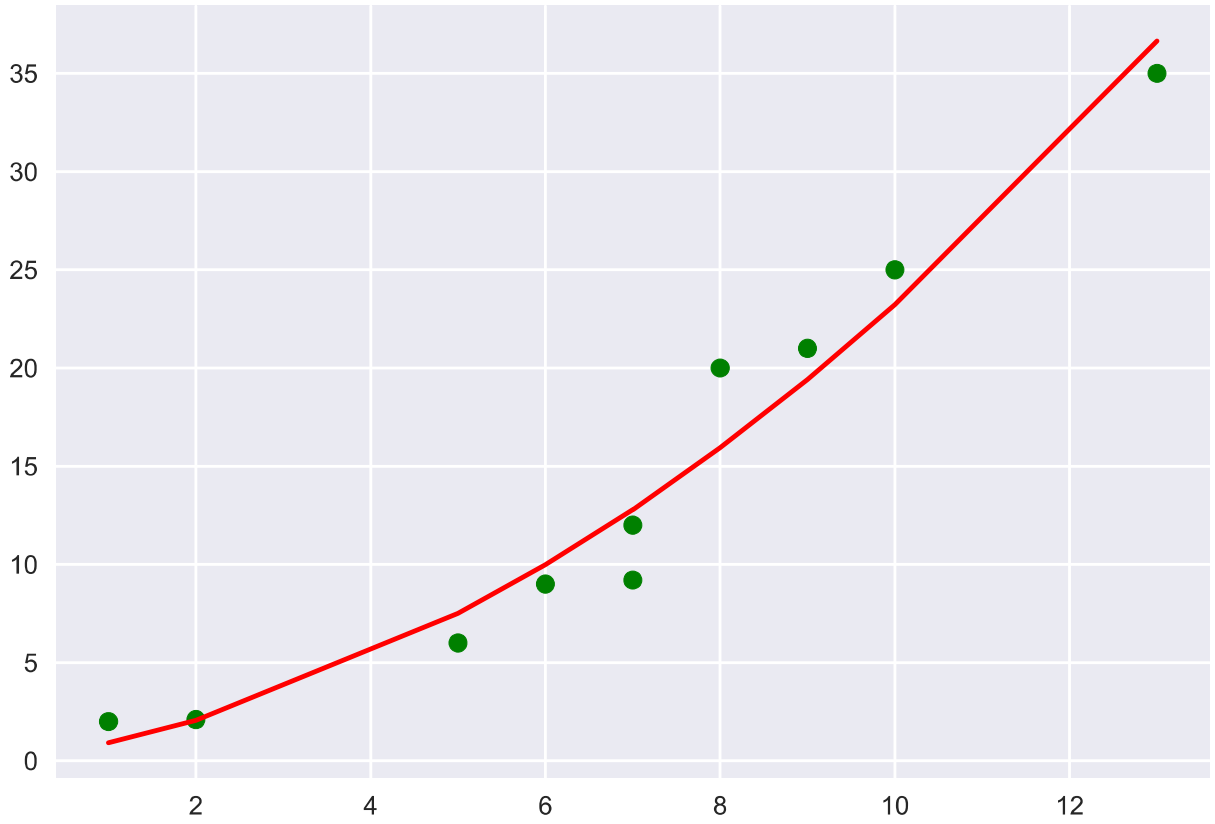
In [3]:

```
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.preprocessing import PolynomialFeatures
import matplotlib.pyplot as plt
import pandas as pd
plt.style.use('seaborn')

df = pd.read_csv("data/polydata.csv")

x_train = df[['x']]
y_train = df['y']

poly = PolynomialFeatures(degree =2)
x_poly = poly.fit_transform(x_train)
model_poly = LinearRegression()
model_poly.fit(x_poly,y_train)

b = model_poly.coef_
a = model_poly.intercept_
y_pred = model_poly.predict(x_poly)
r2 = r2_score(y_train,y_pred)
mse = mean_squared_error(y_train,y_pred)

print("slope (b):",b)
print("intercept (a)",a)
print("R_squared\t : %.2f"%(r2))
print("MSE\t : %.2f"%(mse))

x = model_poly.predict(poly.fit_transform(x_train))
plt.scatter(x_train,y_train,color = "green")
plt.plot(x_train,x,color="red")
plt.show()
```

```
slope (b): [0.         0.65068811 0.1661962 ]
intercept (a) 0.09918047714442046
R_squared        : 0.96
MSE      : 4.28
```

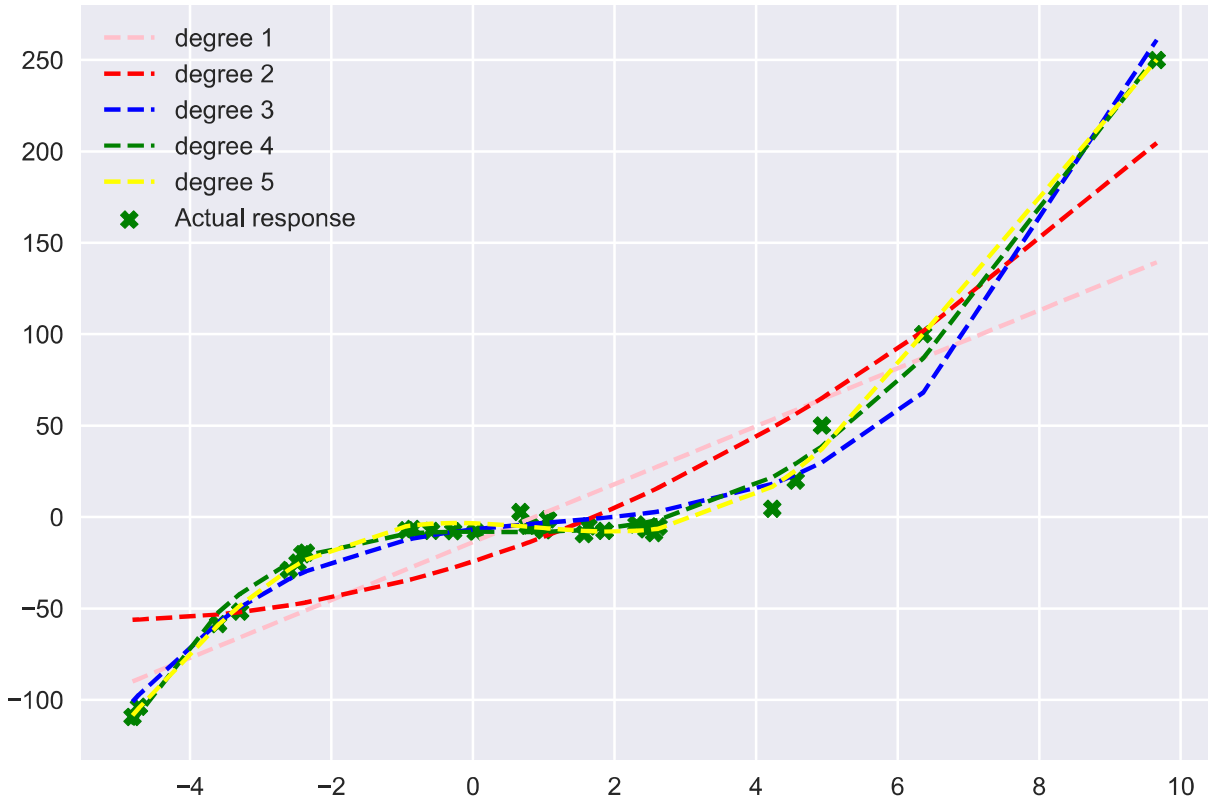


In [47]:

```
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.preprocessing import PolynomialFeatures
import matplotlib.pyplot as plt
import pandas as pd
plt.style.use('seaborn')

df = pd.read_csv("data/rate.csv")

x_train = df[['X']]
y_train = df['Y']

model = LinearRegression()
model.fit(x_train,y_train)
b = model.coef_
a = model.intercept_
print("a\t:",a)
print("b\t:",b)
def calculator(d):
    poly = PolynomialFeatures(degree =d)
    x_poly = poly.fit_transform(x_train)
    model_poly = LinearRegression()
    model_poly.fit(x_poly,y_train)

    b = model_poly.coef_
    a = model_poly.intercept_
    y_pred = model_poly.predict(x_poly)
    r2 = r2_score(y_train,y_pred)
    mse = mean_squared_error(y_train,y_pred)
    print("degree",d)
    print("intercept (a)\t :",a)
```

```
    print("slope (b)\t :",b)
    print("R_squared\t : %.2f"%(r2))
    print("MSE\t\t : %.2f\n"%(mse))

    x = model_poly.predict(poly.fit_transform(x_train))
    return x

cl = ["pink","red","blue","green","yellow"]
for i in range(5):
    x = calculator(i+1)
    plt.plot(x_train,x,color=cl[i], linestyle="dashed",label="degree %d"%(i+1))
plt.scatter(x_train,y_train,color = "green", label="Actual response" ,marker = "X")
plt.legend()
plt.show()
```

```
a       : -13.672444513855144
b       : [15.83788265]
degree 1
intercept (a)    : -13.67244451385514
slope (b)        : [ 0.         15.83788265]
R_squared        : 0.74
MSE              : 922.21

degree 2
intercept (a)    : -24.239740051055623
slope (b)        : [ 0.         12.30363176  1.17853488]
R_squared        : 0.83
MSE              : 603.75

degree 3
intercept (a)    : -6.763340433956019
slope (b)        : [ 0.          4.51474384 -1.29385509  0.38261754]
R_squared        : 0.97
MSE              : 90.30

degree 4
intercept (a)    : -8.055980946752921
slope (b)        : [ 0.         -0.03873876 -0.56502237  0.67230765 -0.0335979 ]
R_squared        : 0.99
MSE              : 38.81

degree 5
intercept (a)    : -3.3269458169115307
slope (b)        : [ 0.         -1.18316686 -2.37404976  0.85960019  0.0386523  -0.00
743062]
R_squared        : 0.99
MSE              : 18.64
```


degree 1
degree 2
degree 3
degree 4
degree 5
Actual response
250
200
150
100
50
0
−50
−100
−4
−2
0
2
4
6
8
10

In [ ]: