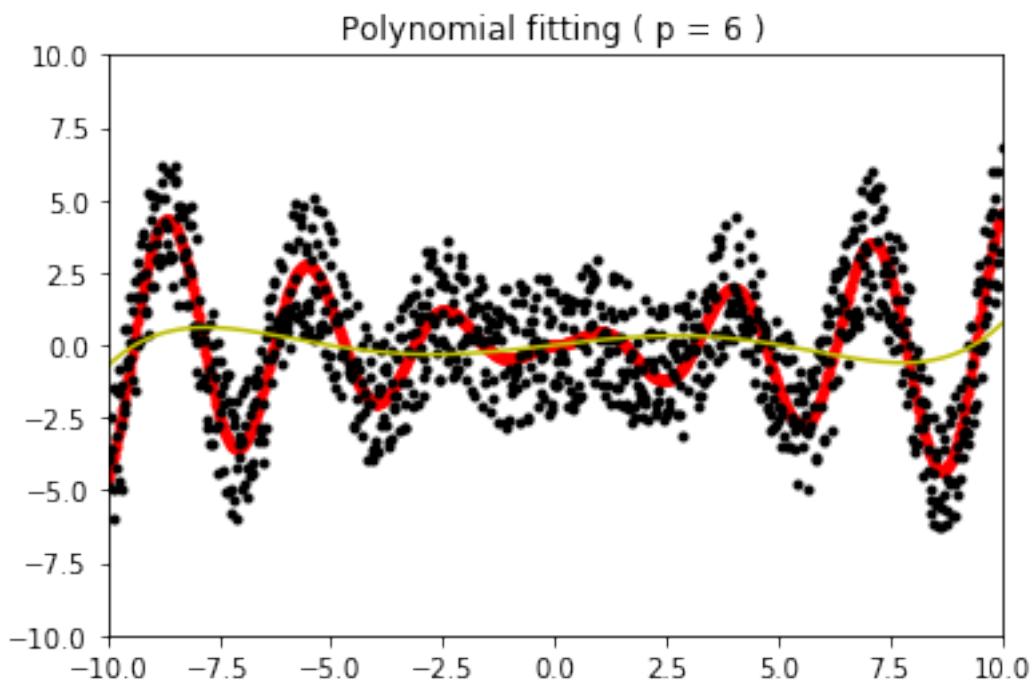


```
(7, 1)
[[ 4.16782562e-18]
 [ 8.13824012e-05]
 [-5.89094474e-16]
 [-9.23801994e-03]
 [ 2.05942034e-14]
 [ 1.82674371e-01]
 [-1.08253684e-13]]
```

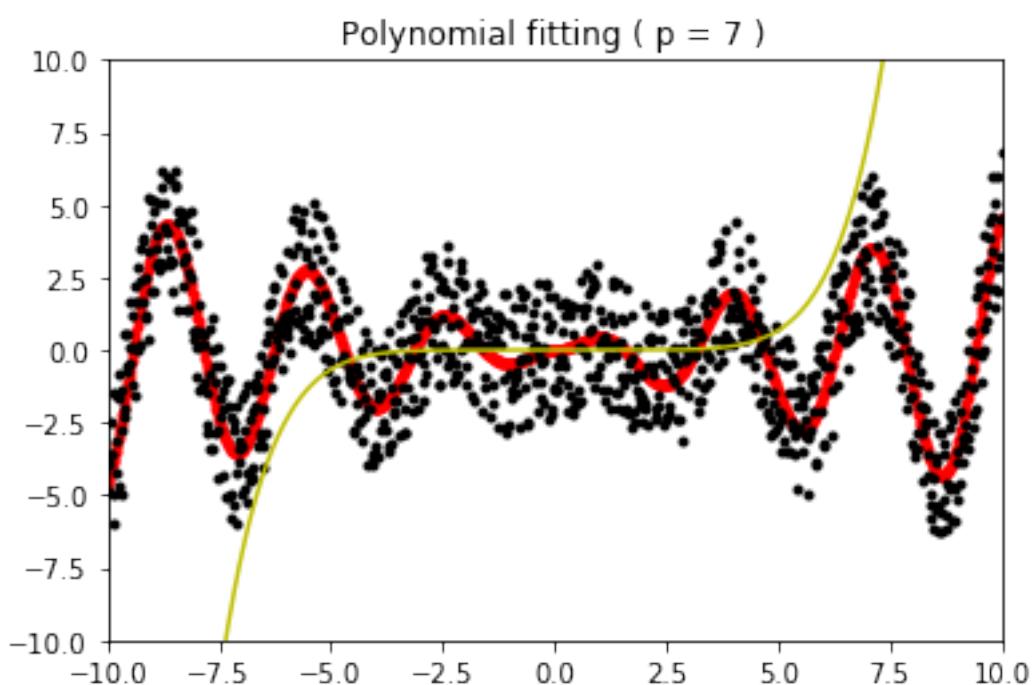
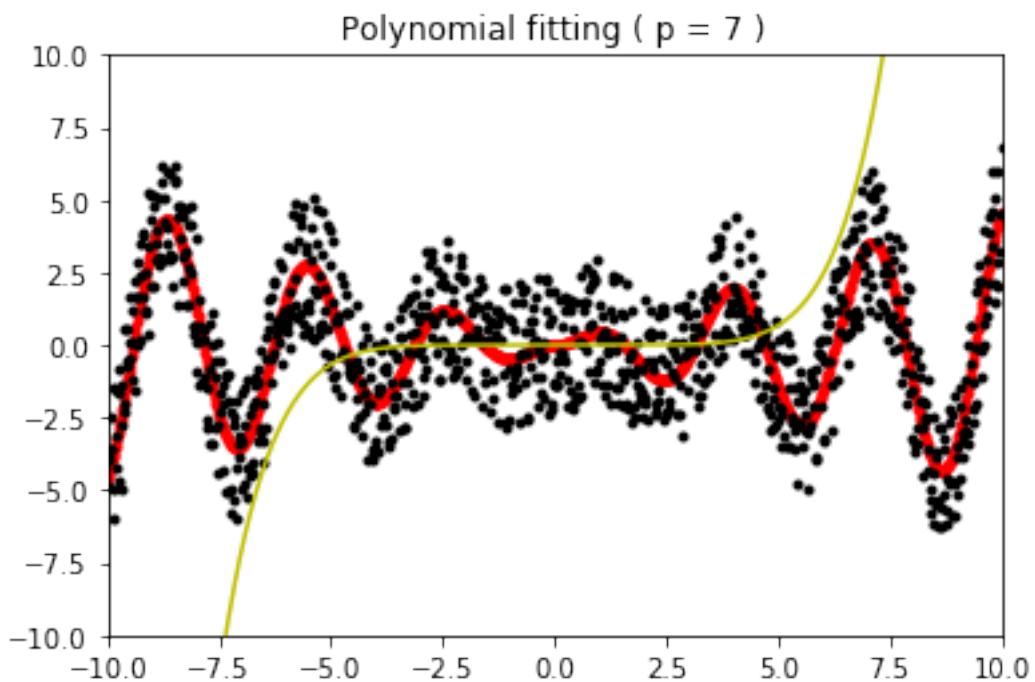


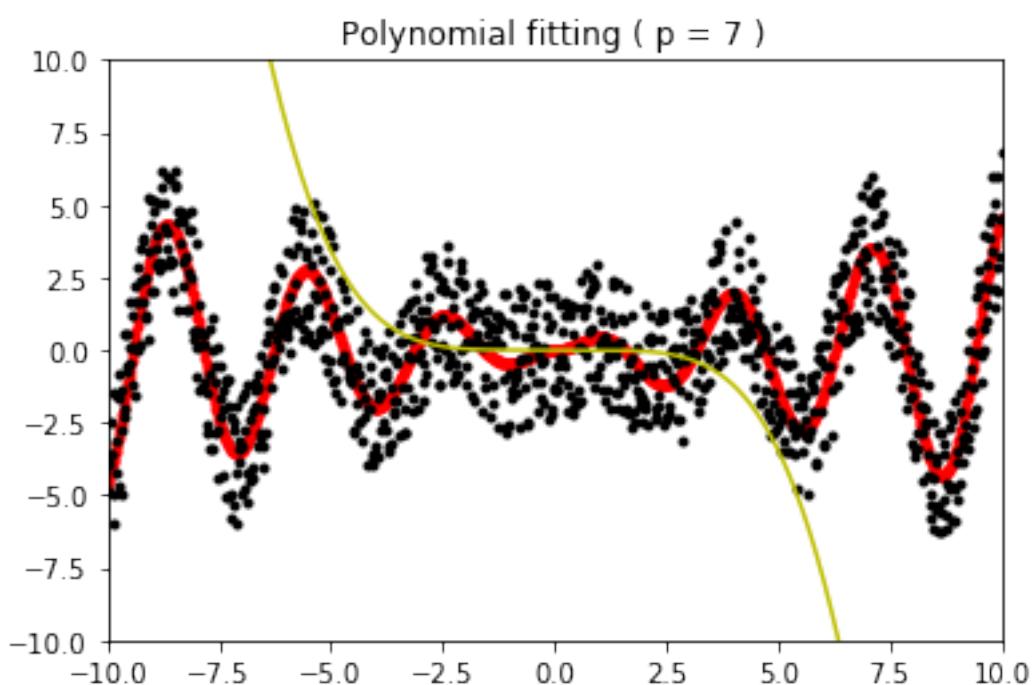
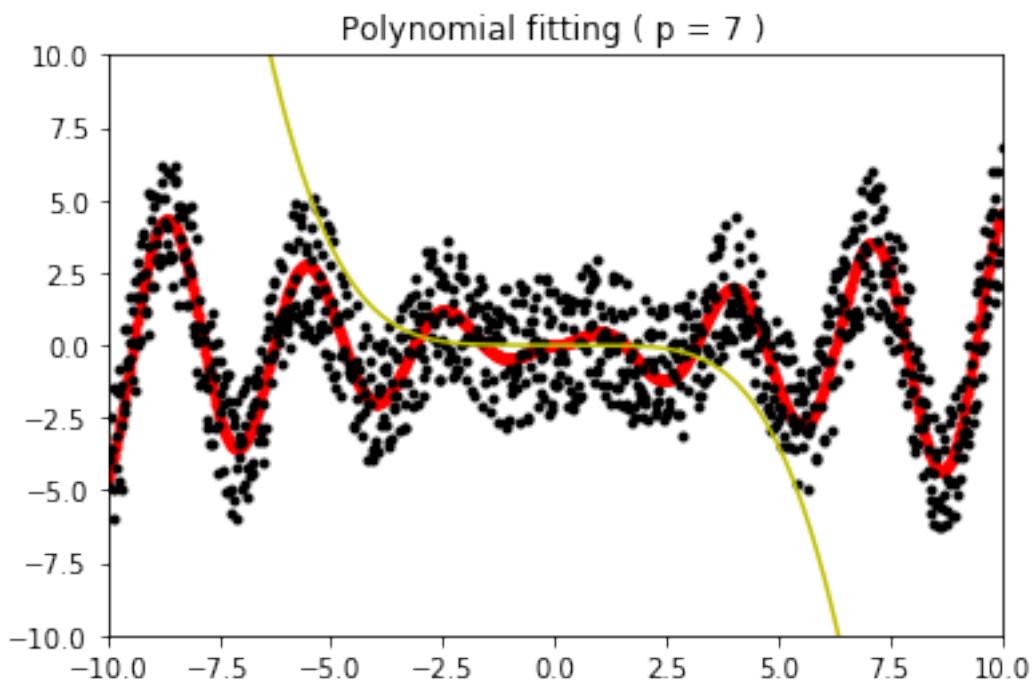


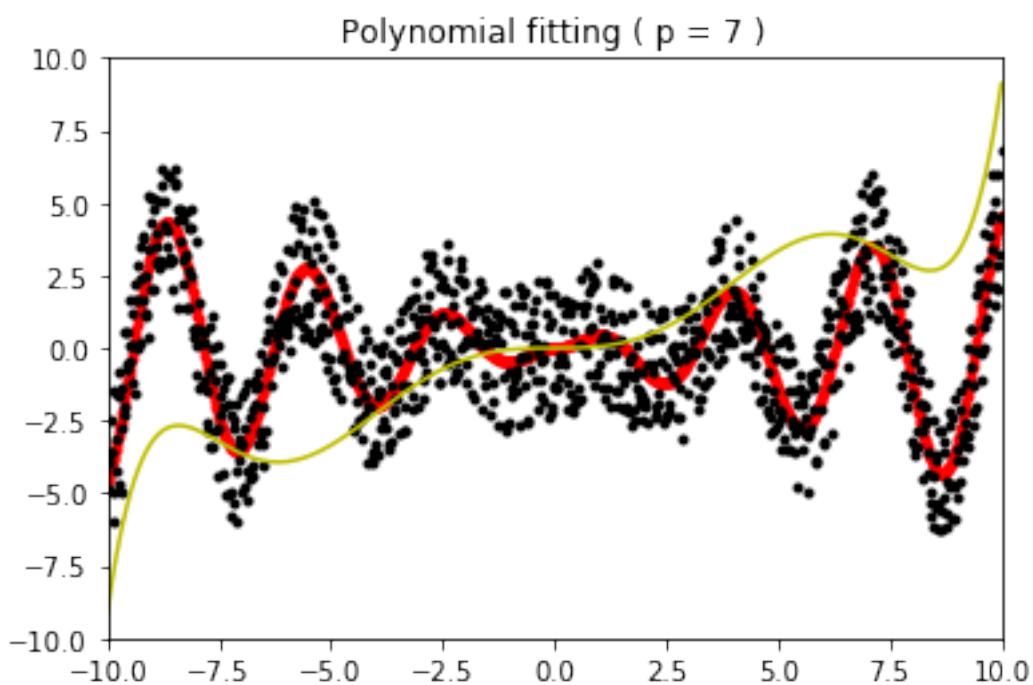
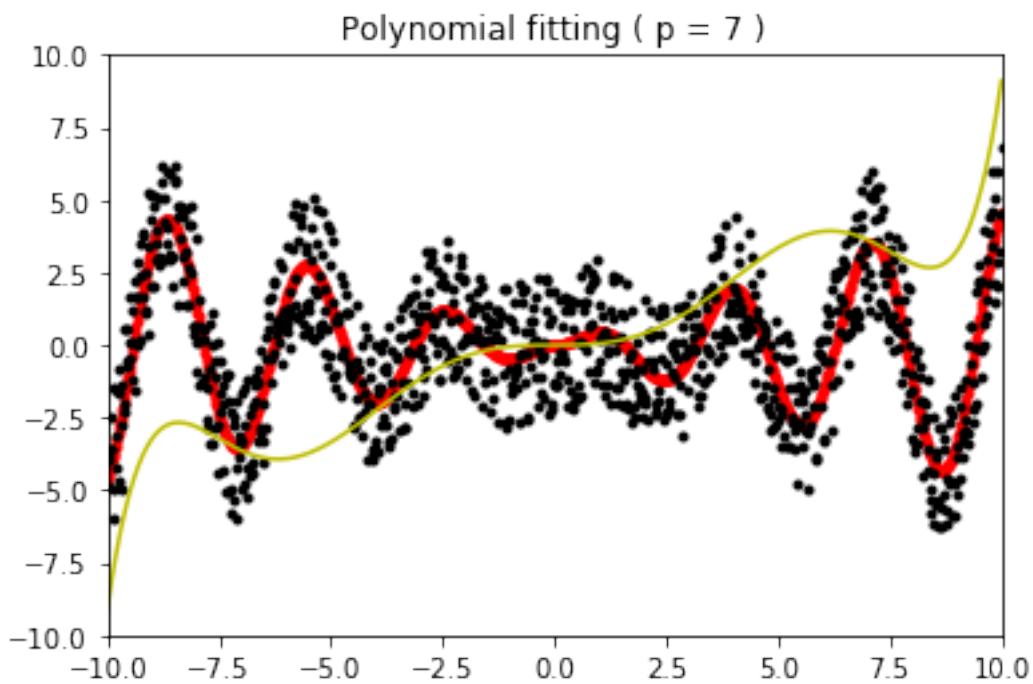


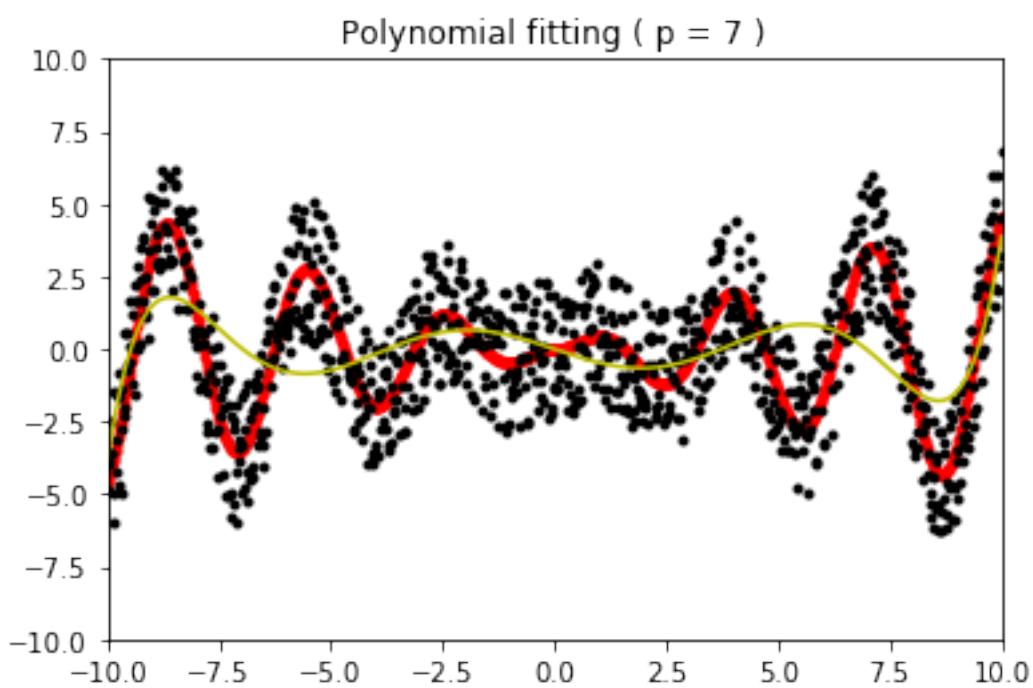
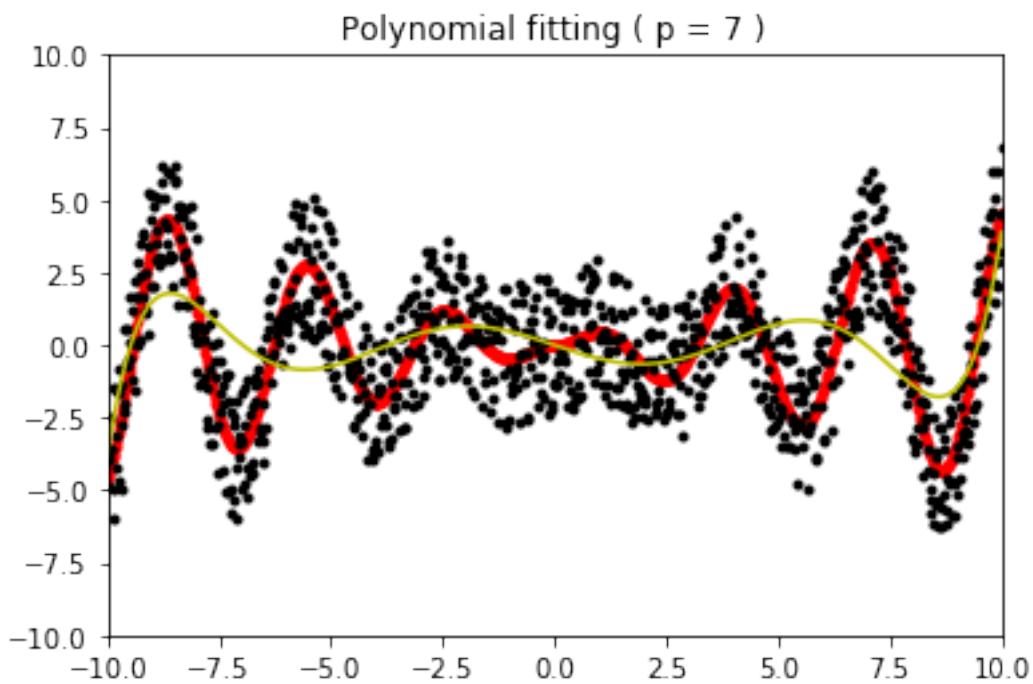


```
(8, 1)
[[ 8.60054853e-06]
 [-3.33352993e-17]
 [-1.31069984e-03]
 [ 4.78504497e-15]
 [ 5.41641951e-02]
 [-1.72747450e-13]
 [-5.23193739e-01]
 [ 9.58264718e-13]]
```



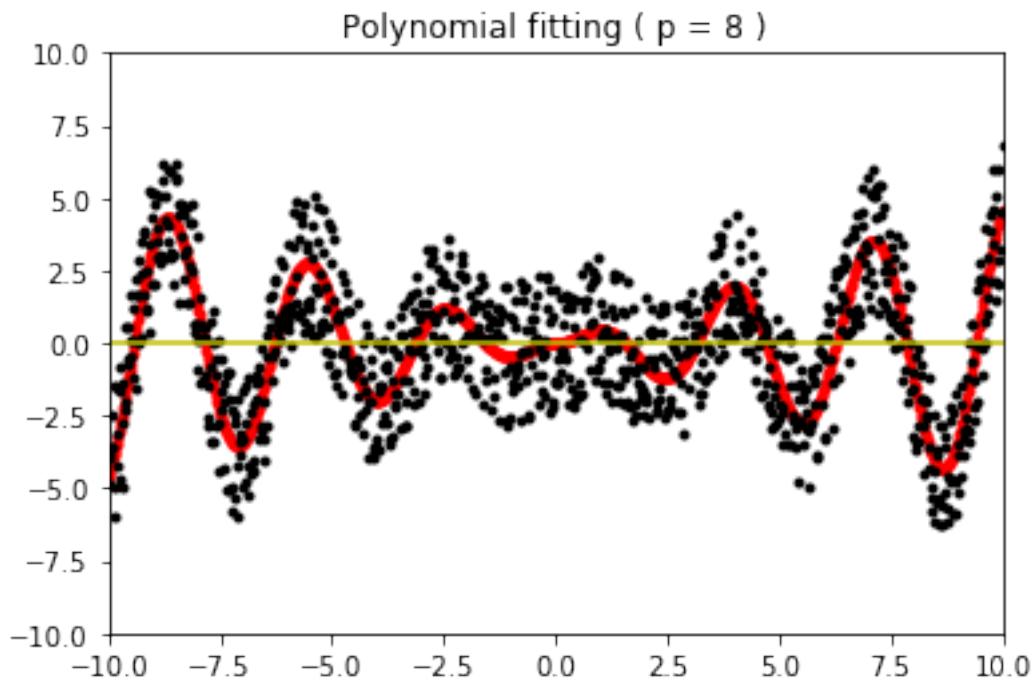


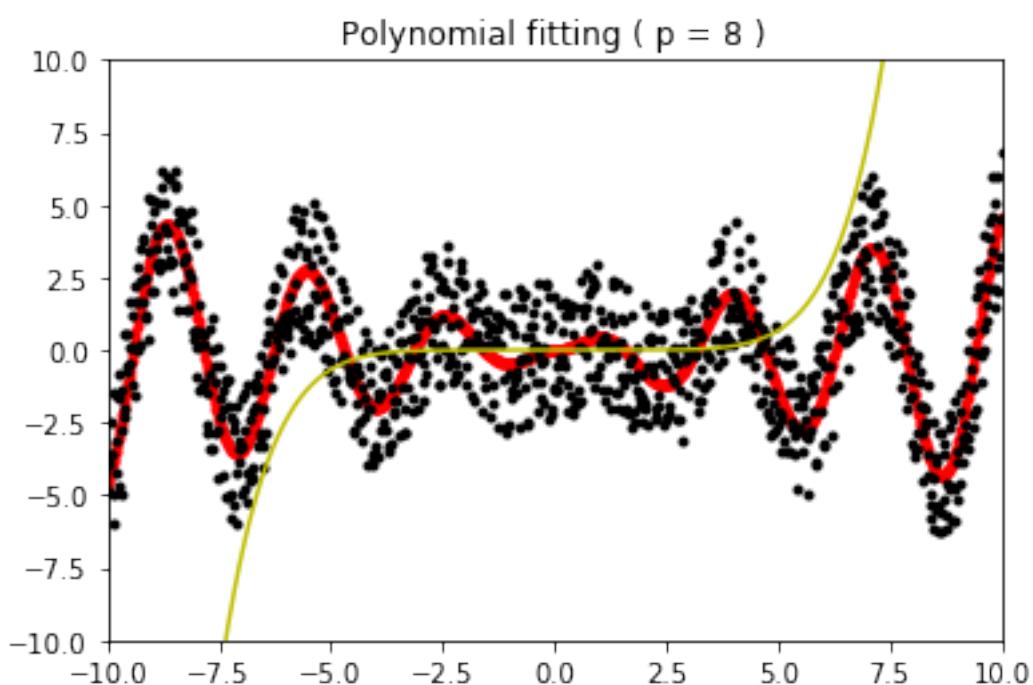
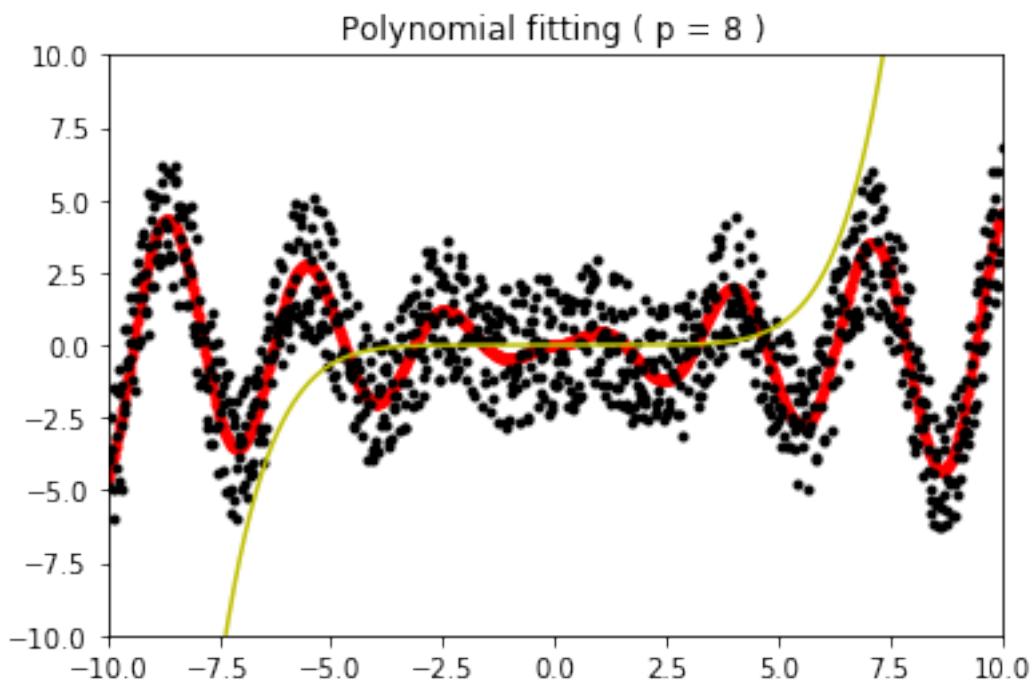


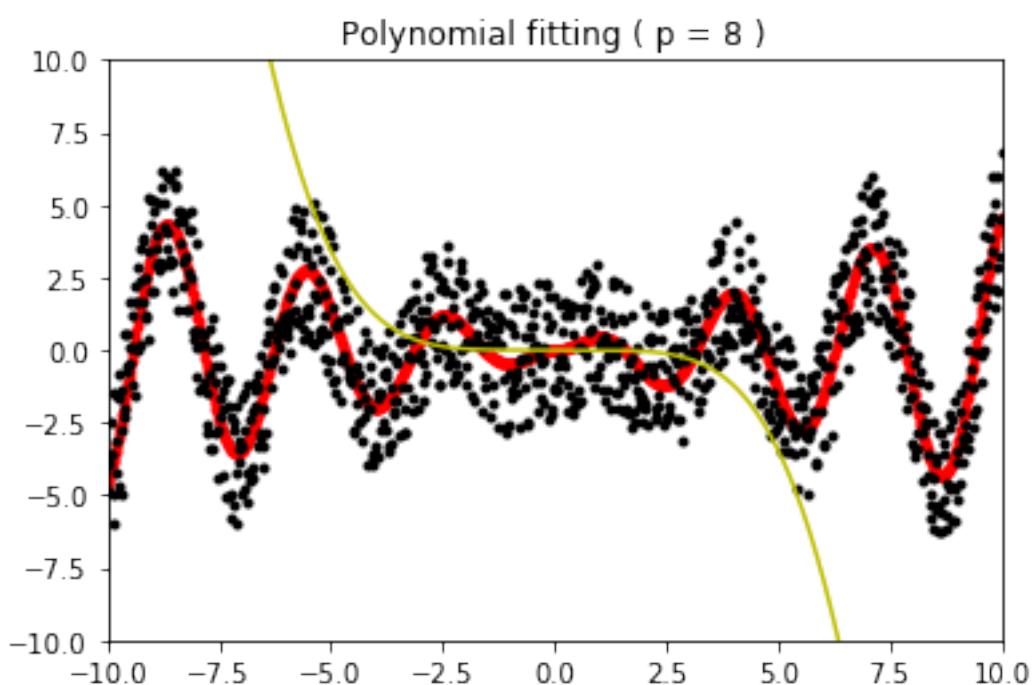
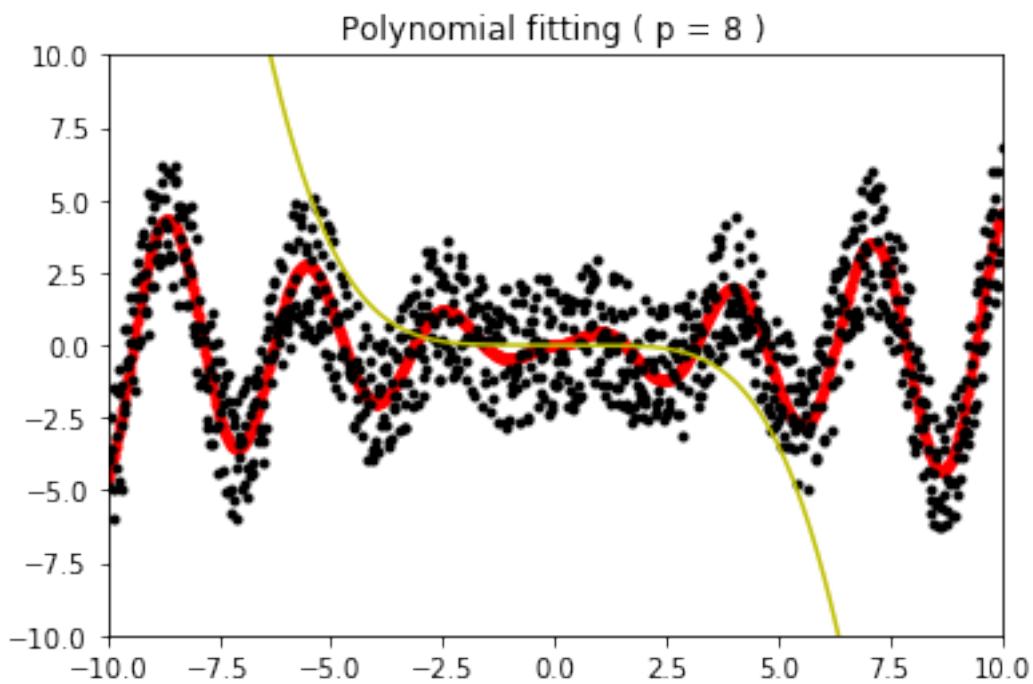


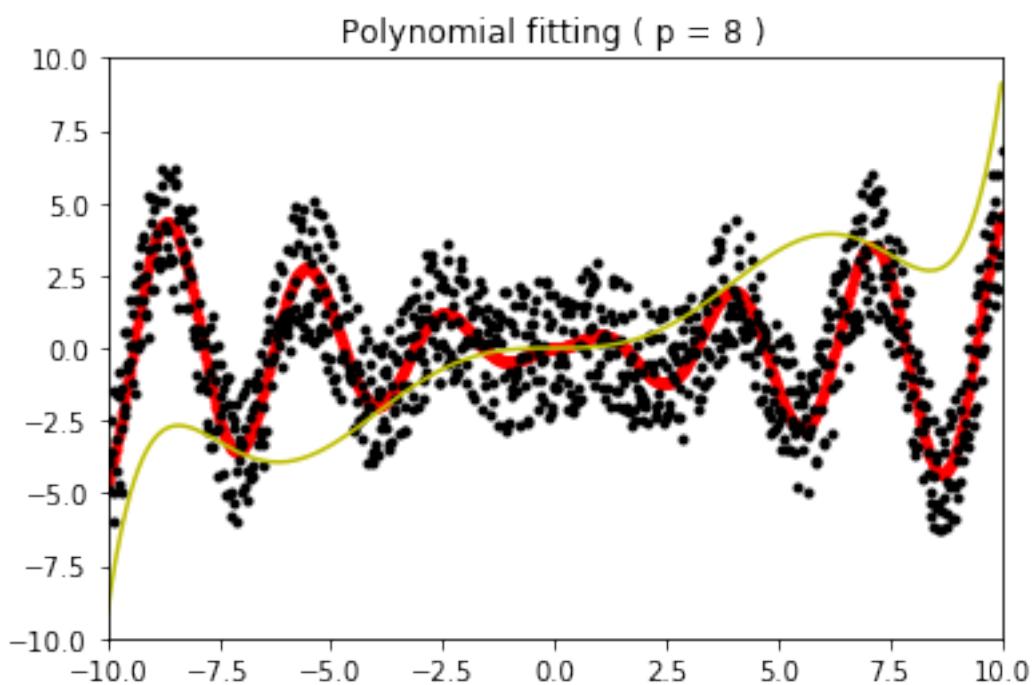
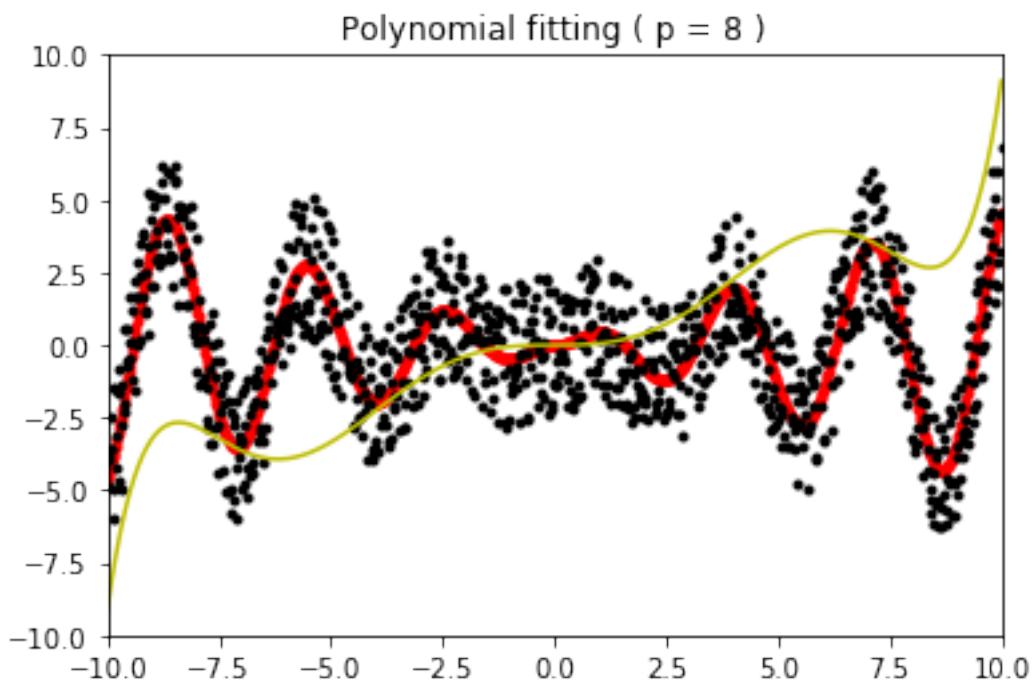
(9, 1)
[[2.50238017e-18]

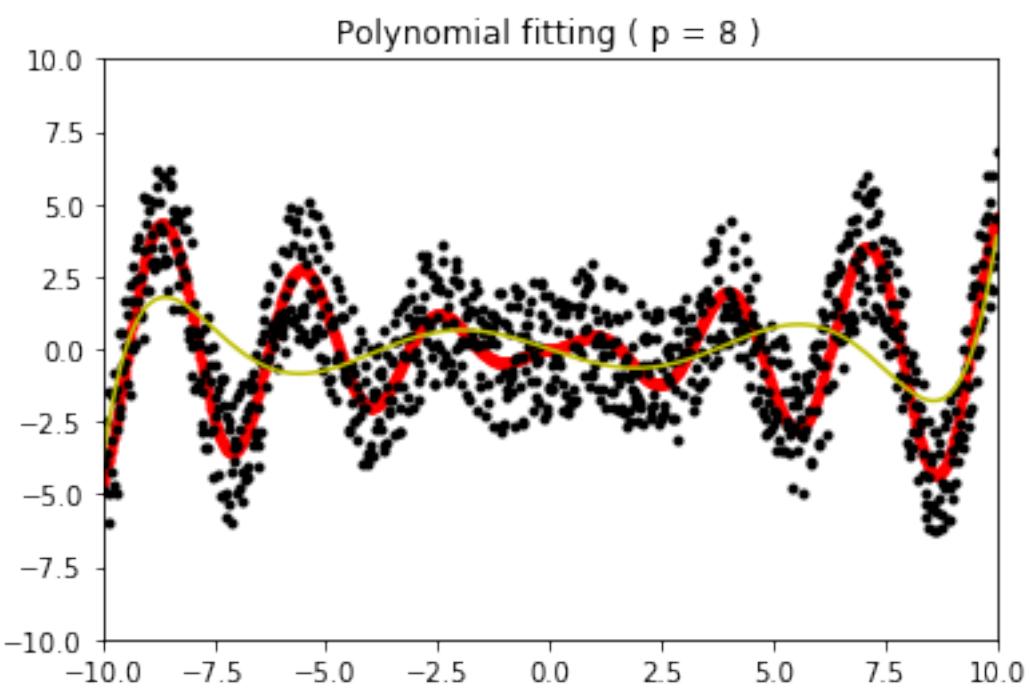
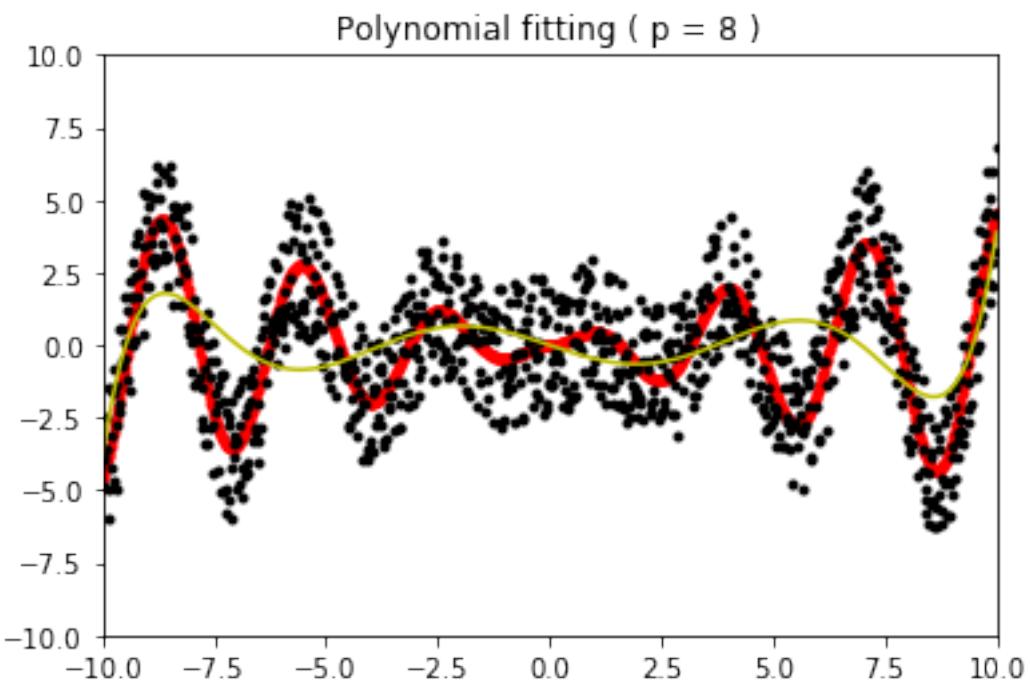
```
[ 8.60054853e-06]
[-5.01773424e-16]
[-1.31069984e-03]
[ 3.18444815e-14]
[ 5.41641951e-02]
[-6.66310323e-13]
[-5.23193739e-01]
[ 2.33219954e-12]]
```





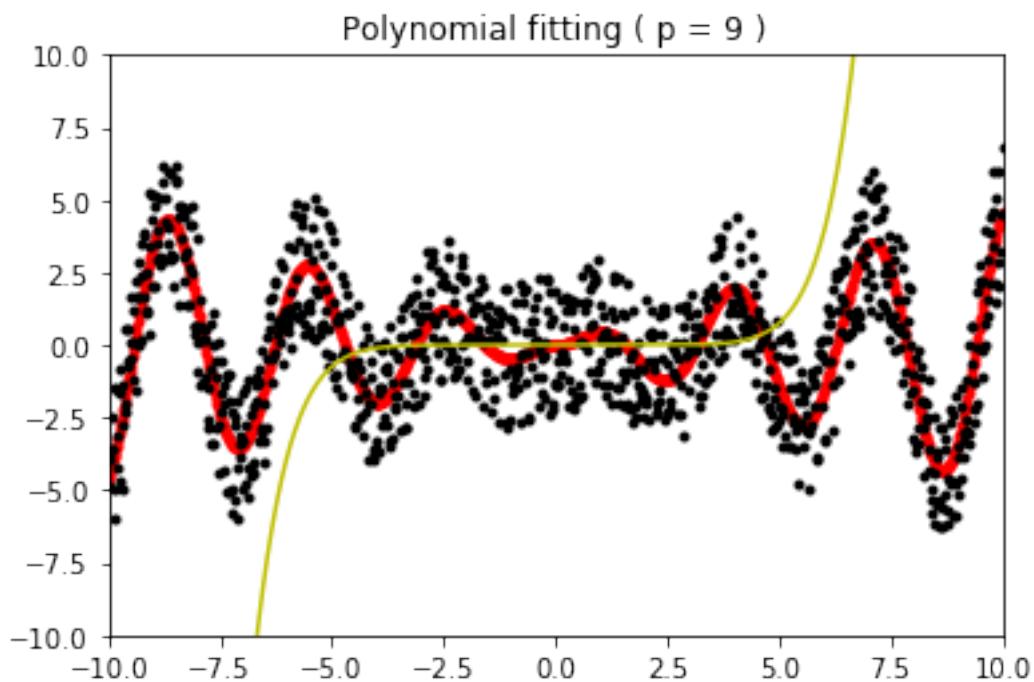


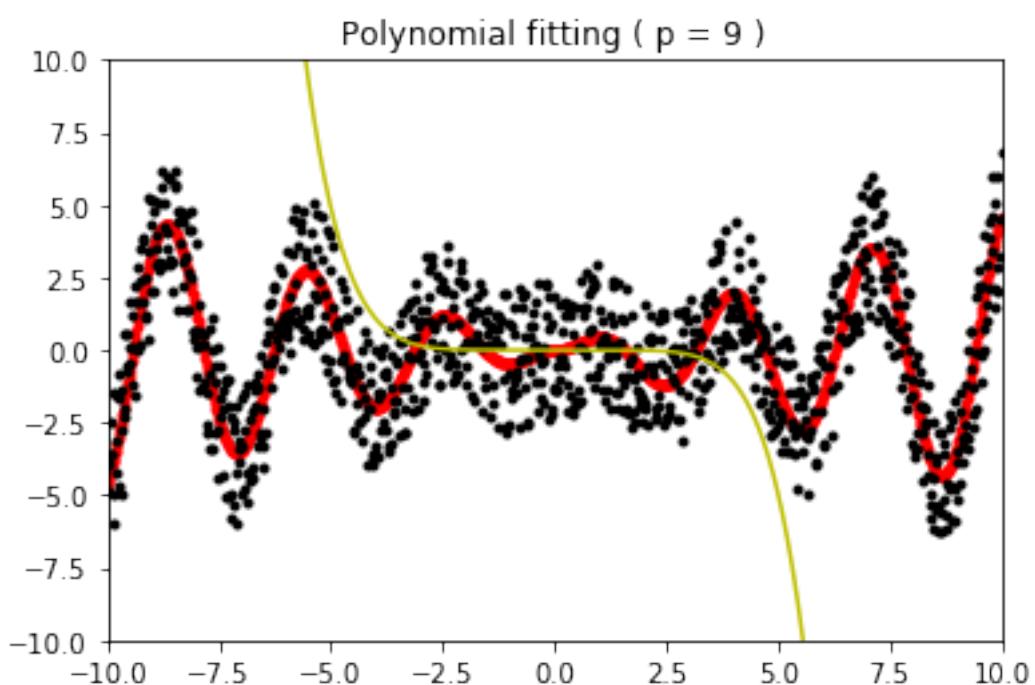
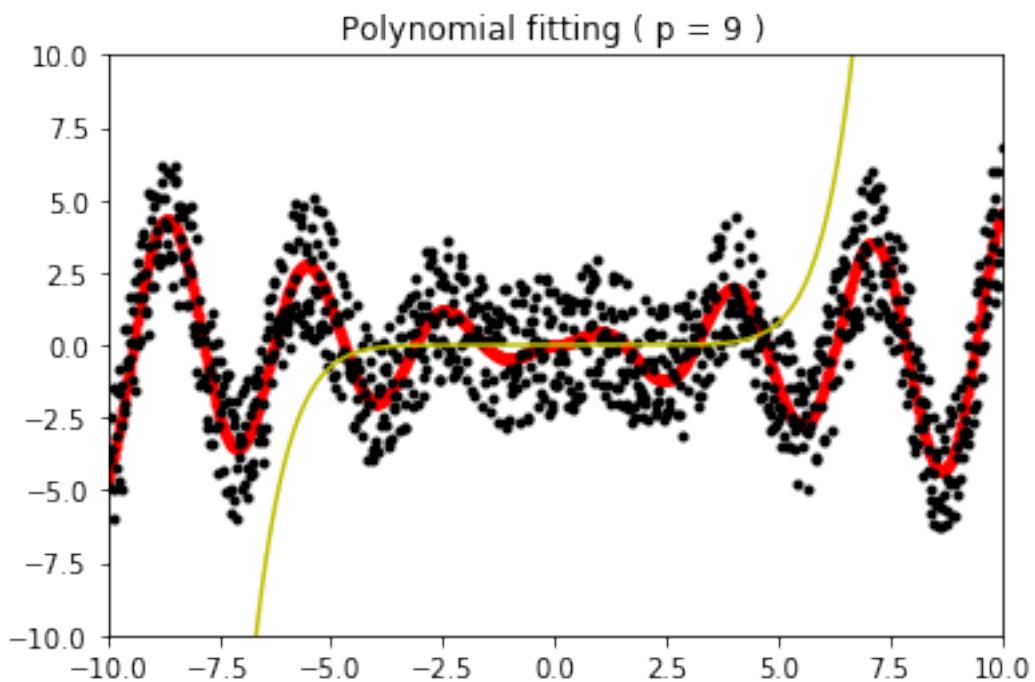


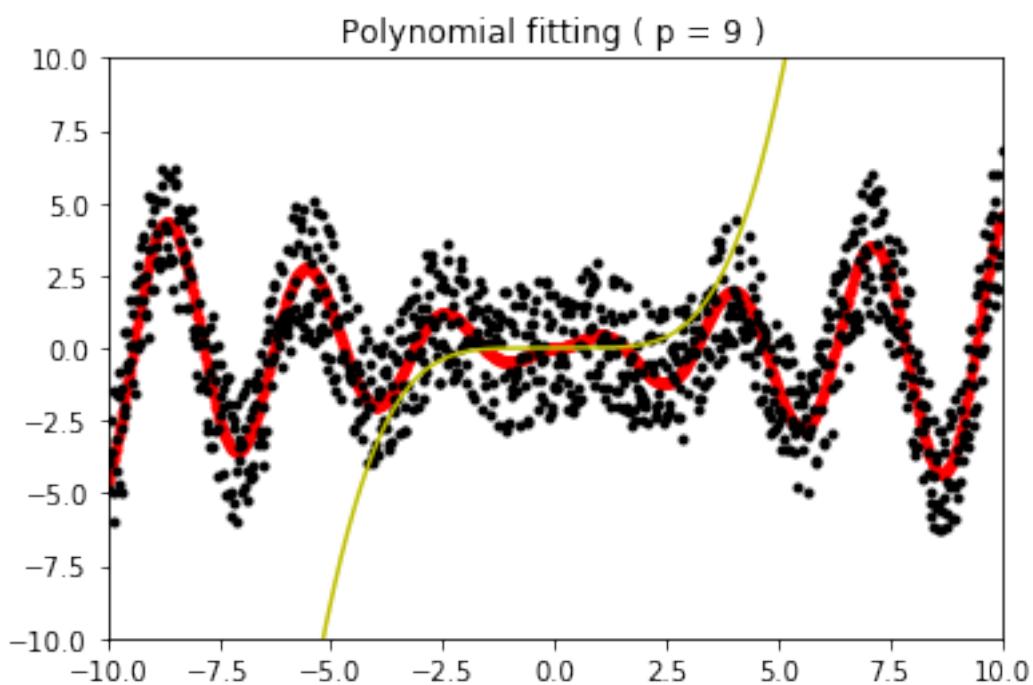
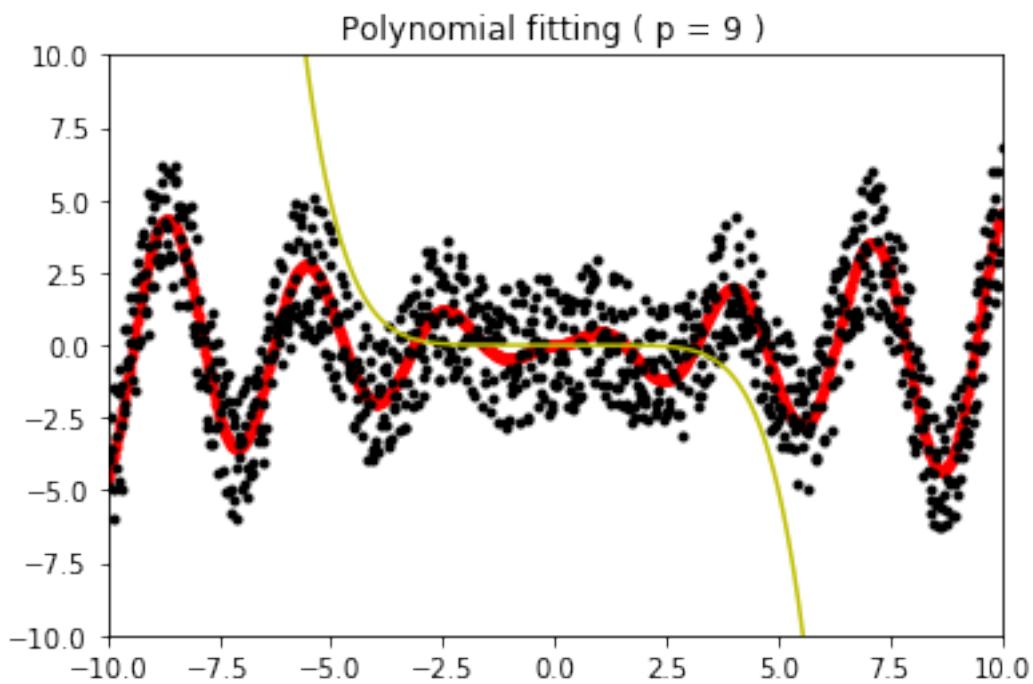


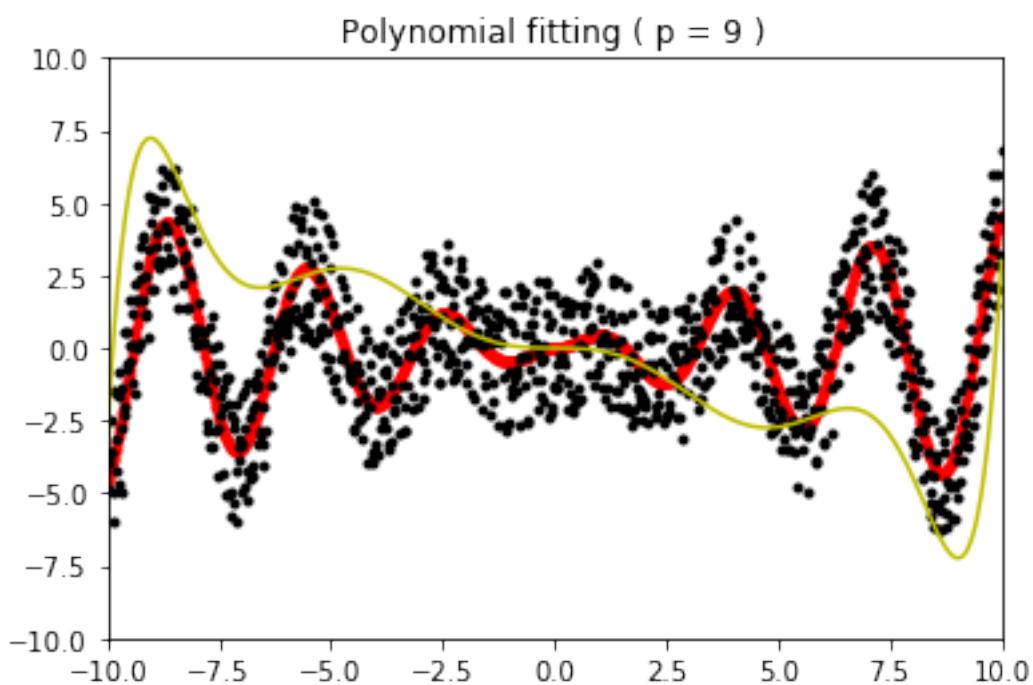
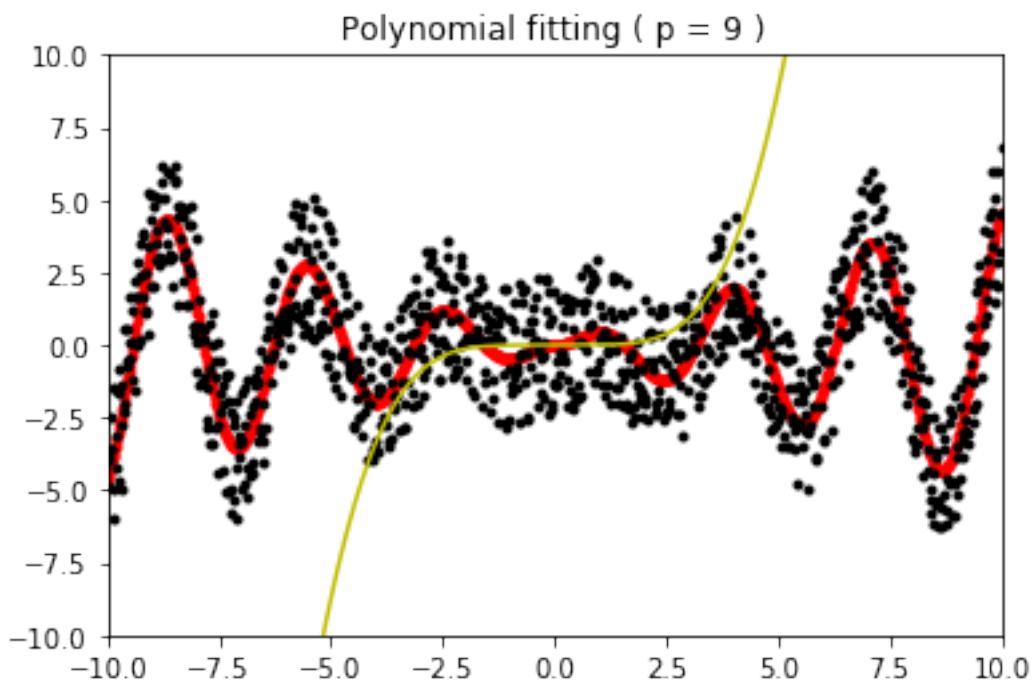
(10, 1)
[[3.81499520e-07]

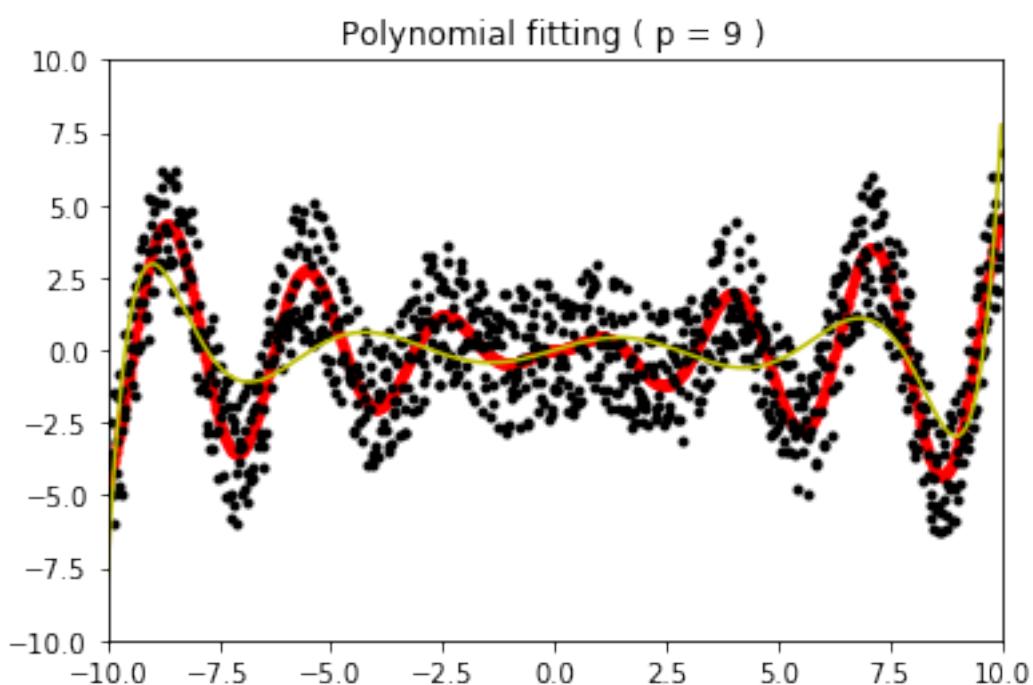
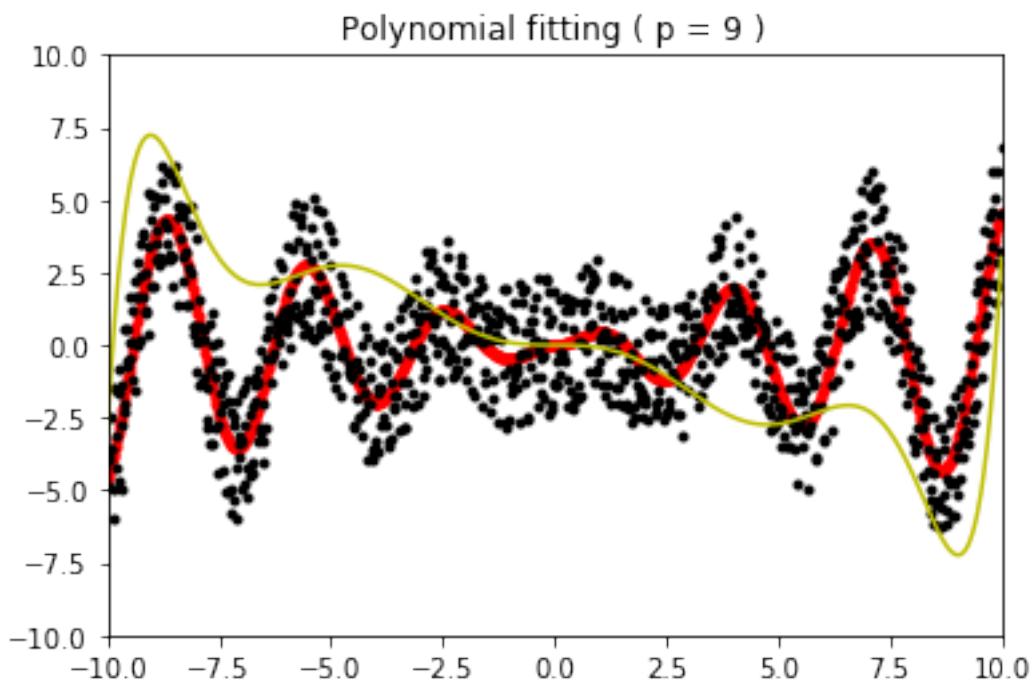
```
[ 2.31239664e-18]
[-7.23474344e-05]
[-3.89011739e-16]
[ 4.36686298e-03]
[ 1.82576935e-14]
[-9.17019945e-02]
[-1.84105769e-13]
[ 4.73312299e-01]
[-5.73032941e-13]]
```

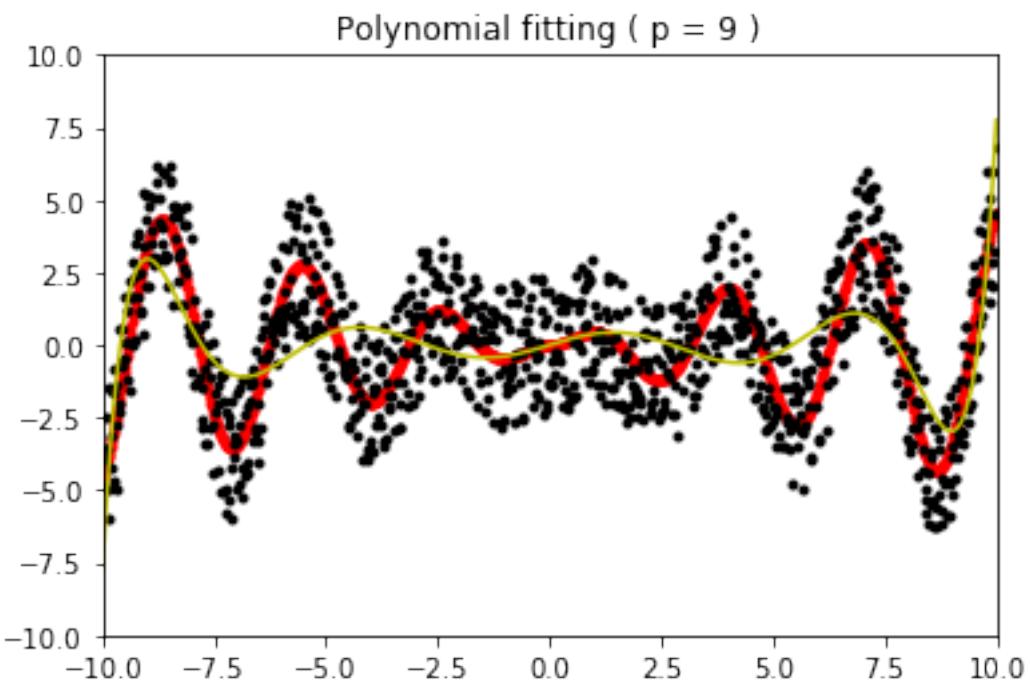




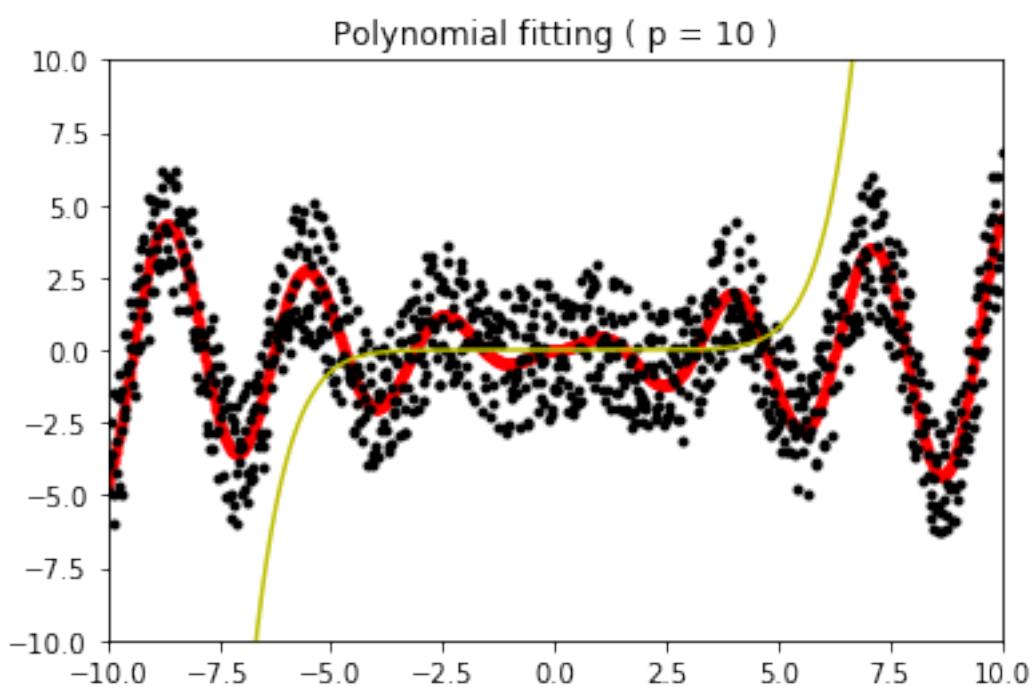
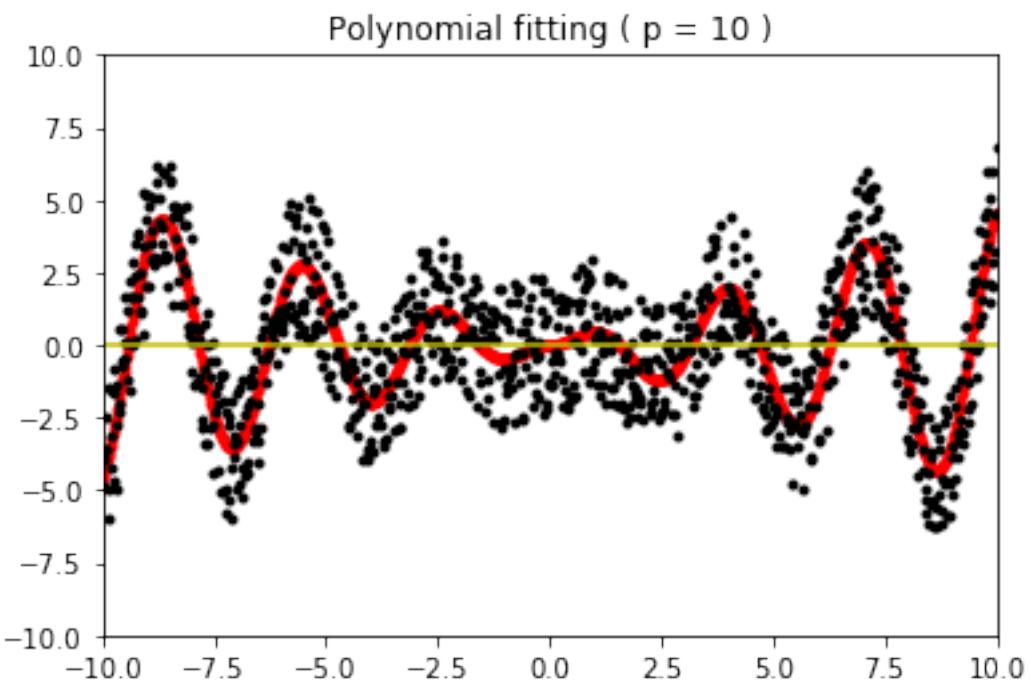


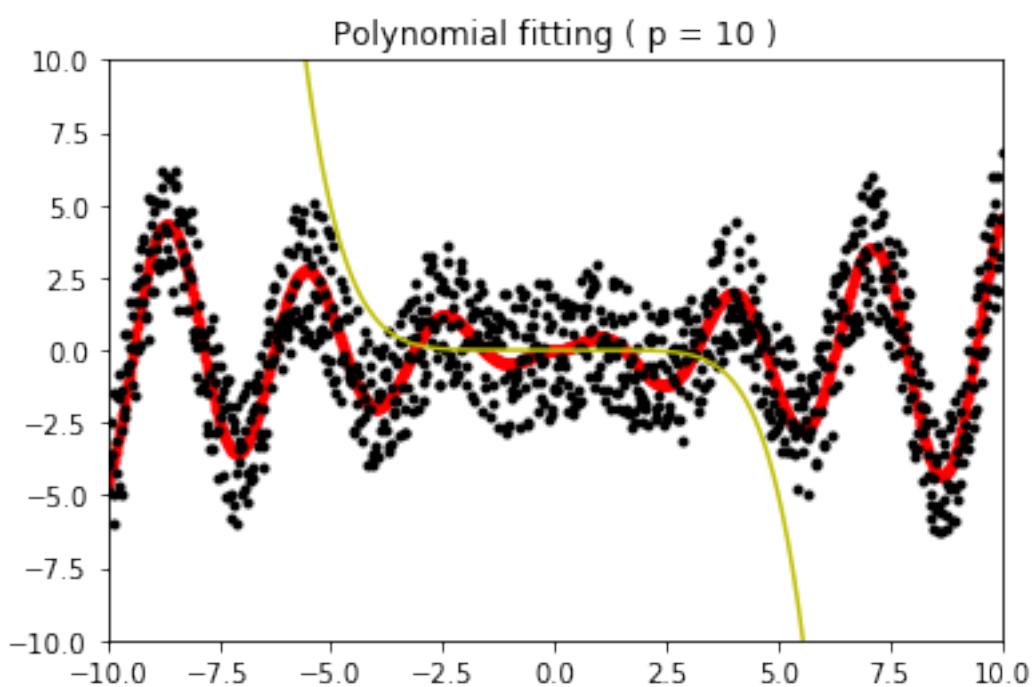
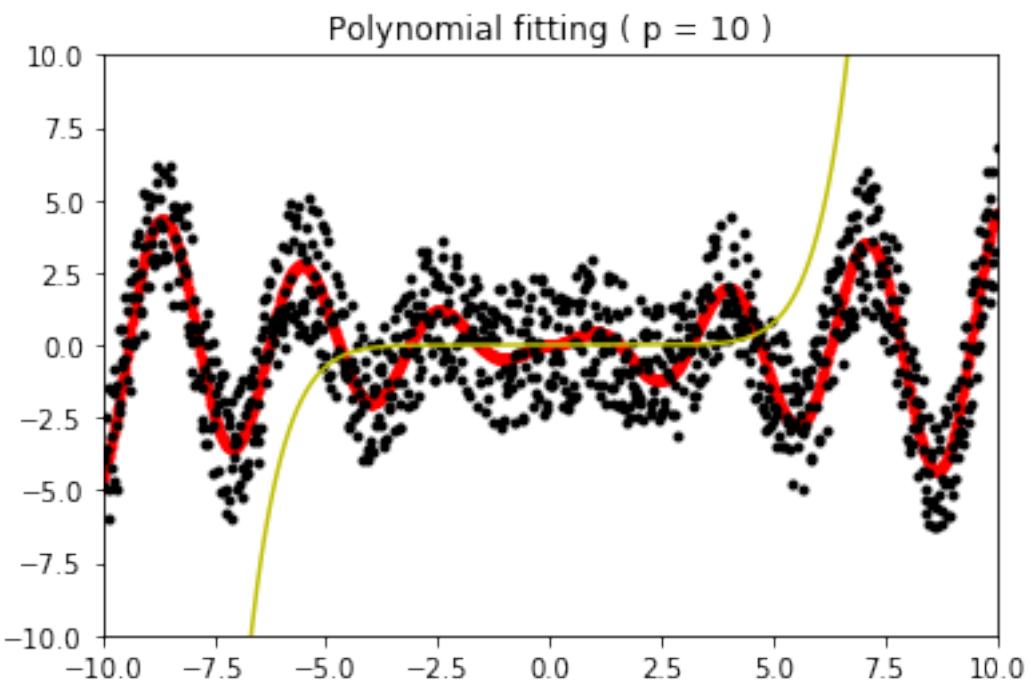


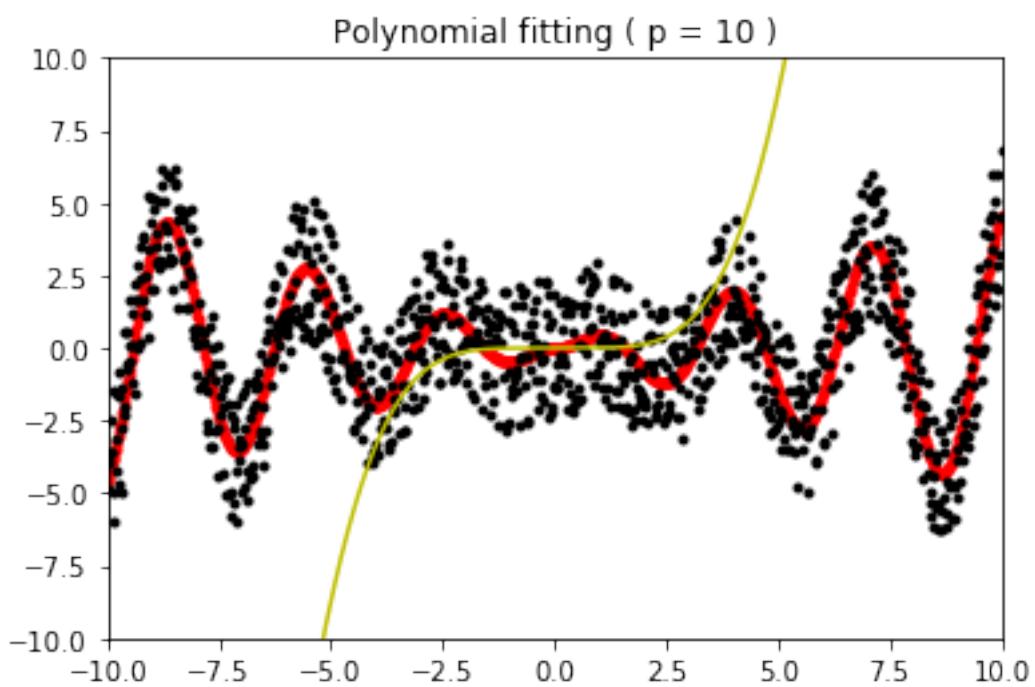
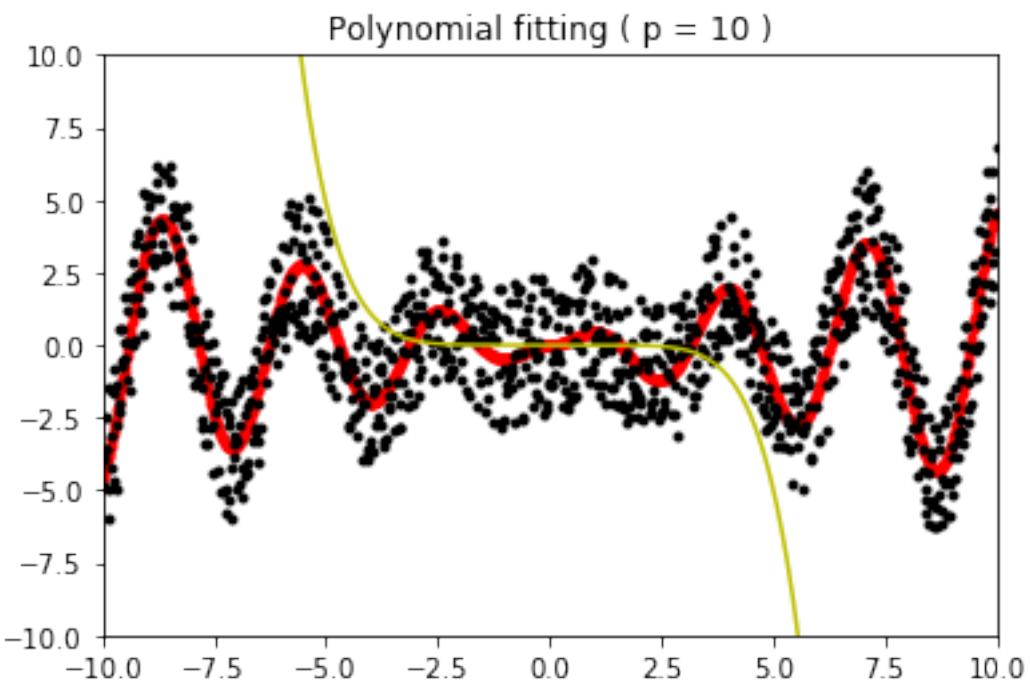


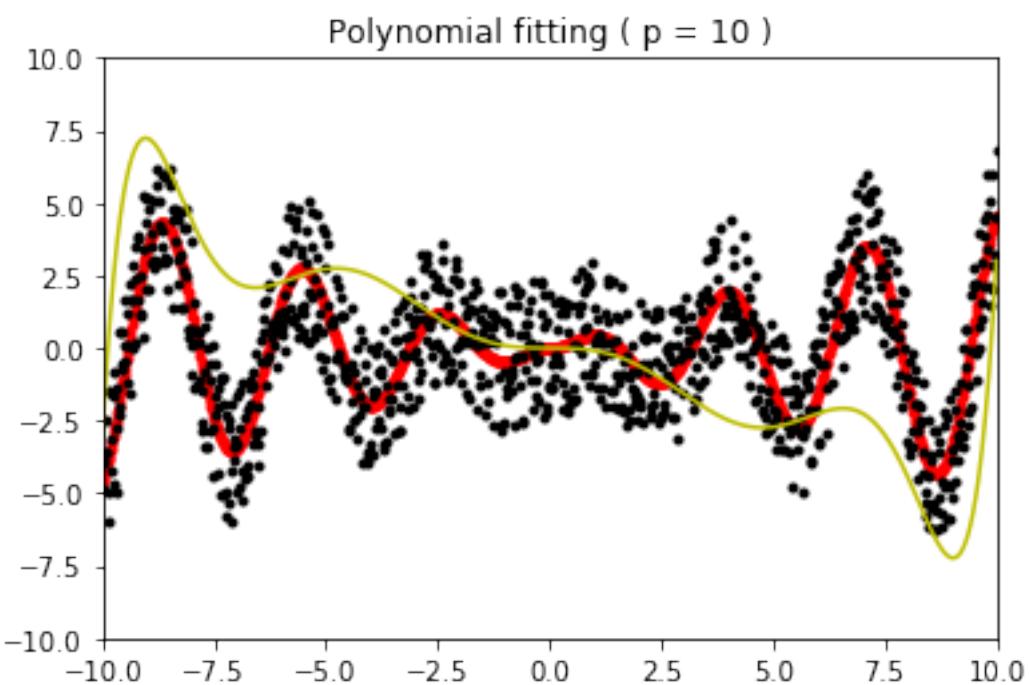
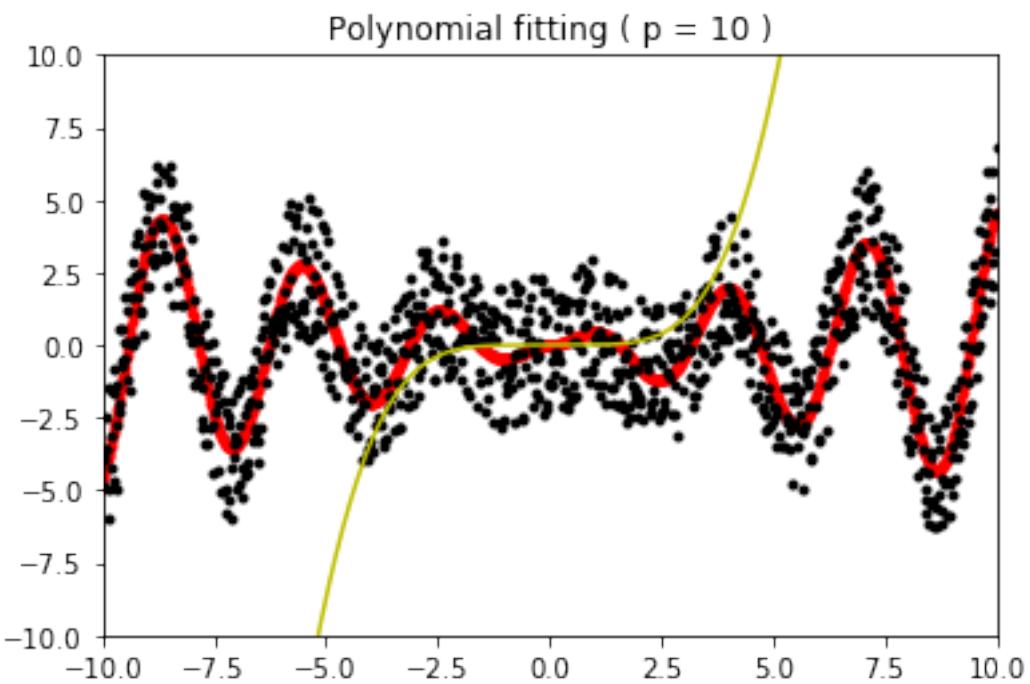


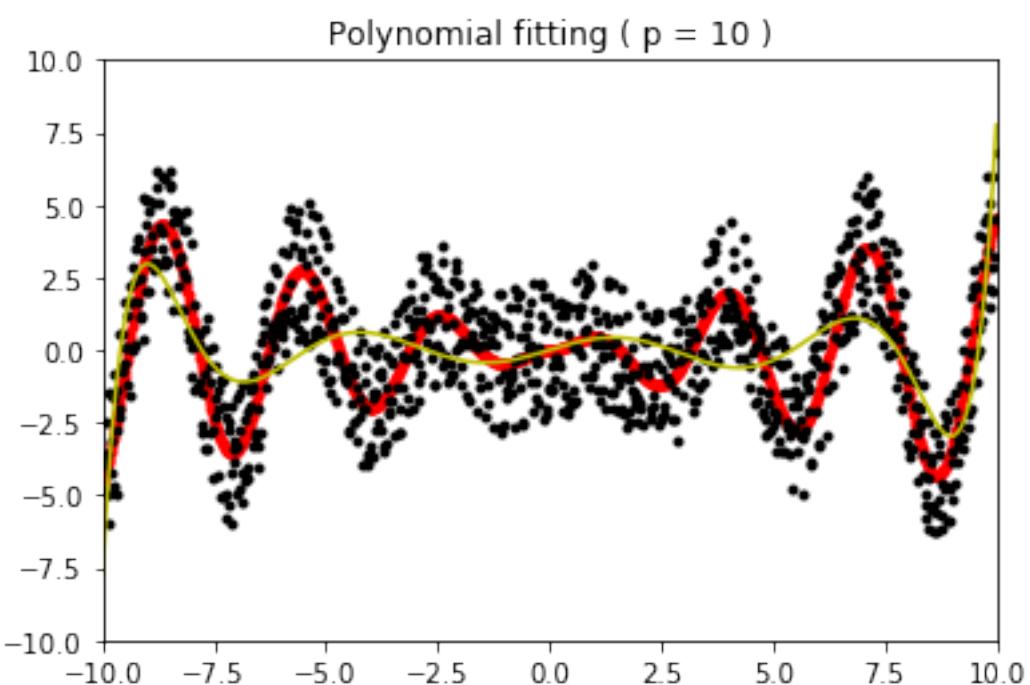
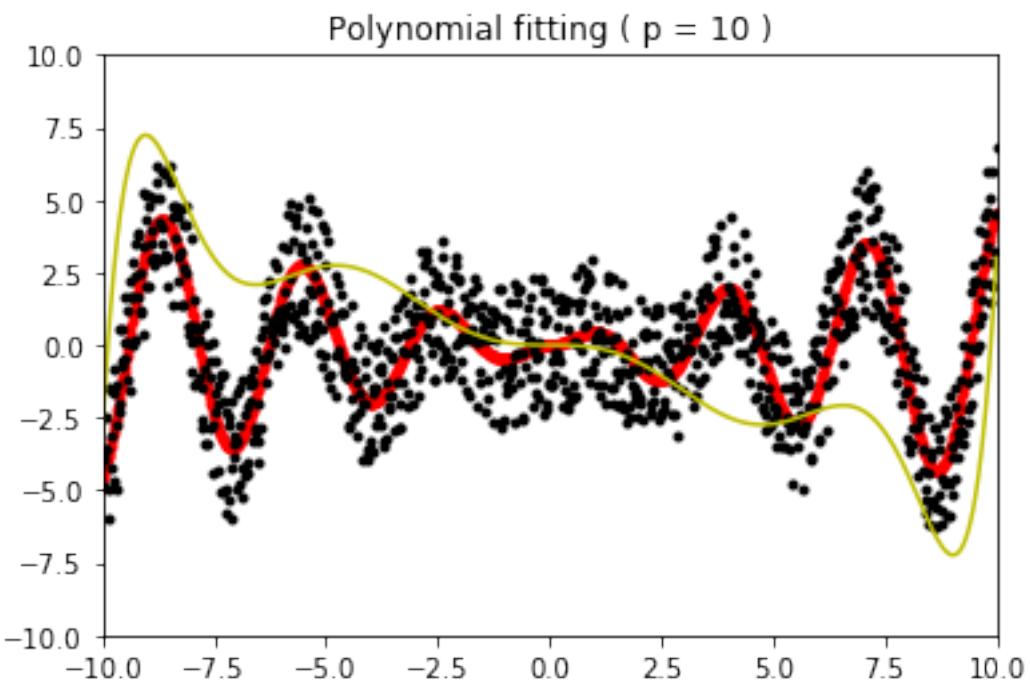
```
(11, 1)
[[-4.10791954e-19]
 [ 3.81499520e-07]
 [ 9.98995423e-17]
 [-7.23474344e-05]
 [-8.43654980e-15]
 [ 4.36686298e-03]
 [ 2.87023985e-13]
 [-9.17019945e-02]
 [-3.29252511e-12]
 [ 4.73312299e-01]
 [ 5.07424571e-12]]
```

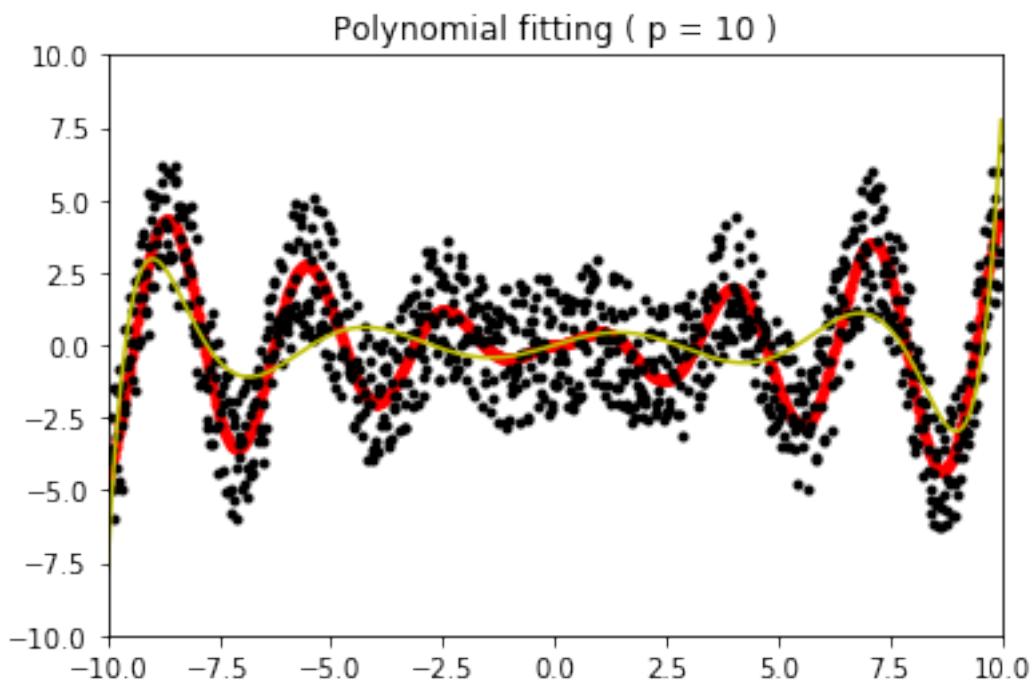




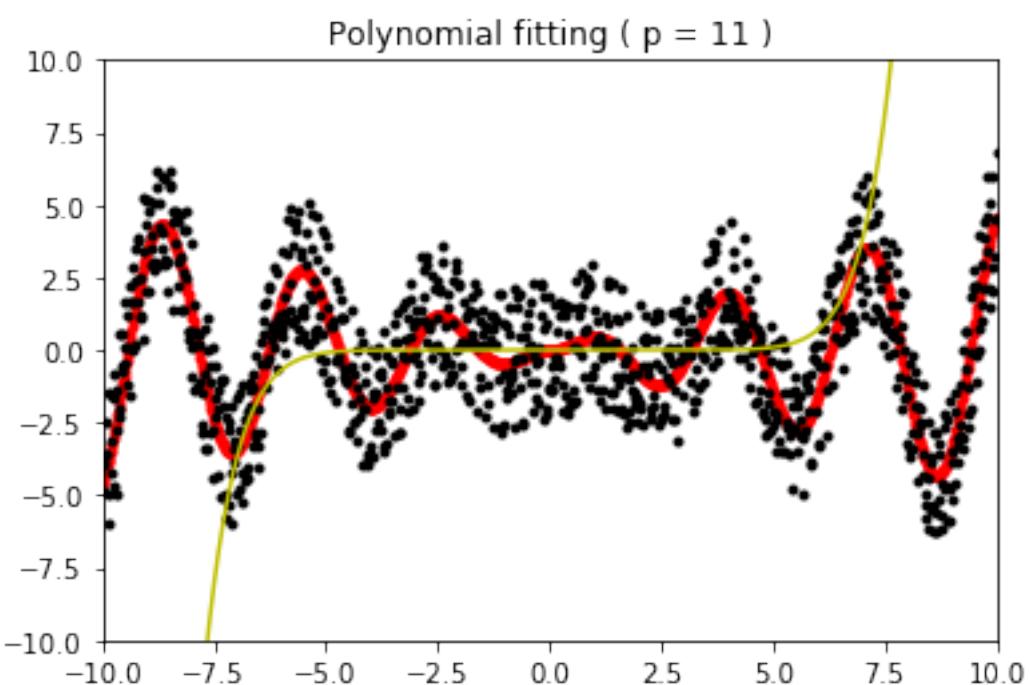
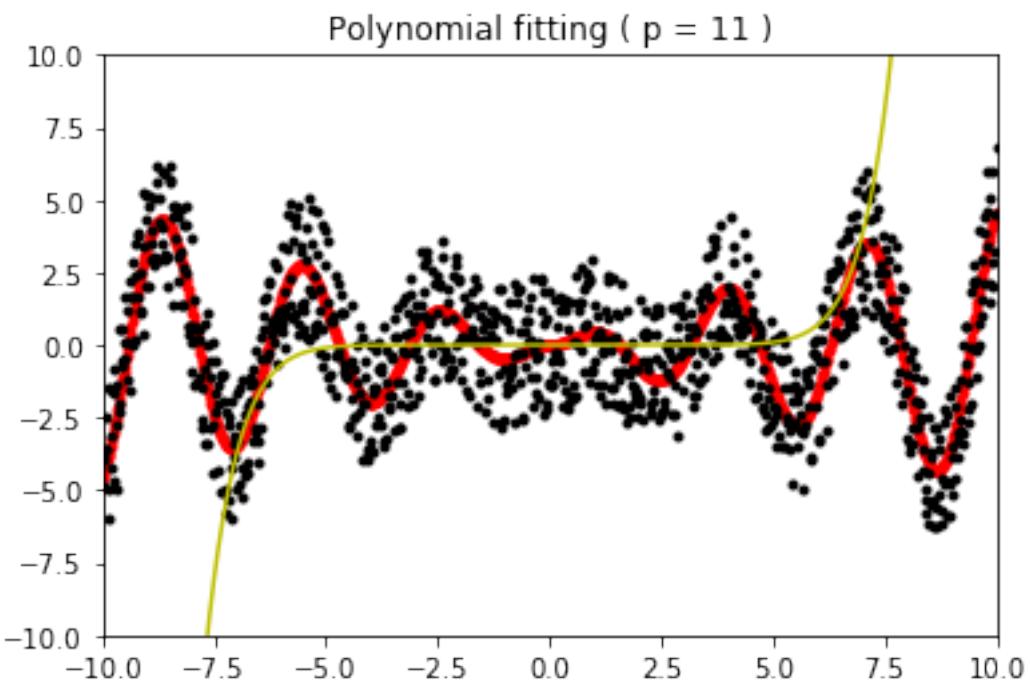


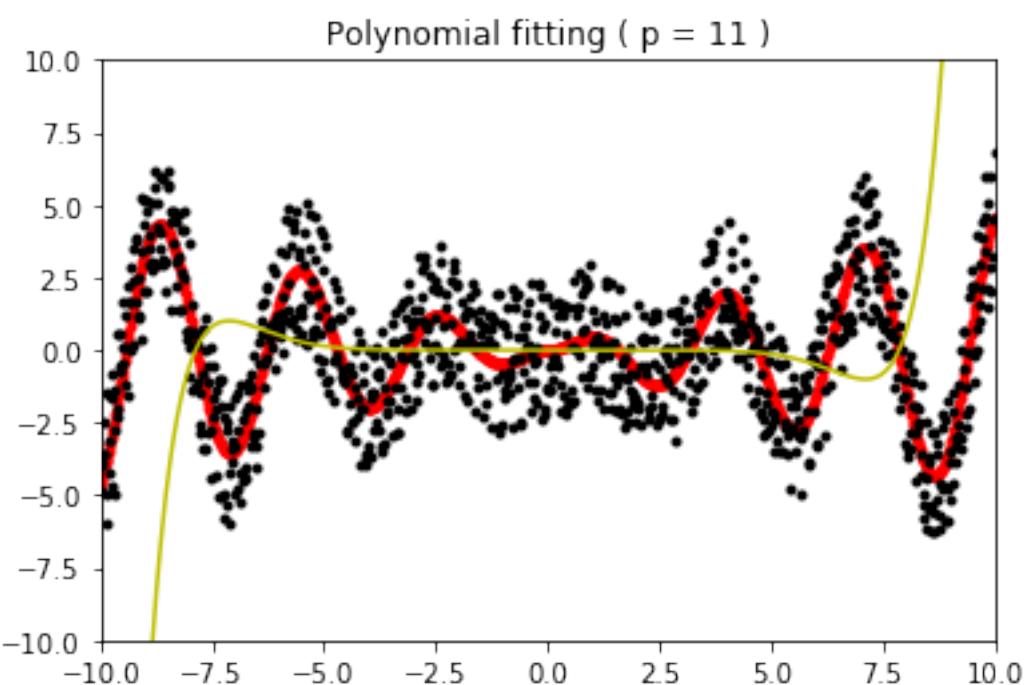
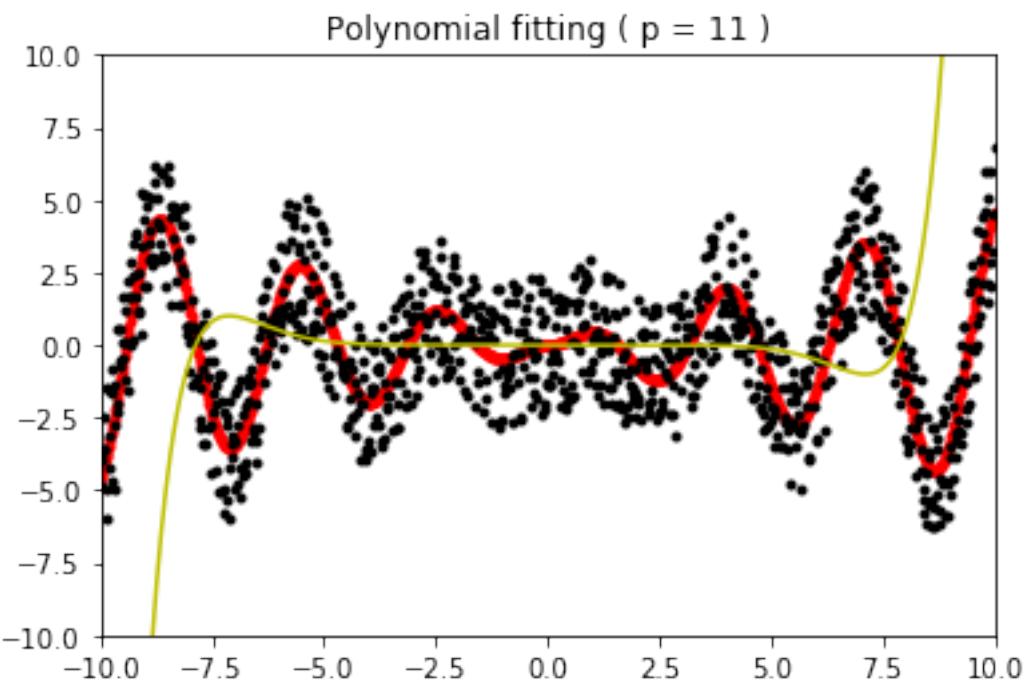


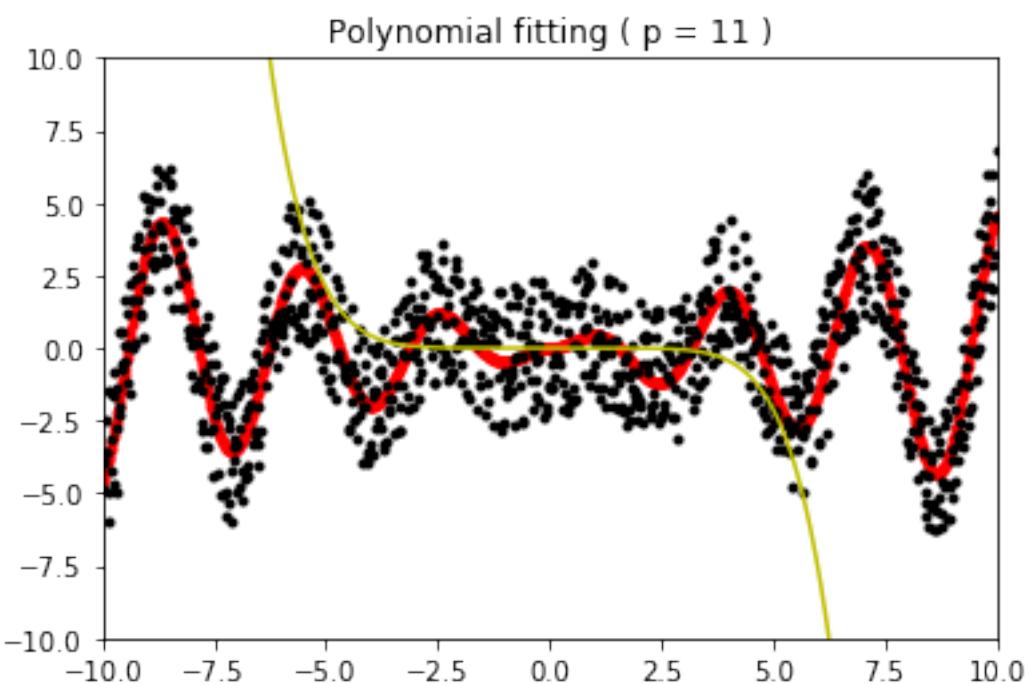
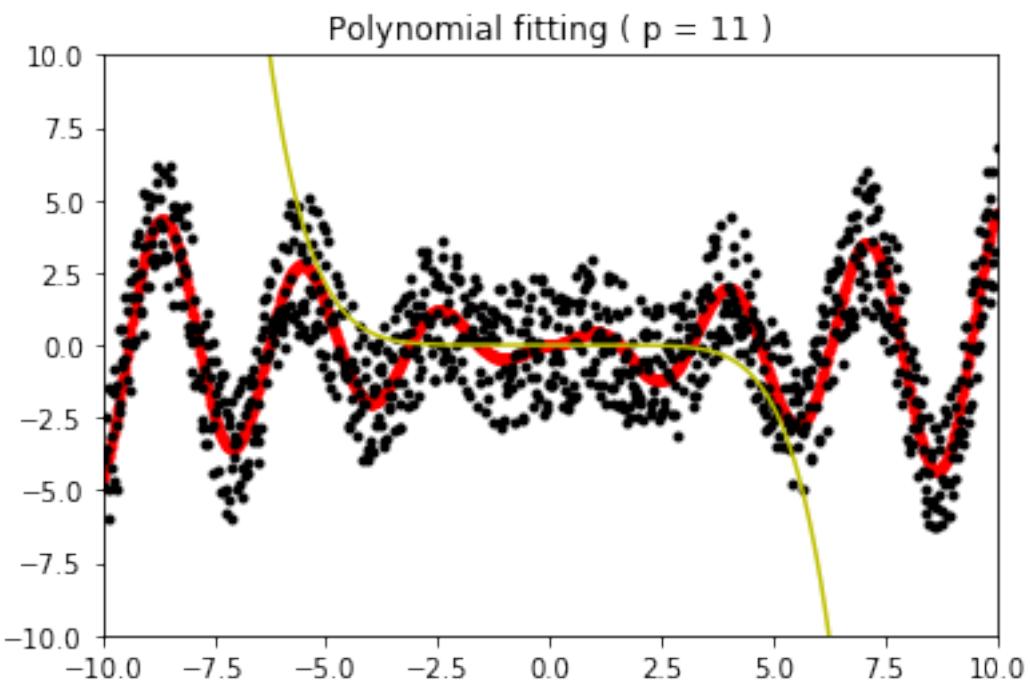


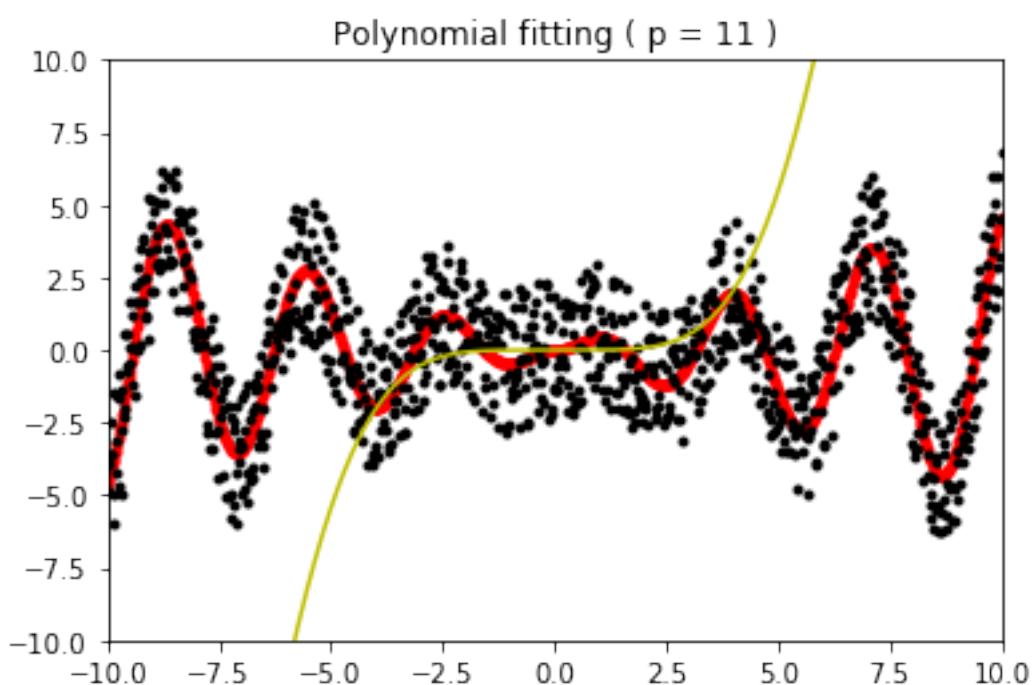
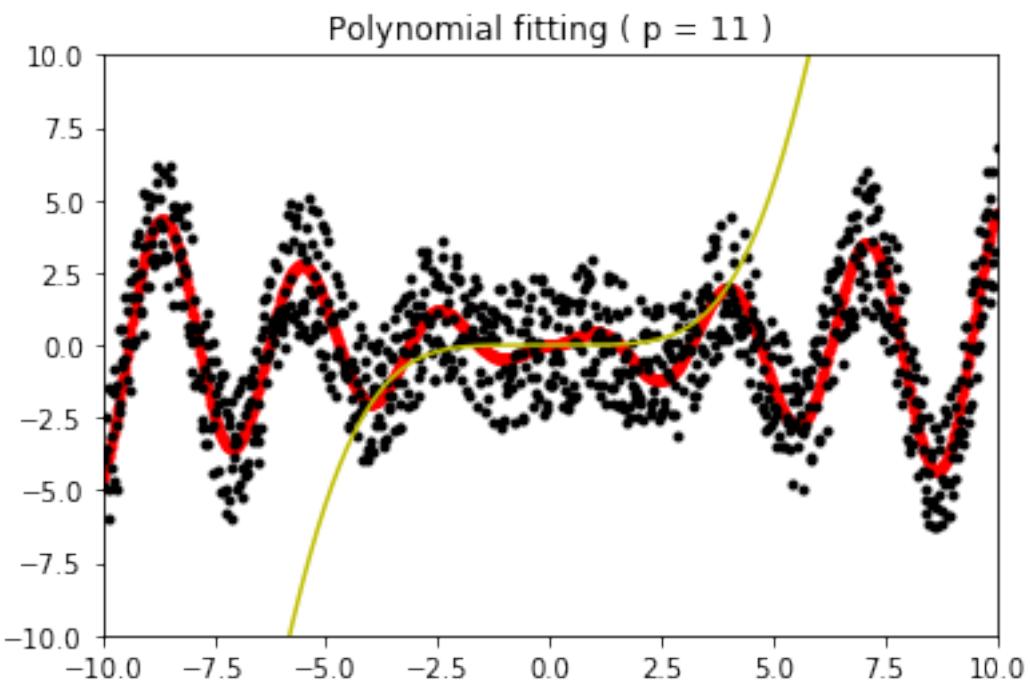


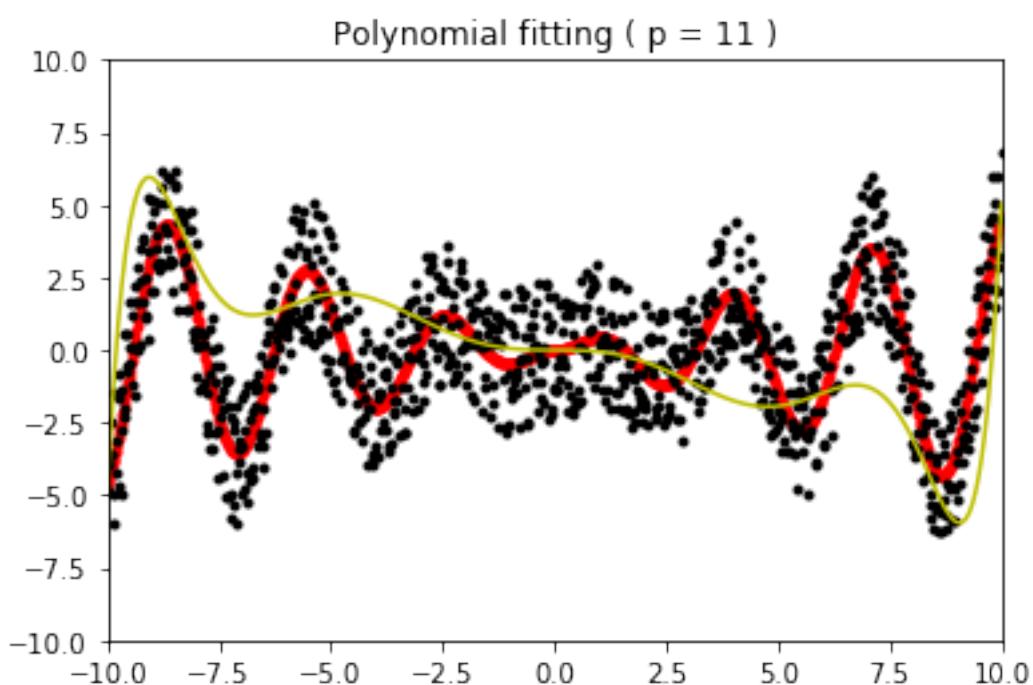
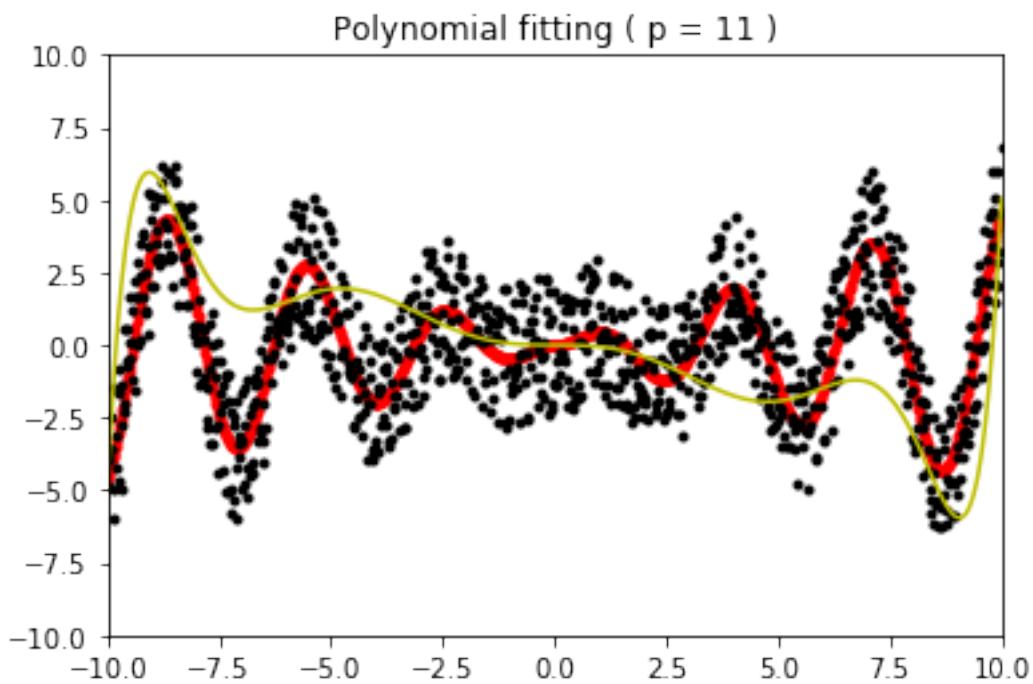
```
(12, 1)
[[ 1.90285972e-09]
 [-6.60584266e-19]
 [-1.17848258e-07]
 [ 1.57924476e-16]
 [-2.49478645e-05]
 [-1.31272319e-14]
 [ 2.41128420e-03]
 [ 4.39121992e-13]
 [-5.90450700e-02]
 [-5.00178284e-12]
 [ 3.22292525e-01]
 [ 8.32098453e-12]]
```

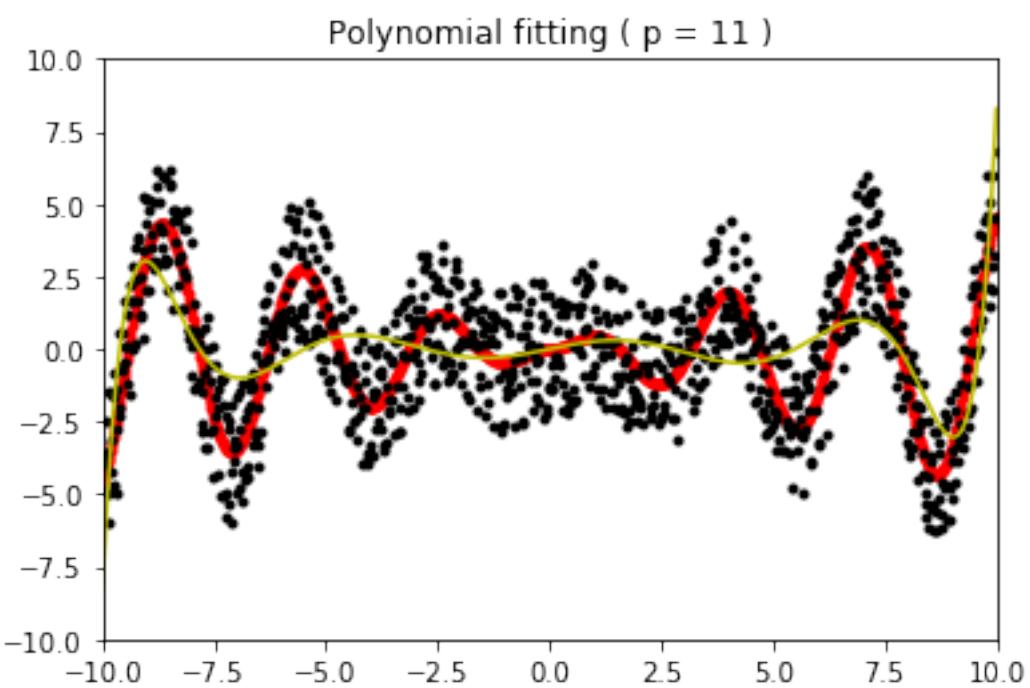
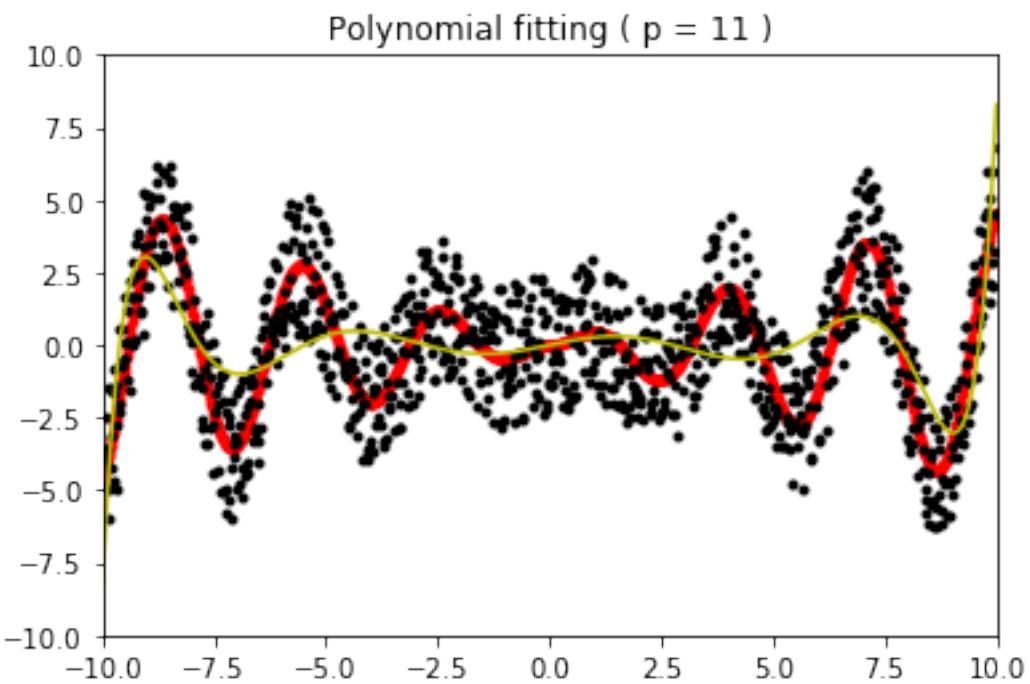






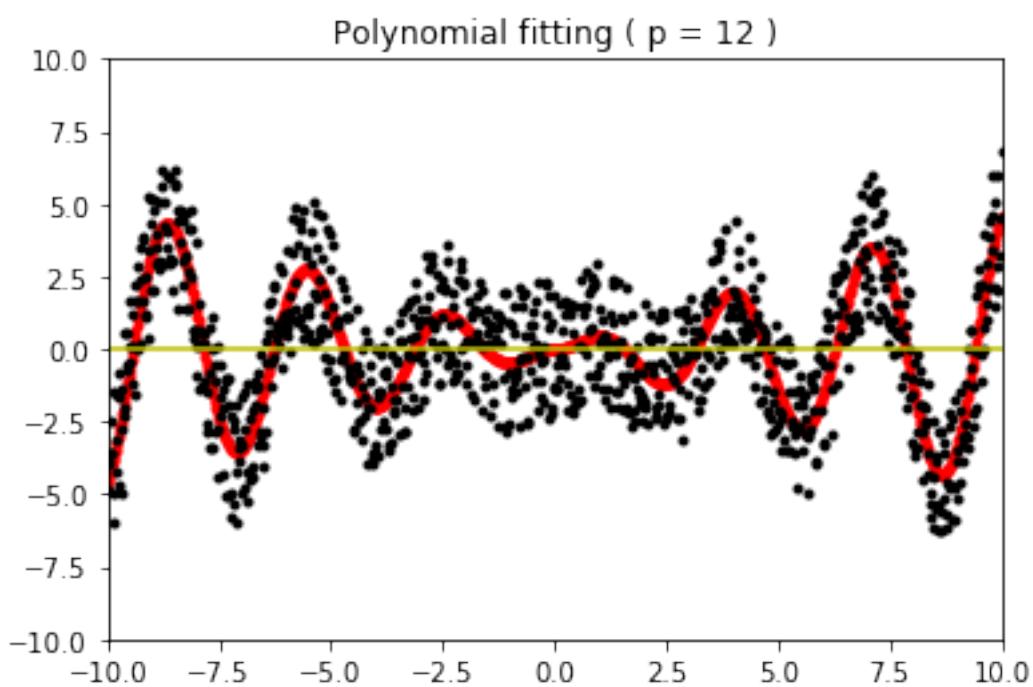


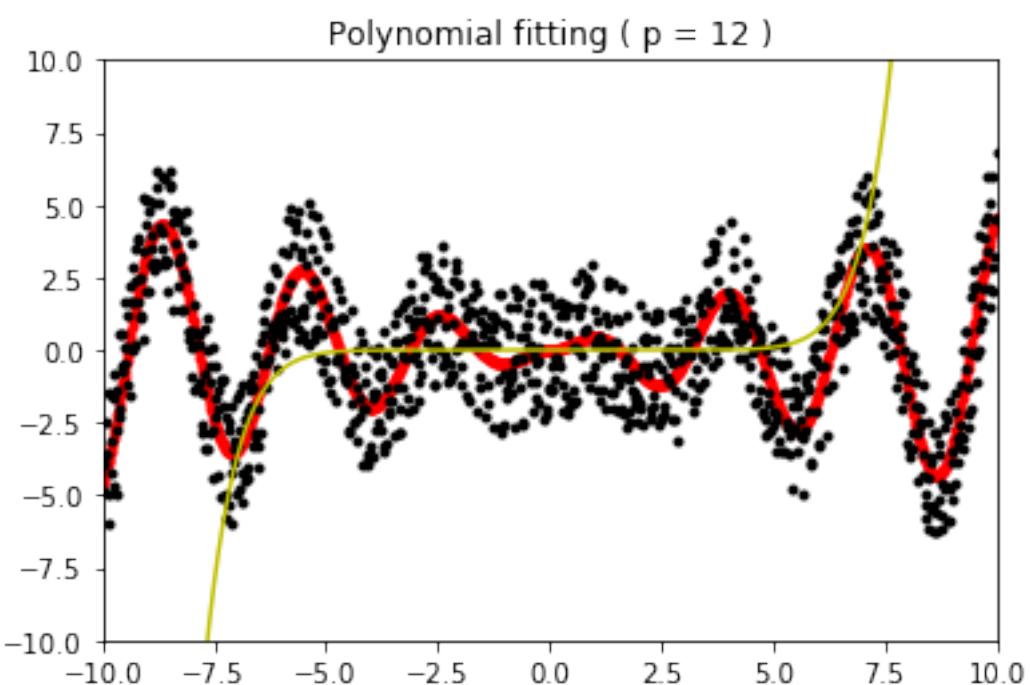
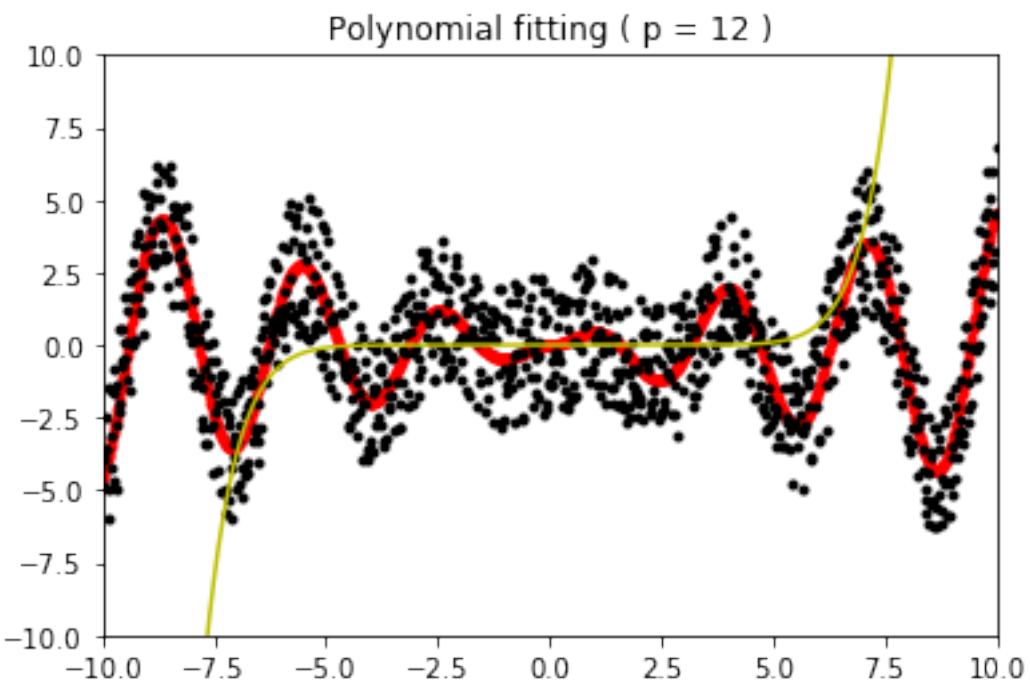


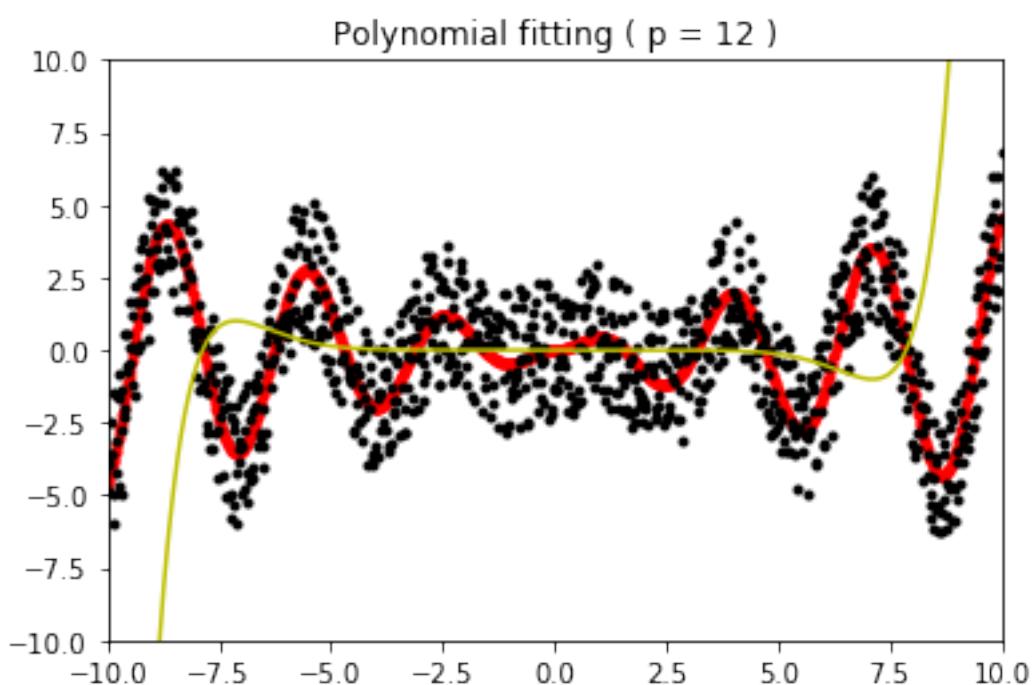
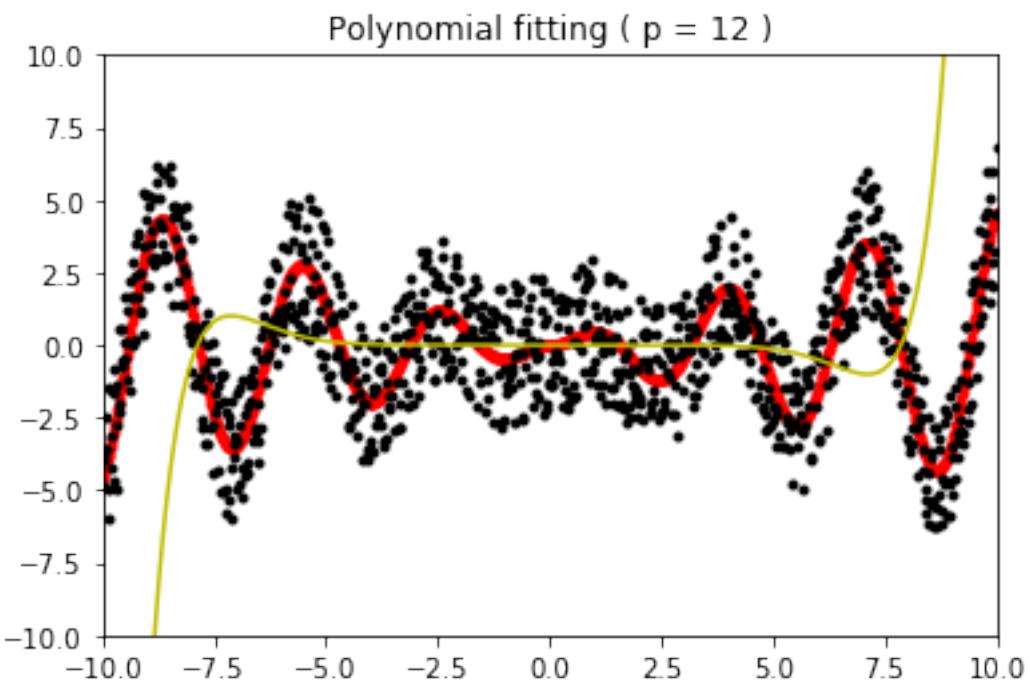


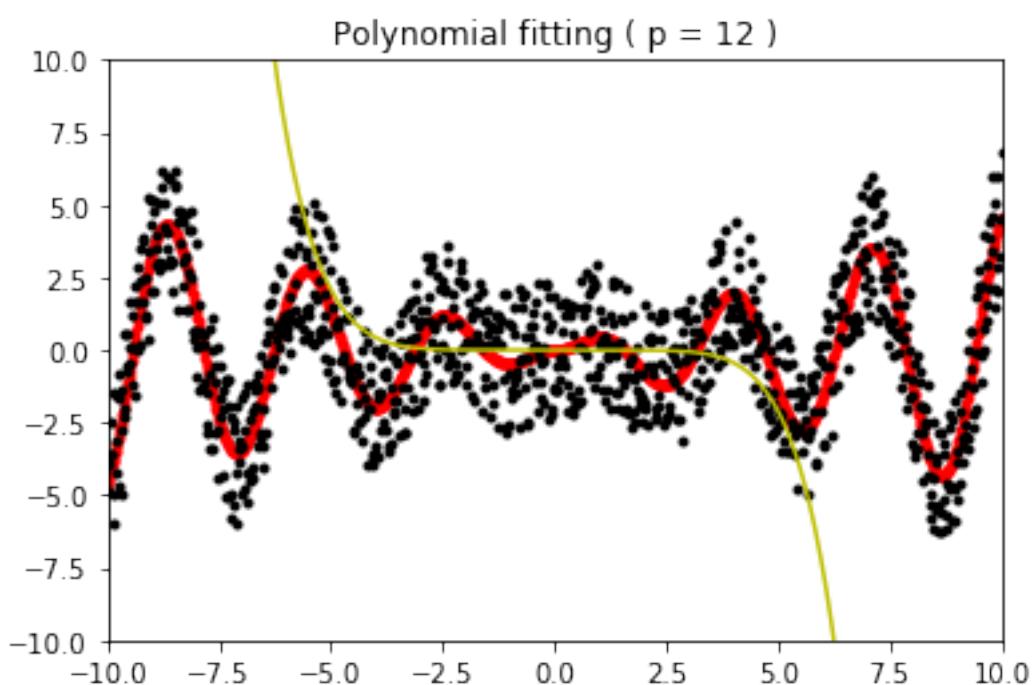
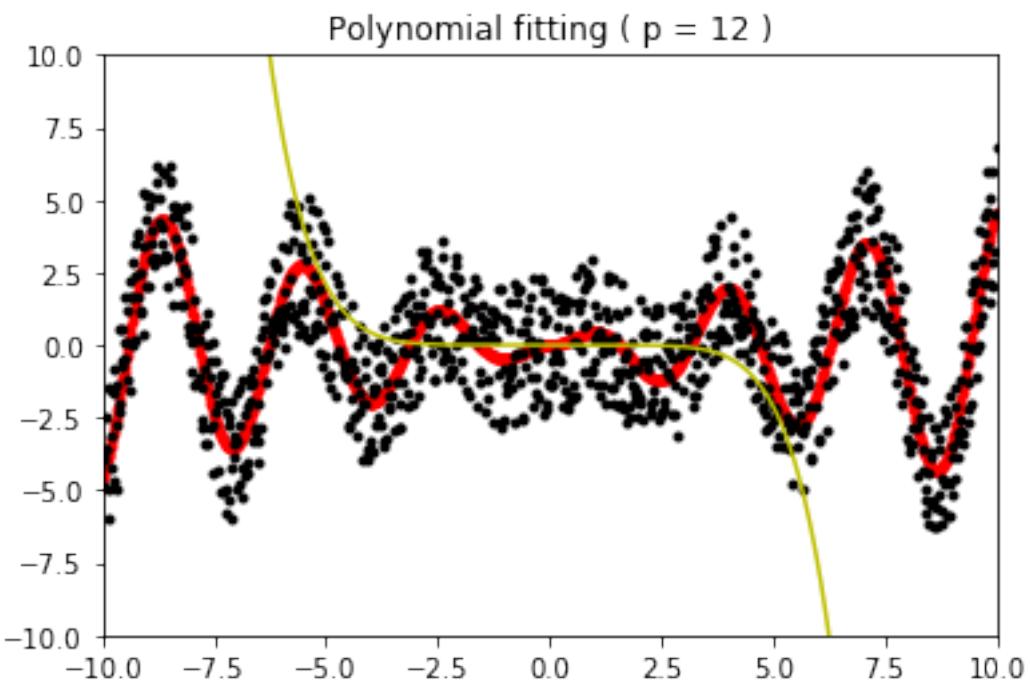
(13, 1)
[[-5.72000240e-20]]

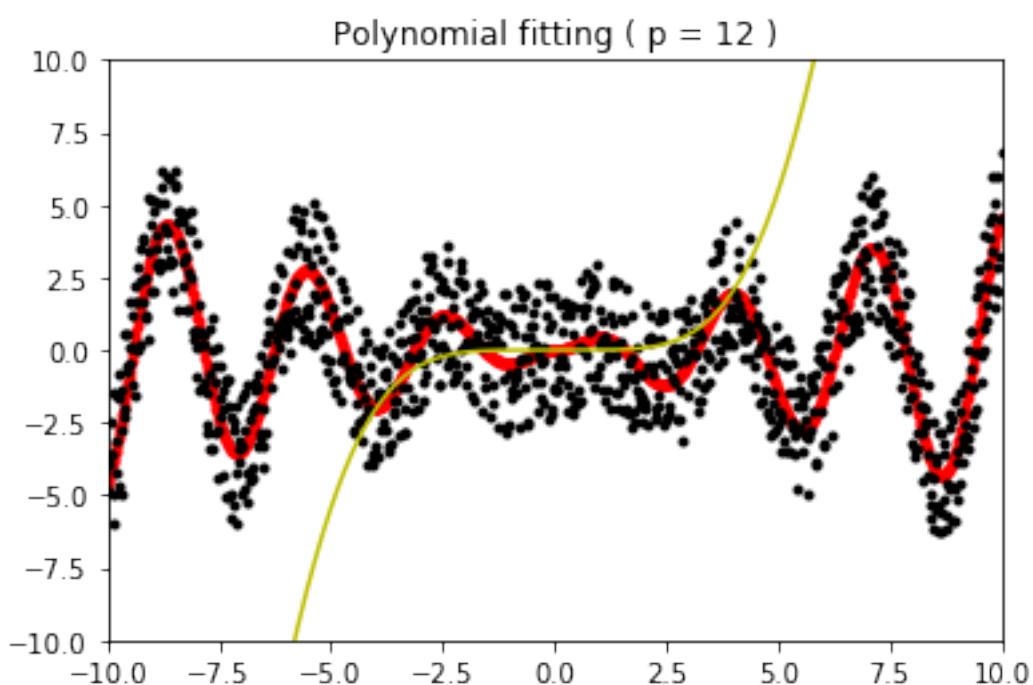
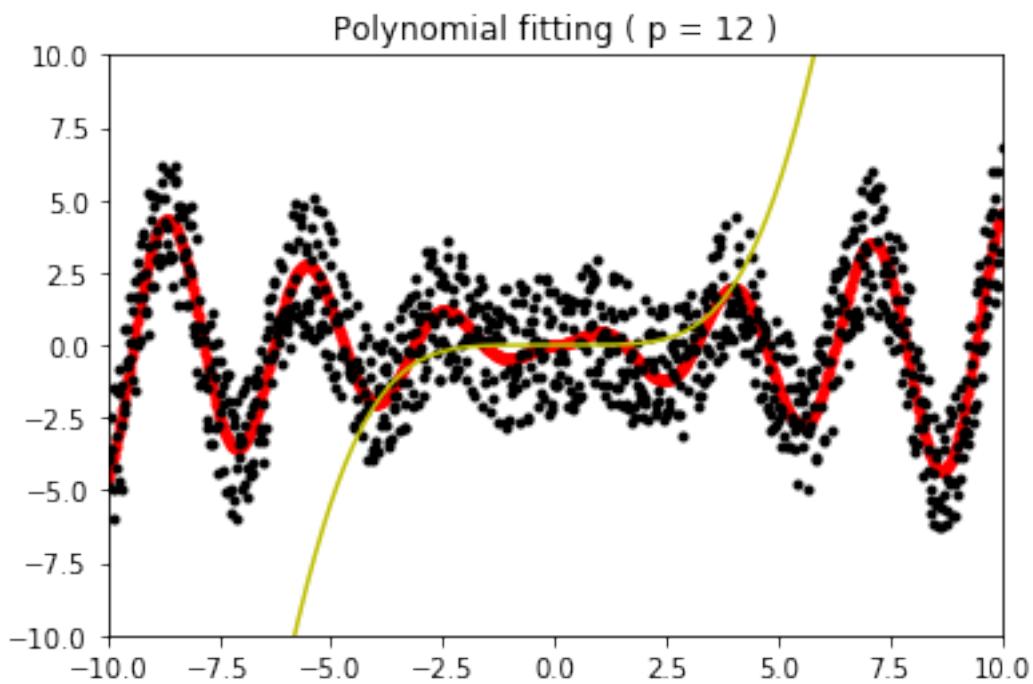
```
[ 1.90285972e-09]
[ 1.58147669e-17]
[-1.17848258e-07]
[-1.61197781e-15]
[-2.49478645e-05]
[ 7.39731762e-14]
[ 2.41128420e-03]
[-1.48740081e-12]
[-5.90450700e-02]
[ 1.04001165e-11]
[ 3.22292525e-01]
[-1.14578095e-11]]
```

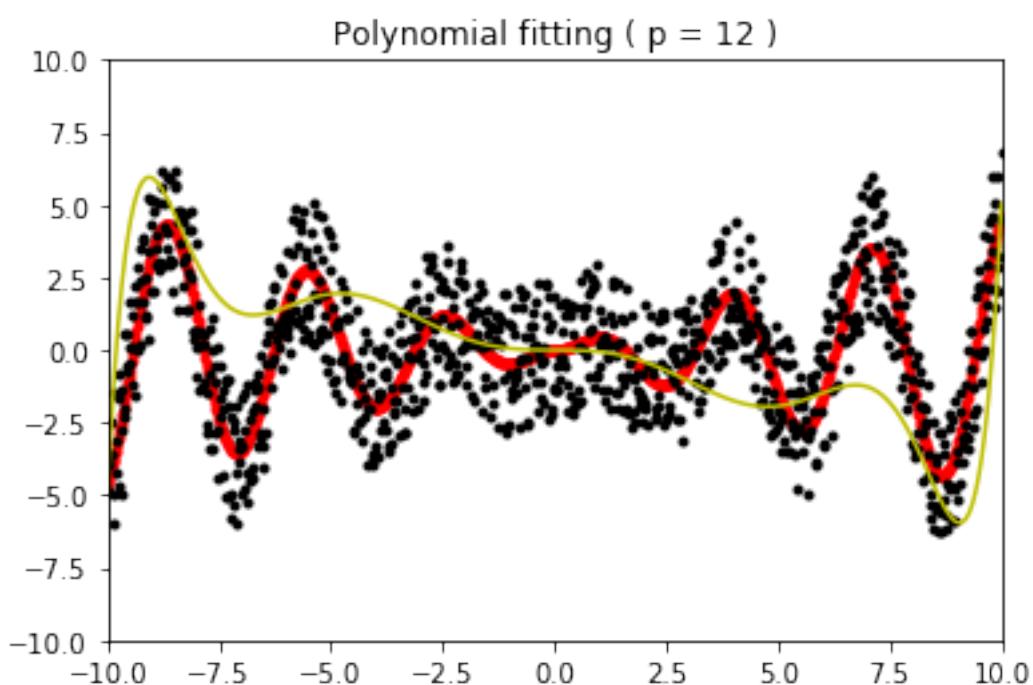
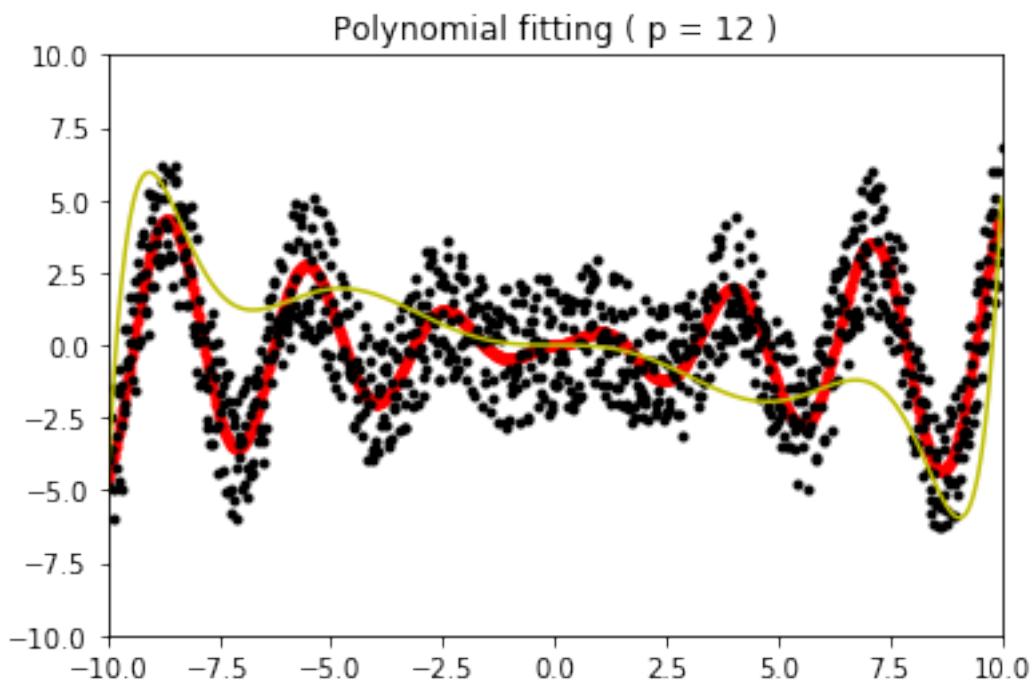


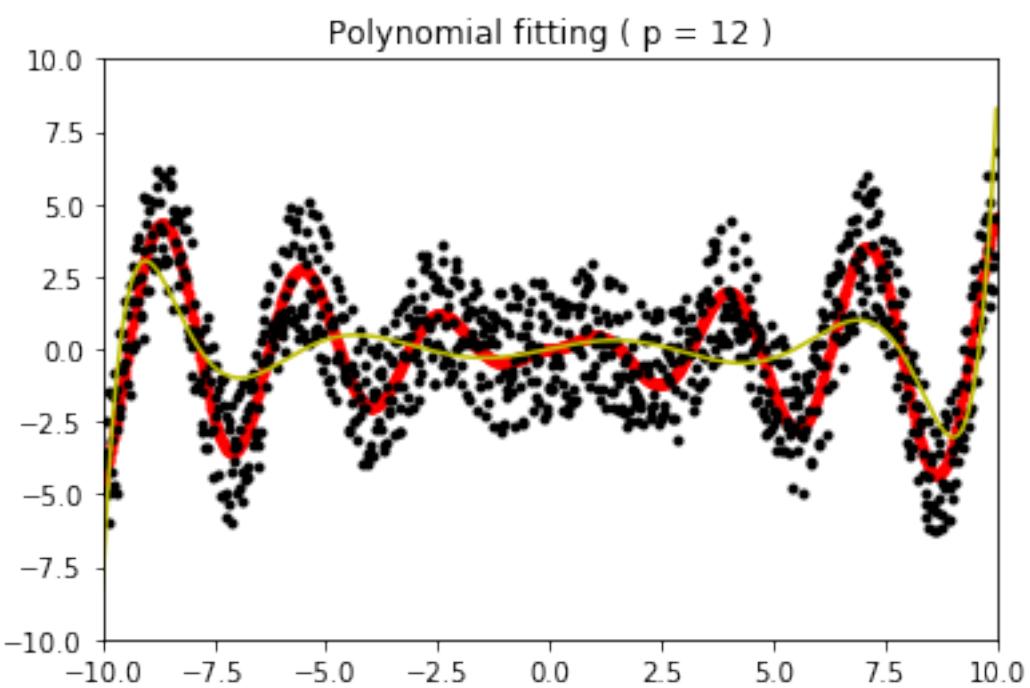
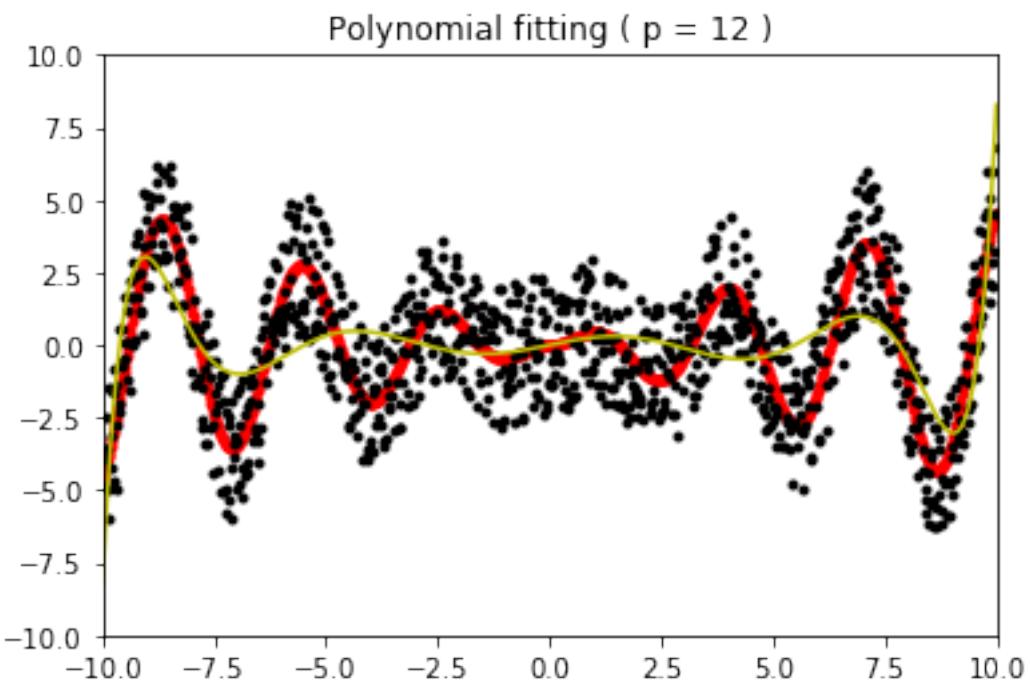






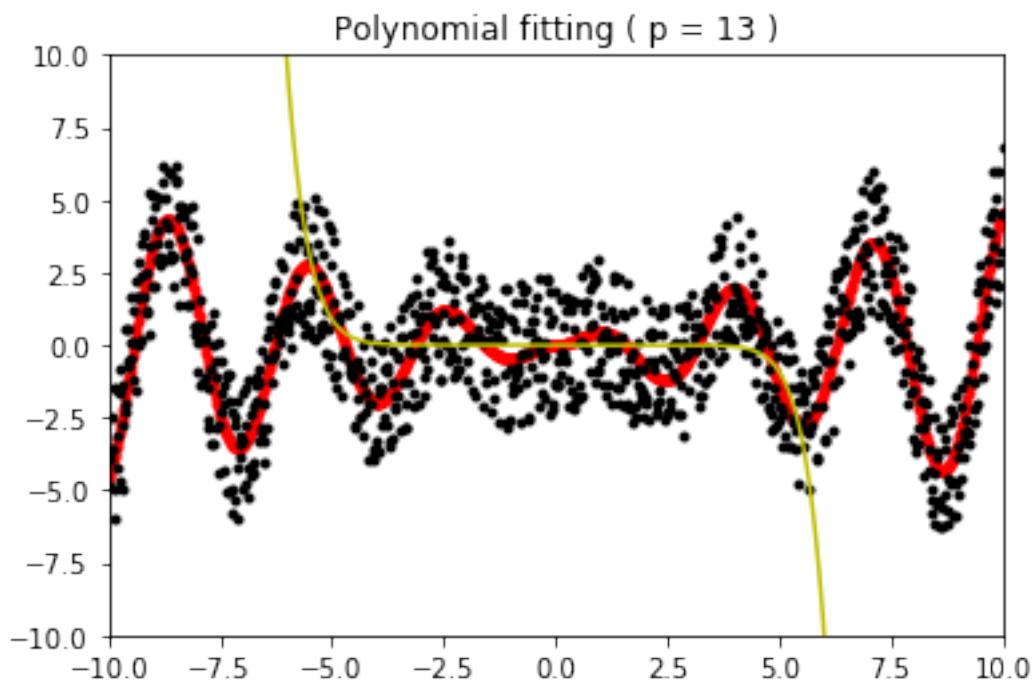


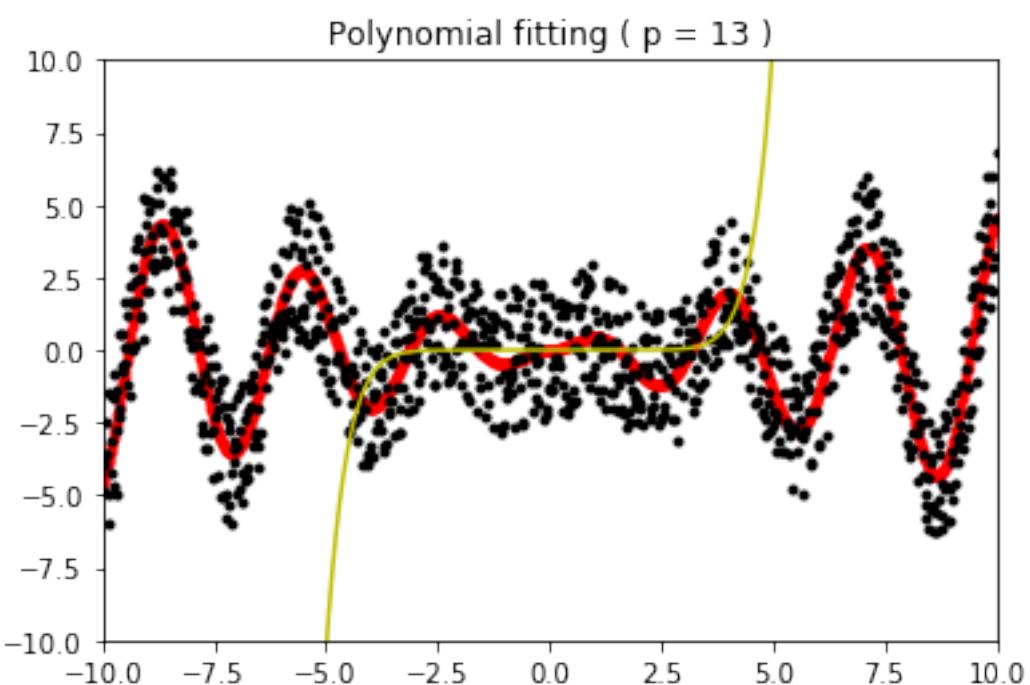
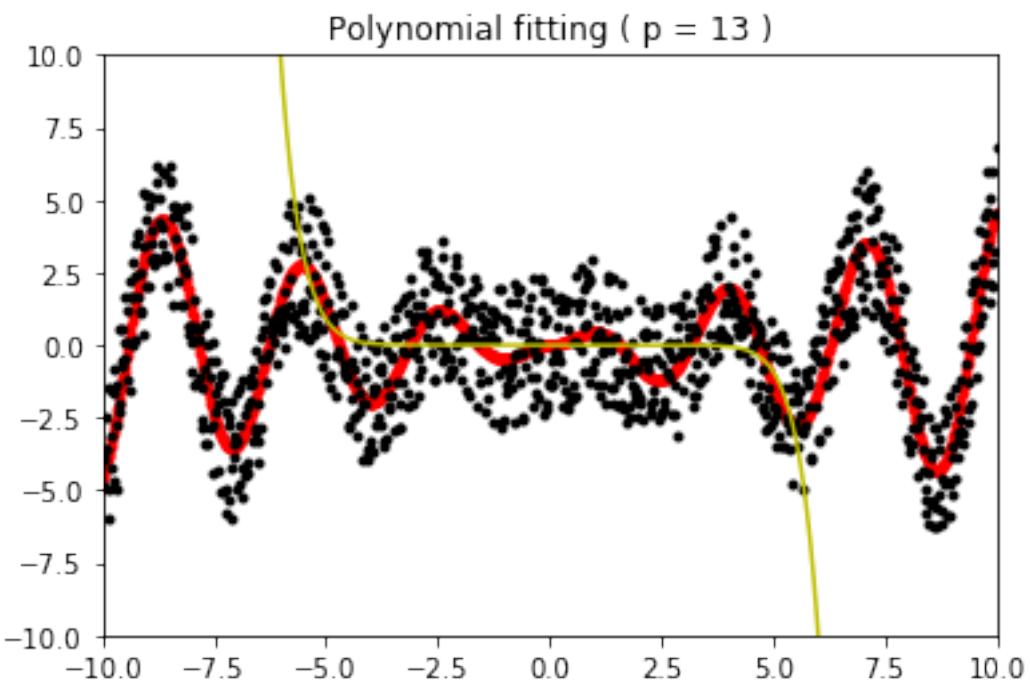


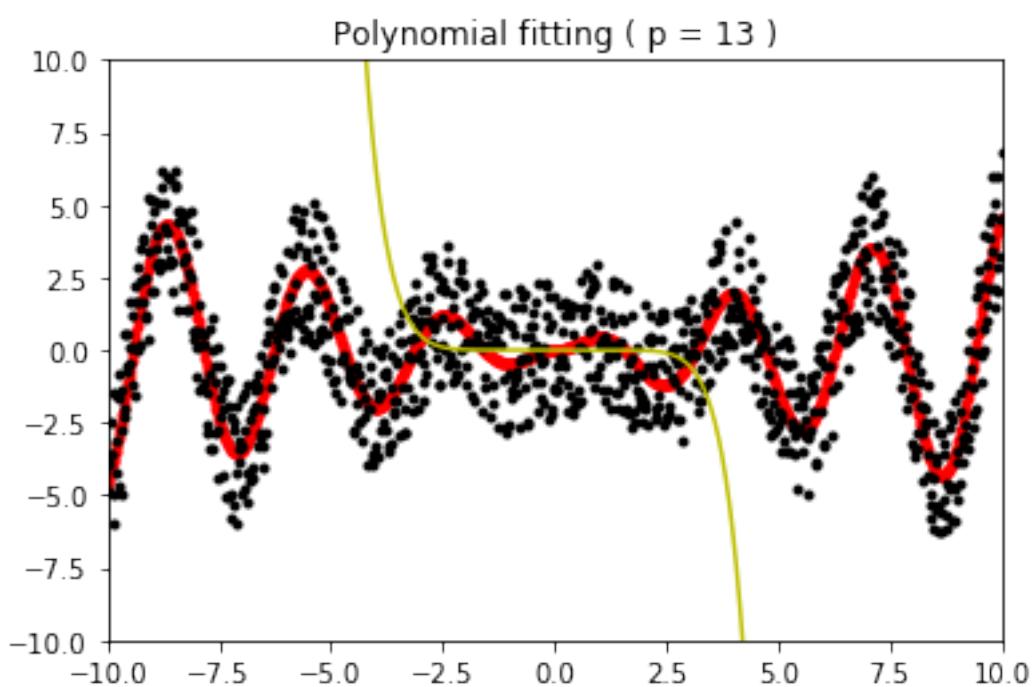
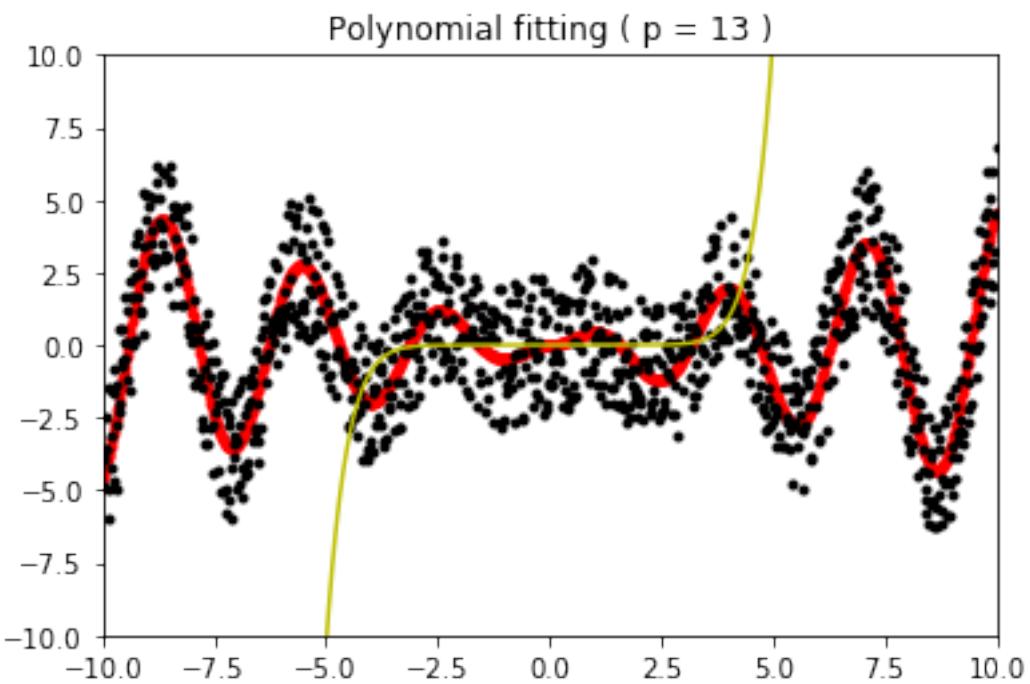


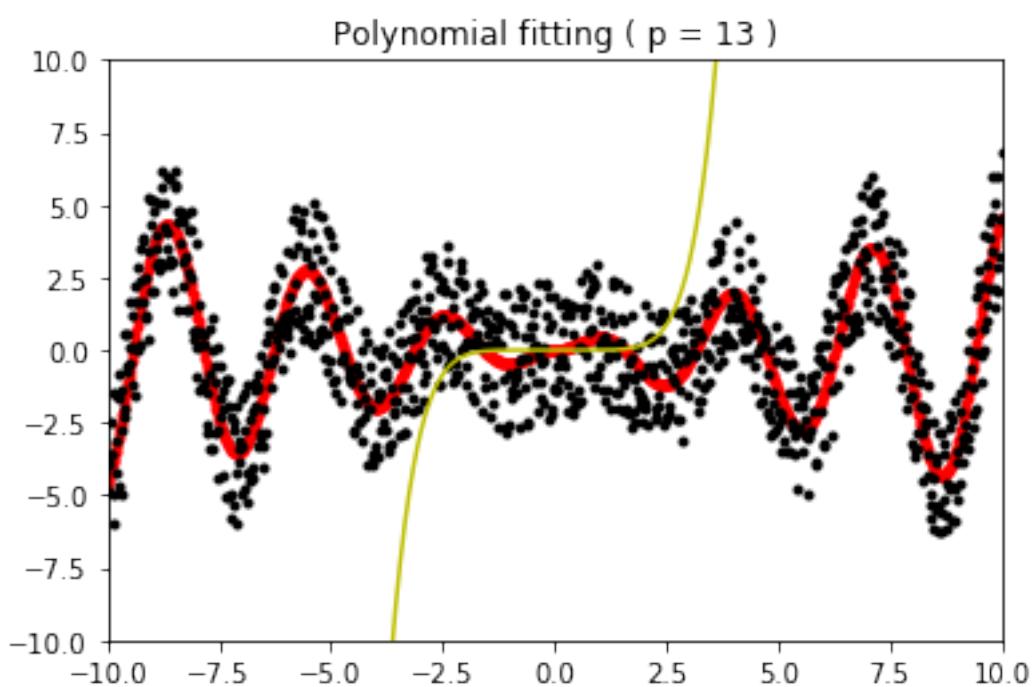
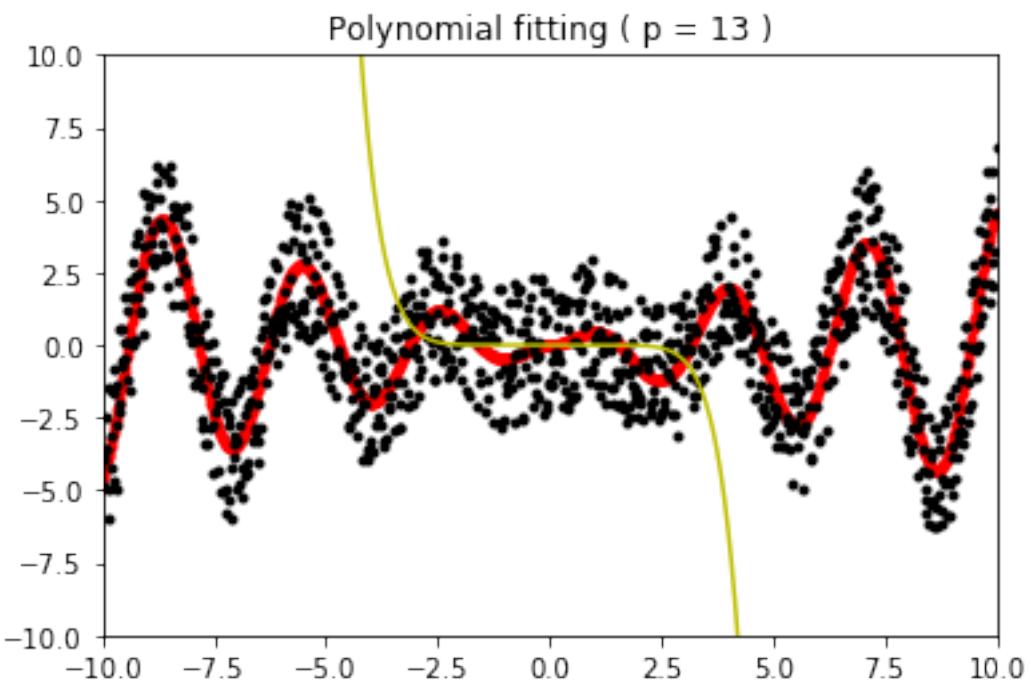
(14, 1)
[-7.38776373e-10]

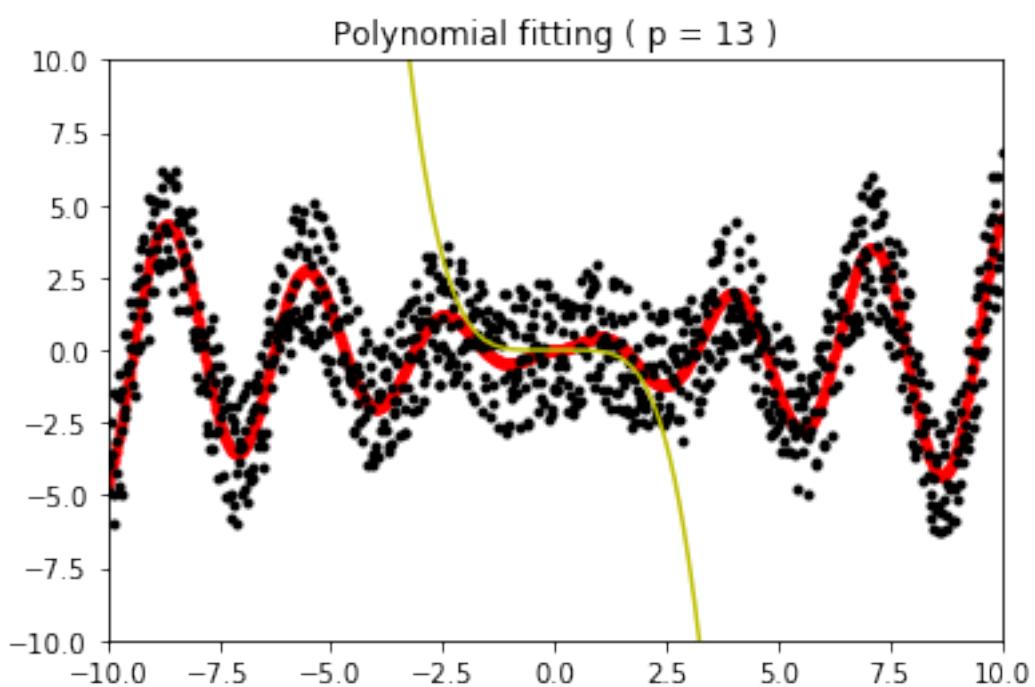
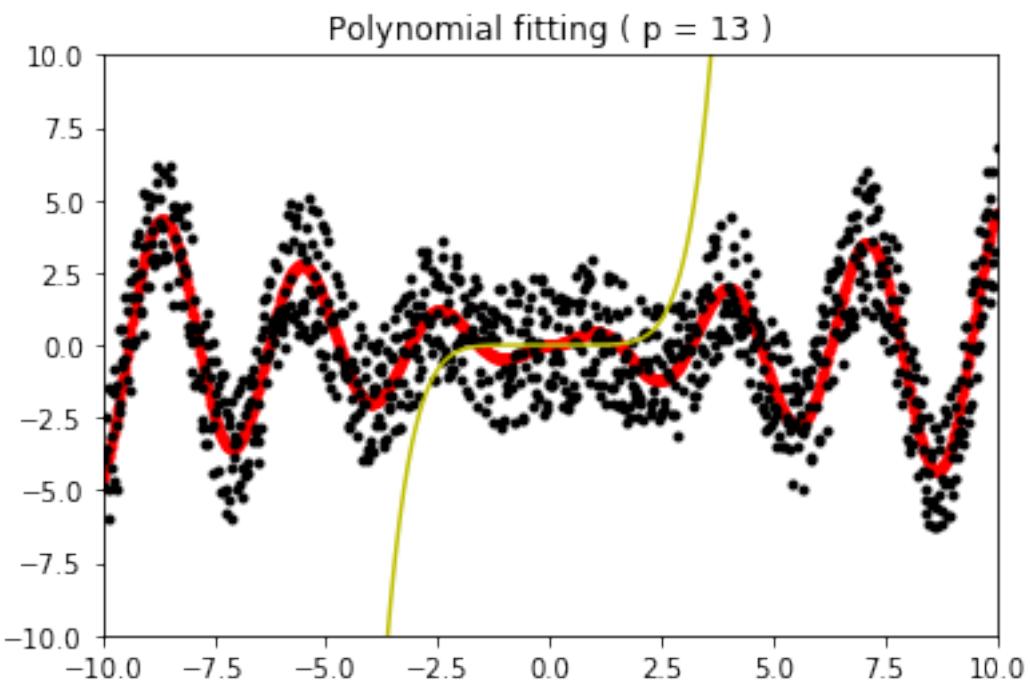
```
[-1.13731296e-17]
[ 2.32850690e-07]
[ 3.25864661e-15]
[-2.77850005e-05]
[-3.48248994e-13]
[ 1.55911184e-03]
[ 1.70363476e-11]
[-4.14439875e-02]
[-3.73346312e-10]
[ 4.57902928e-01]
[ 2.95517294e-09]
[-1.40421709e+00]
[-3.71769331e-09]]
```

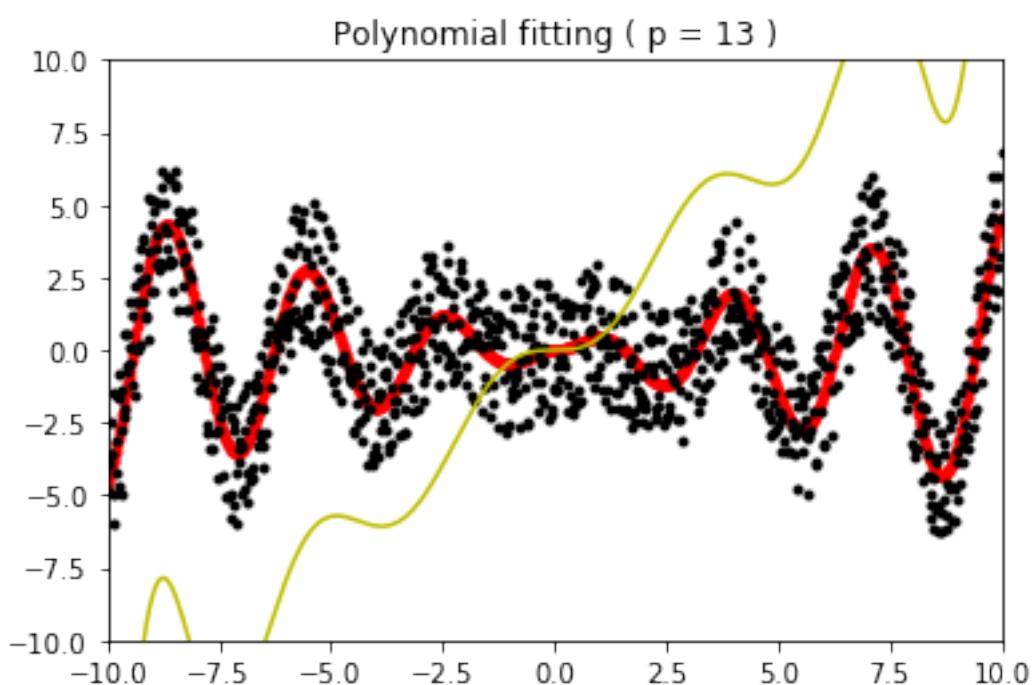
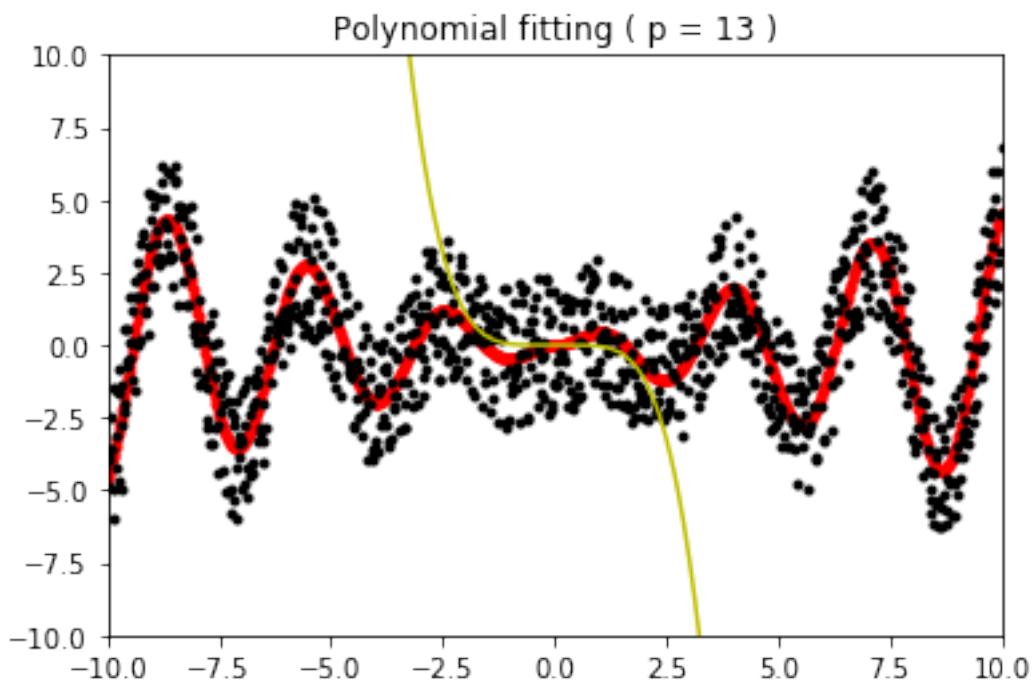


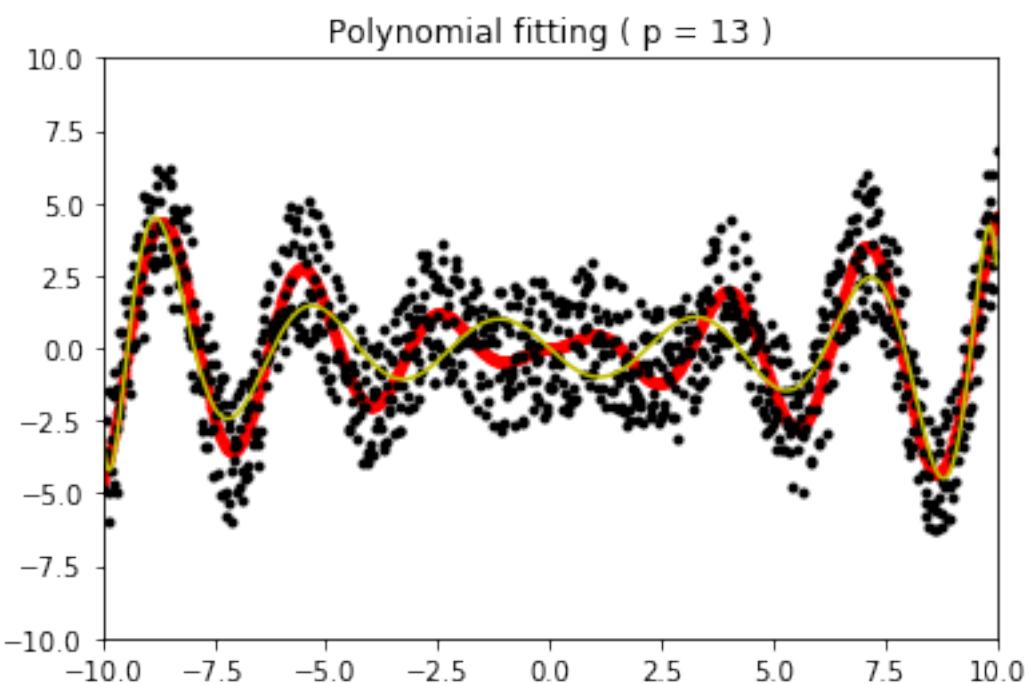
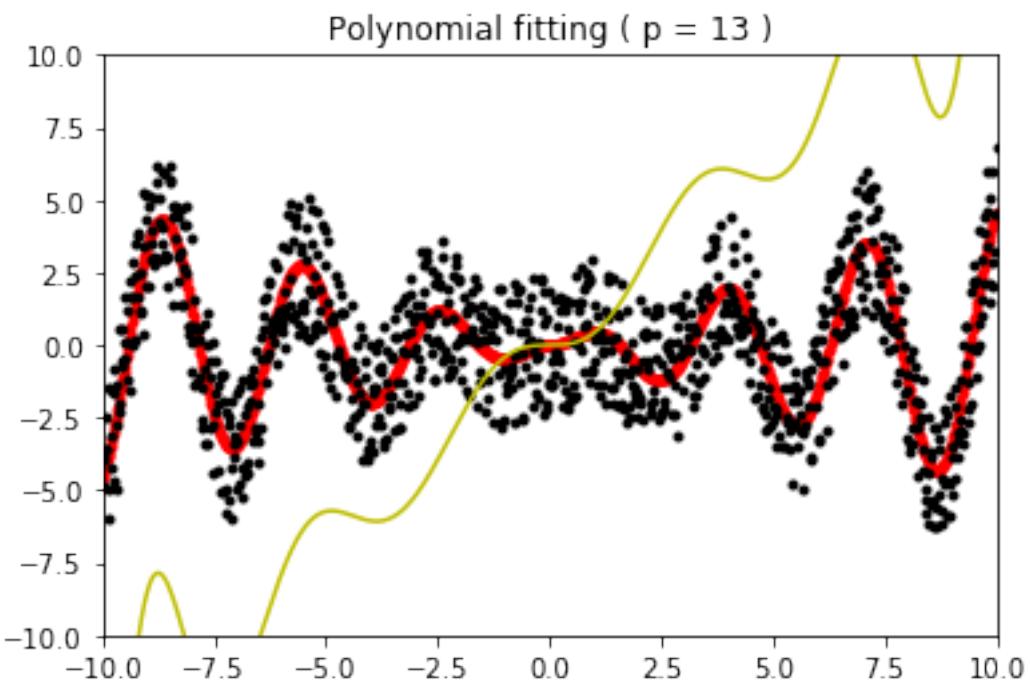


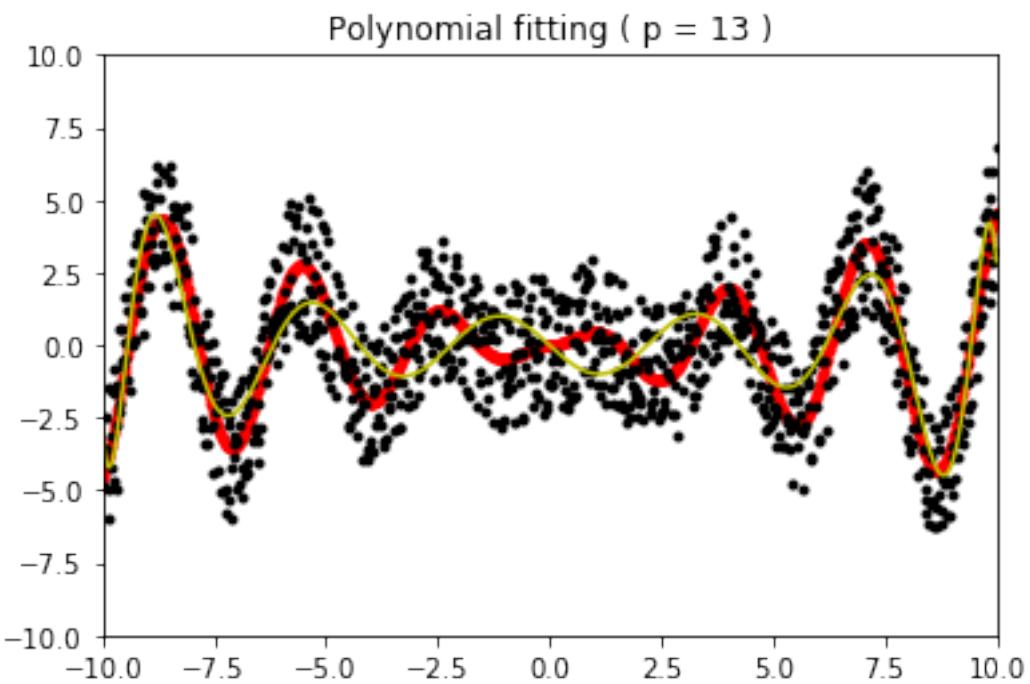




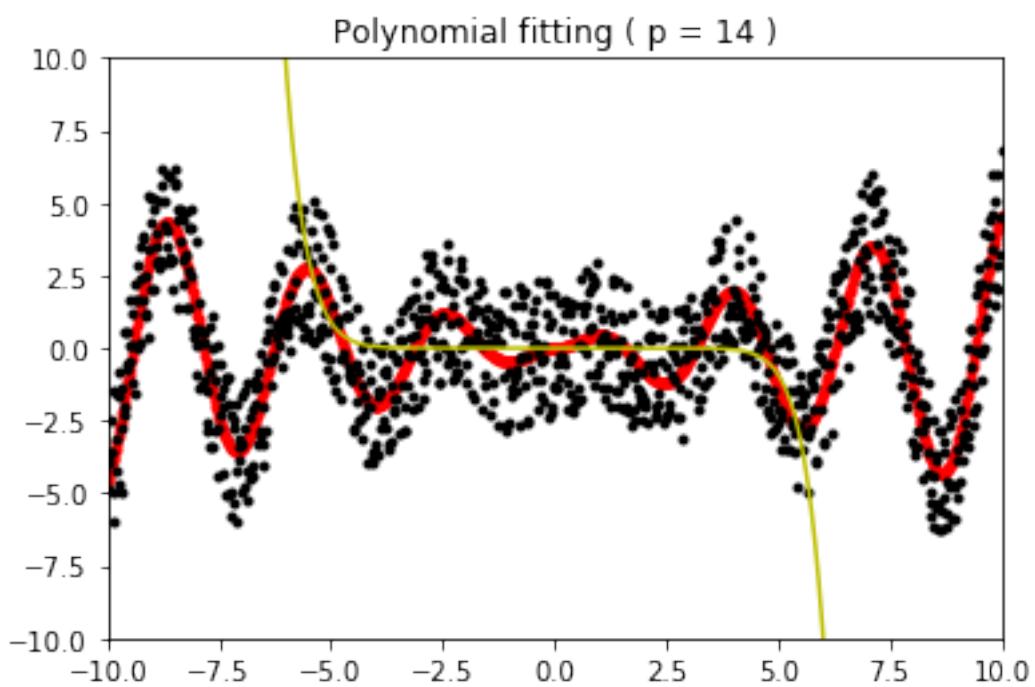
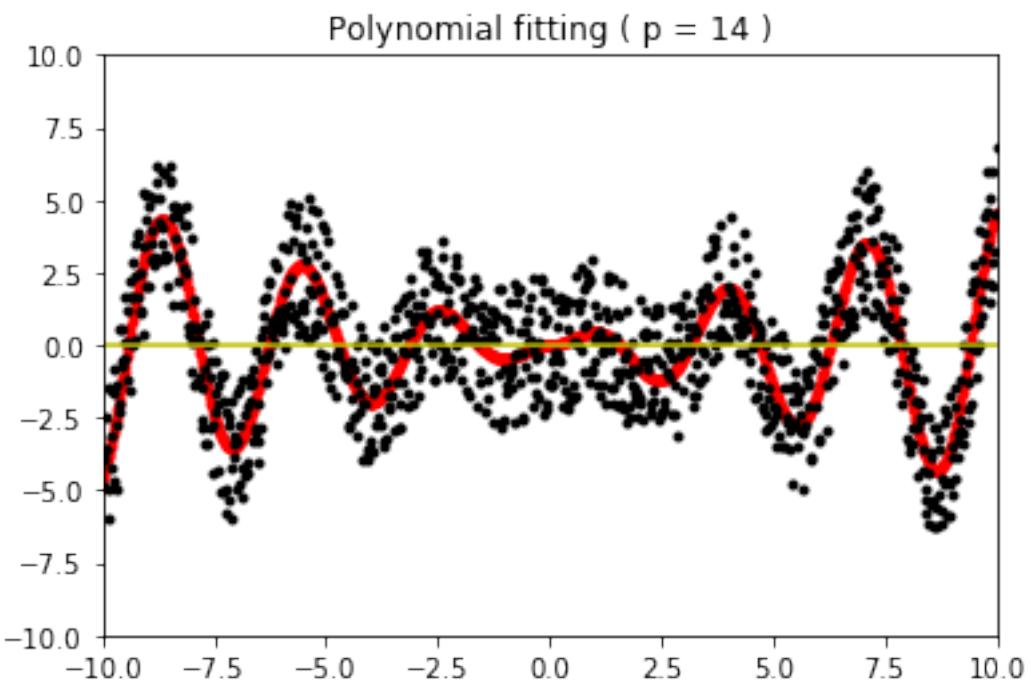


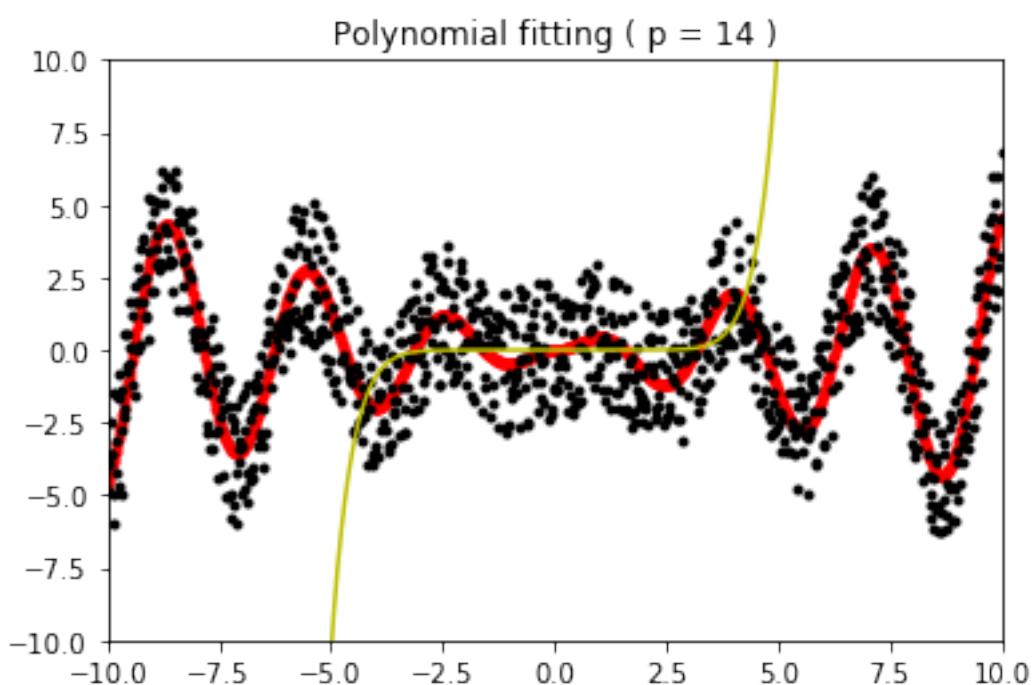
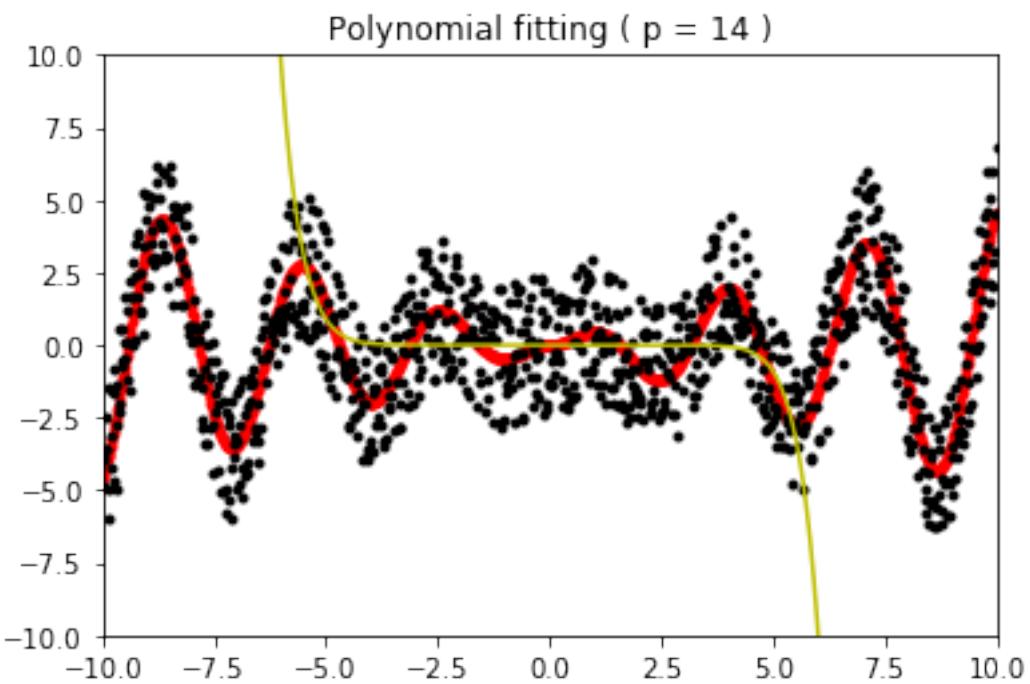


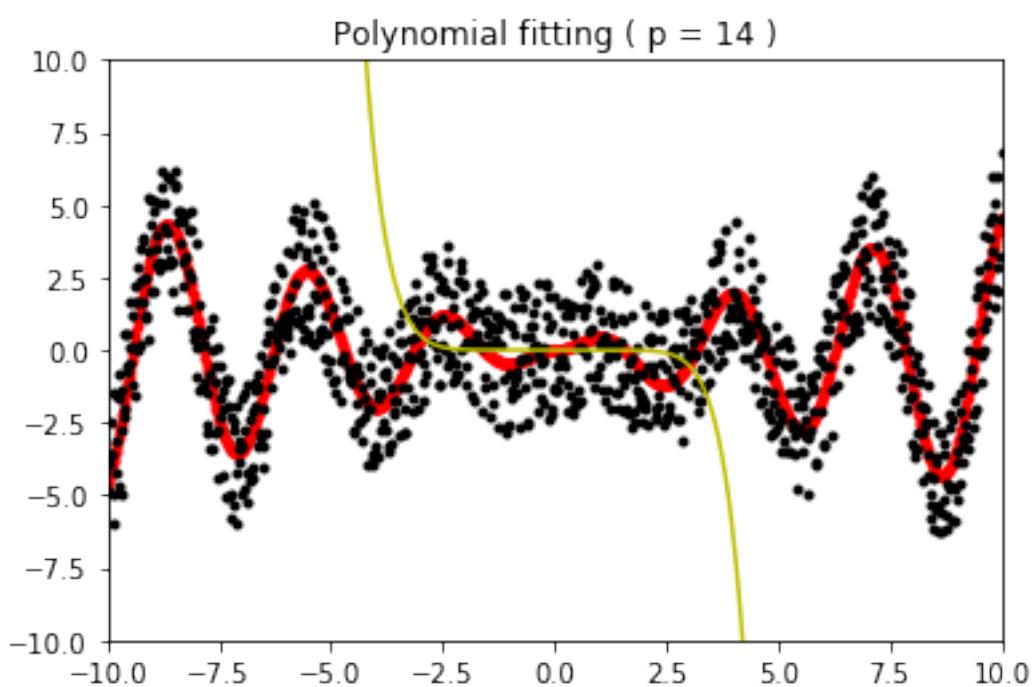
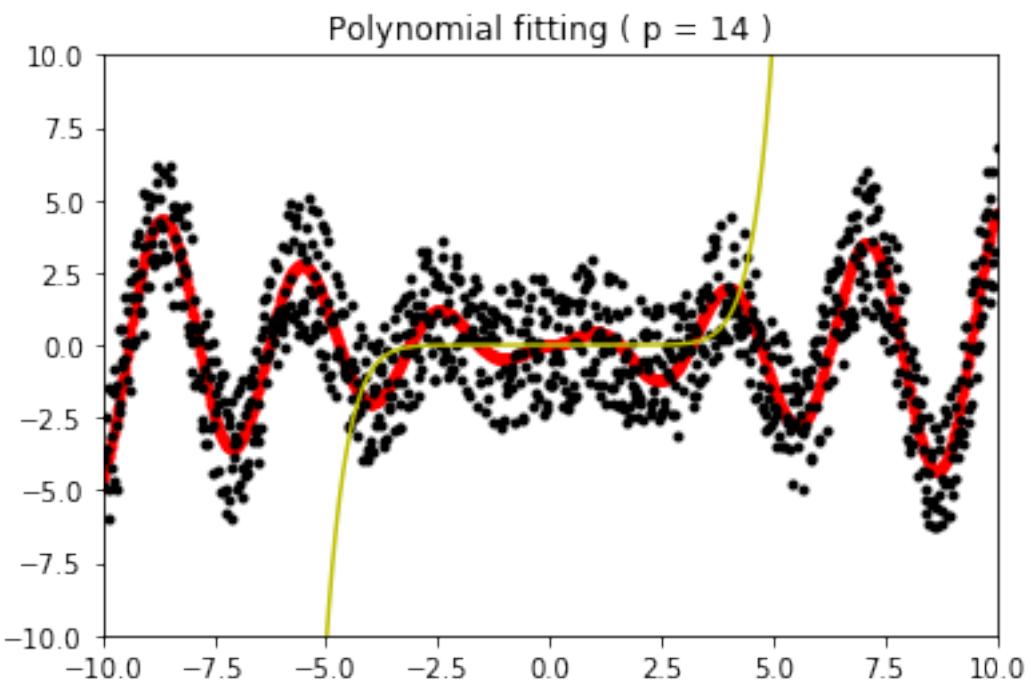


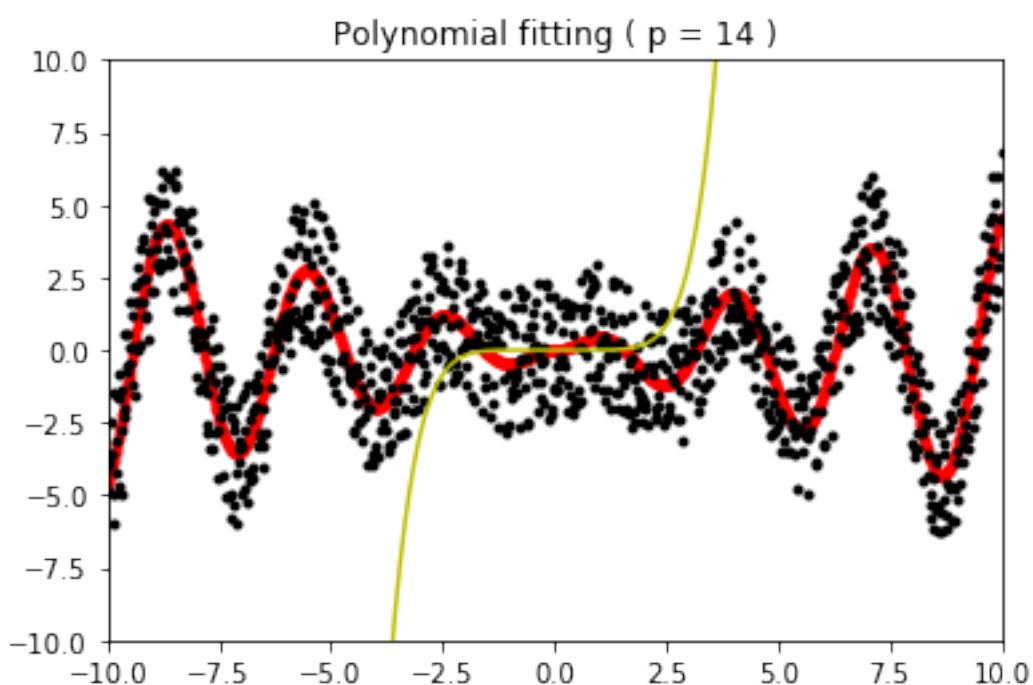
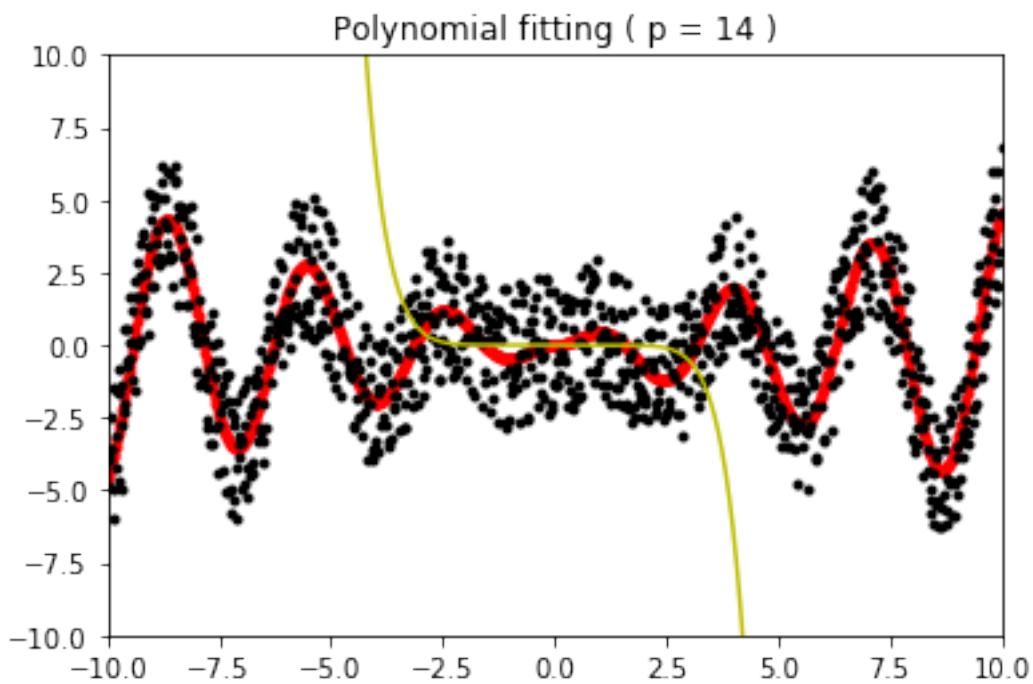


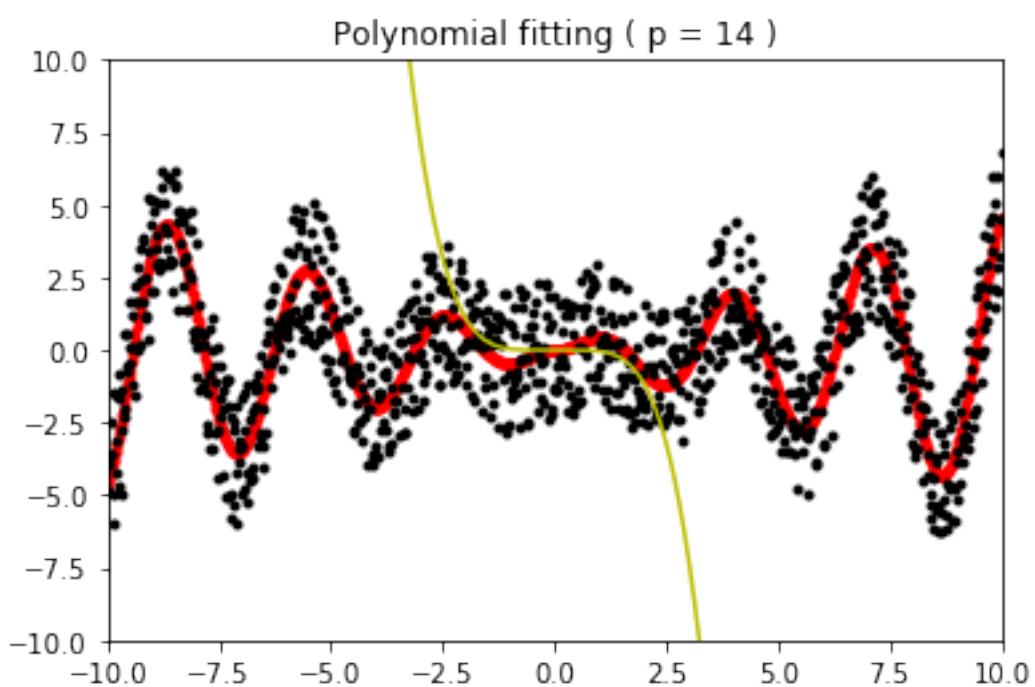
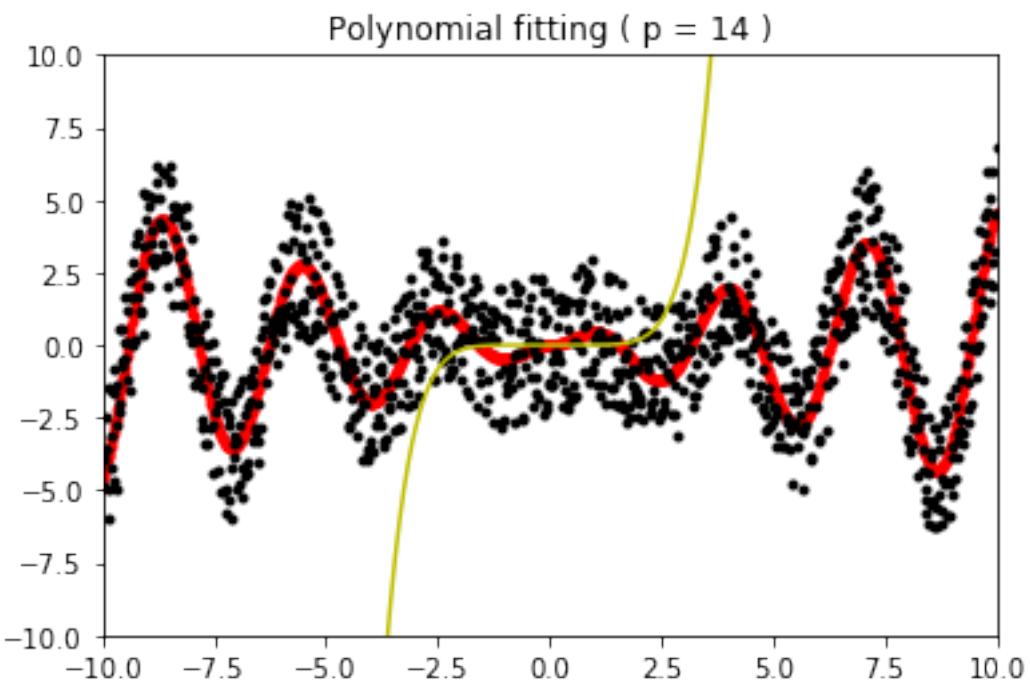
```
(15, 1)
[[ 5.78679979e-18]
 [-7.38776373e-10]
 [-1.96552829e-15]
 [ 2.32850690e-07]
 [ 2.61707268e-13]
 [-2.77850005e-05]
 [-1.72363041e-11]
 [ 1.55911184e-03]
 [ 5.81061187e-10]
 [-4.14439875e-02]
 [-9.29627137e-09]
 [ 4.57902928e-01]
 [ 5.55445945e-08]
 [-1.40421709e+00]
 [-5.38998471e-08]]
```

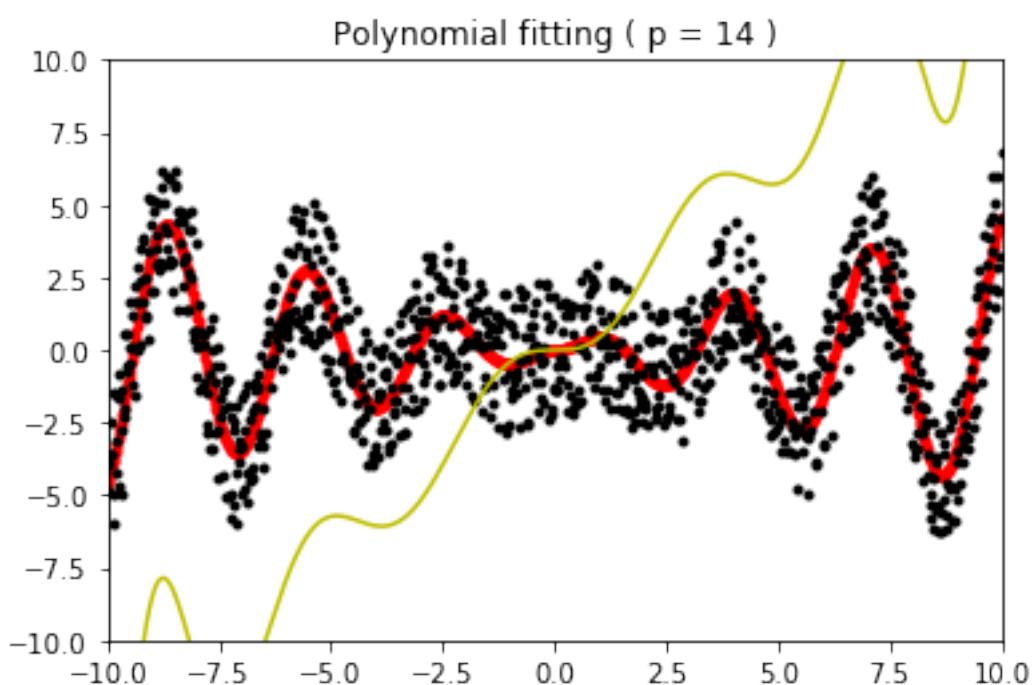
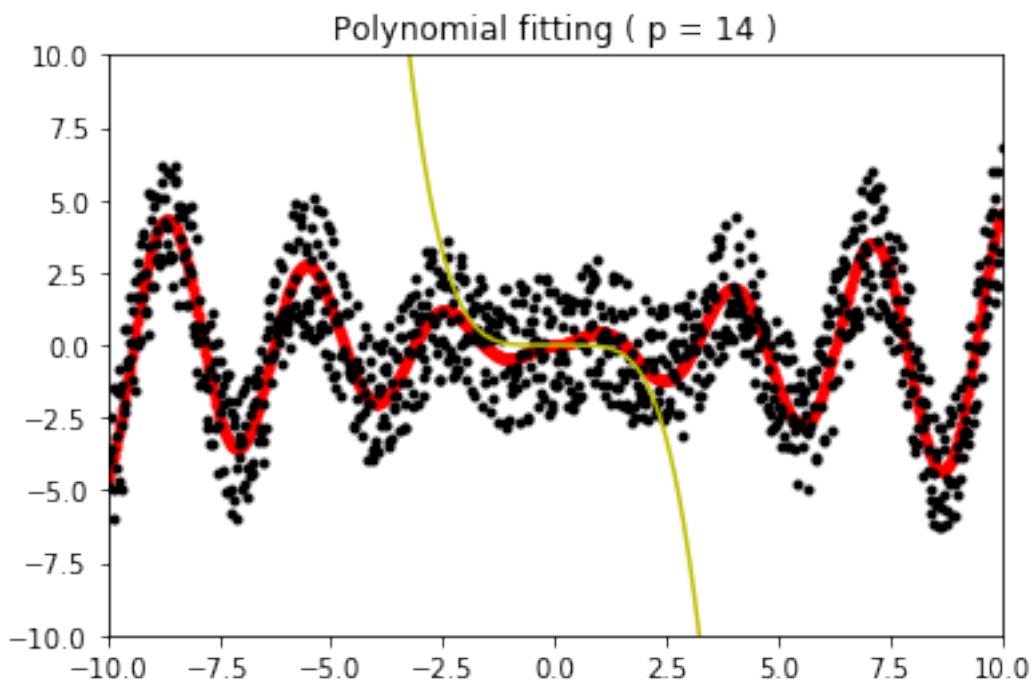


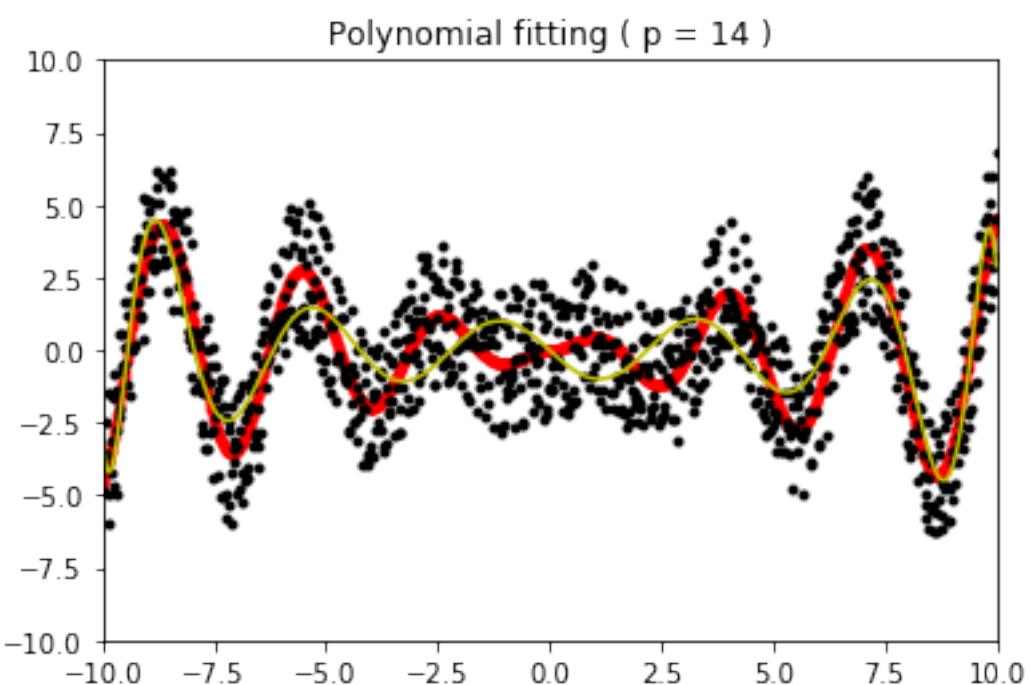
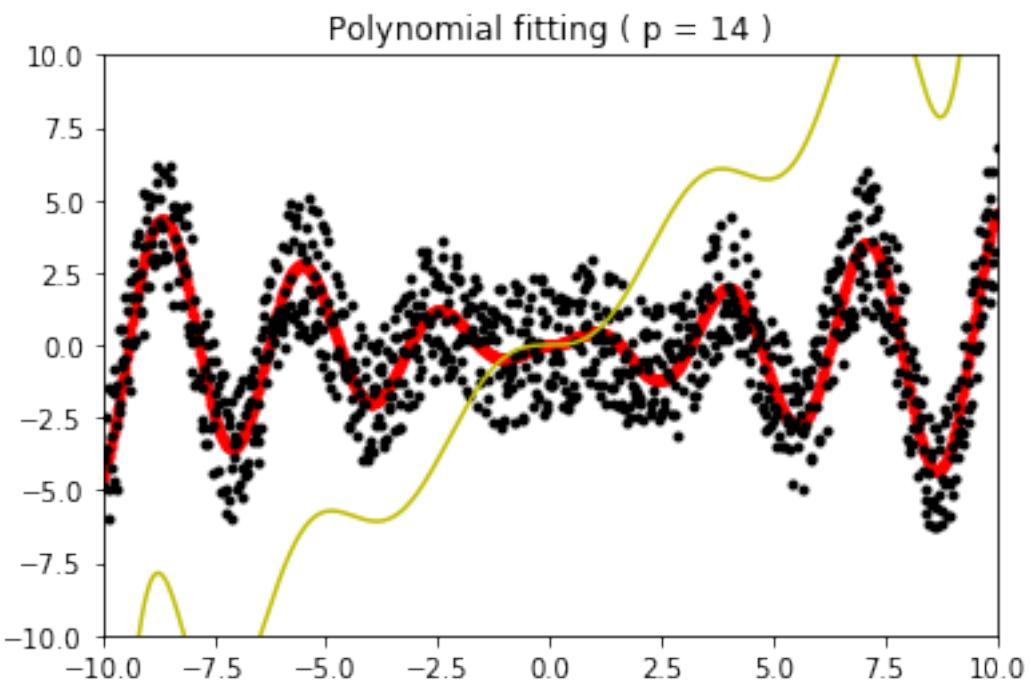


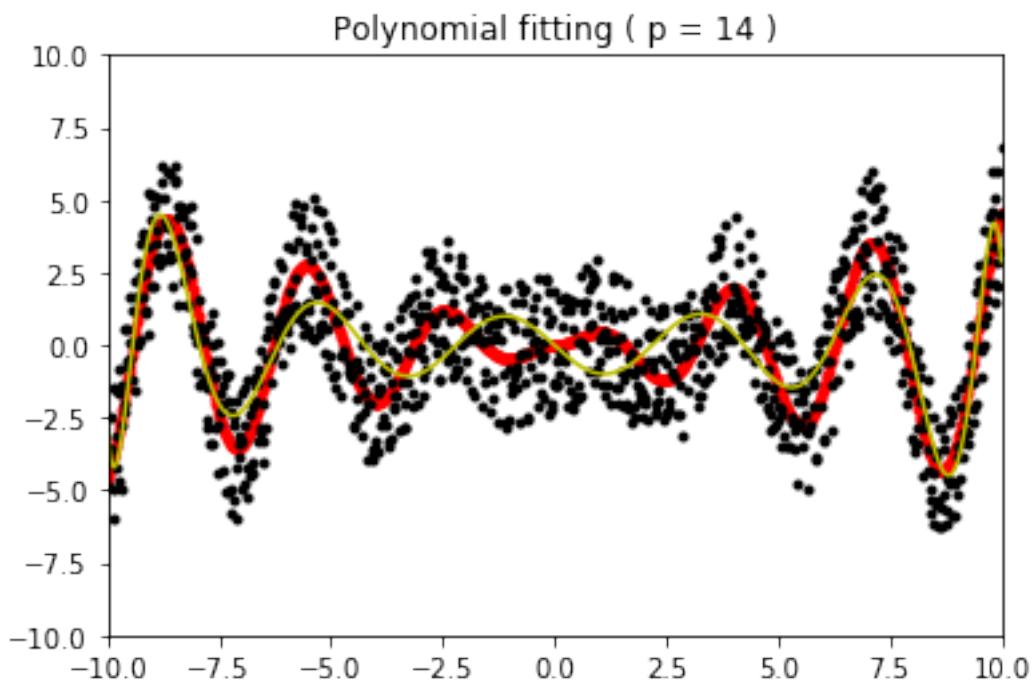




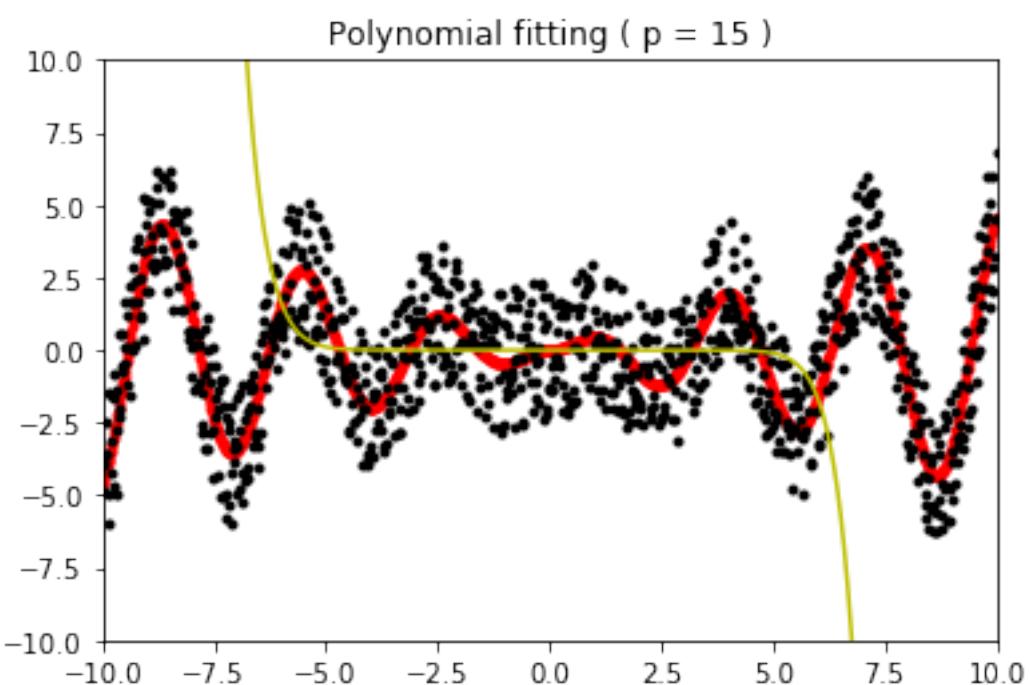
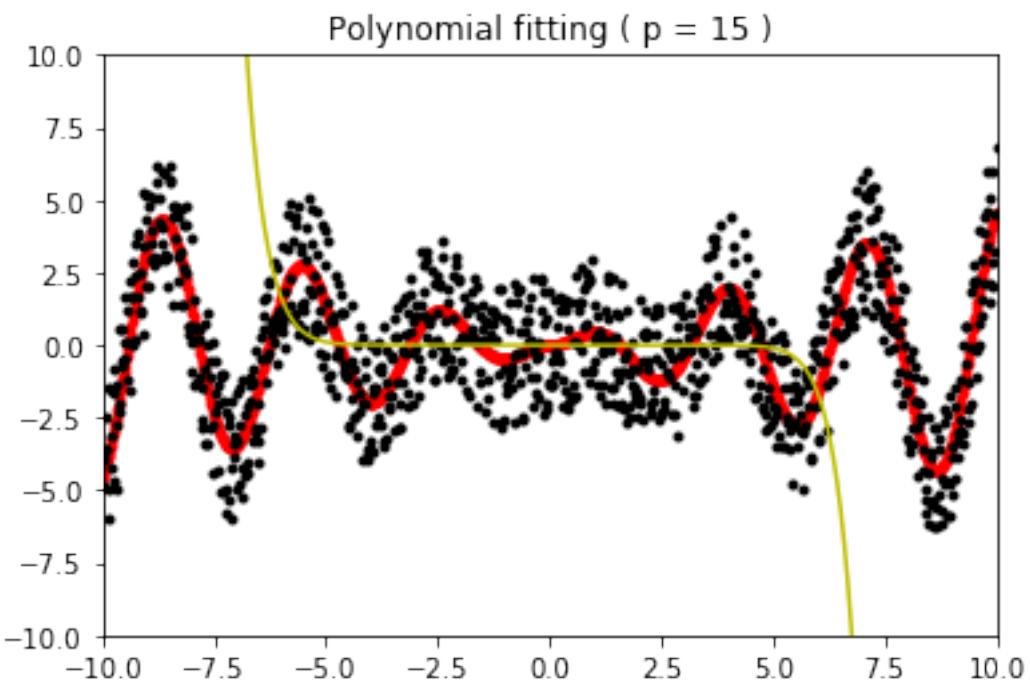


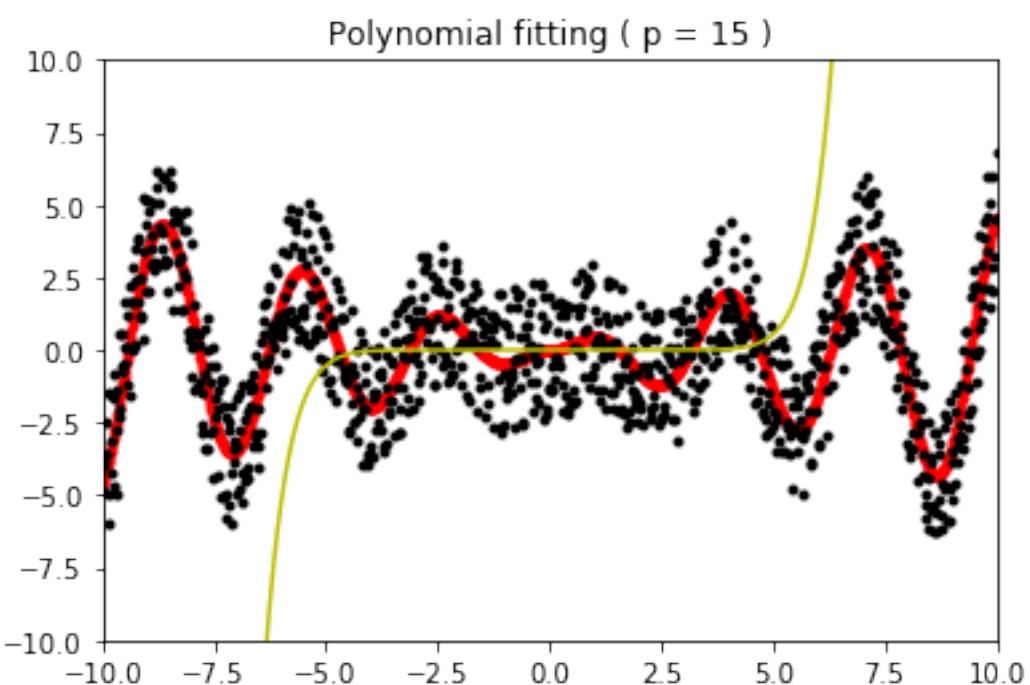
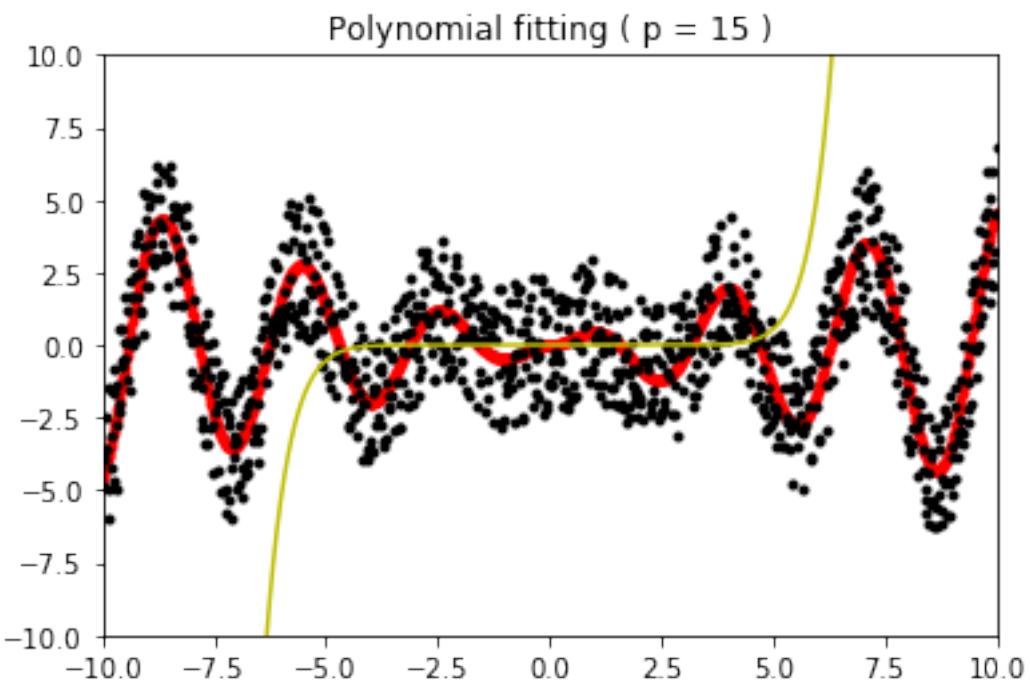


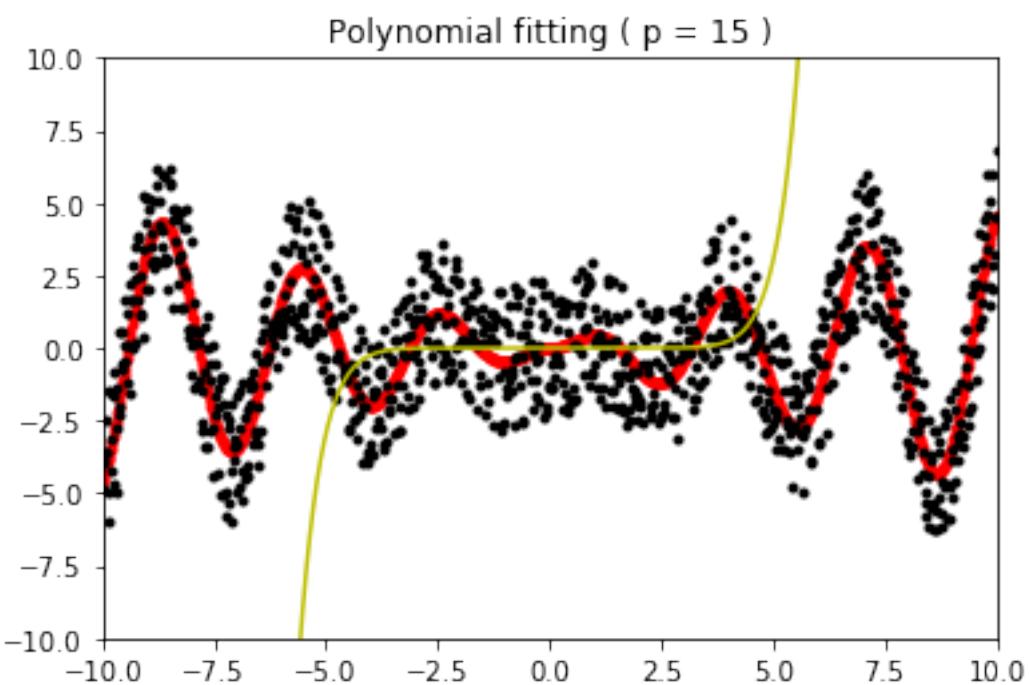
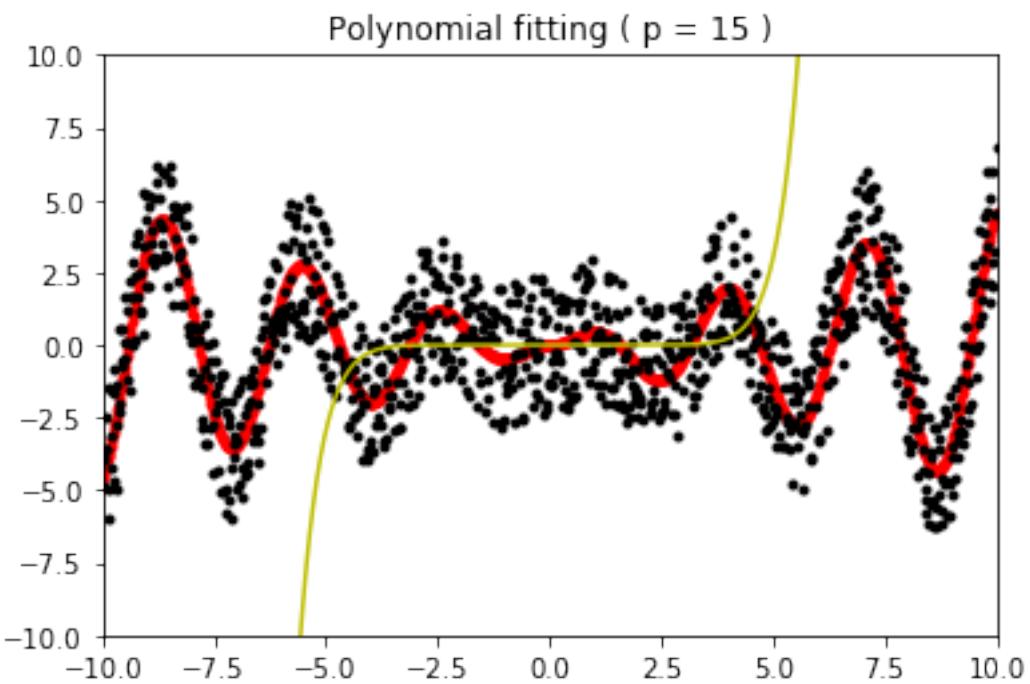


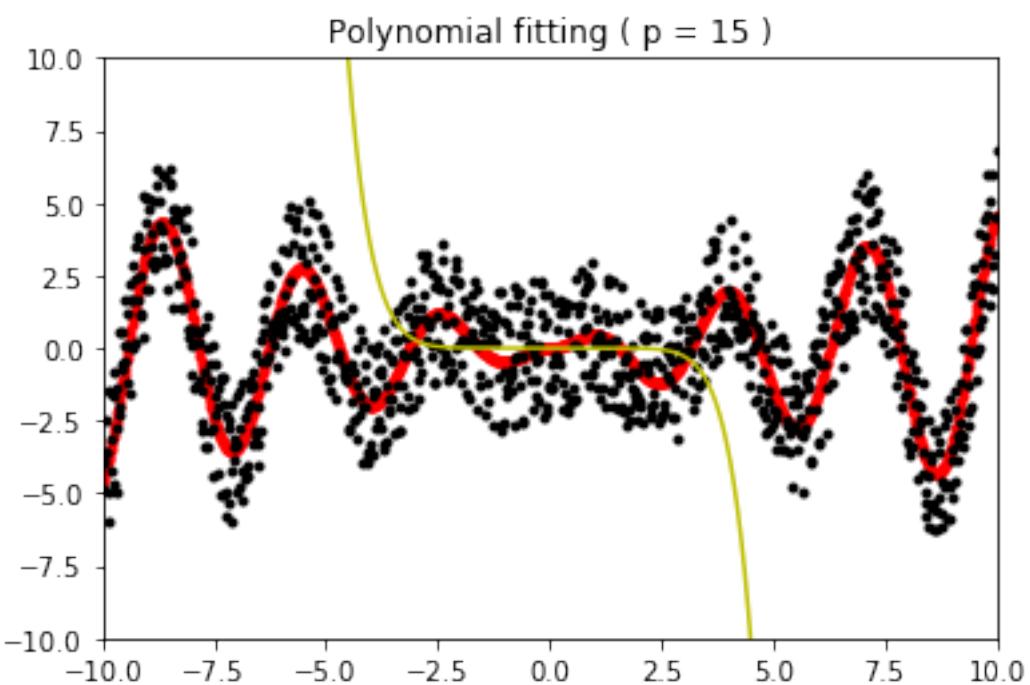
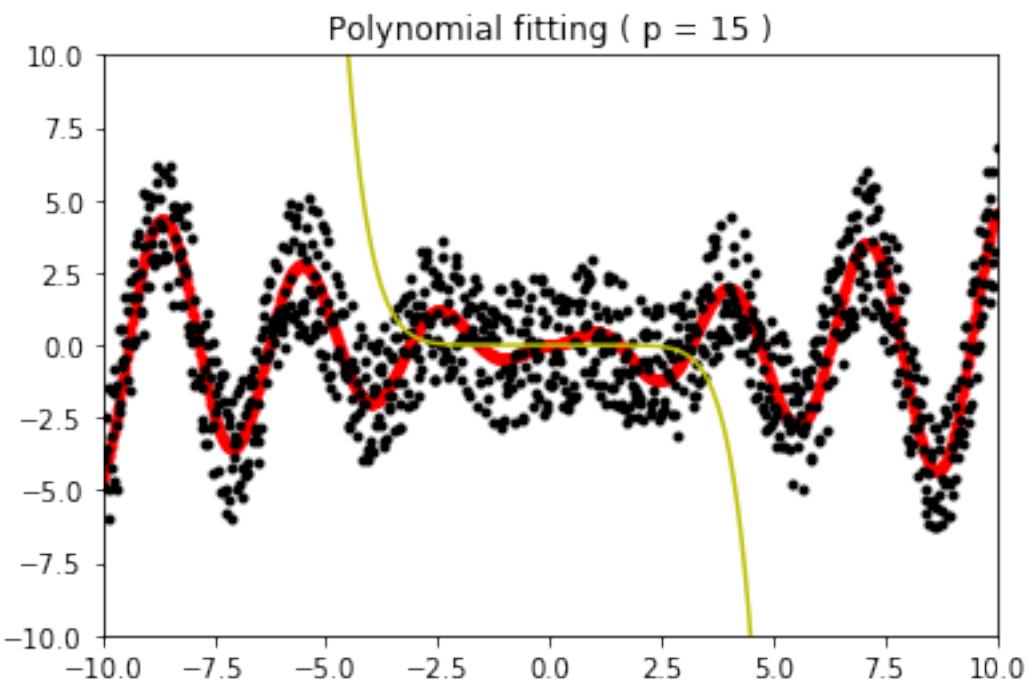


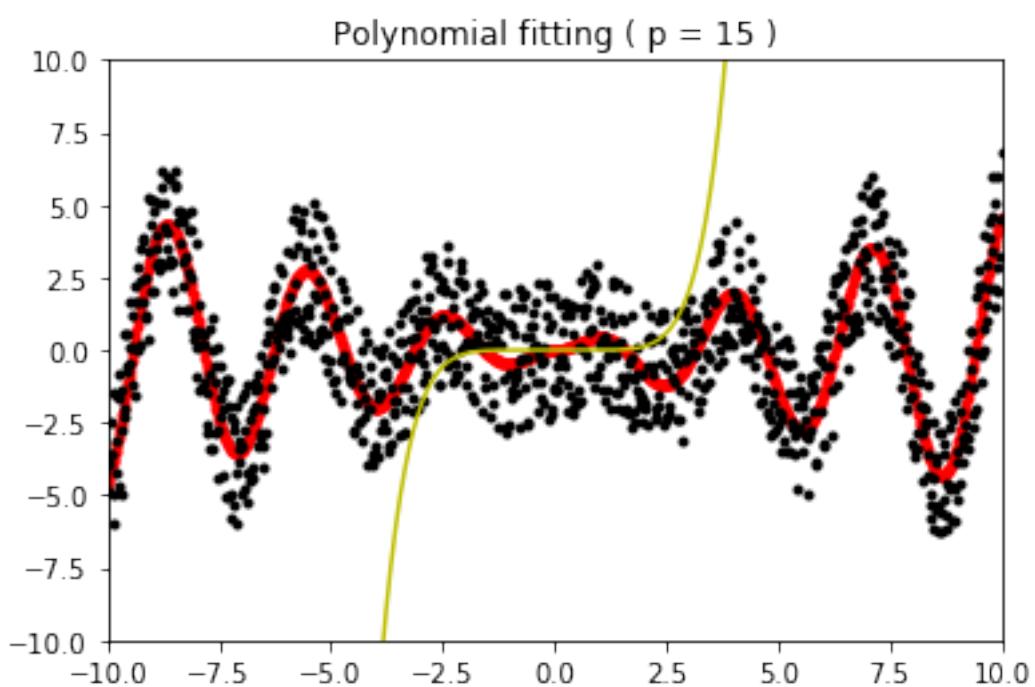
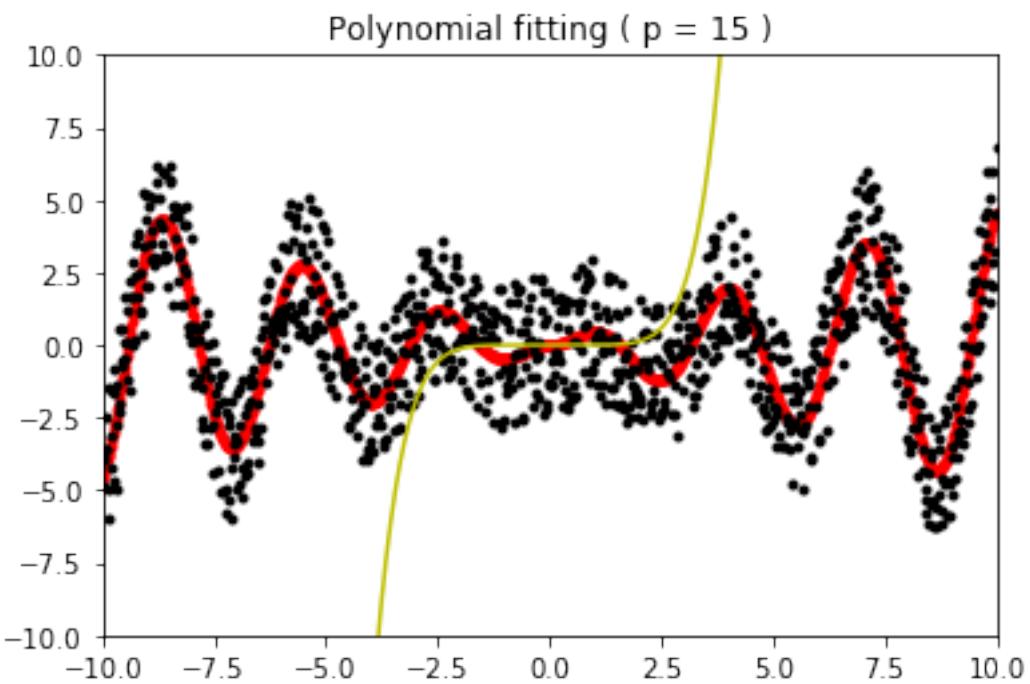
```
(16, 1)
[[-3.48975727e-12]
 [ 1.17987875e-18]
 [ 5.27198236e-10]
 [-3.94125584e-16]
 [ 4.96345487e-08]
 [ 5.14195499e-14]
 [-1.43232210e-05]
 [-3.30155030e-12]
 [ 1.03133069e-03]
 [ 1.07725347e-10]
 [-3.08680220e-02]
 [-1.65006407e-09]
 [ 3.64952639e-01]
 [ 9.27740609e-09]
 [-1.16943767e+00]
 [-8.21643290e-09]]
```

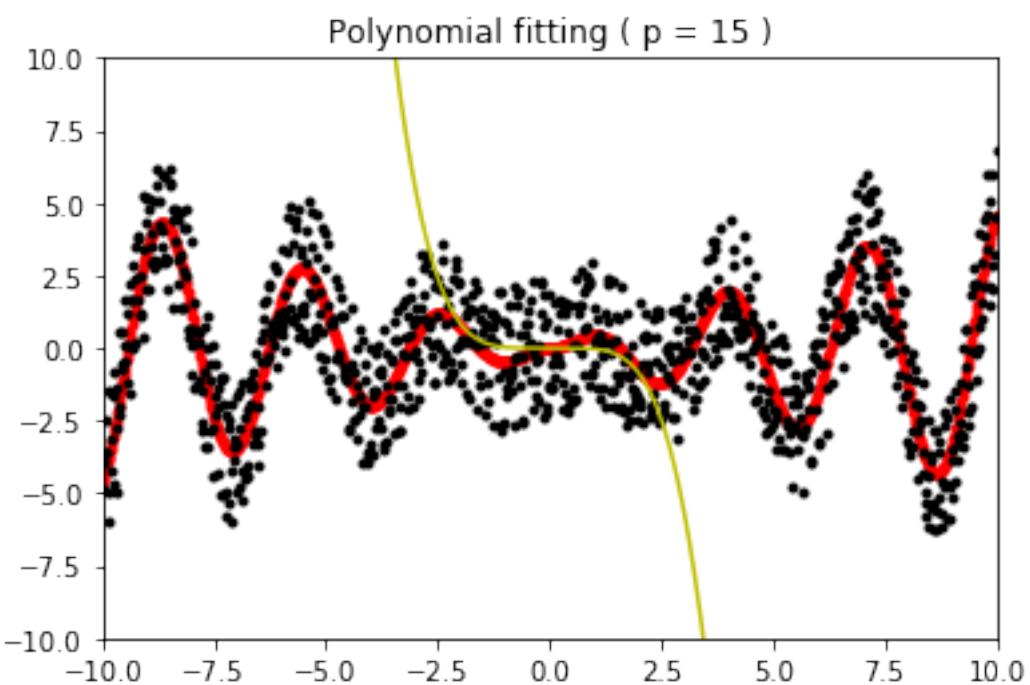
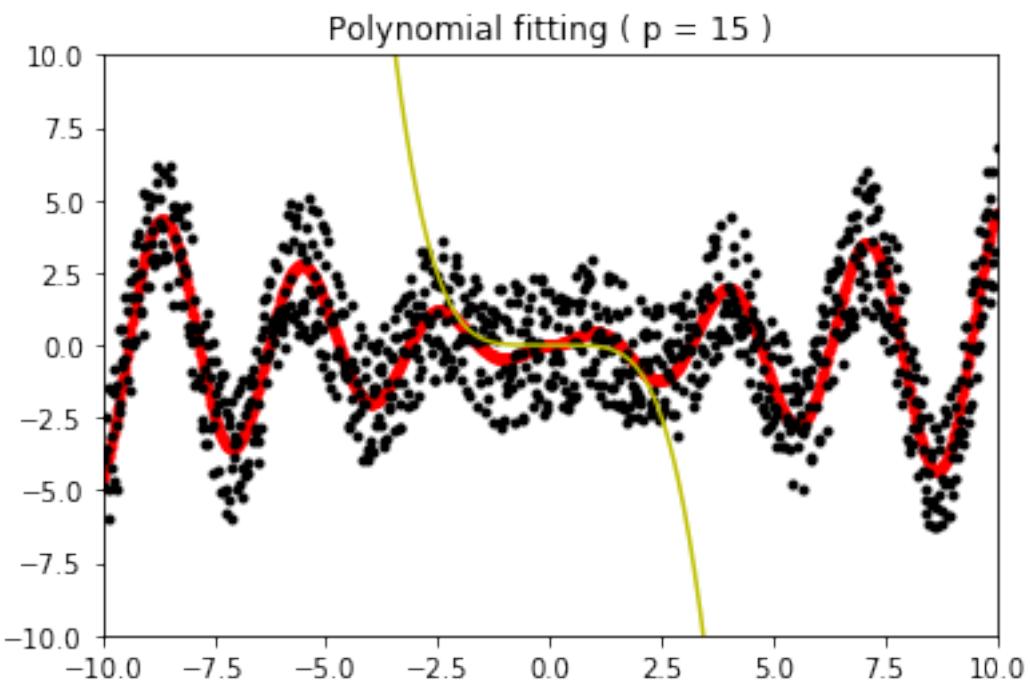


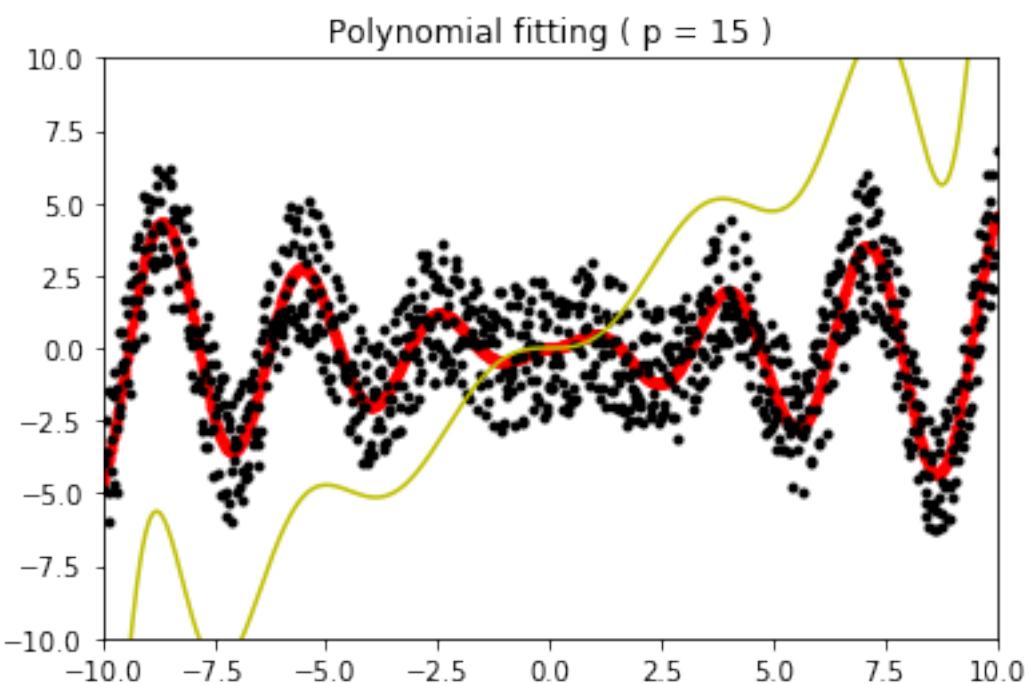
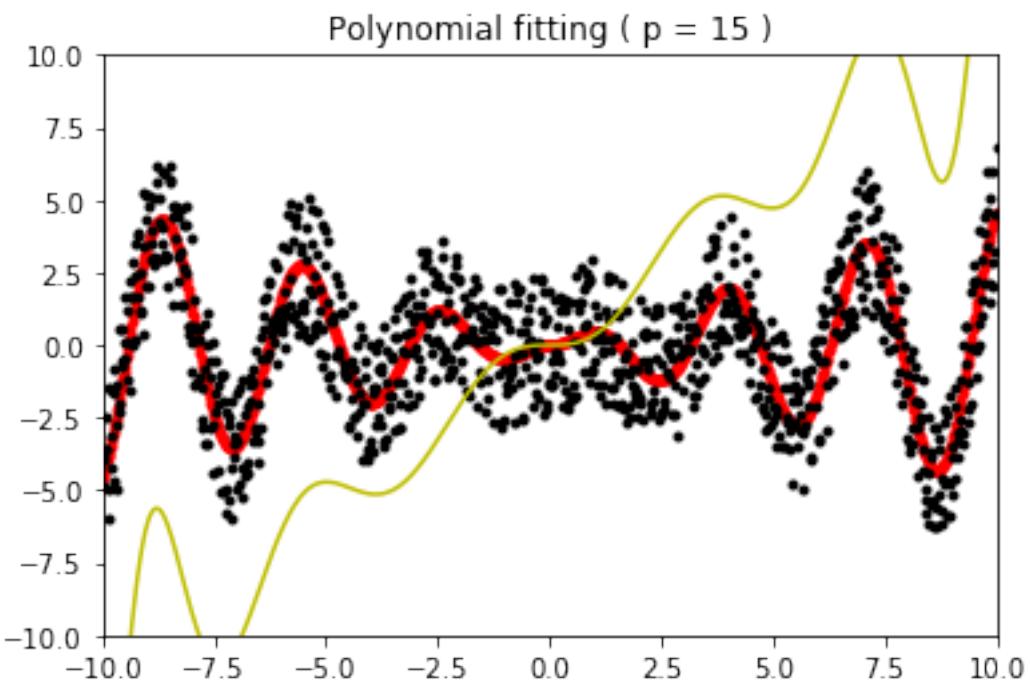


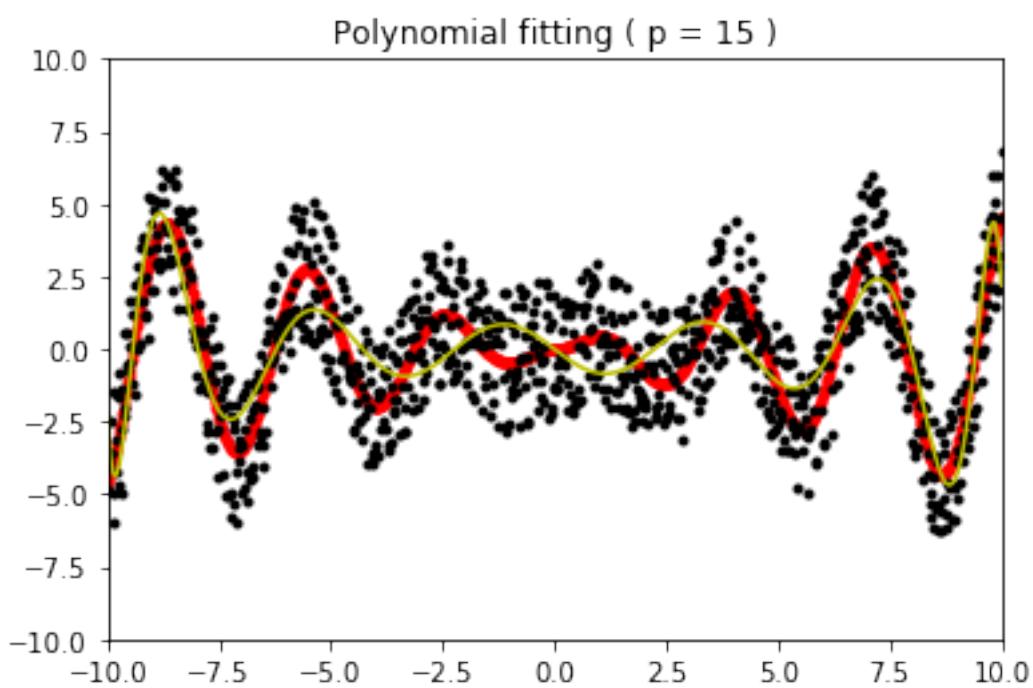
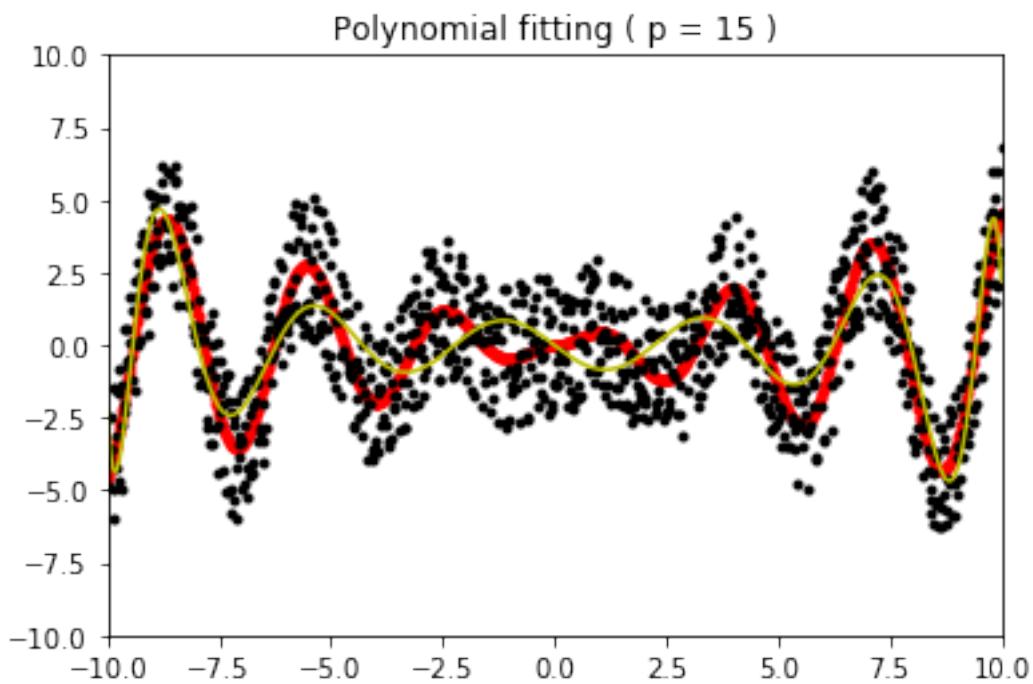






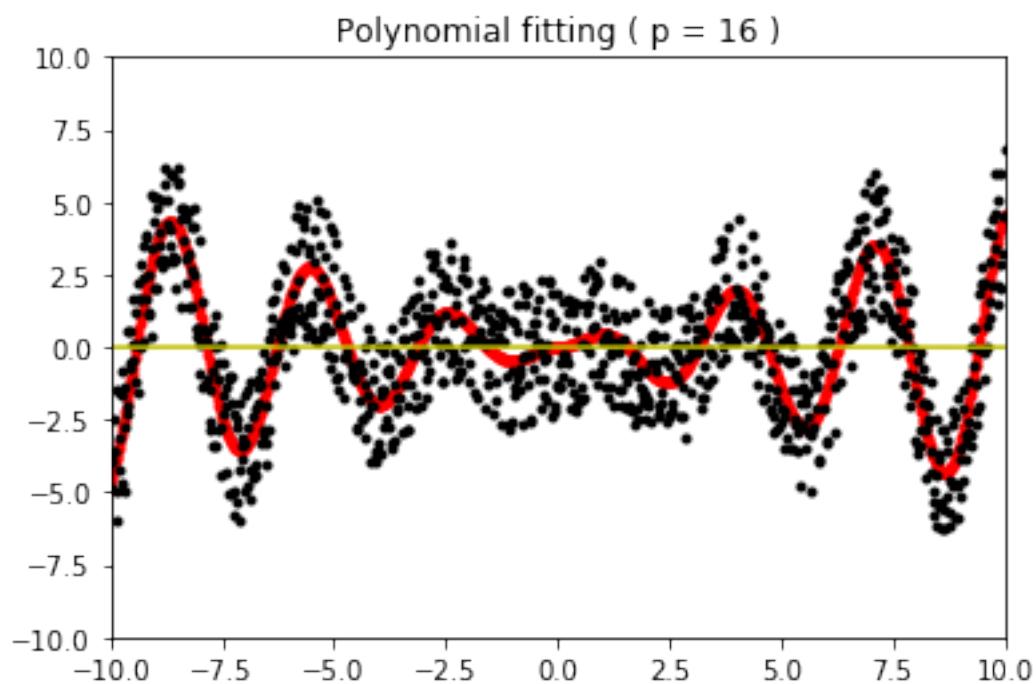


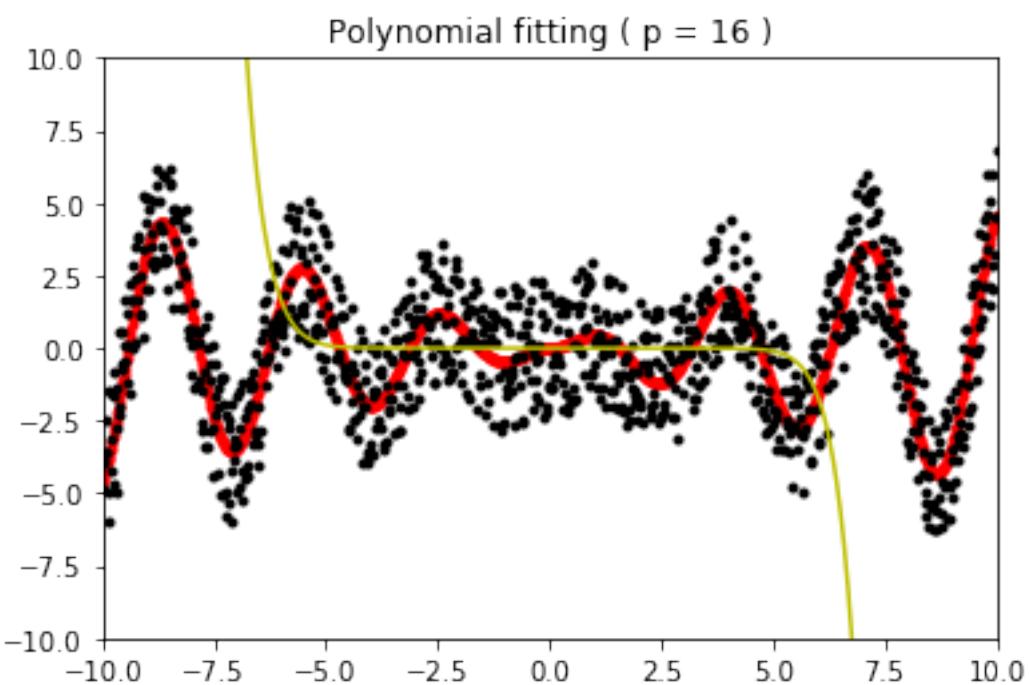
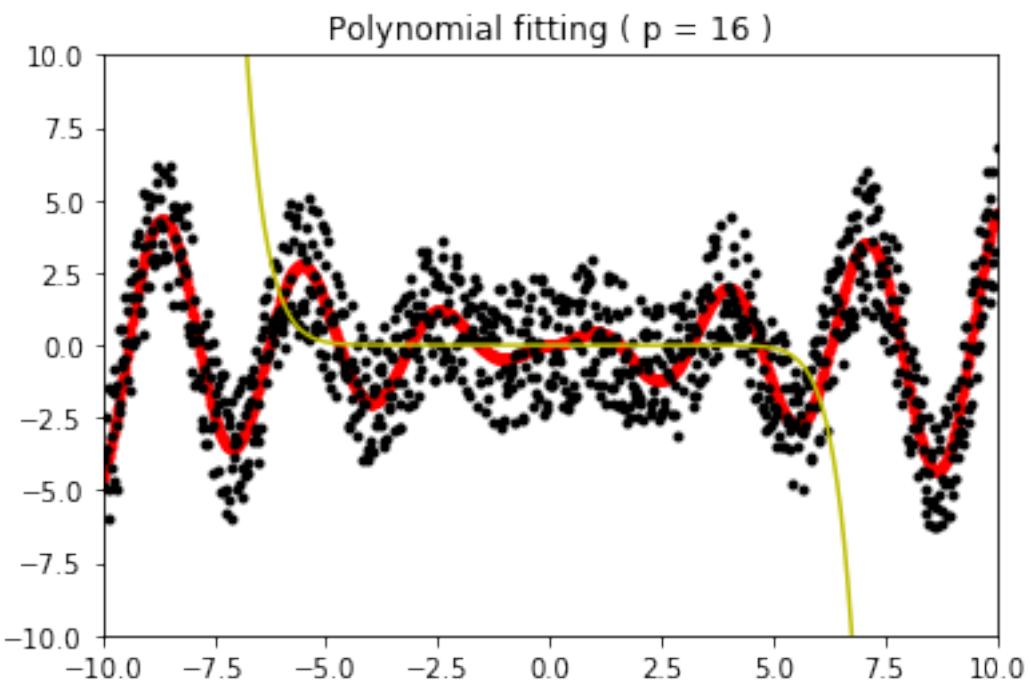


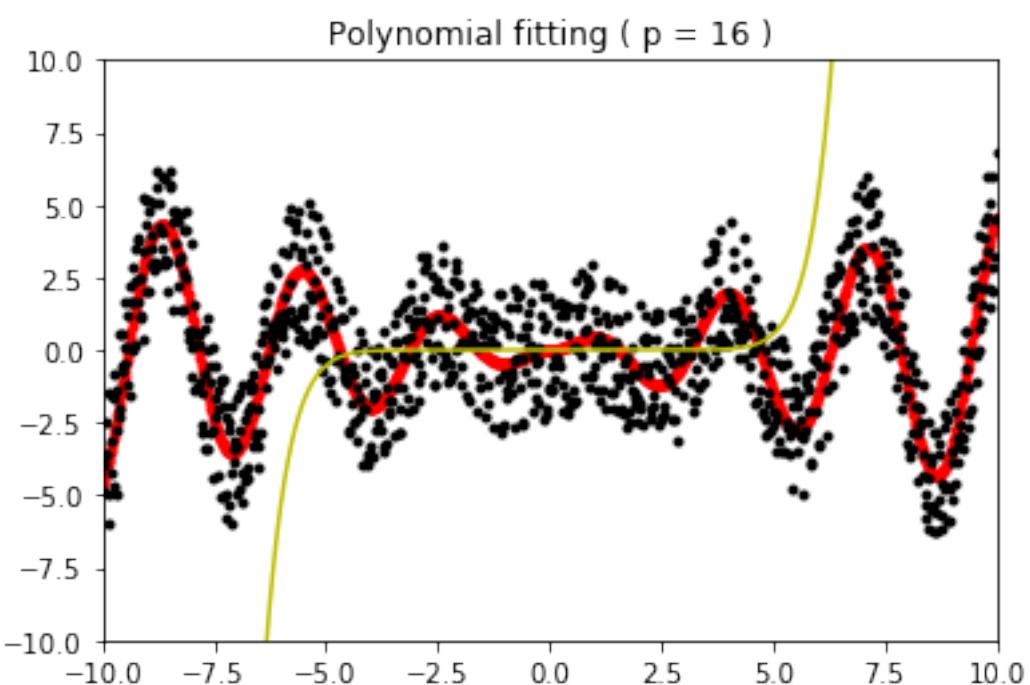
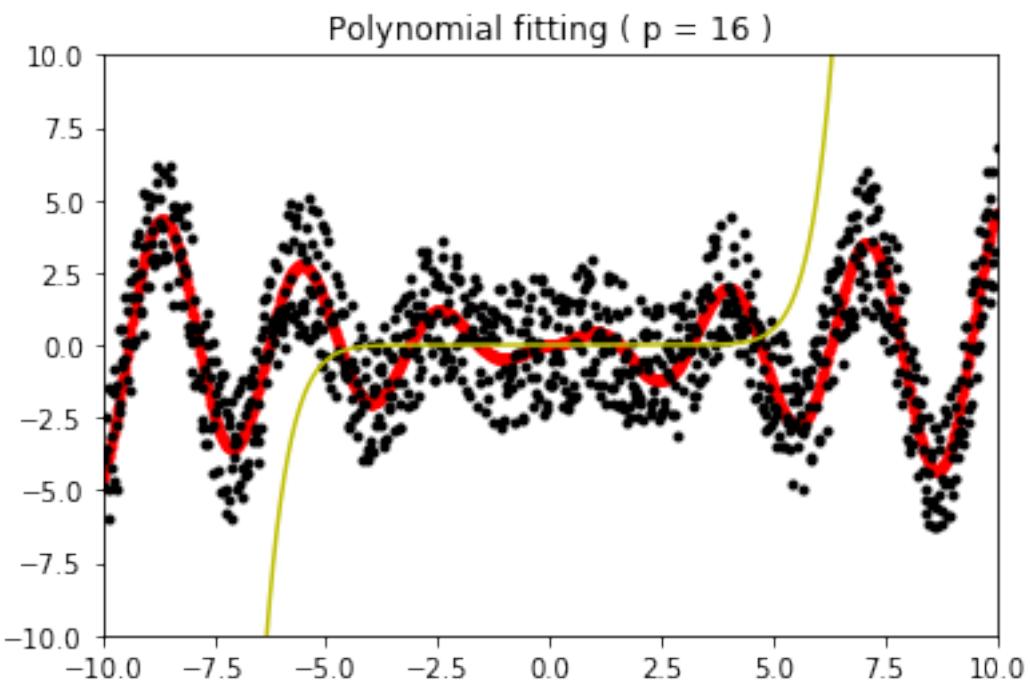


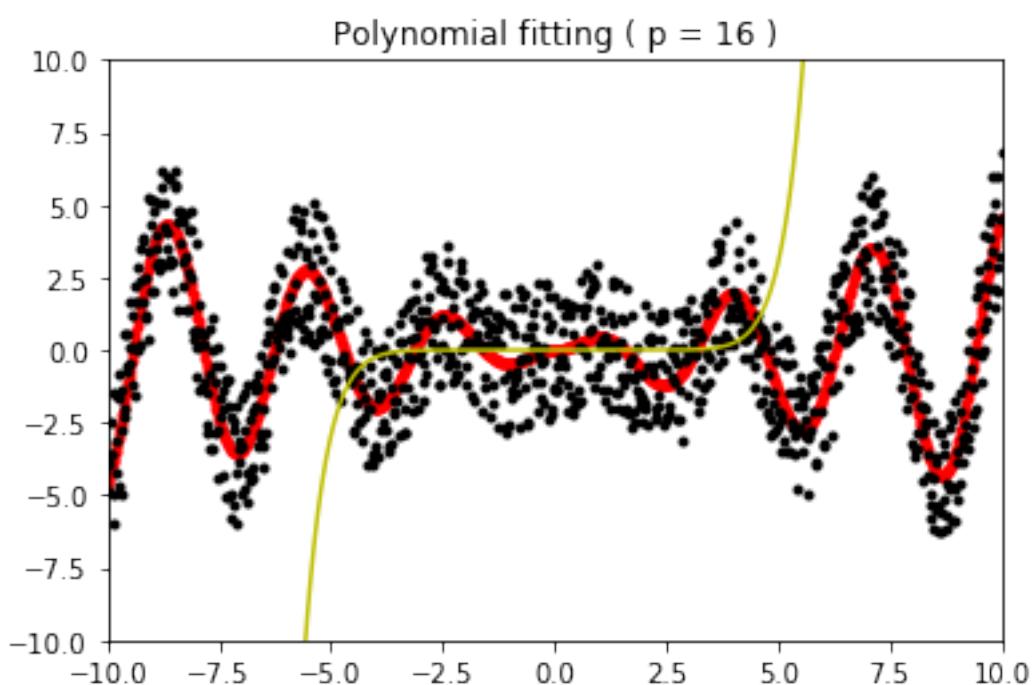
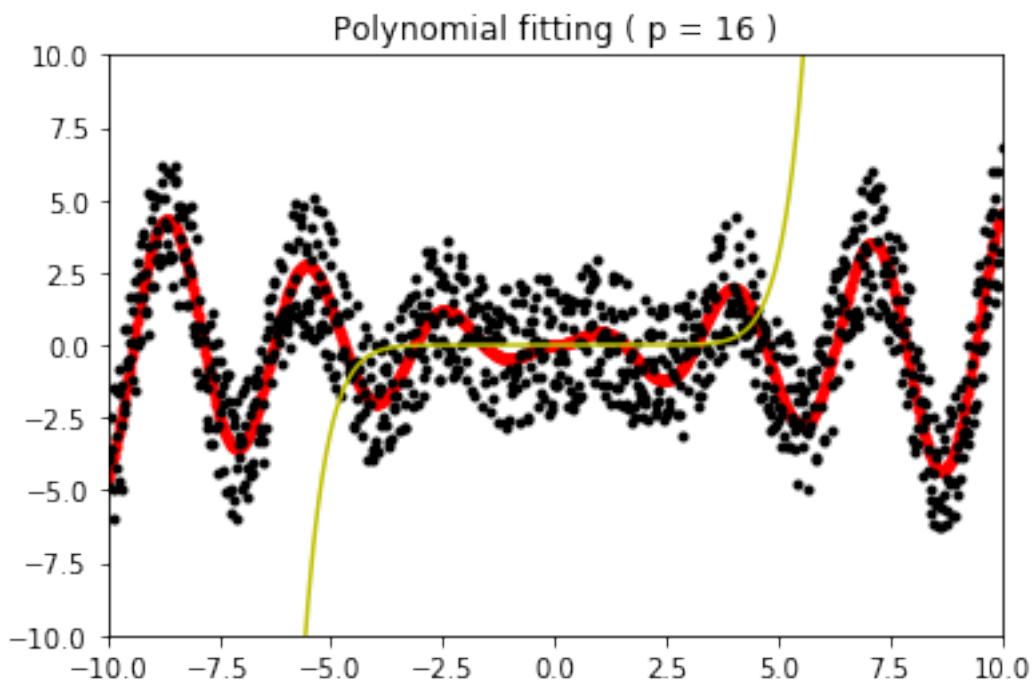
(17, 1)
[-6.43770172e-19]

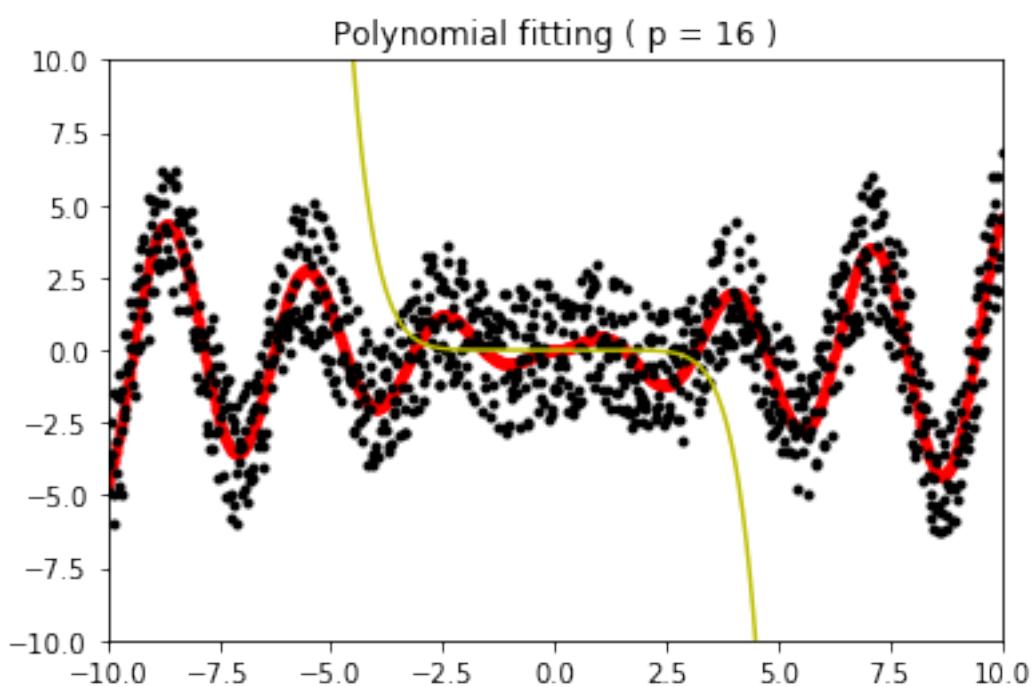
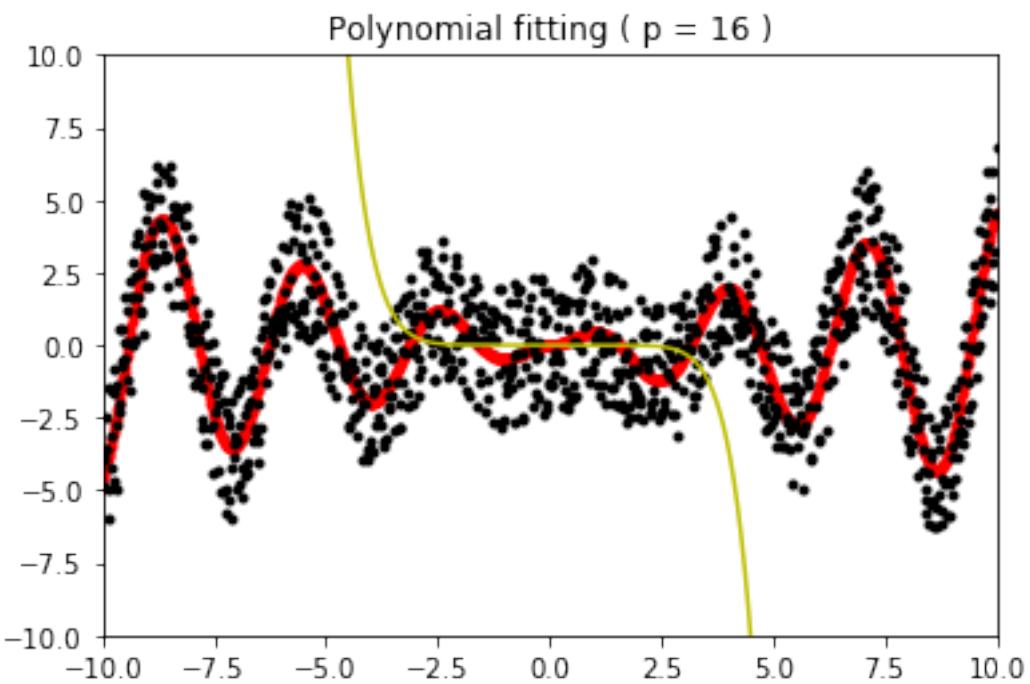
```
[-3.48975727e-12]
[ 2.50861749e-16]
[ 5.27198236e-10]
[-3.96435323e-14]
[ 4.96345487e-08]
[ 3.25565039e-12]
[-1.43232210e-05]
[-1.47769252e-10]
[ 1.03133069e-03]
[ 3.63194112e-09]
[-3.08680220e-02]
[-4.36853855e-08]
[ 3.64952639e-01]
[ 1.99273219e-07]
[-1.16943767e+00]
[-1.48186371e-07]]
```

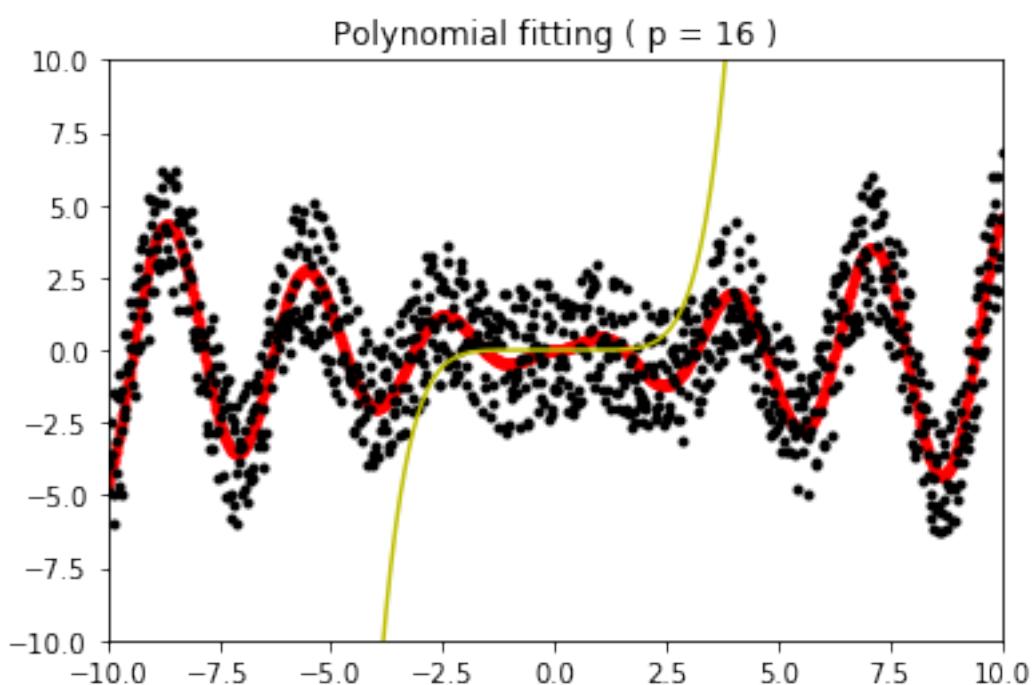
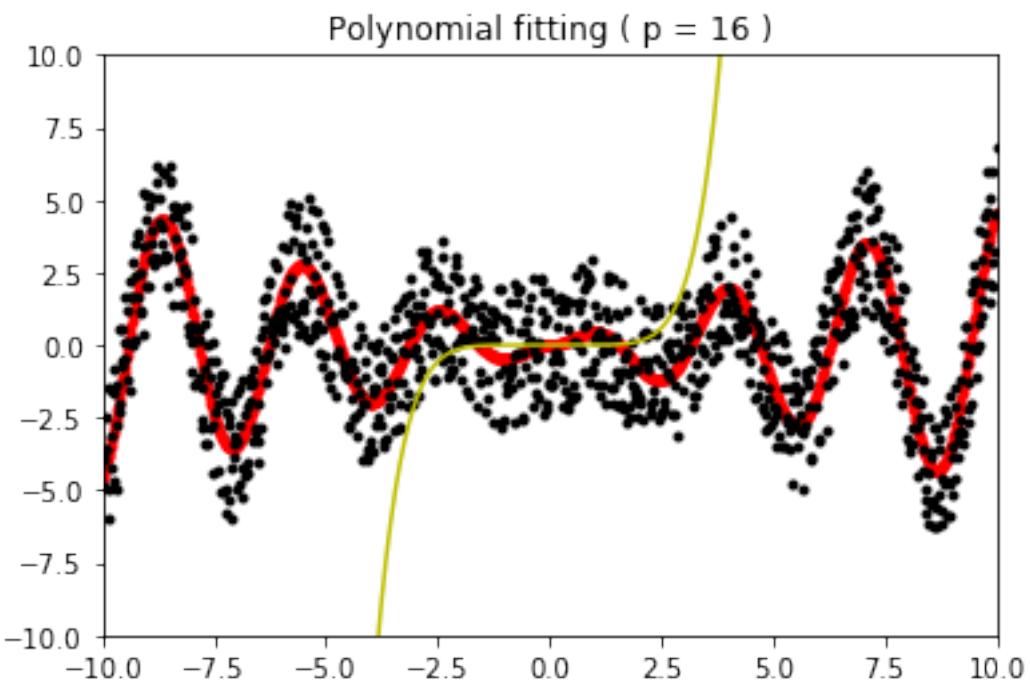


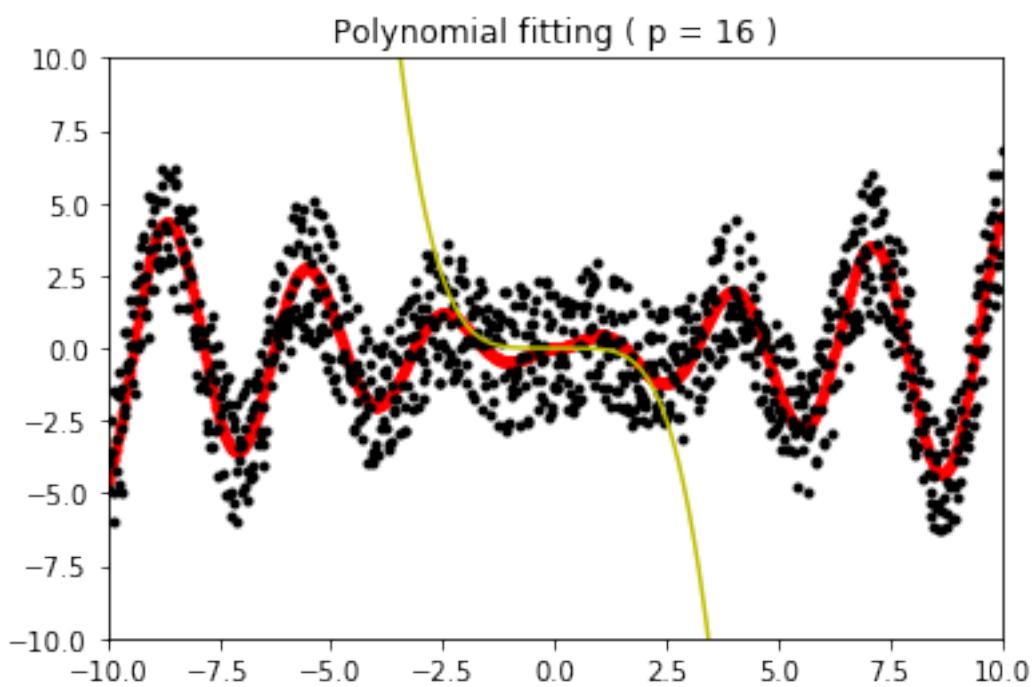
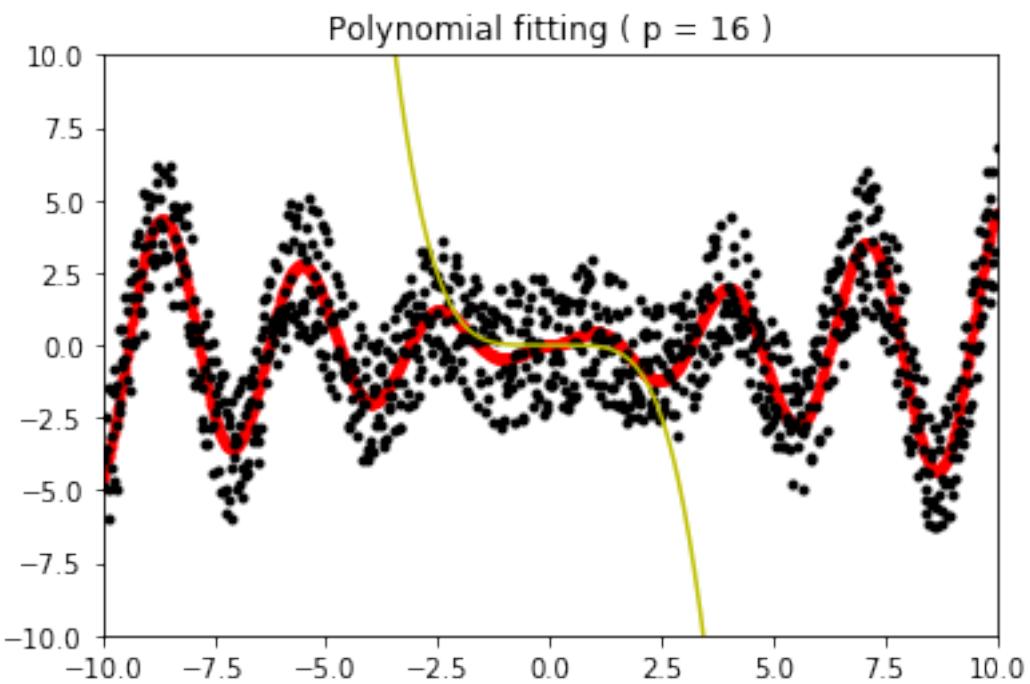


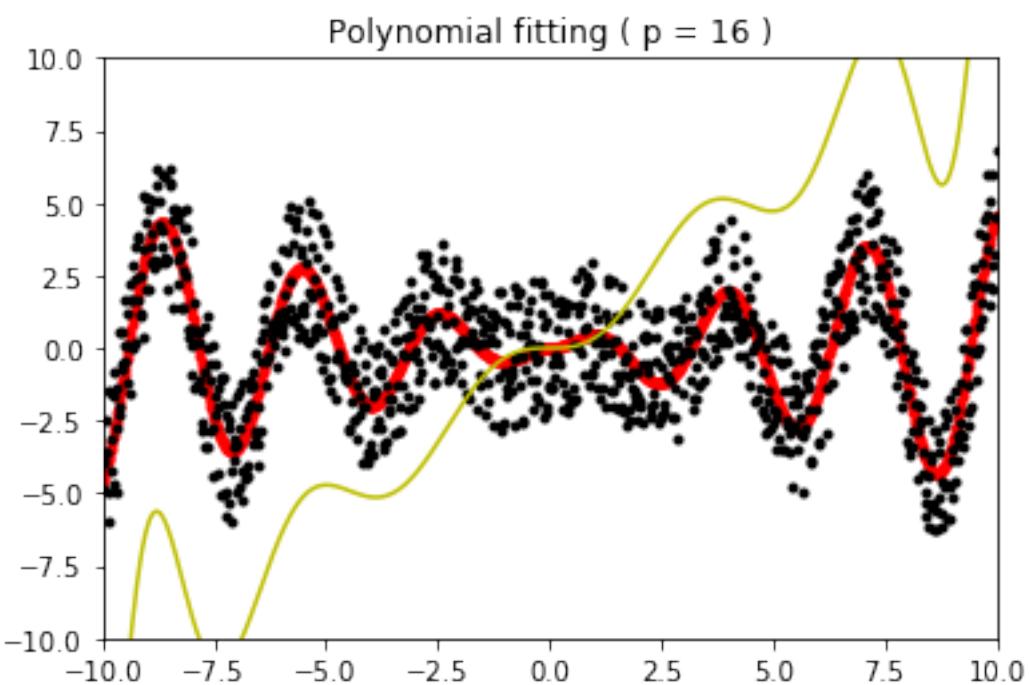
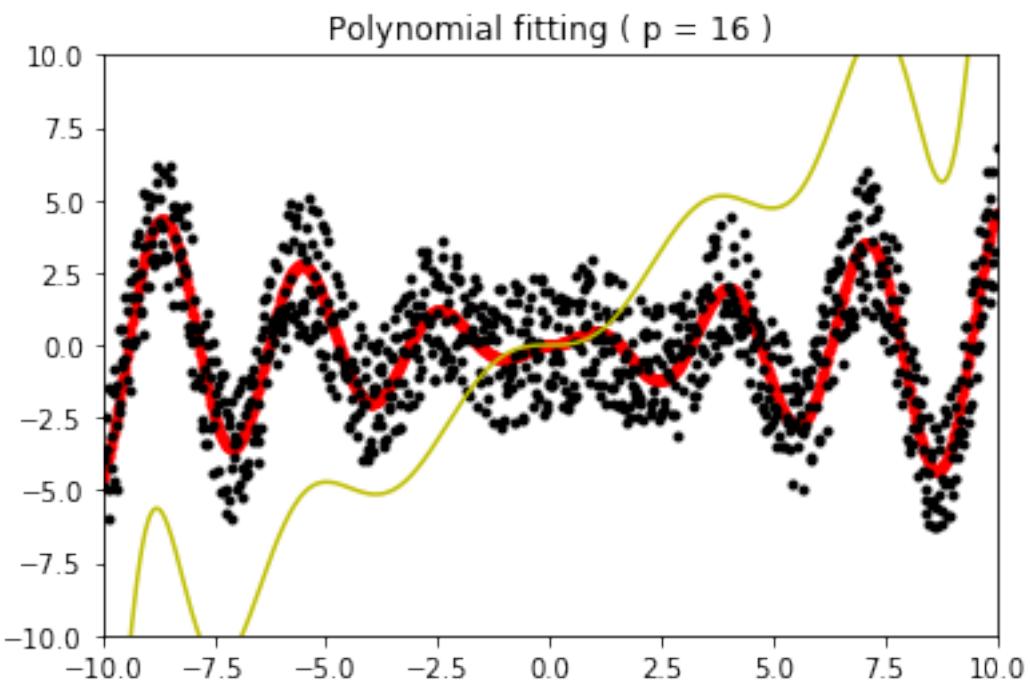


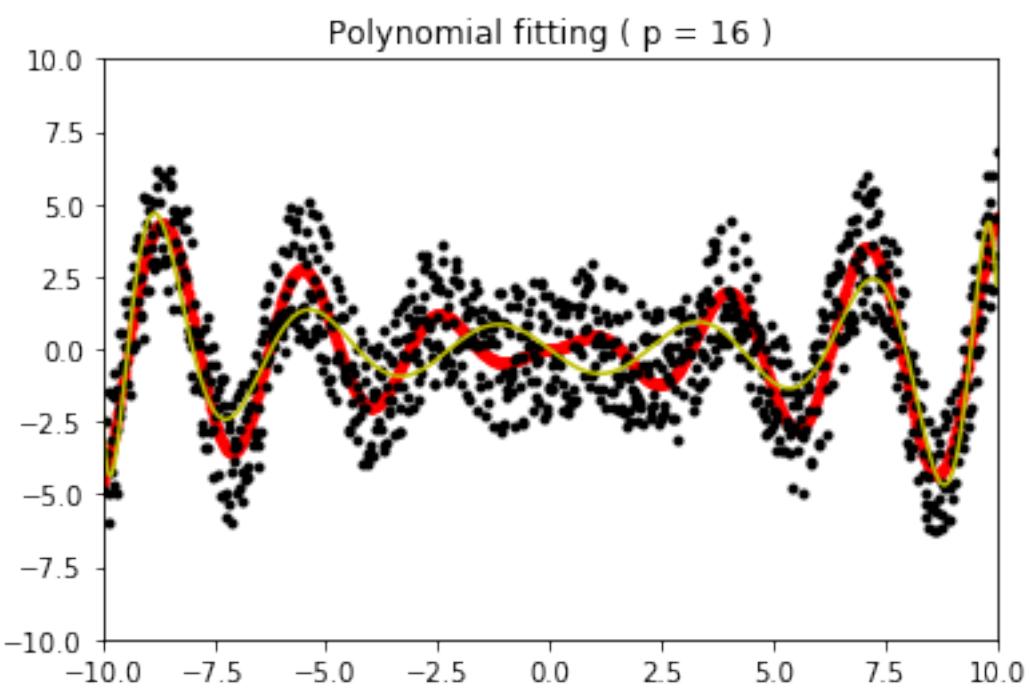
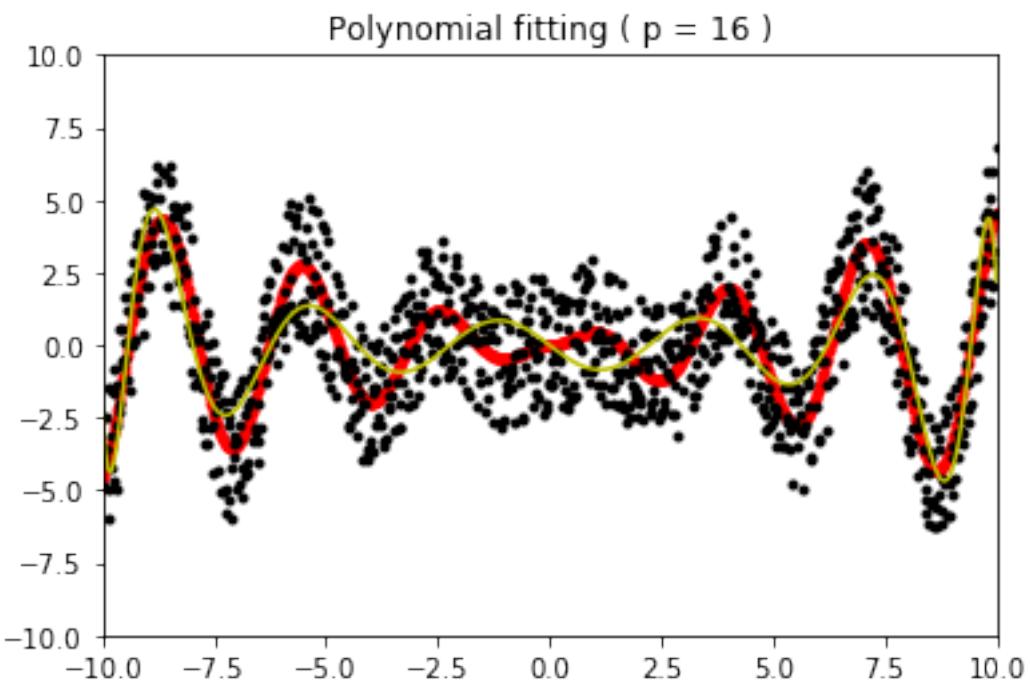






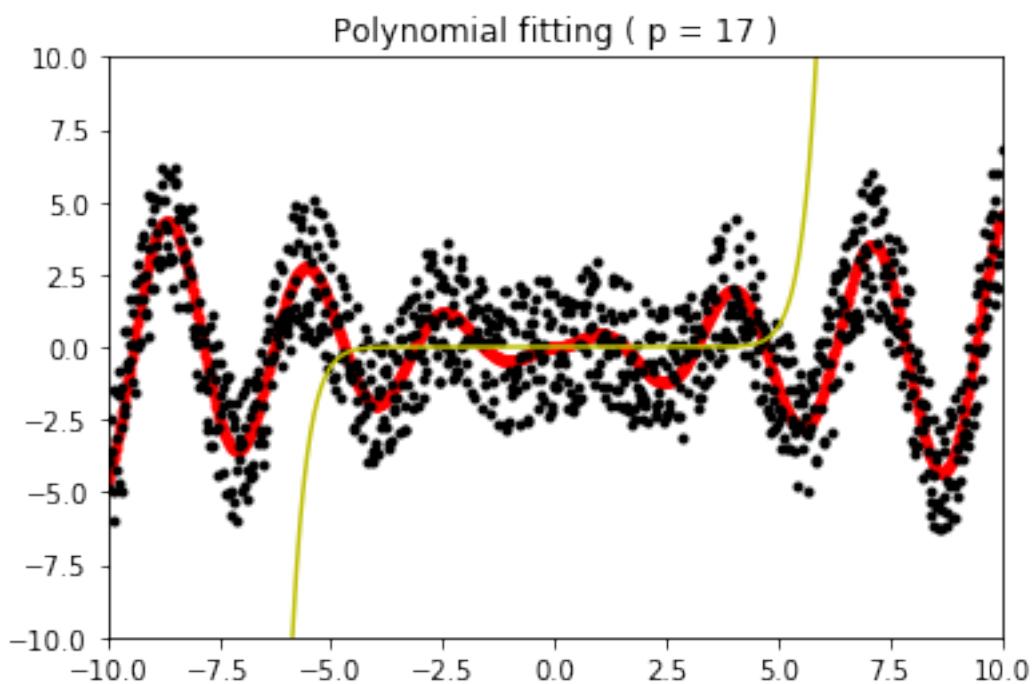


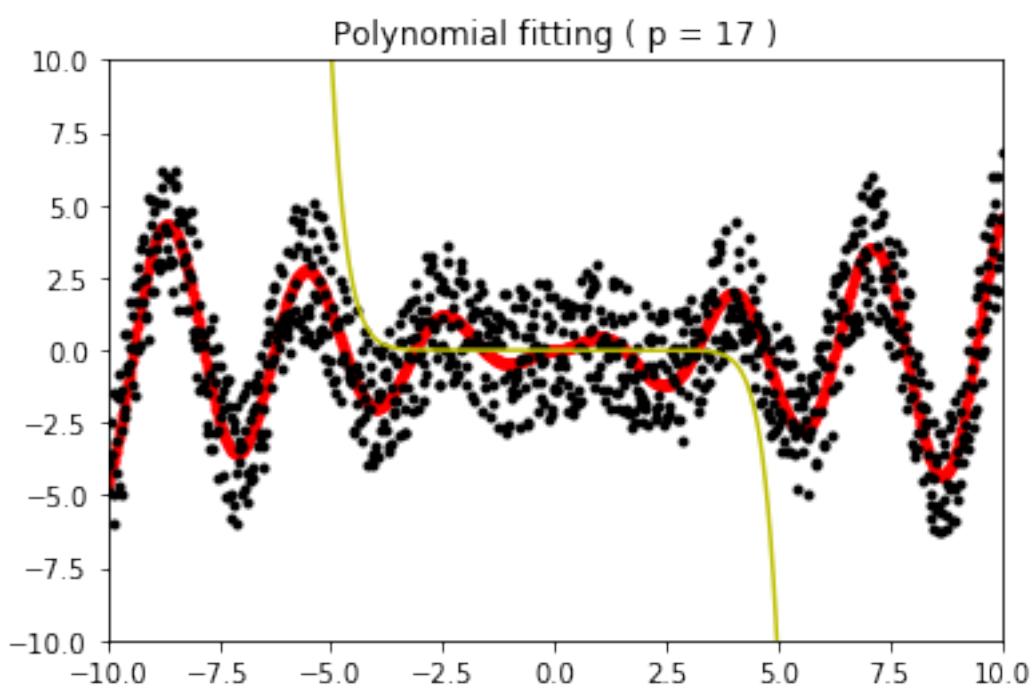
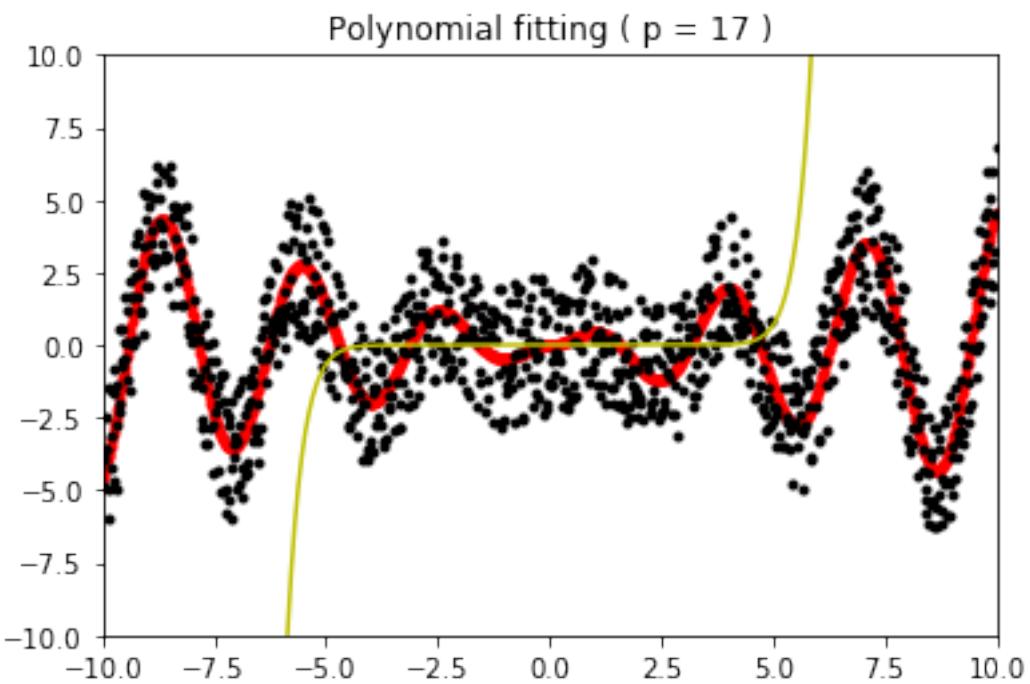


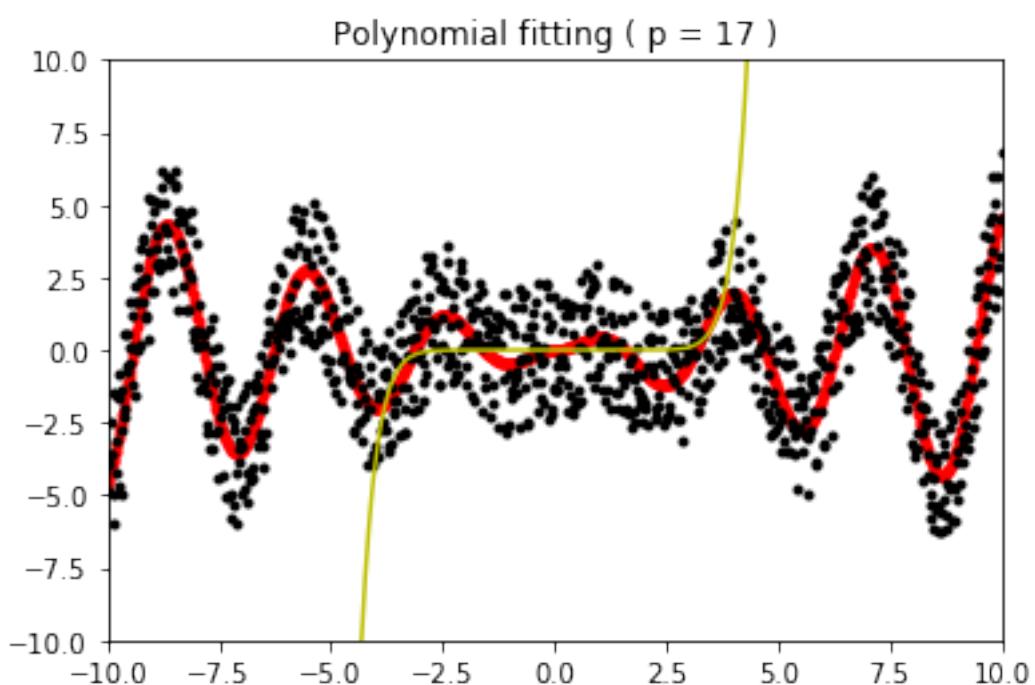
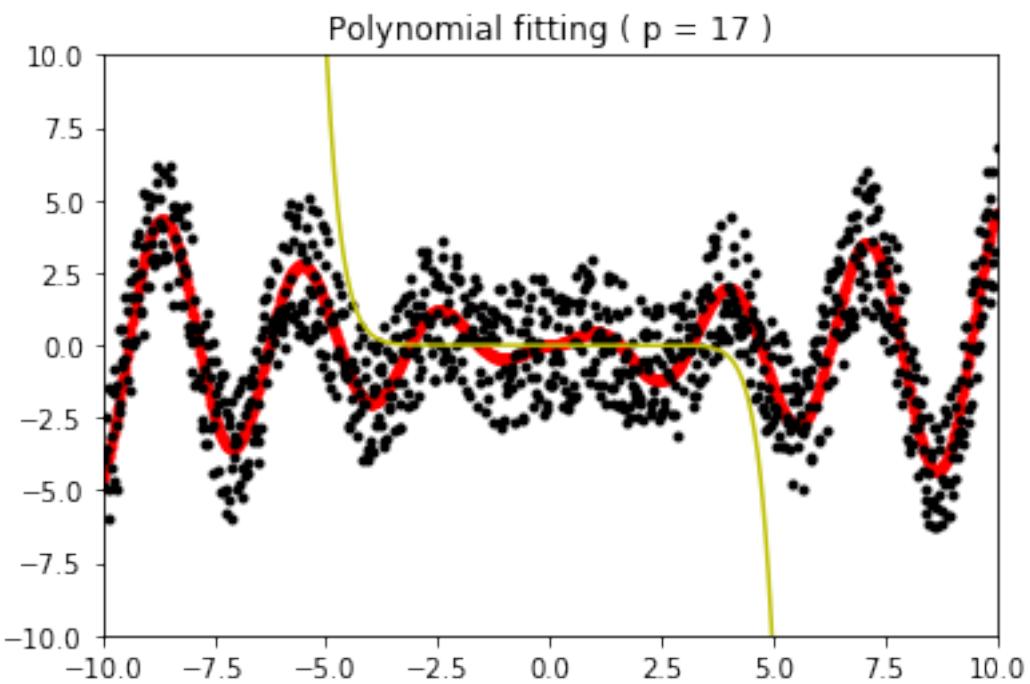


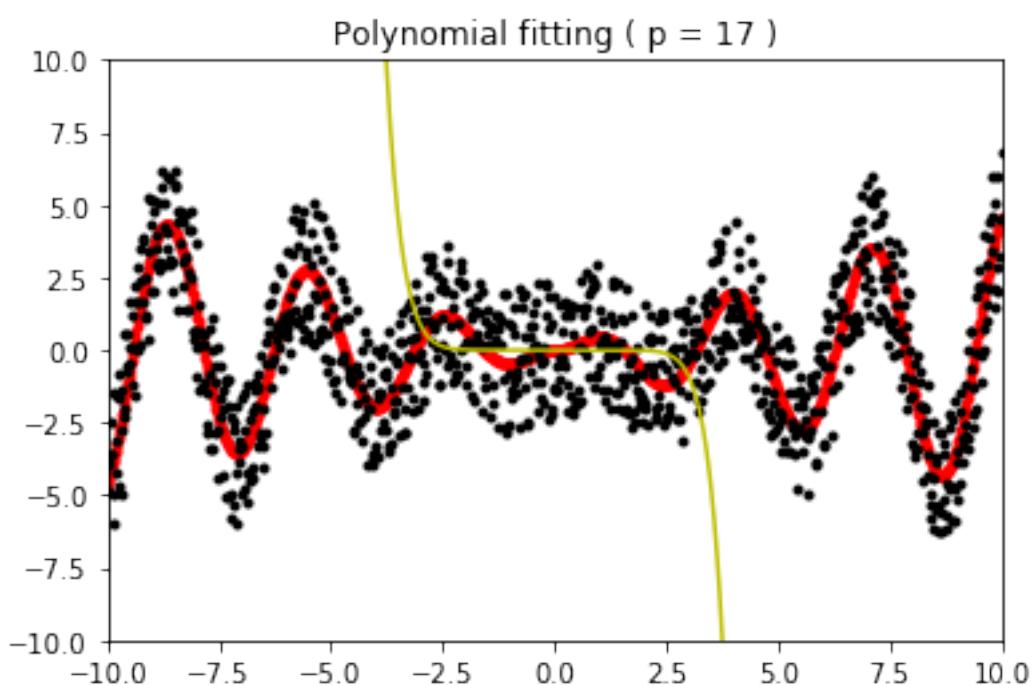
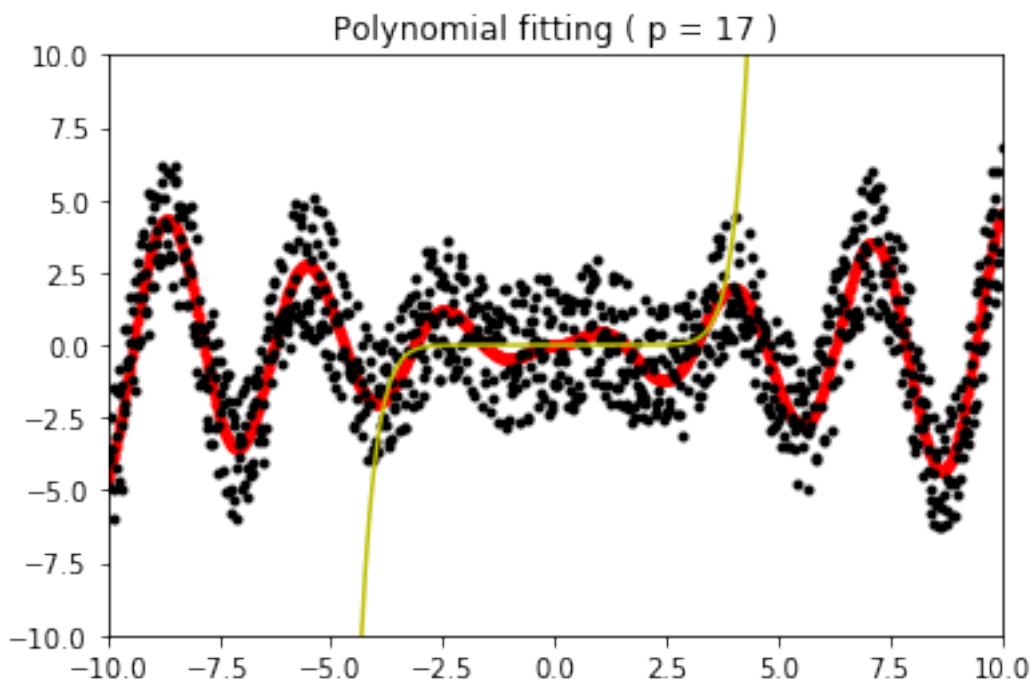
(18, 1)
[[8.84529387e-13]

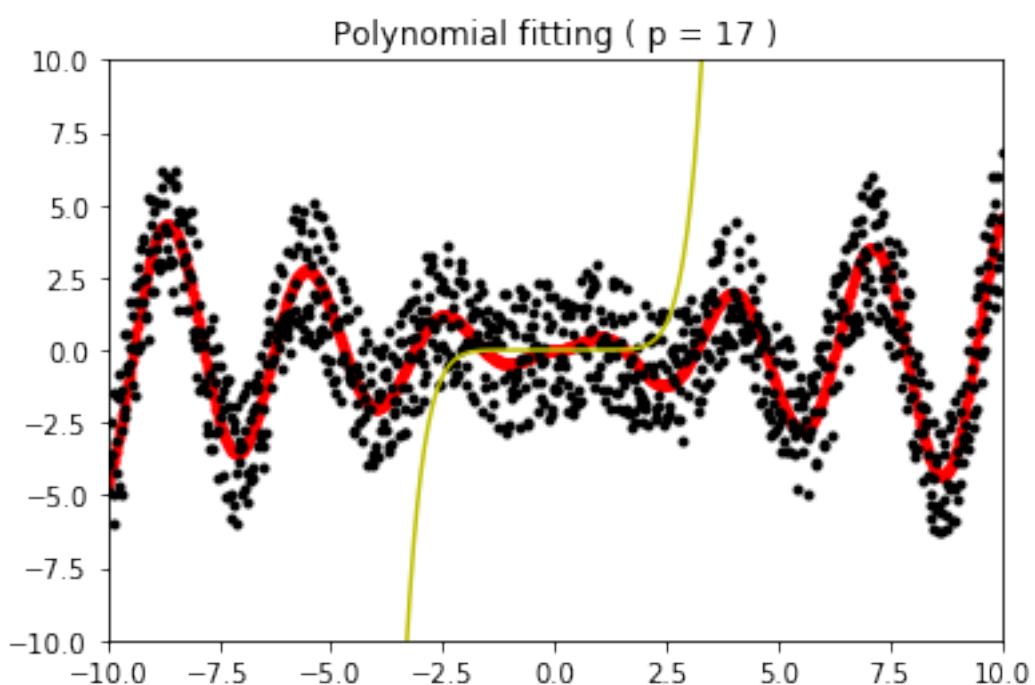
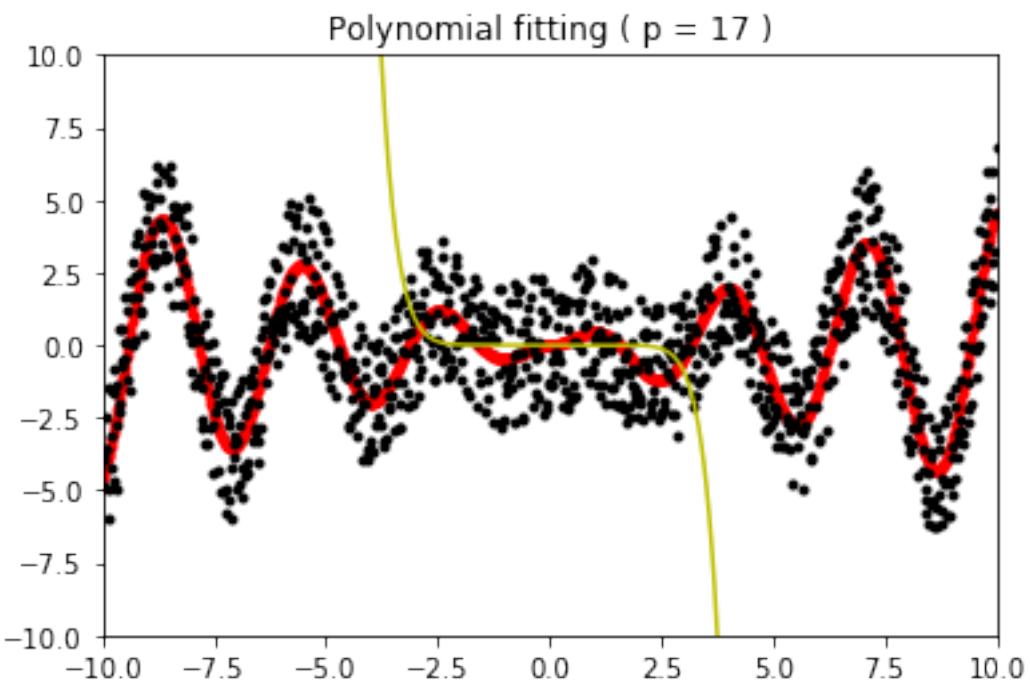
```
[ 2.50532808e-17]
[-3.68719981e-10]
[-9.86587083e-15]
[ 6.24988693e-08]
[ 1.57852168e-12]
[-5.51705318e-06]
[-1.31563763e-10]
[ 2.69706317e-04]
[ 6.07968664e-09]
[-7.16431388e-03]
[-1.52800085e-07]
[ 9.40858864e-02]
[ 1.89096606e-06]
[-4.86692009e-01]
[-8.95646184e-06]
[ 5.14637097e-01]
[ 7.01684122e-06]]
```

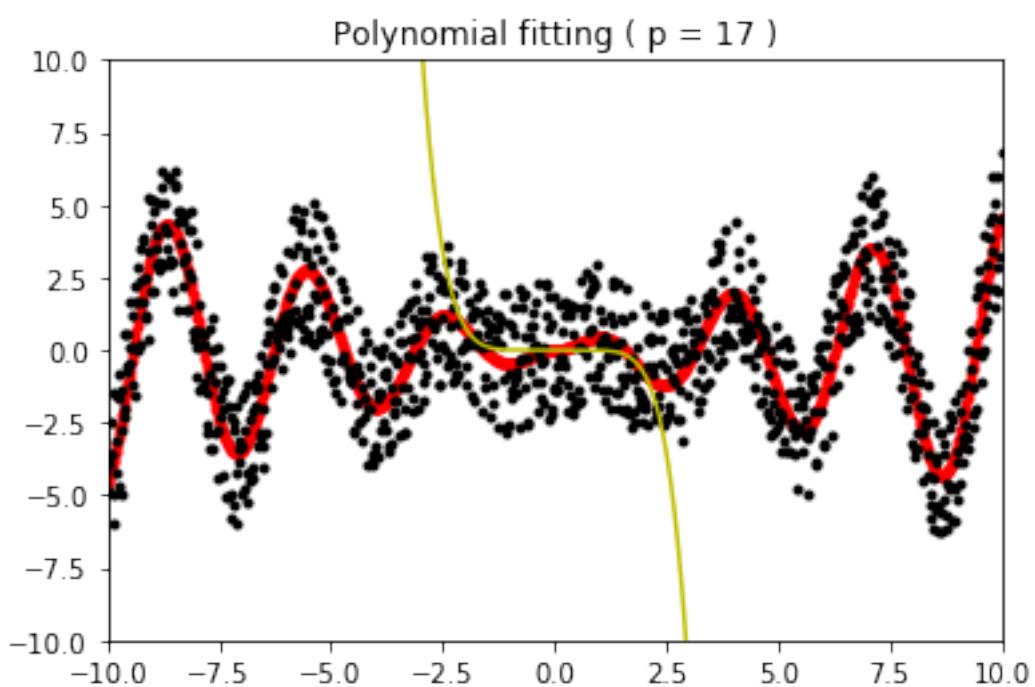
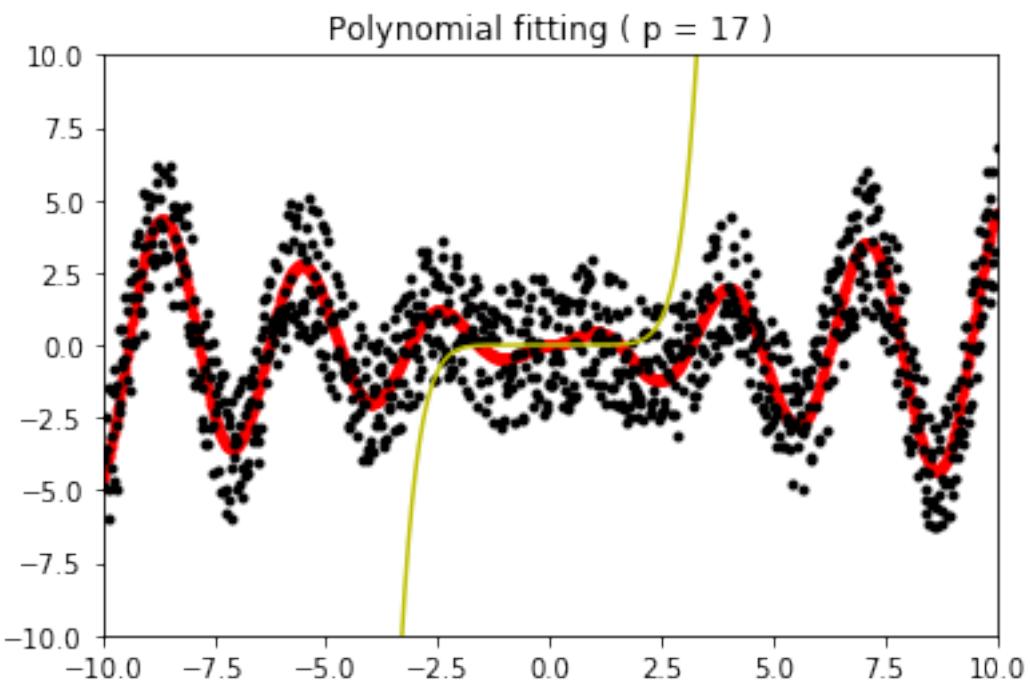


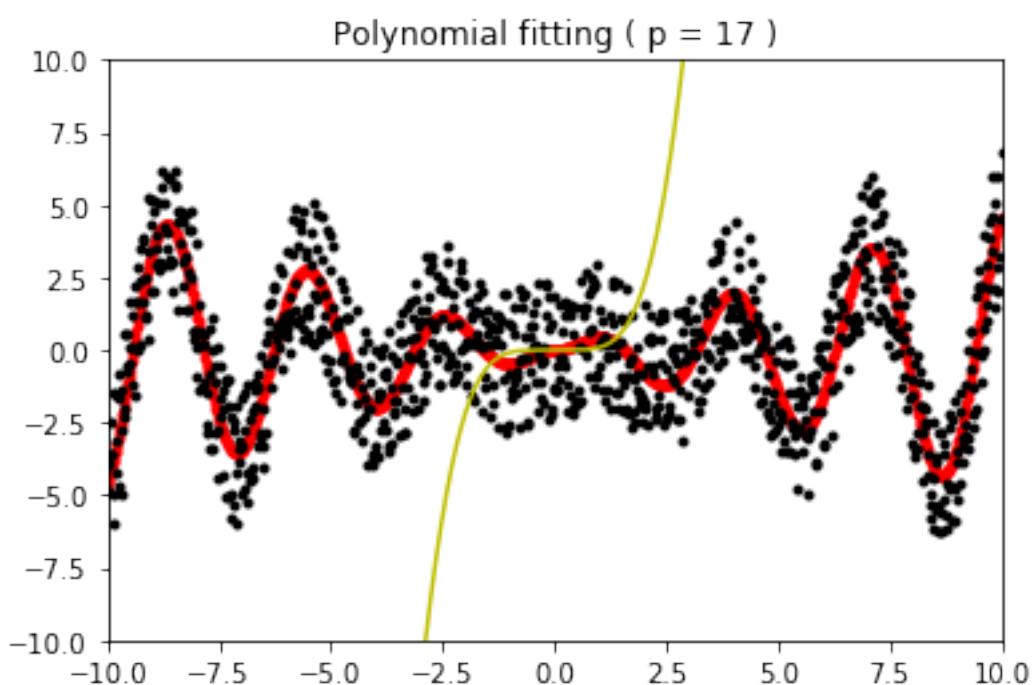
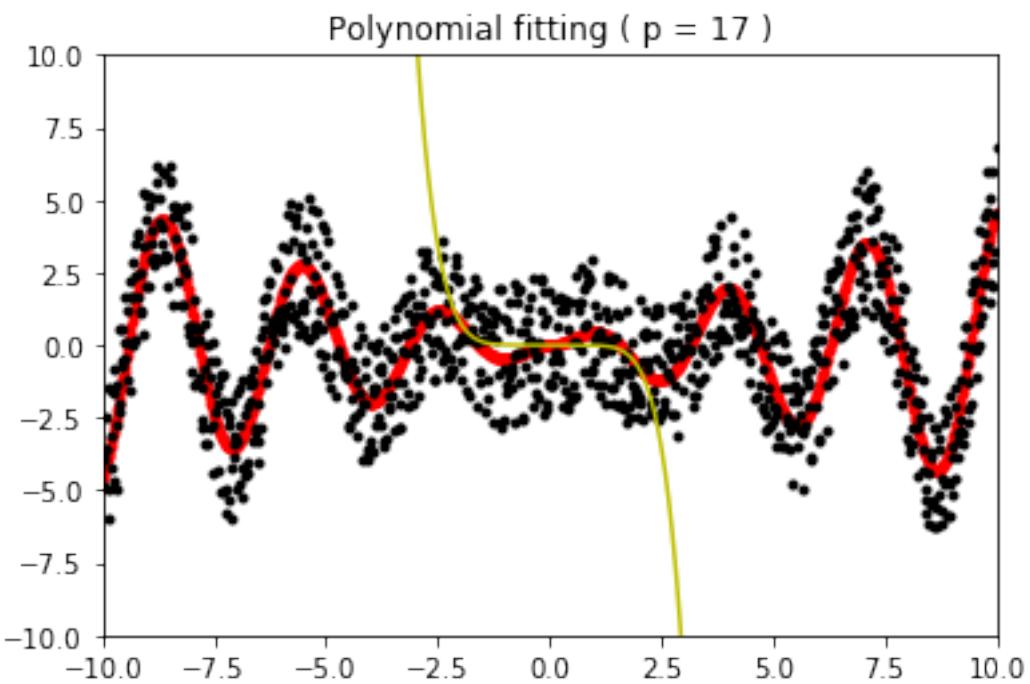


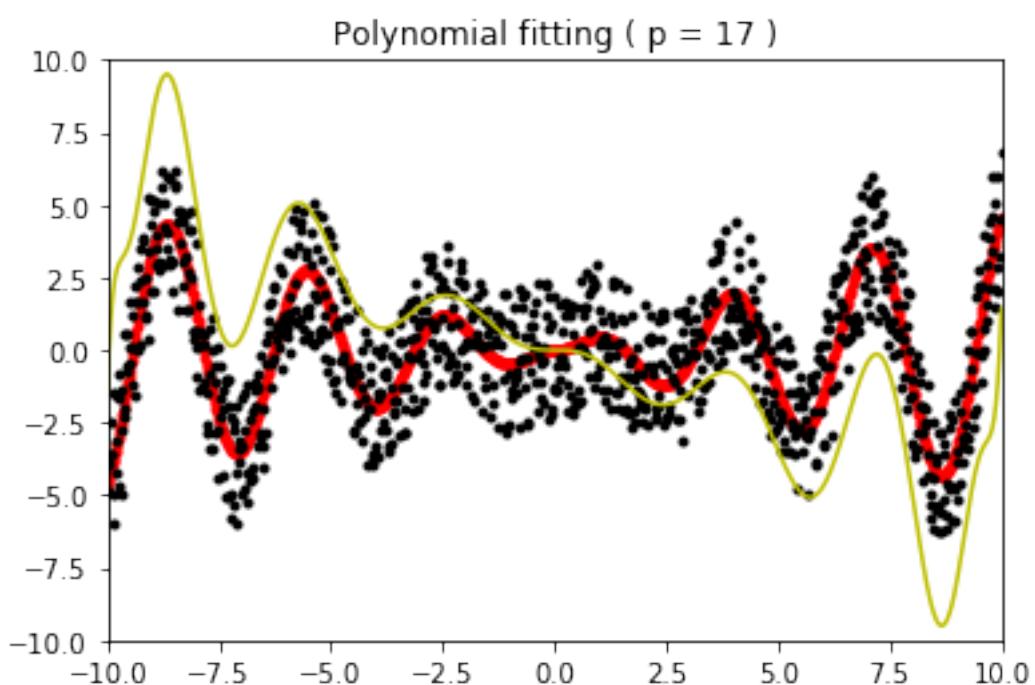
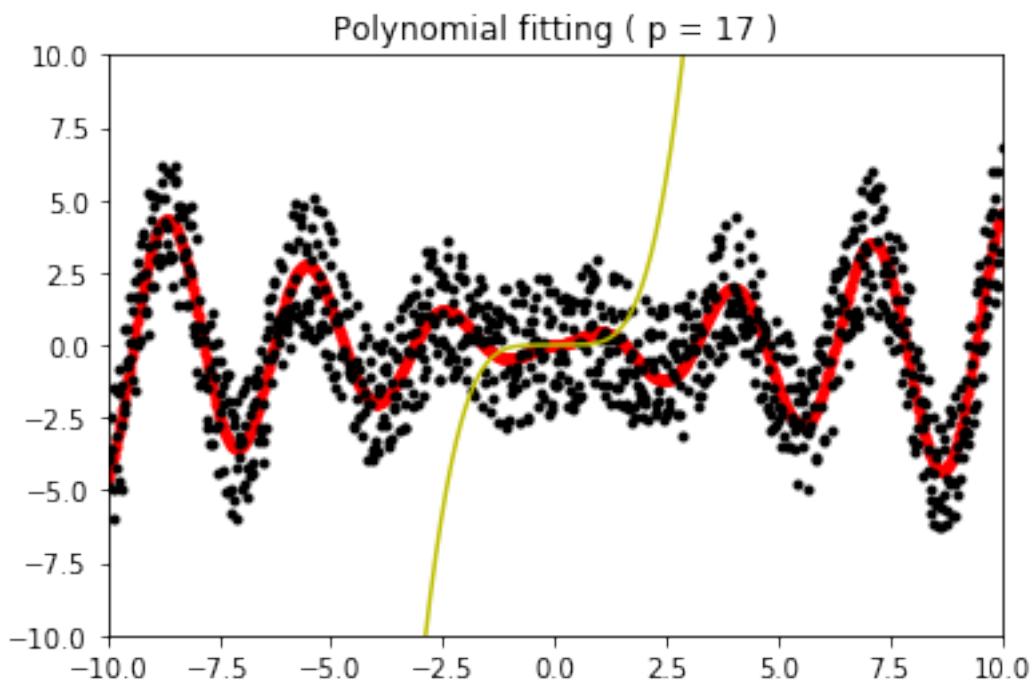


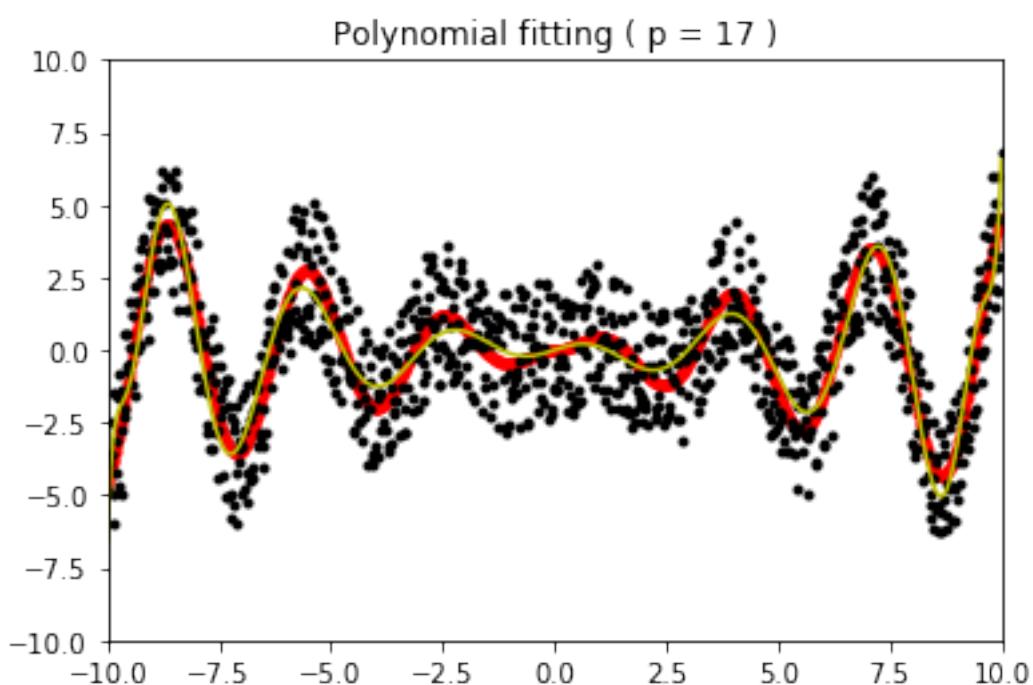
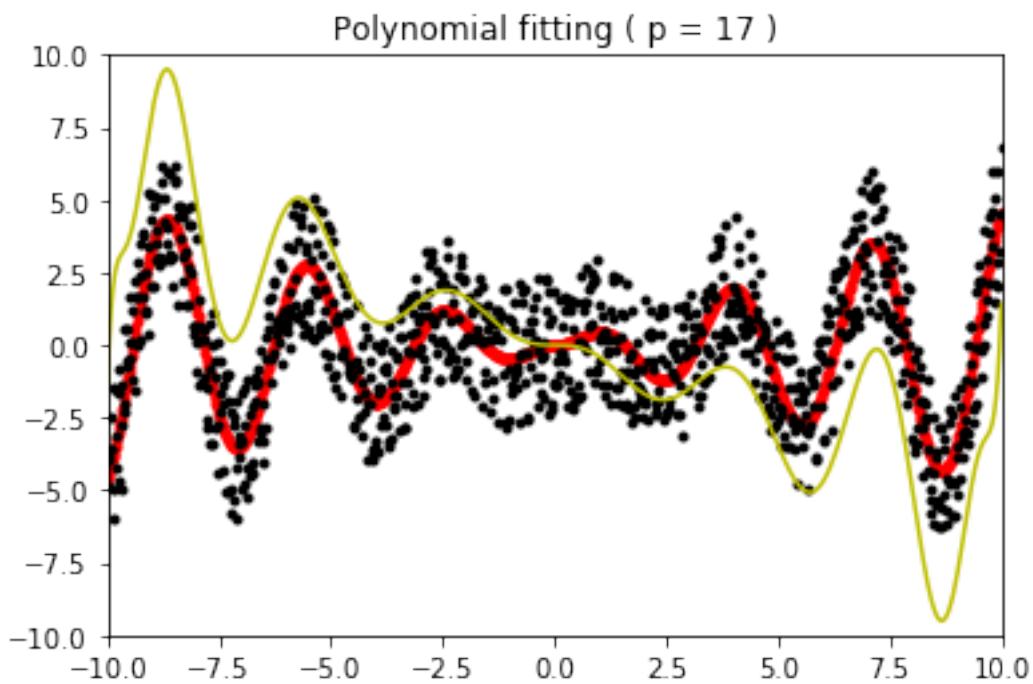


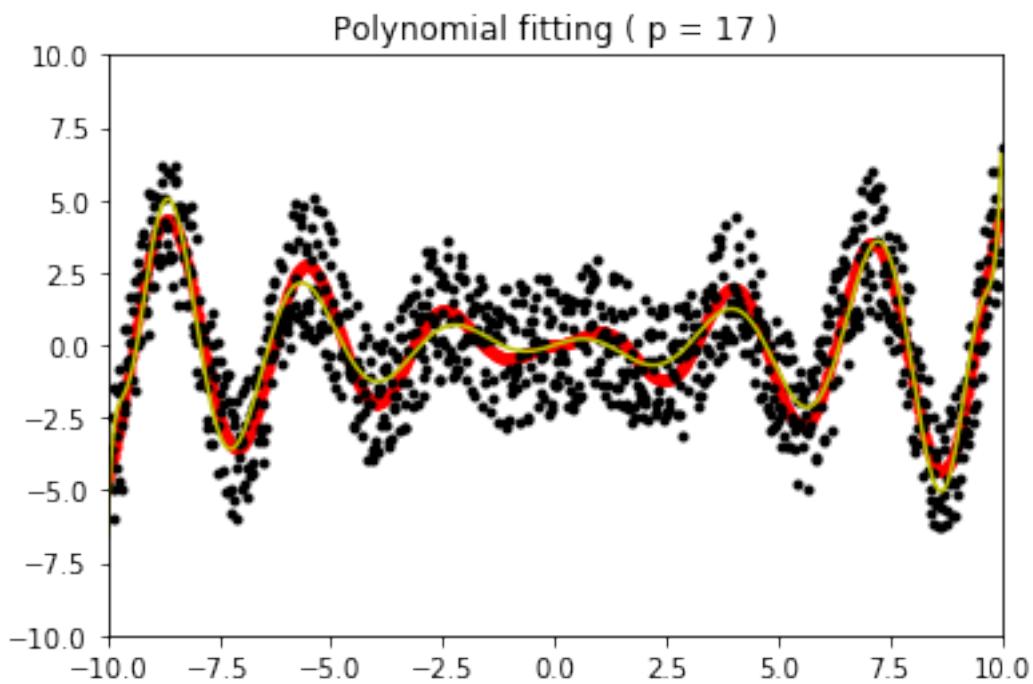




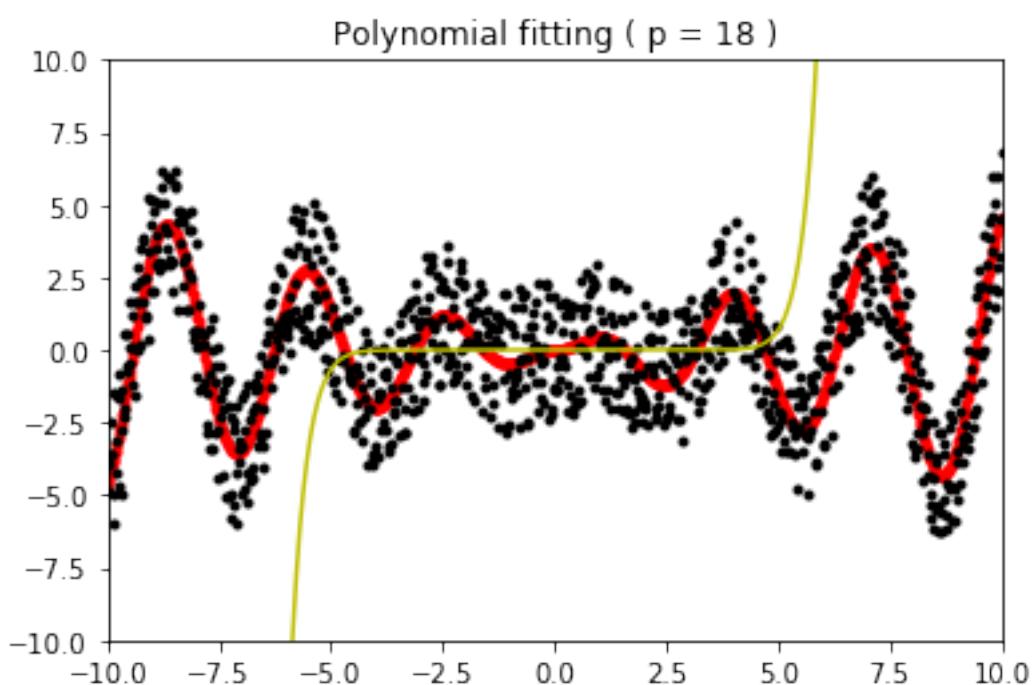
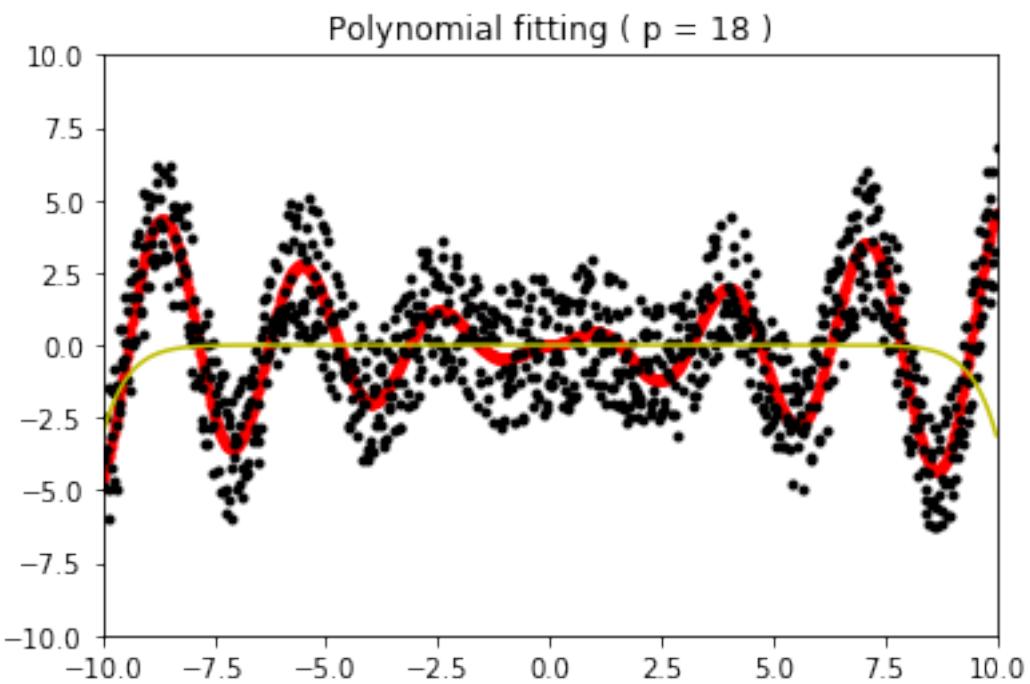


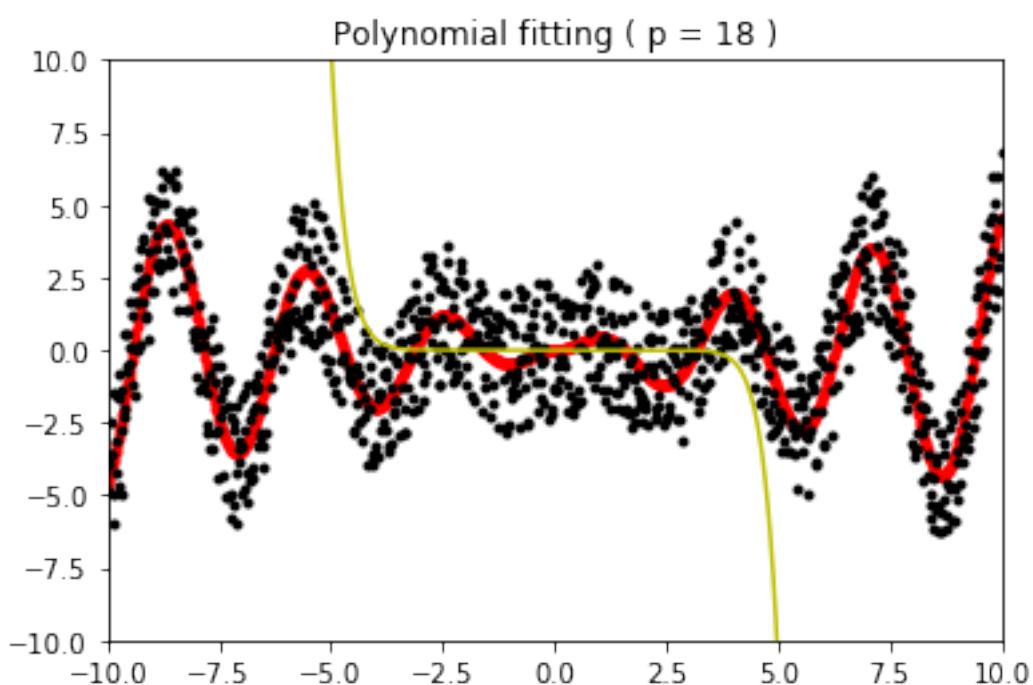
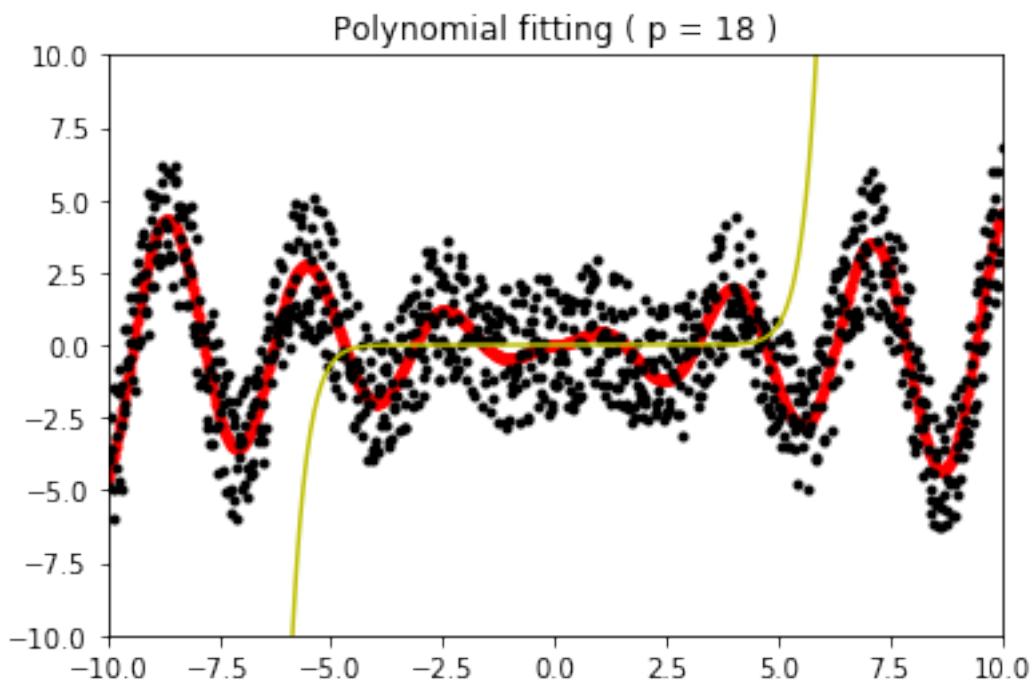


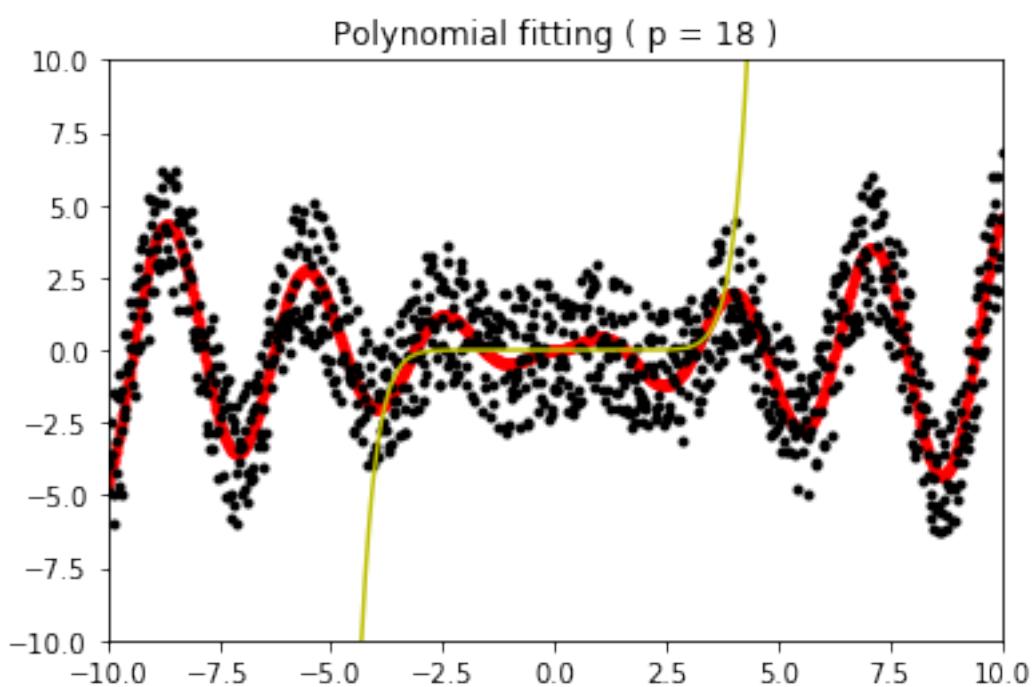
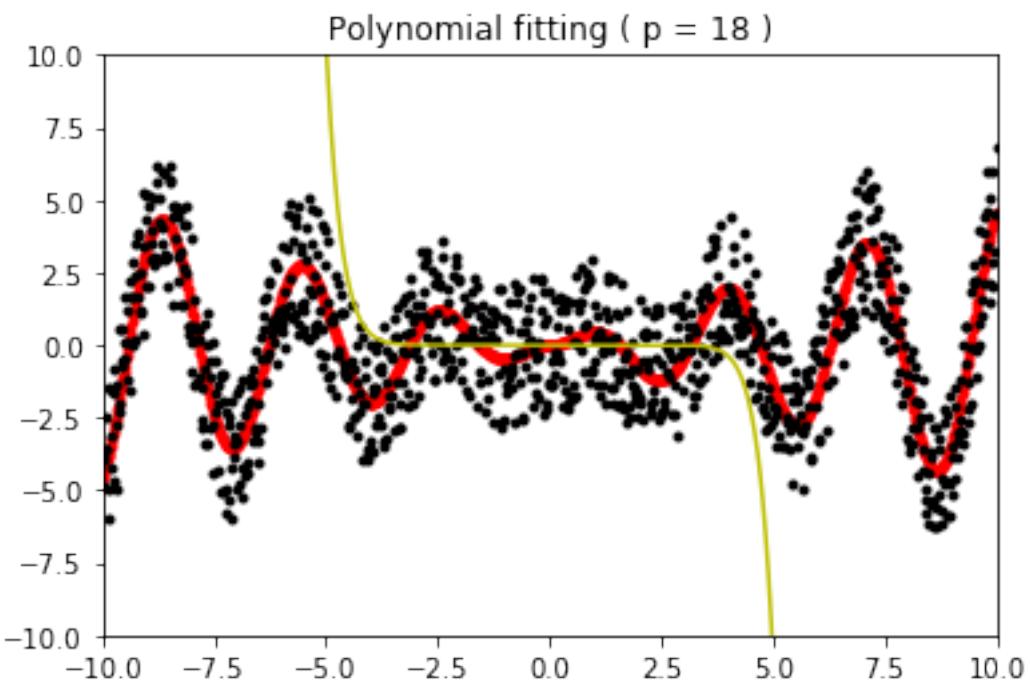


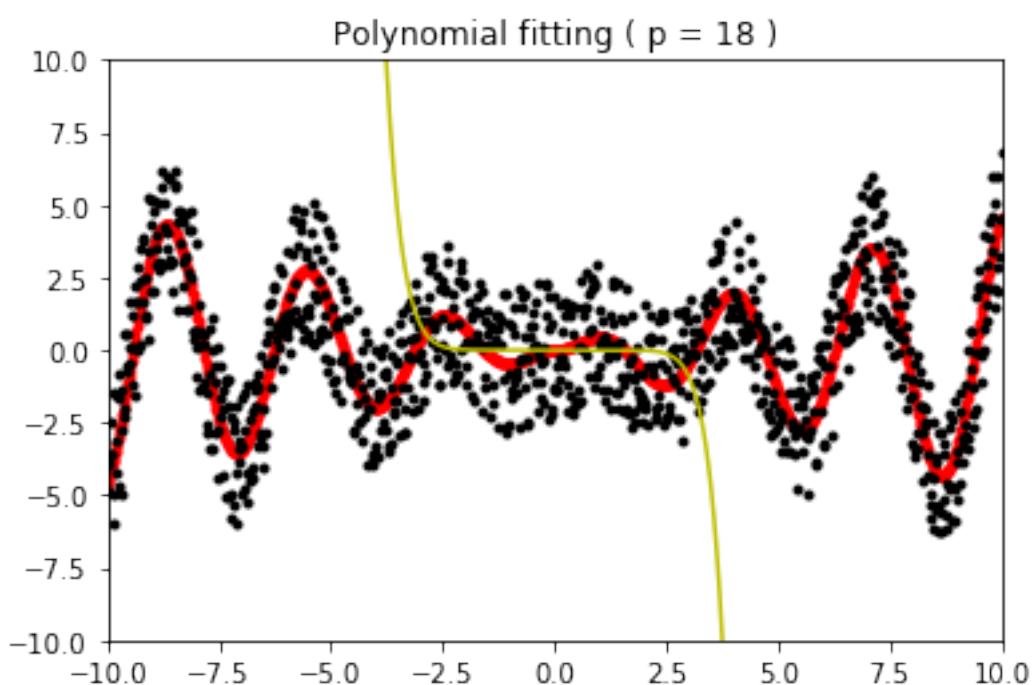
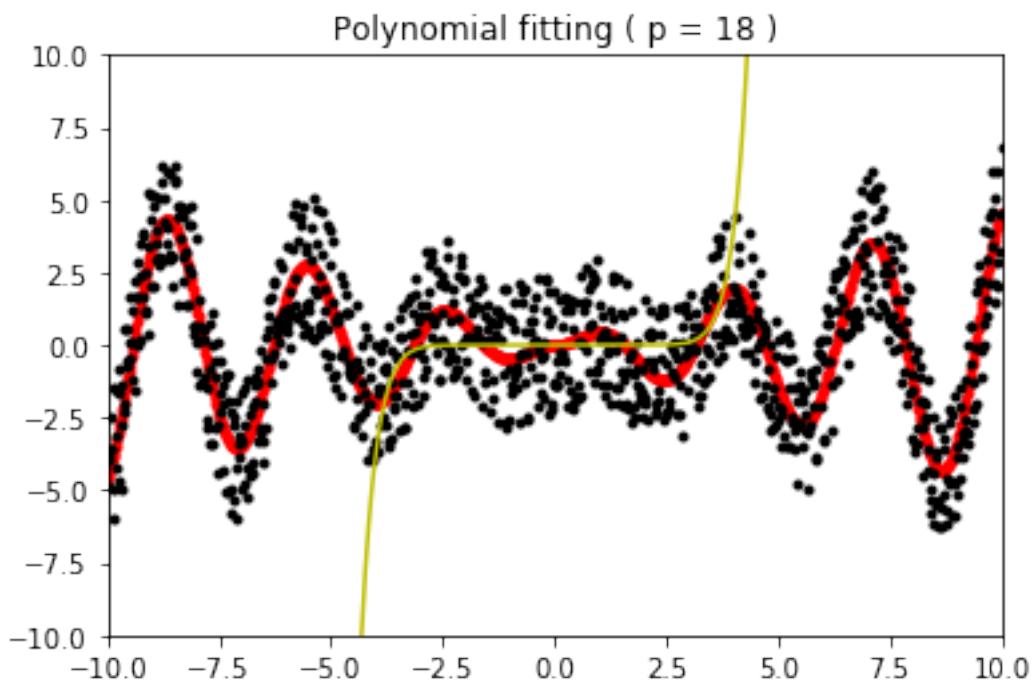


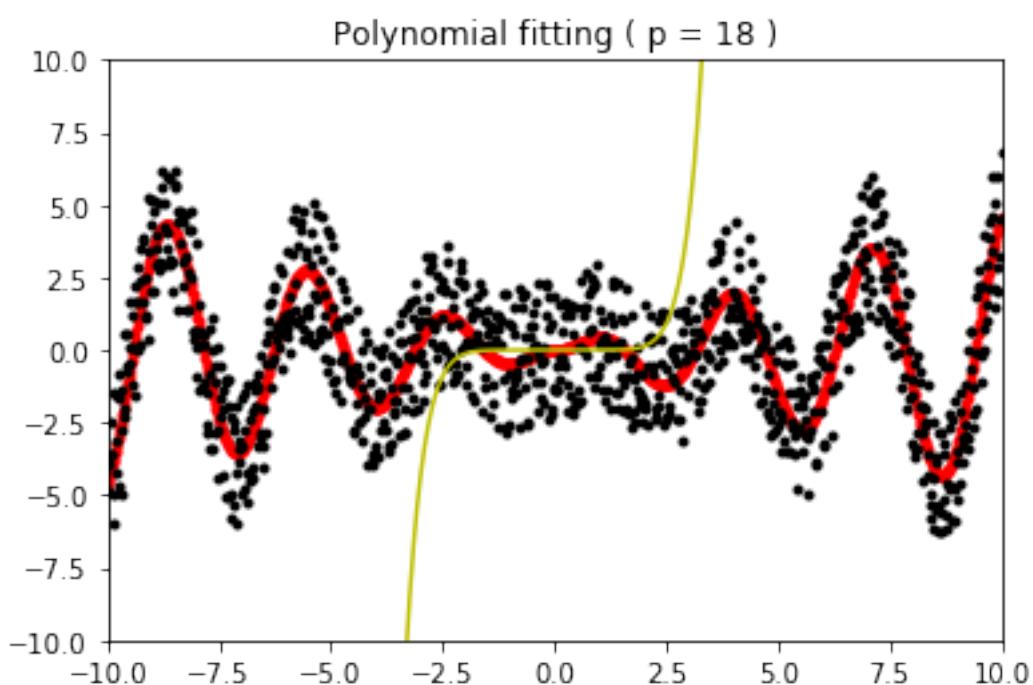
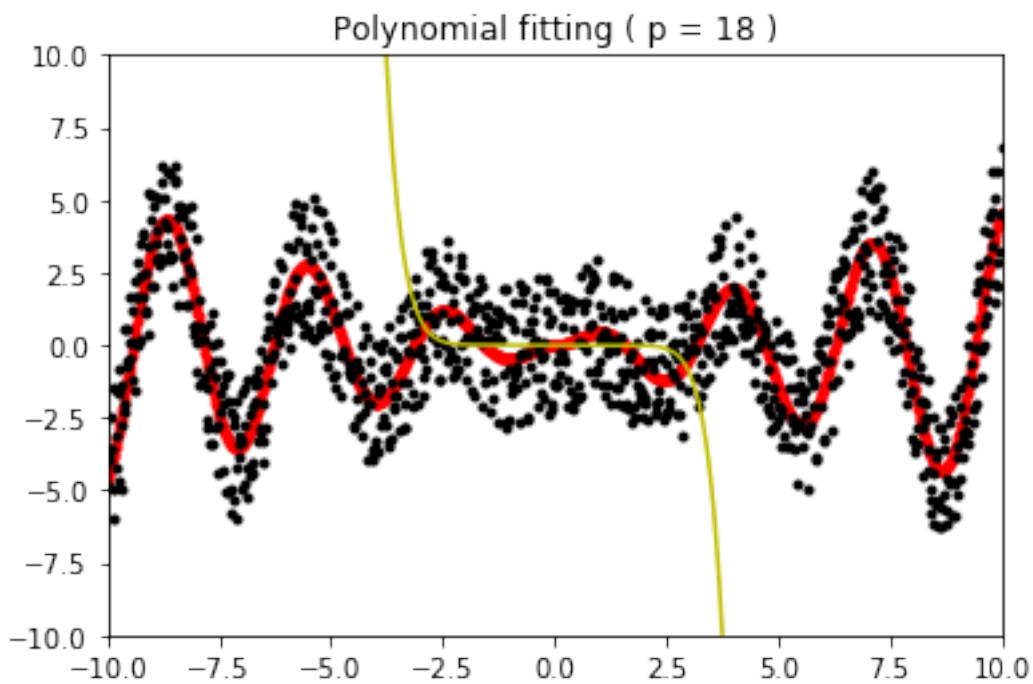
```
(19, 1)
[[-3.11101339e-18]
 [ 8.84529388e-13]
 [ 1.38759559e-15]
 [-3.68719981e-10]
 [-2.58071375e-13]
 [ 6.24988694e-08]
 [ 2.59114004e-11]
 [-5.51705318e-06]
 [-1.51864652e-09]
 [ 2.69706317e-04]
 [ 5.24034404e-08]
 [-7.16431388e-03]
 [-1.01914864e-06]
 [ 9.40858865e-02]
 [ 9.97781993e-06]
 [-4.86692009e-01]
 [-3.78927367e-05]
 [ 5.14637098e-01]
 [ 2.39706673e-05]]
```

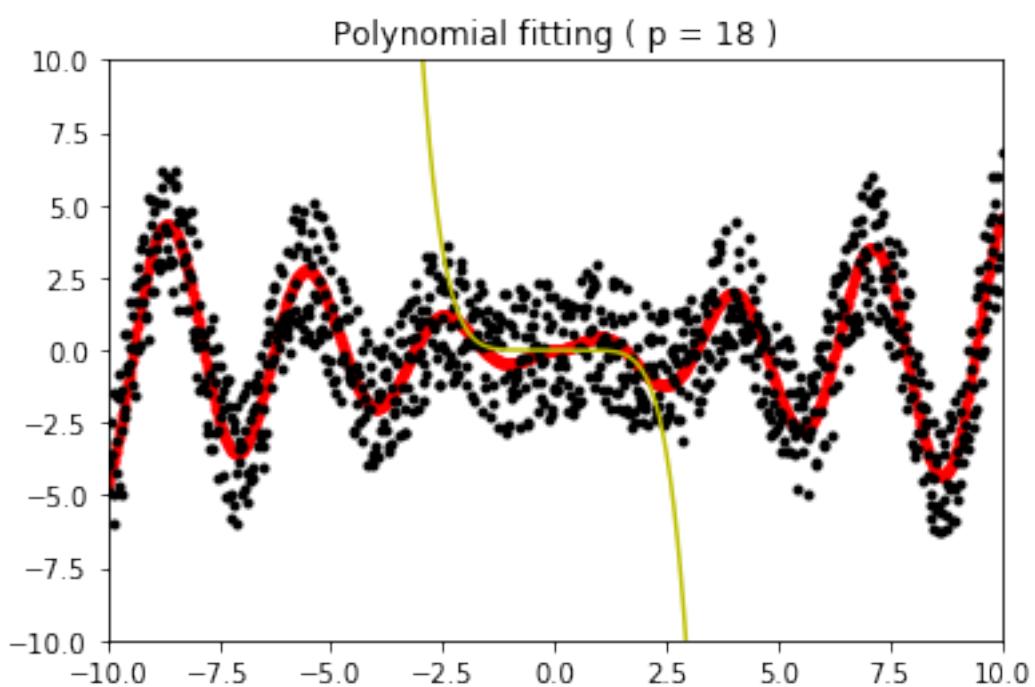
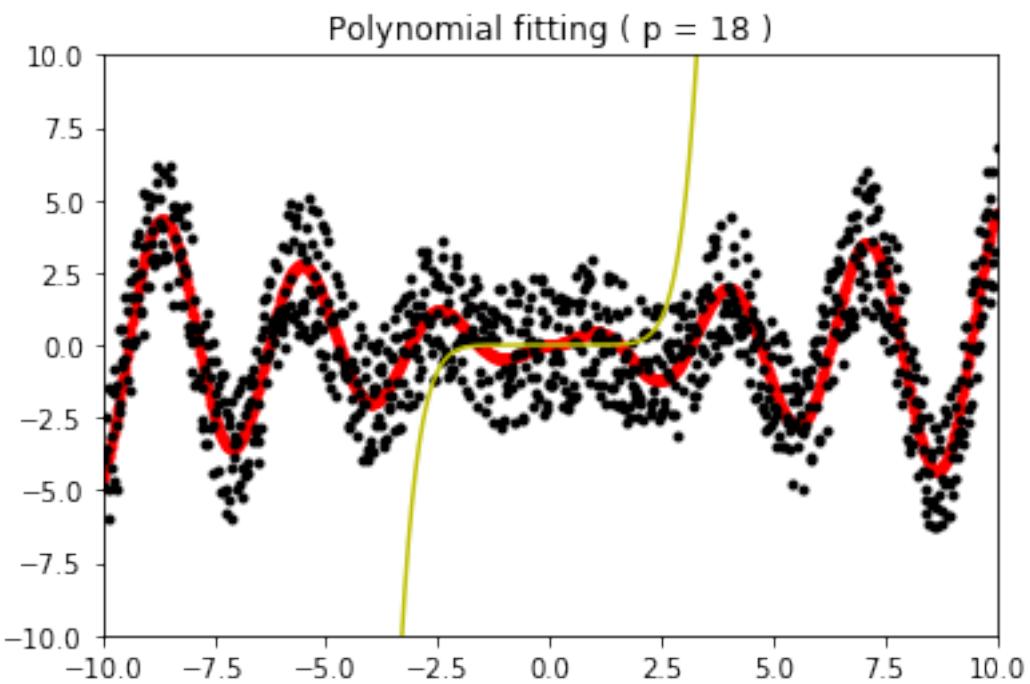


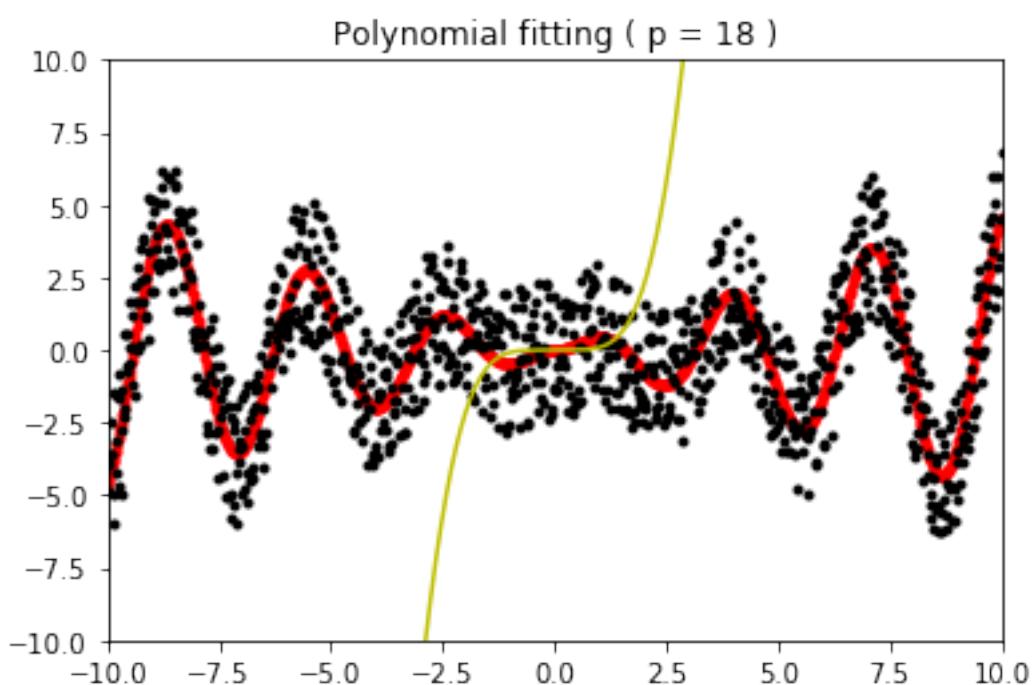
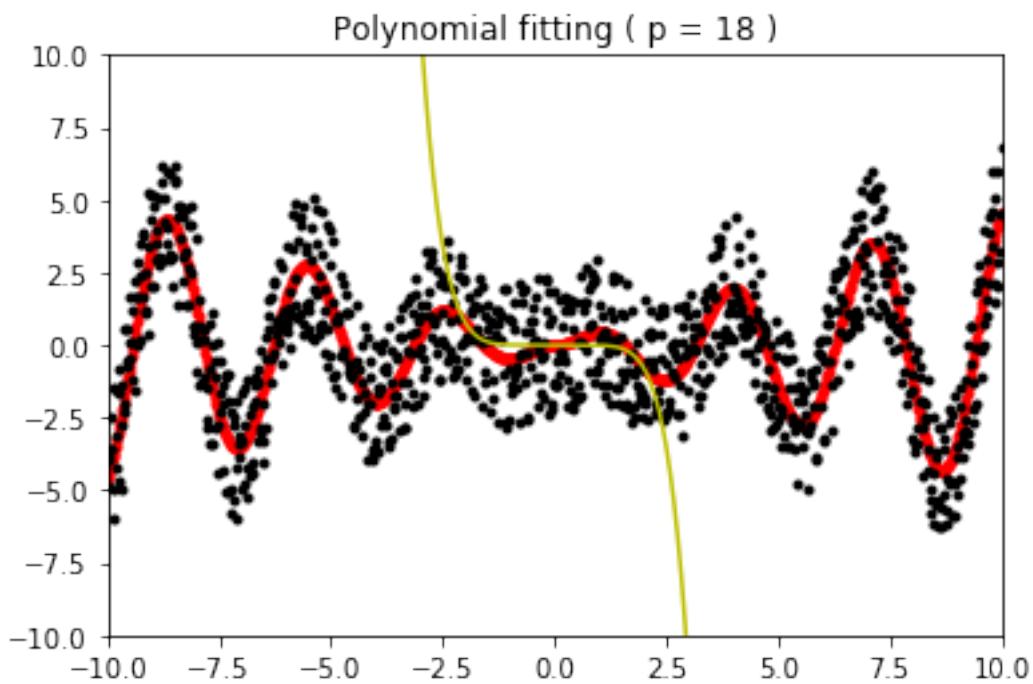


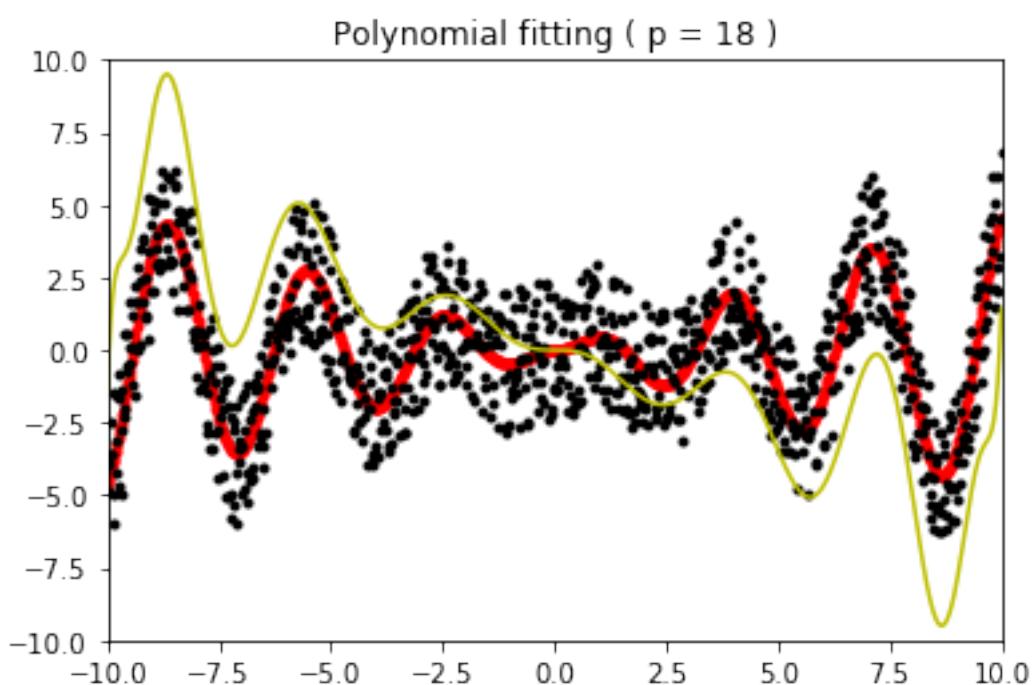
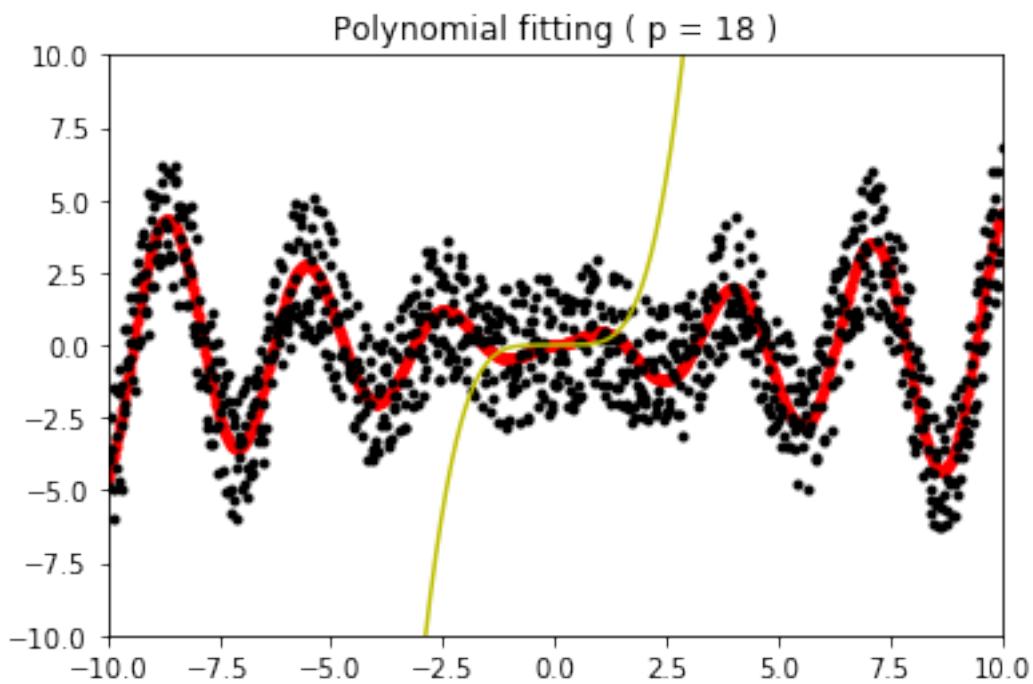


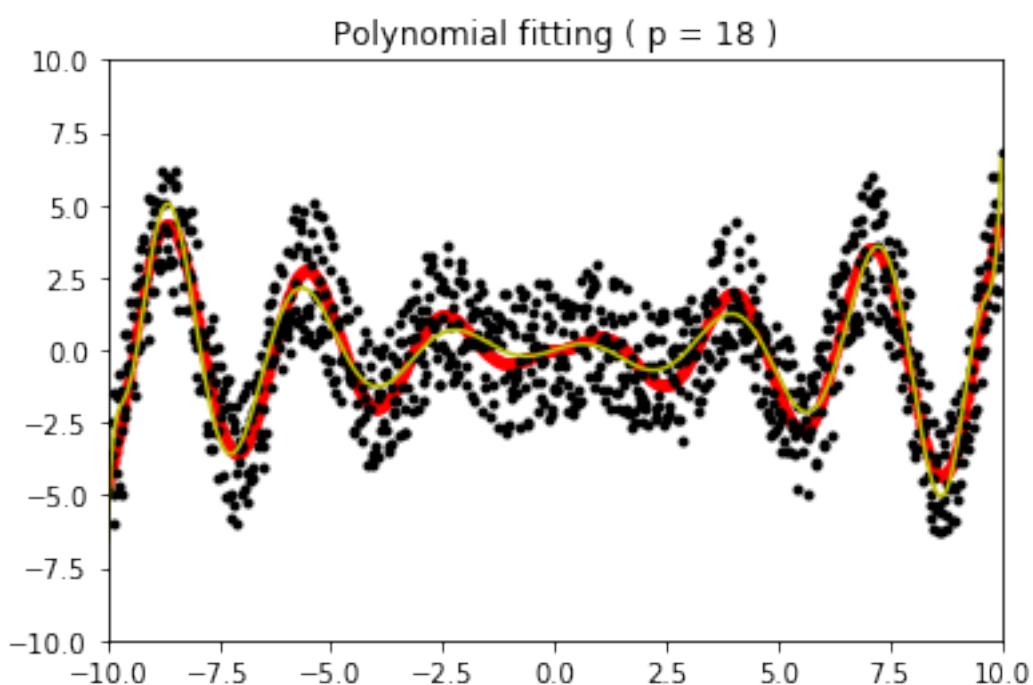
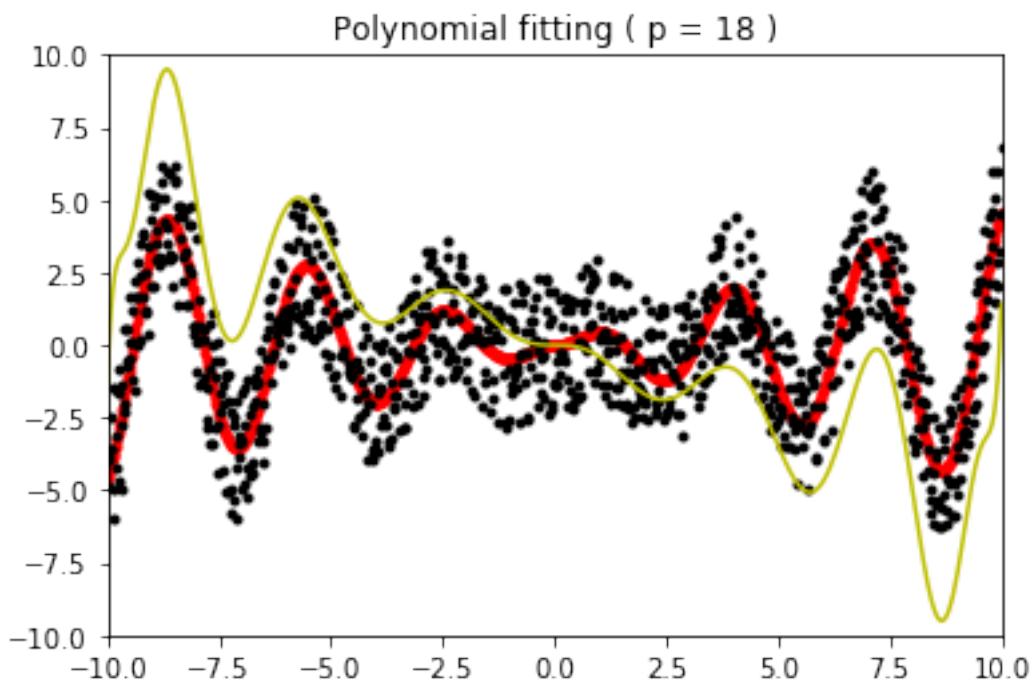


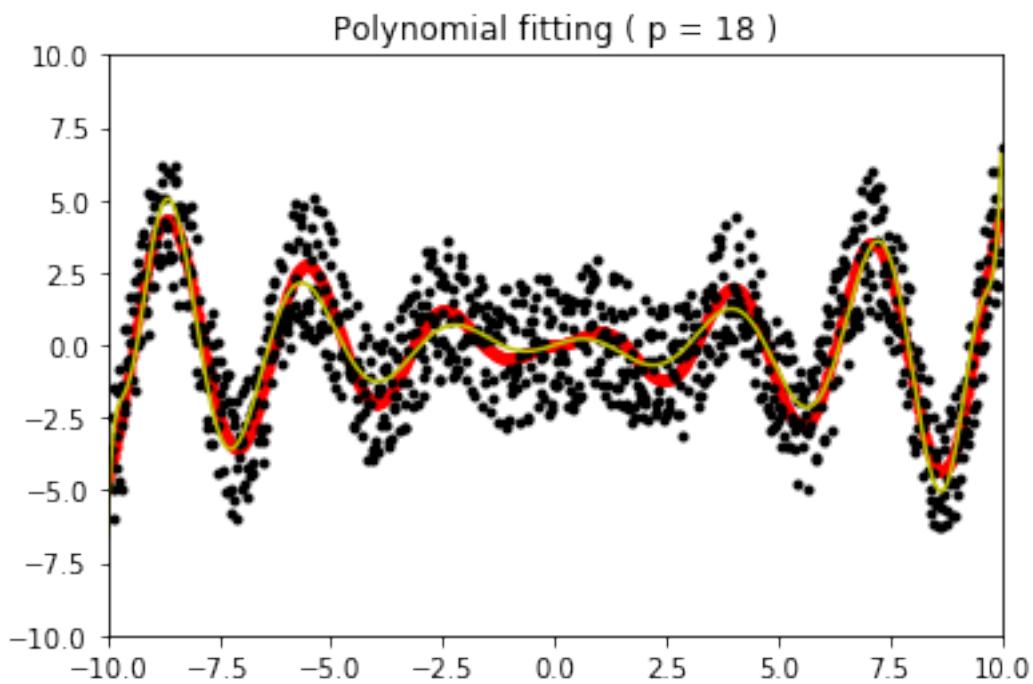




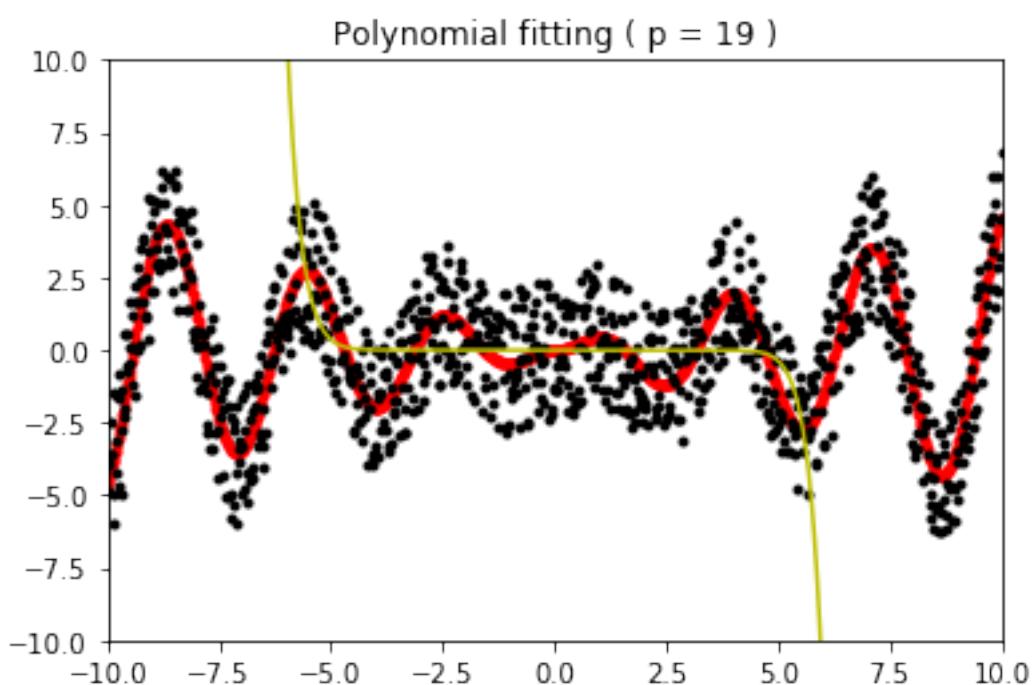
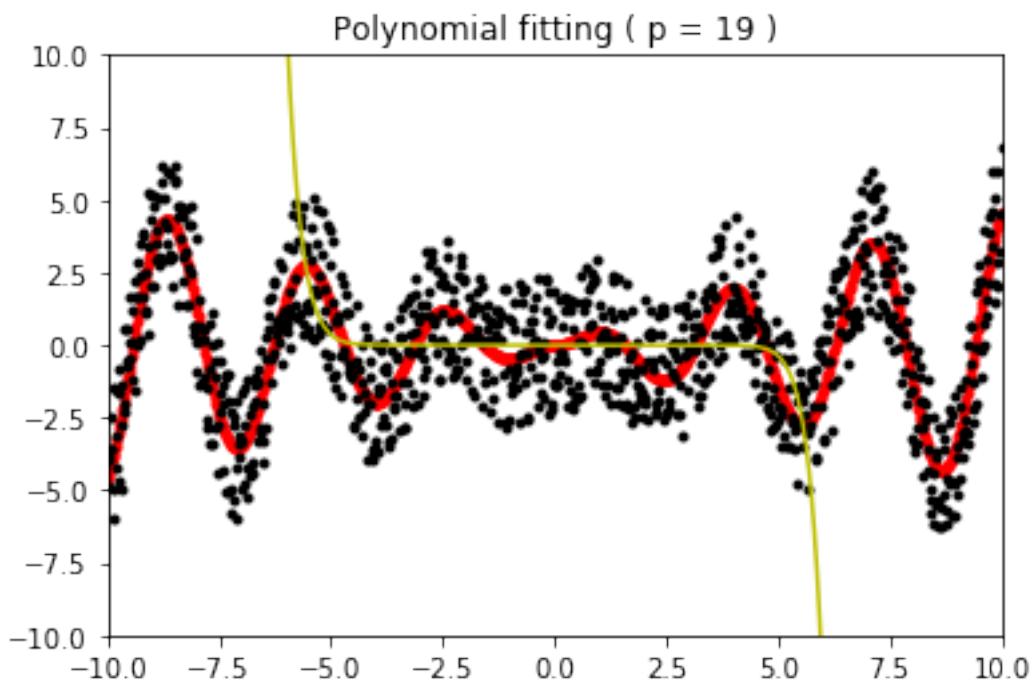


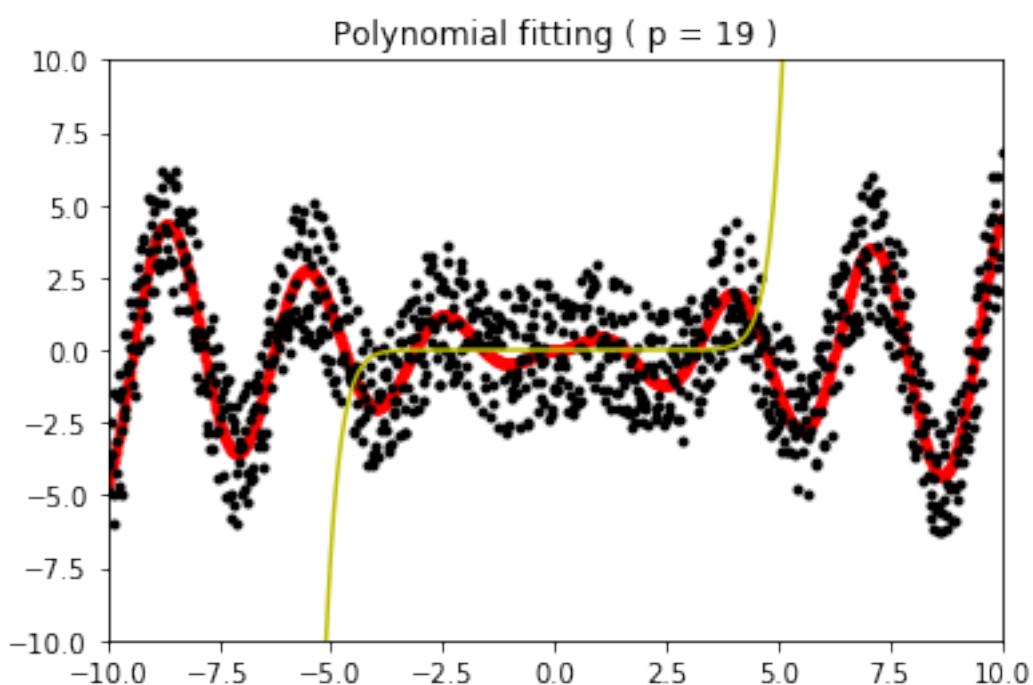
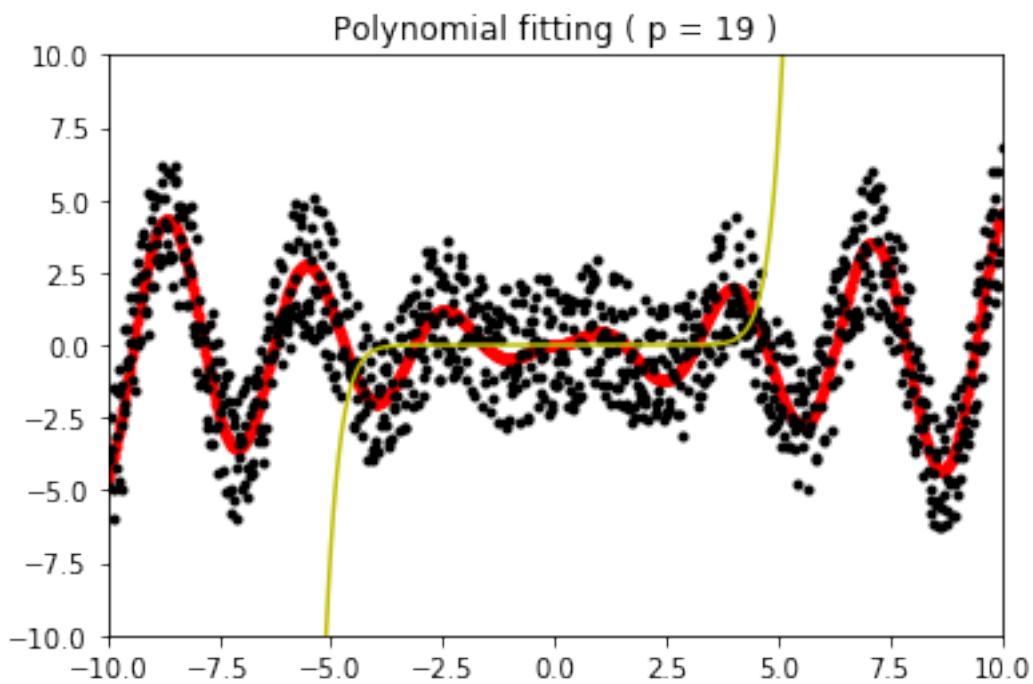


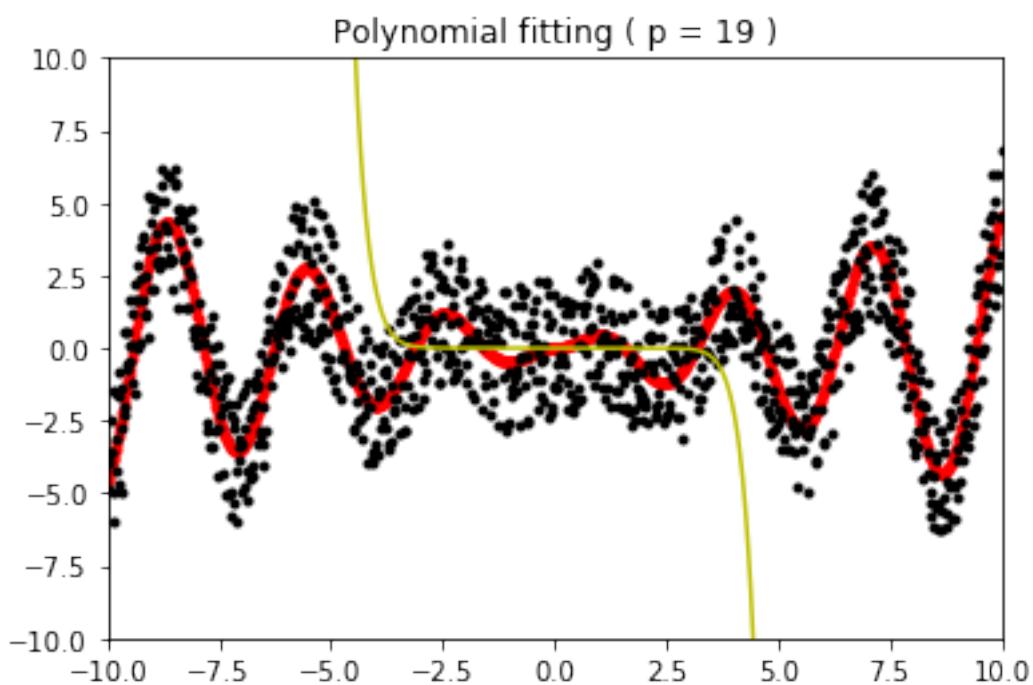
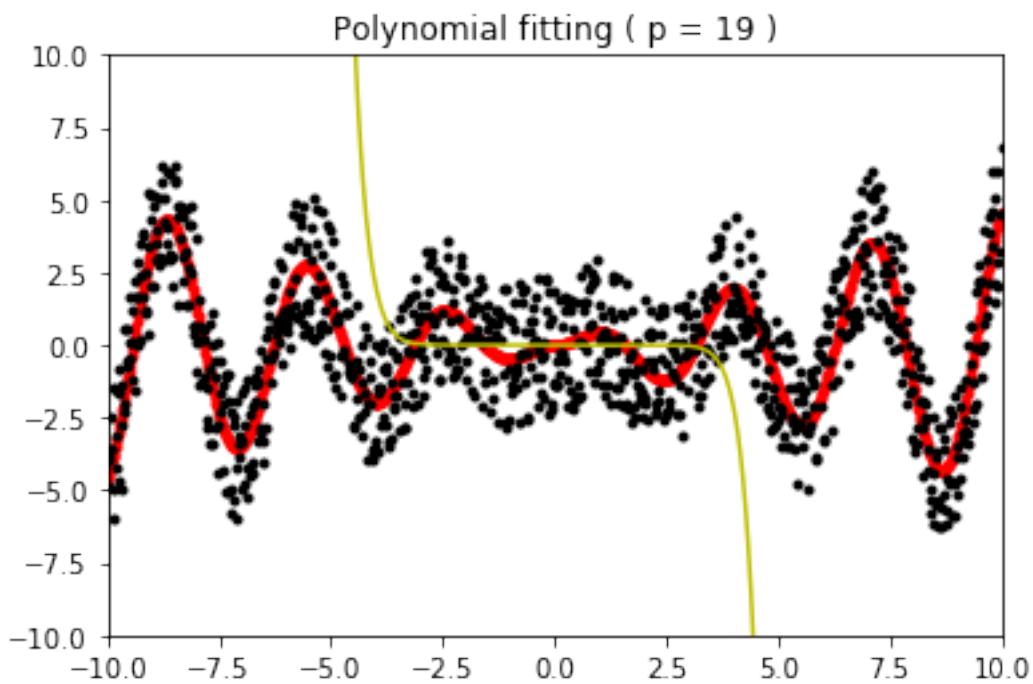


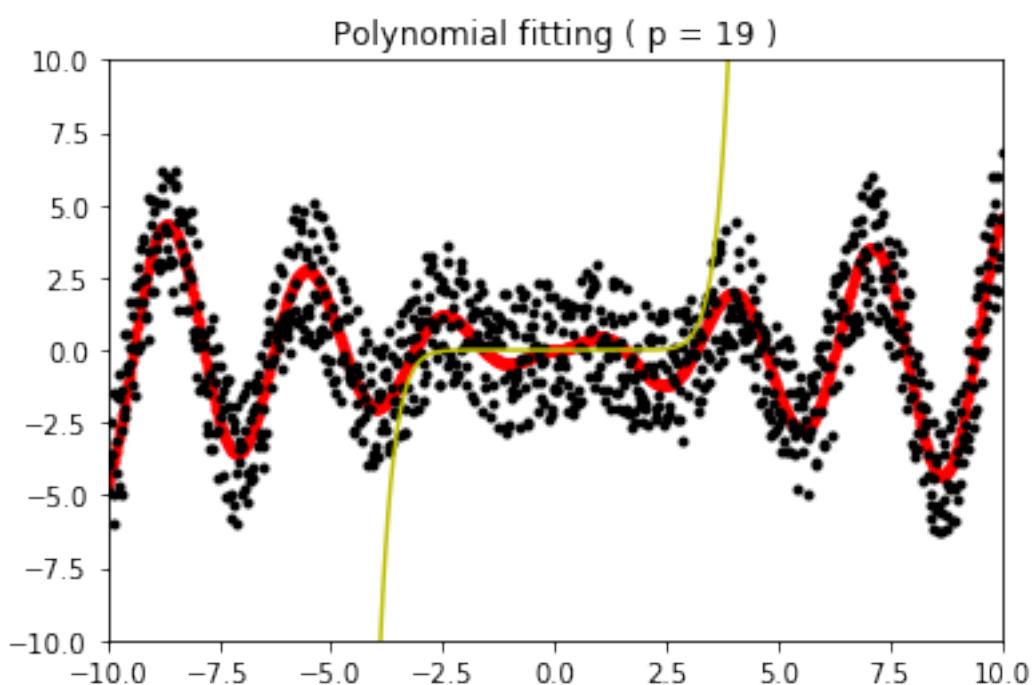
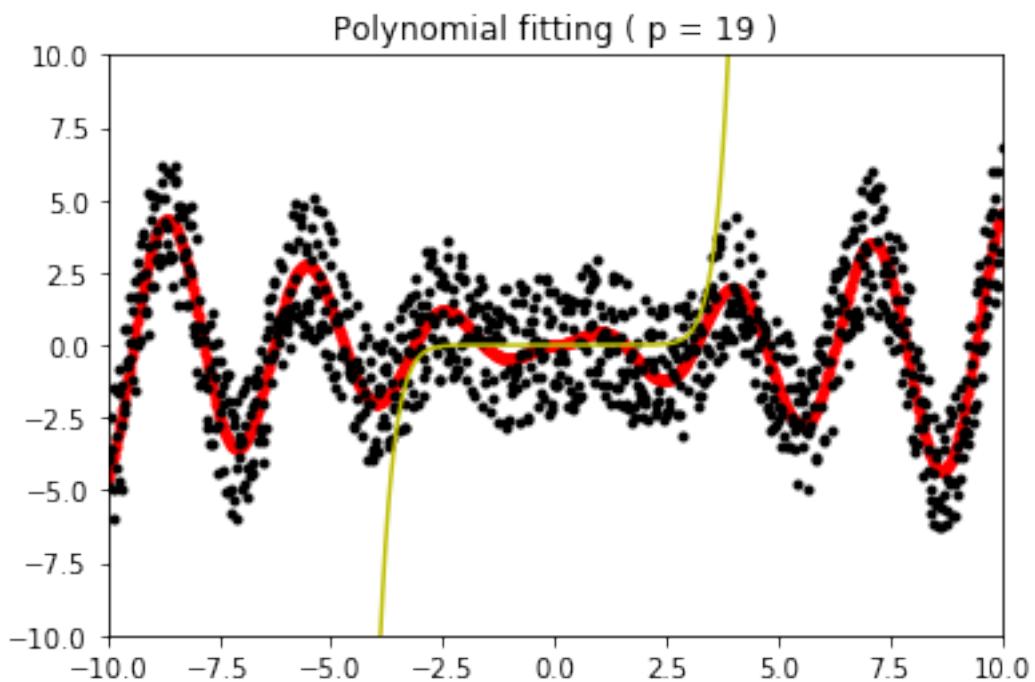


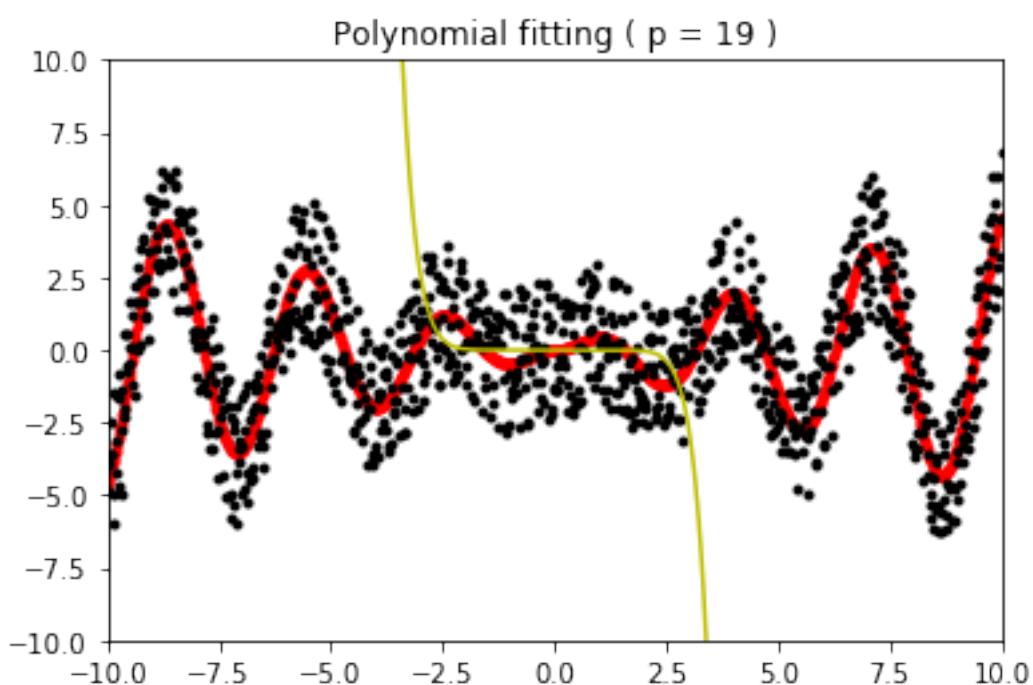
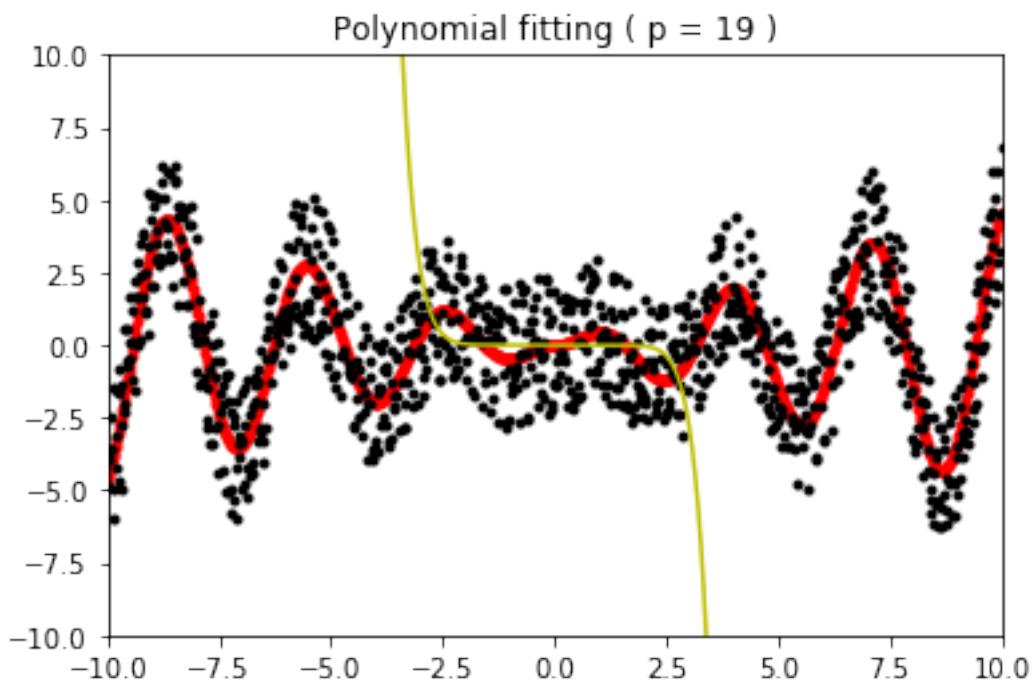
```
(20, 1)
[[-1.89079218e-14]
 [-5.09435142e-18]
 [ 9.63960519e-12]
 [ 2.30353645e-15]
 [-2.07292664e-09]
 [-4.34999578e-13]
 [ 2.43590204e-07]
 [ 4.44277221e-11]
 [-1.69298429e-05]
 [-2.65456213e-09]
 [ 7.03424656e-04]
 [ 9.36351383e-08]
 [-1.68207483e-02]
 [-1.86773426e-06]
 [ 2.10182637e-01]
 [ 1.88350850e-05]
 [-1.11884929e+00]
 [-7.40952236e-05]
 [ 1.51996780e+00]
 [ 4.89346239e-05]]
```

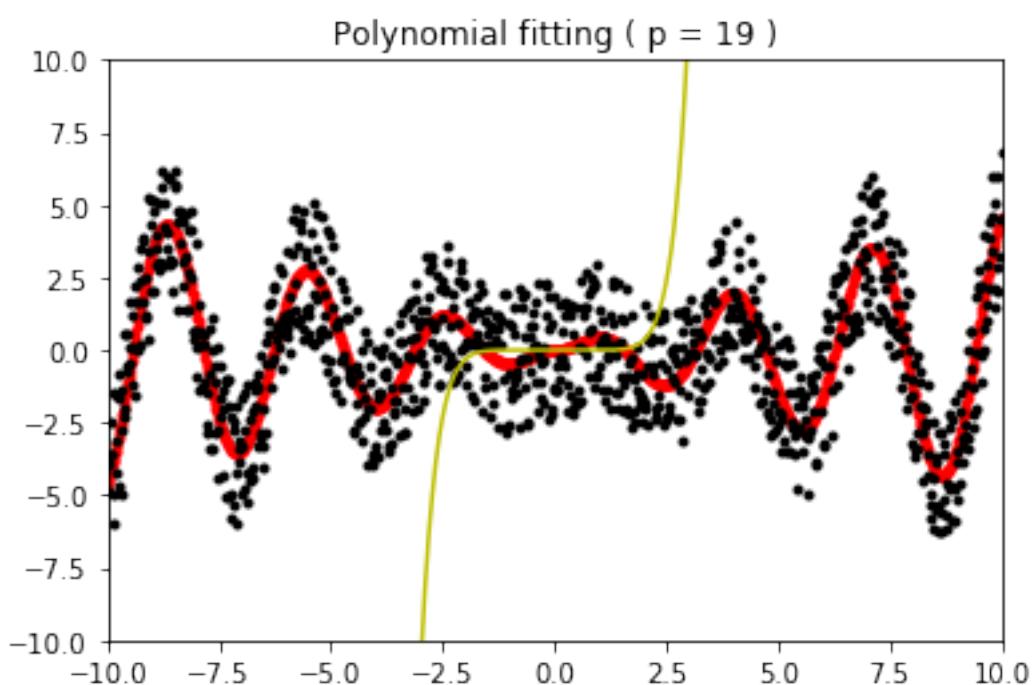
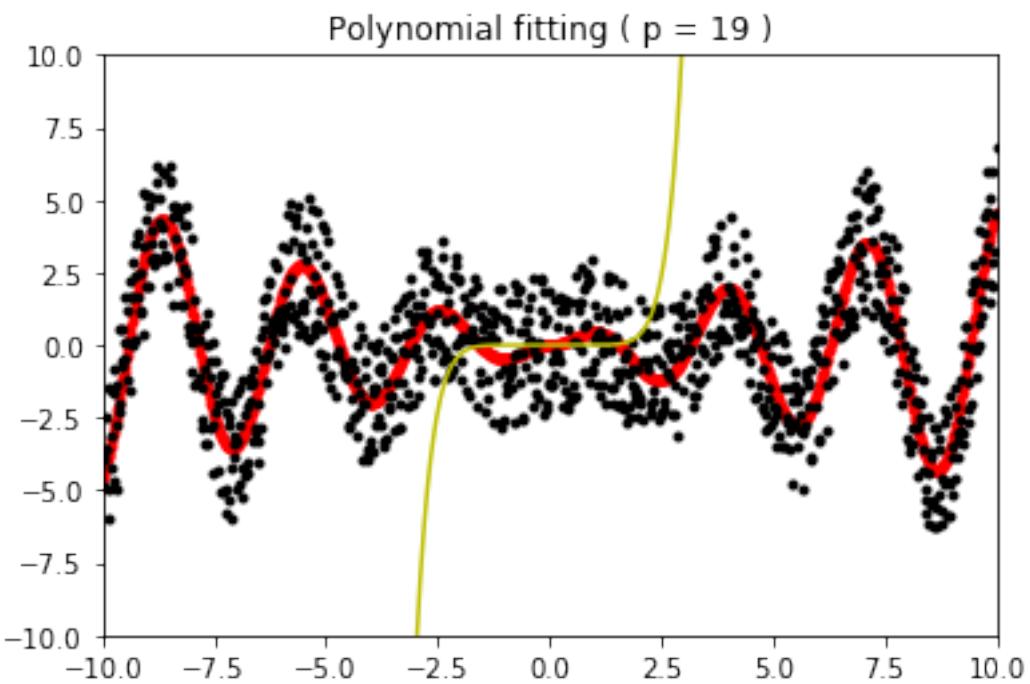


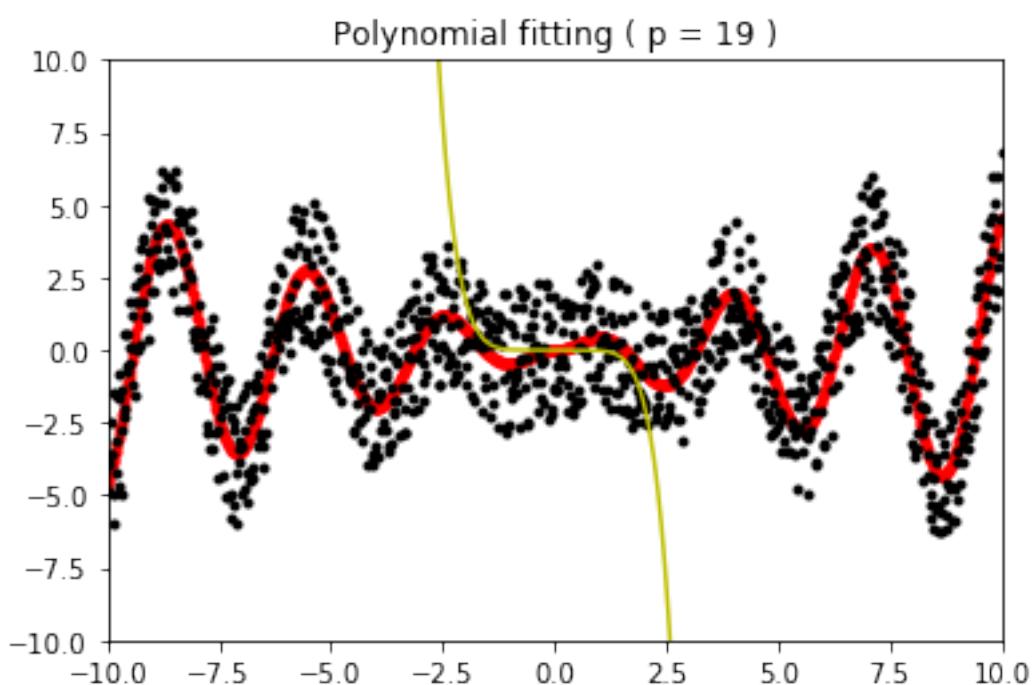
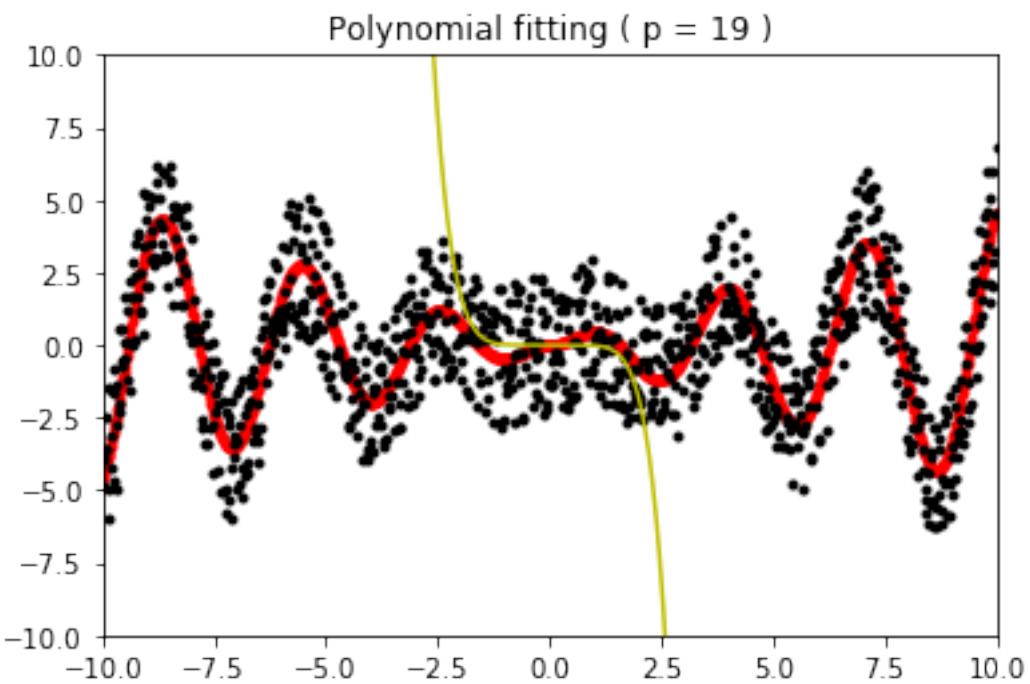


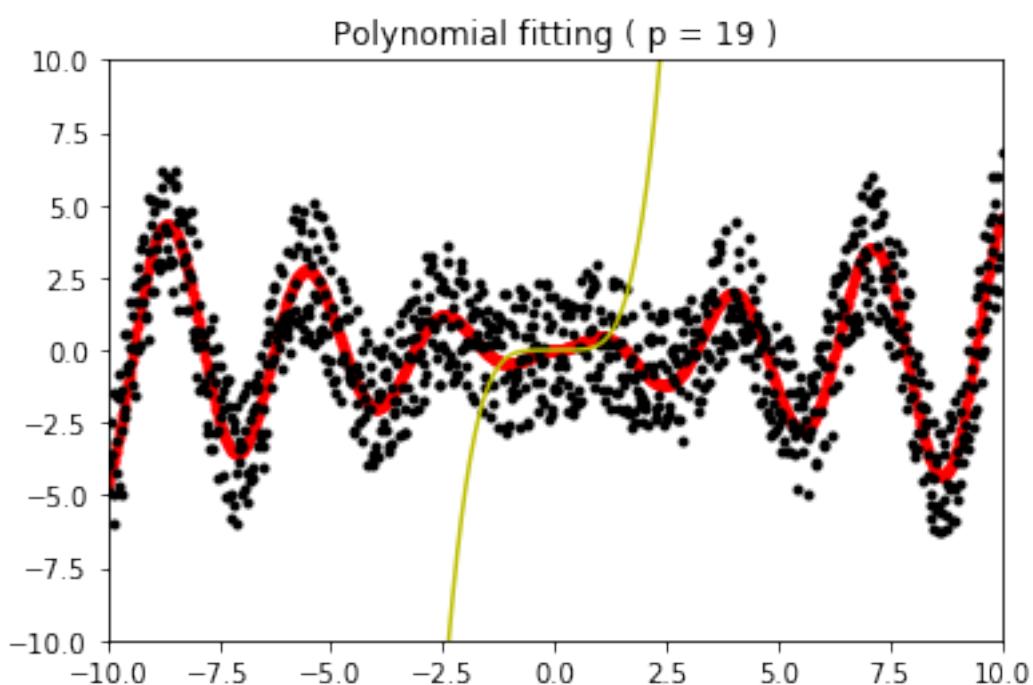
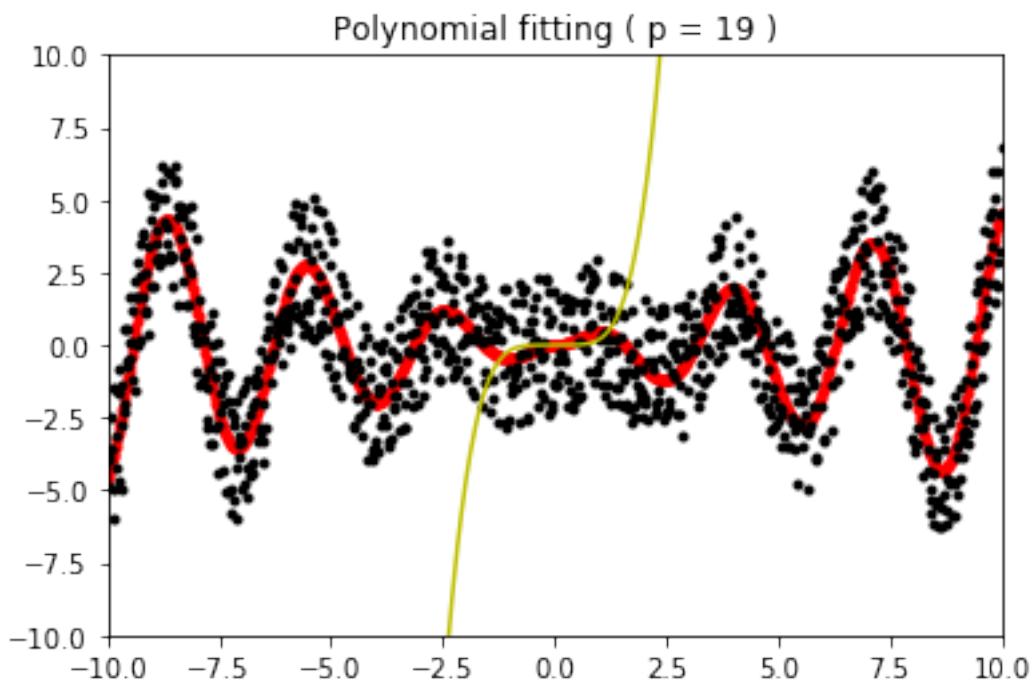


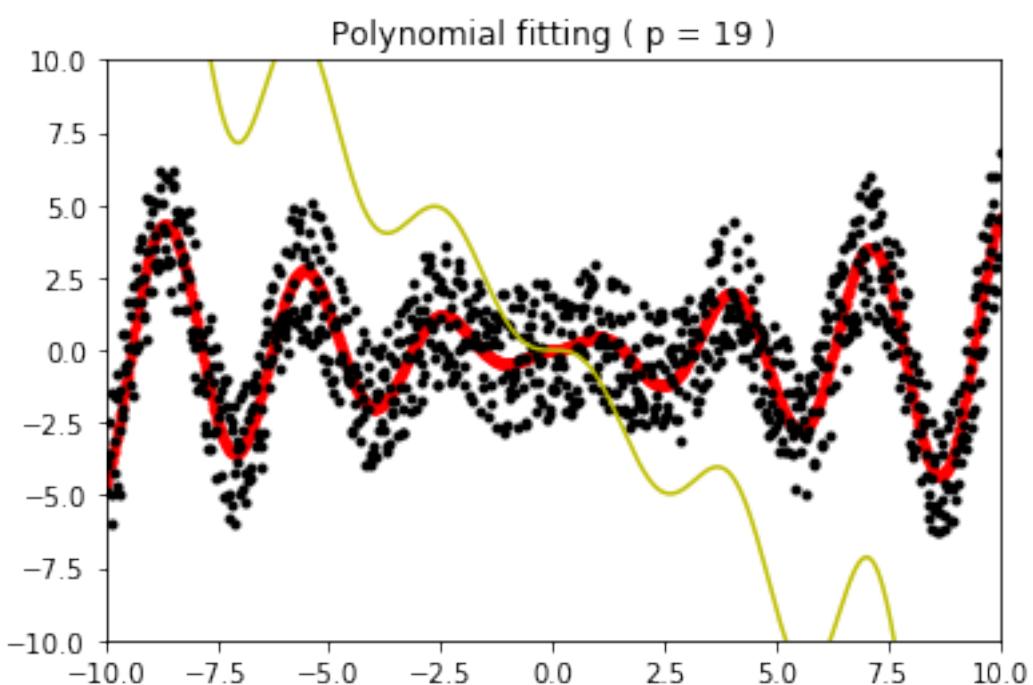
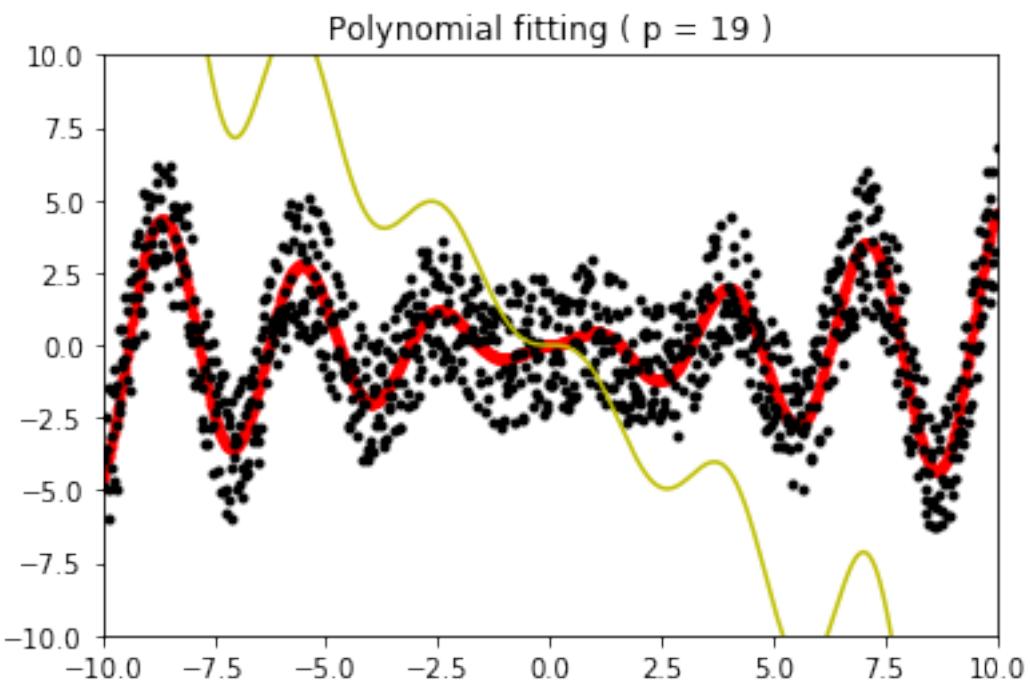


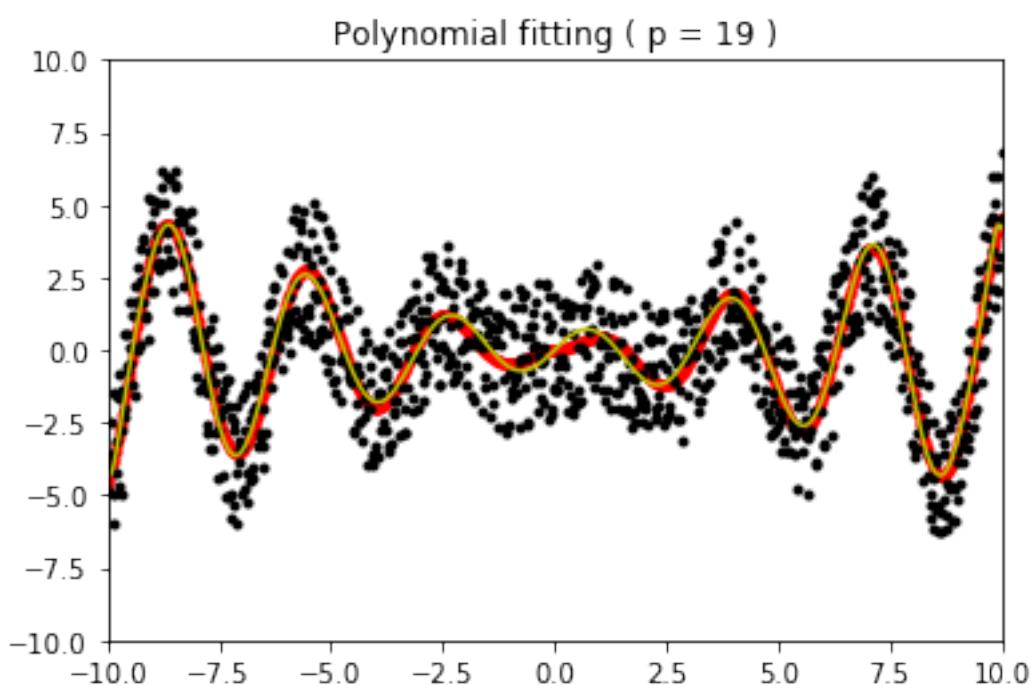
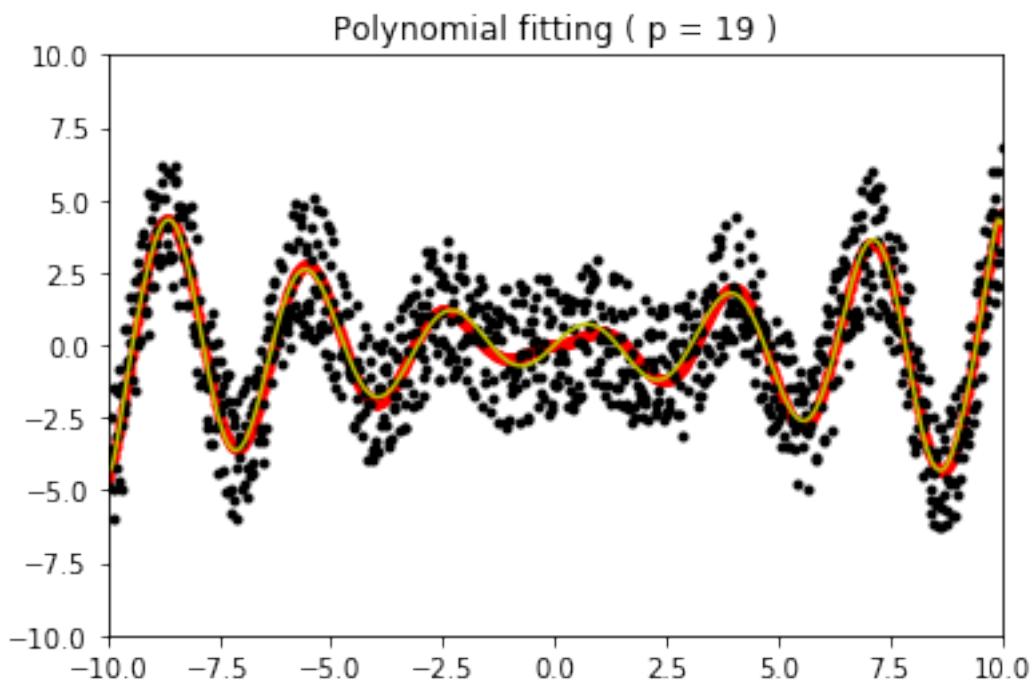












1.4 4. Check Error

```
In [75]: plt.plot(res_x,res_y,'r')
plt.show()
```

