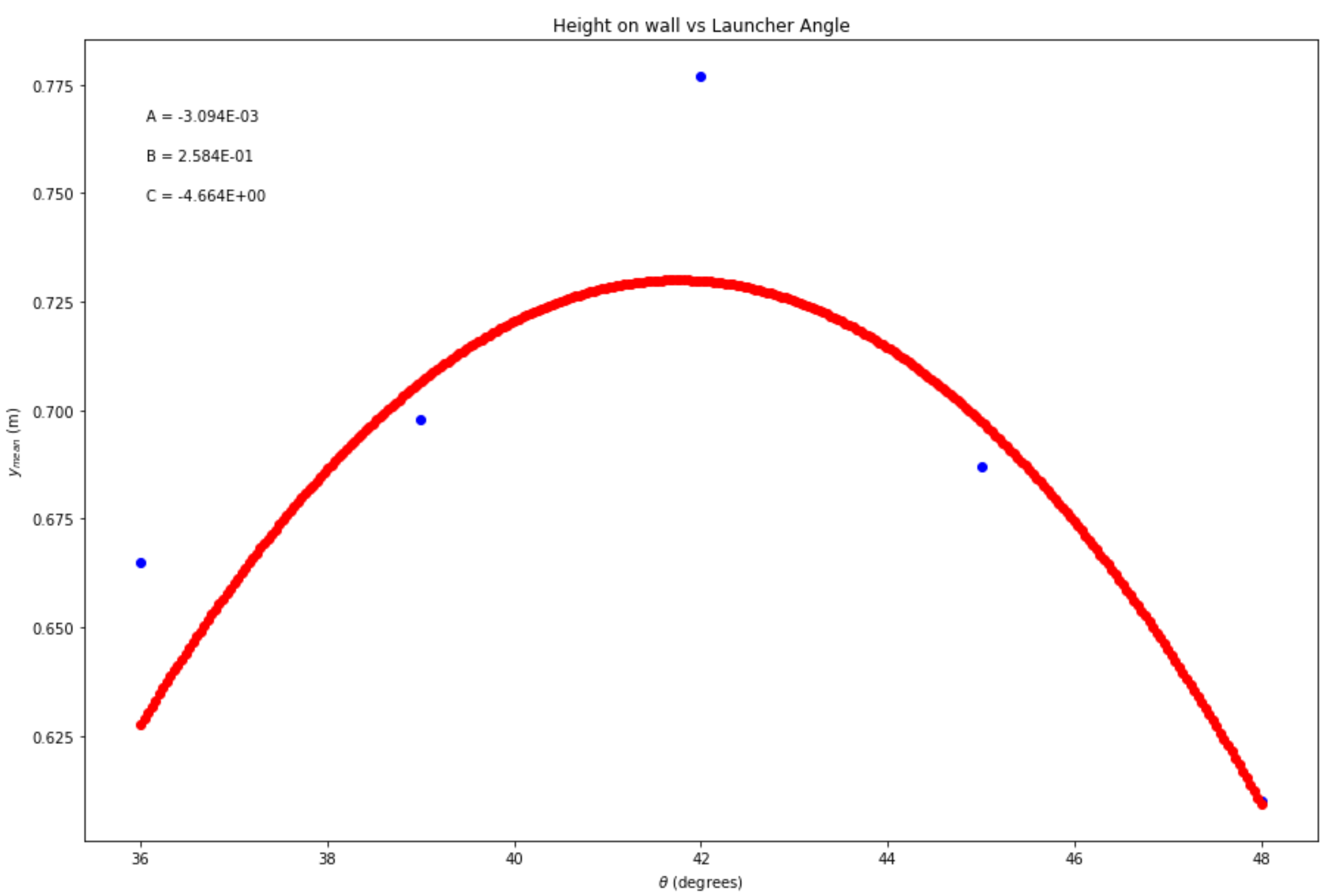
In [1]:

```
import numpy as np
import matplotlib.pyplot as plt


%matplotlib inline


#Create arrays for your data
theta_array=np.array([36, 39, 42, 45, 48])
ymean_array=np.array([.665, .698, .777, .687, .610])
yerr_array=np.array([.003, .014, .008, .015, .031])

#Create an array for your y-axis uncertainties

#Reassign variables
x = theta_array
y = ymean_array
dy = yerr_array

#size the plot
plt.figure(figsize=(15,10))

#create scatter plot
plt.scatter(x, y, color='blue', marker='o')

#create labels
plt.xlabel('$\\theta$ (degrees)')
plt.ylabel('$y_{mean}$ (m)')
plt.title('Height on wall vs Launcher Angle')

#fitting to a 2nd degree polynomial
c,b,a=np.polynomial.polynomial.polyfit(x,y,2,w=dy)

#Annotate with values of A, B, C from best fit polynomial
plt.annotate('A = {value:.{digits}E}'.format(value=a, digits=3),
             (0.05, 0.9), xycoords='axes fraction')

plt.annotate('B = {value:.{digits}E}'.format(value=b, digits=3),
             (0.05, 0.85), xycoords='axes fraction')

plt.annotate('C = {value:.{digits}E}'.format(value=c, digits=3),
             (0.05, 0.8), xycoords='axes fraction')
#Create fit line
xnew = np.linspace(x.min(), x.max(), 300)
fit = a*xnew**2 + b*xnew +c

plt.scatter(xnew, fit, color='red')
plt.show()

print ("C = ",c , "B = ",b, "A = ",a)
```




```
C =  -4.6643820584488225 B =  0.2583922596534471 A =  -0.00309424607587506
```

In [ ]: