In [20]:
```
# We are aware that if one game is played, it is impossible to get two losses
 in a row
# Hence set F(1) = 0 where F is the probability
```

In [21]:
```
# F(2) can only happen when both games are losses. Since each loss is an indep
endent event
# F(2) = P_L * P_L where P_L = 0.2 and P_W = 0.8
```

In [22]:
```
# A function can be created to recursively calculate F(82) by checking the pre
vious outcomes
# since we need to have two losses in a row.
# F(n) = P_W * F(n-1) + P_L * ( P_L + P_W * F(n-2))
# The first part of the equation states that if game n is a win, you multiply
 the probability it's a win times the
# outcome of the previous game. The second part states that if game n is a los
s, you the probability it's a loss
# times the probability that the previous game was a loss (hence two losses in
 a row) plus the probability that
# two games before it was a win and thus you continue recursively.
```

In [23]:
```
# Using this formula and the constants for F(1) and F(2) we can iteratively fi
nd the solution using a loop
# In addition, the formula can be solved analytically to get F(82) directly.
```

In [24]:
```
import numpy as np
# create an iterative list from games 3 to 82 to run the formula through
games = np.arange(3,83)
games
```

Out[24]:
```
array([ 3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,
       20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36,
       37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53,
       54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70,
       71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82])
```

In [25]:
```
# create a list that will store the percentage of consecutive losses and hard
 code the first two values
consecutive_losses = []
consecutive_losses.append(0)
consecutive_losses.append(0.2*0.2)
```

In [26]:
```
consecutive_losses
```

Out[26]:
```
[0, 0.04000000000000001]
```

In [33]:
```
# initialize variable
percent = 0
for game in games:
    # F(n) = P_W * F(n-1) + P_L * ( P_L + P_W * F(n-2))
    percent = 0.8 * consecutive_losses[-1] + 0.2 * (0.2 + 0.8 * consecutive_lo
sses[-2])
    consecutive_losses.append(percent)
```

In [32]:
```
consecutive_losses
```

```
Out[32]: [0,
          0.04000000000000001,
          0.07200000000000001,
          0.10400000000000001,
          0.13472,
          0.164416,
          0.193088,
          0.22077696000000002,
          0.24751564800000003,
          0.27333683200000003,
          0.29827196928000005,
          0.3223514685440001,
          0.3456046899200001,
          0.36805998690304015,
          0.38974473990963215,
          0.41068538983219216,
          0.4309074702512949,
          0.45043563857418667,
          0.46929370609955656,
          0.4875046670515152,
          0.5050907266171412,
          0.5220733280219554,
          0.538473178676307,
          0.5543102754245585,
          0.5696039289278559,
          0.5843727872102141,
          0.5986348583966283,
          0.612407532670937,
          0.62570760348021,
          0.638551288011518,
          0.650954246966048,
          0.6629316036546813,
          0.6744979624383127,
          0.6856674265353992,
          0.6964536152184495,
          0.7068696804204235,
          0.7169283227712908,
          0.7266418070843004,
          0.736021977310847,
          0.7450802709821657,
          0.753827733155468,
          0.7622750298815211,
          0.7704324612100919,
          0.778309973749117,
          0.7859171727929084,
          0.7932633340341855,
          0.8003574148742137,
          0.8072080653448407,
          0.8138236386557468,
          0.8202122013797719,
          0.826381543288737,
          0.8323391868517532,
          0.8380923964076006,
          0.843648187022361,
          0.849013333043105,
          0.8541943763580618,
          0.8591976343733463,
```

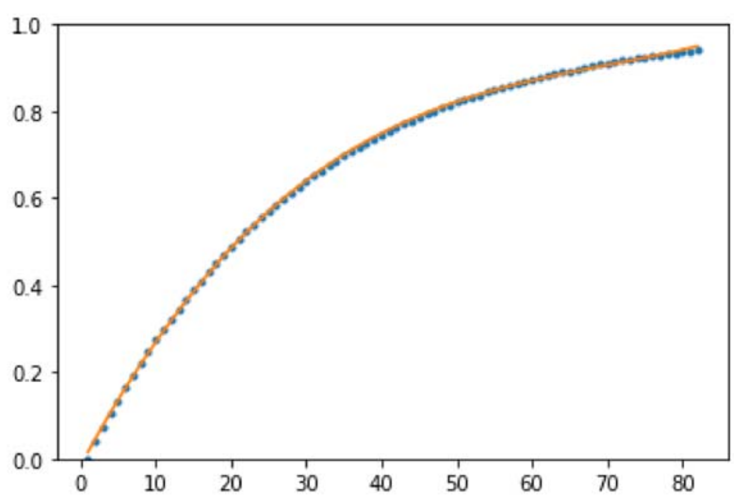
```
 0.864029207715967,
 0.8686949876725092,
 0.8732006633725621,
 0.8775517287256511,
 0.881753489120131,
 0.8858110678922091,
 0.8897294125729883,
 0.8935133009211442,
 0.8971673467485937,
 0.9006960055462581,
 0.9041035799167816,
 0.9073942248208267,
 0.9105719526433464,
 0.9136406380860094,
 0.916604022891743,
 0.919465720407156,
 0.9222292199884037,
 0.924897891255868,
 0.927474988202839,
 0.9299636531632102,
 0.9323669206430224,
 0.9346877210205315,
 0.9369288841193089,
 0.9390931426587322,
 0.9411831355860754]
```

In [29]:

```
full_games = np.arange(1,83)
```

In [31]:

```
import matplotlib.pyplot as plt
%matplotlib inline
z = np.polyfit(full_games, consecutive_losses, 3)
p = np.poly1d(z)
xp = np.linspace(1, 82, 82)
_ = plt.plot(full_games, consecutive_losses, '.', xp, p(xp), '-',)
plt.ylim(0,1)
plt.show()
```



In [ ]: