THE REAL WORLD HAS NOISE

(or finite samples, usually both)

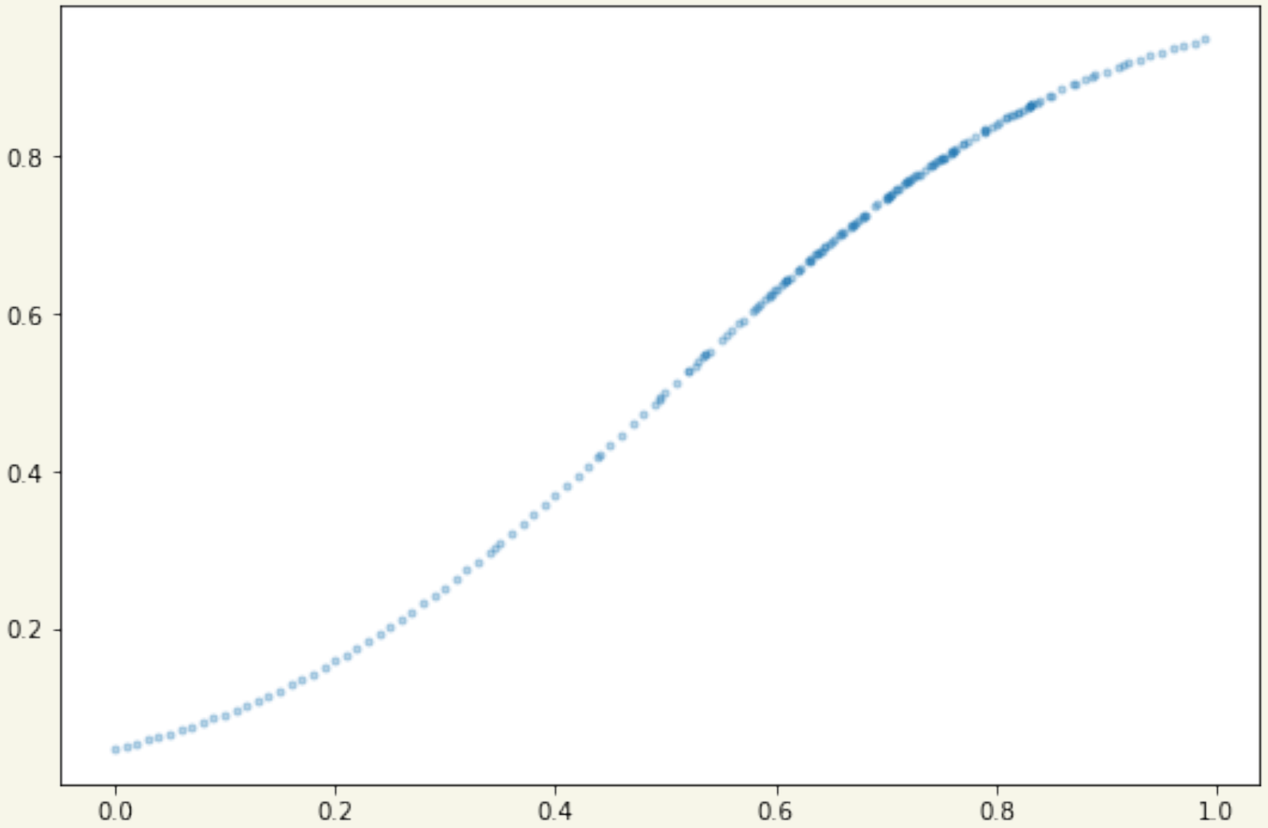
Part of:

The Essence of Learning

All of AI is the estimation of functions.

Univ.AI

Our Example



Consider a very simple scenario, where the probability of voting for Romney is a function only of how religious the population in a county is.

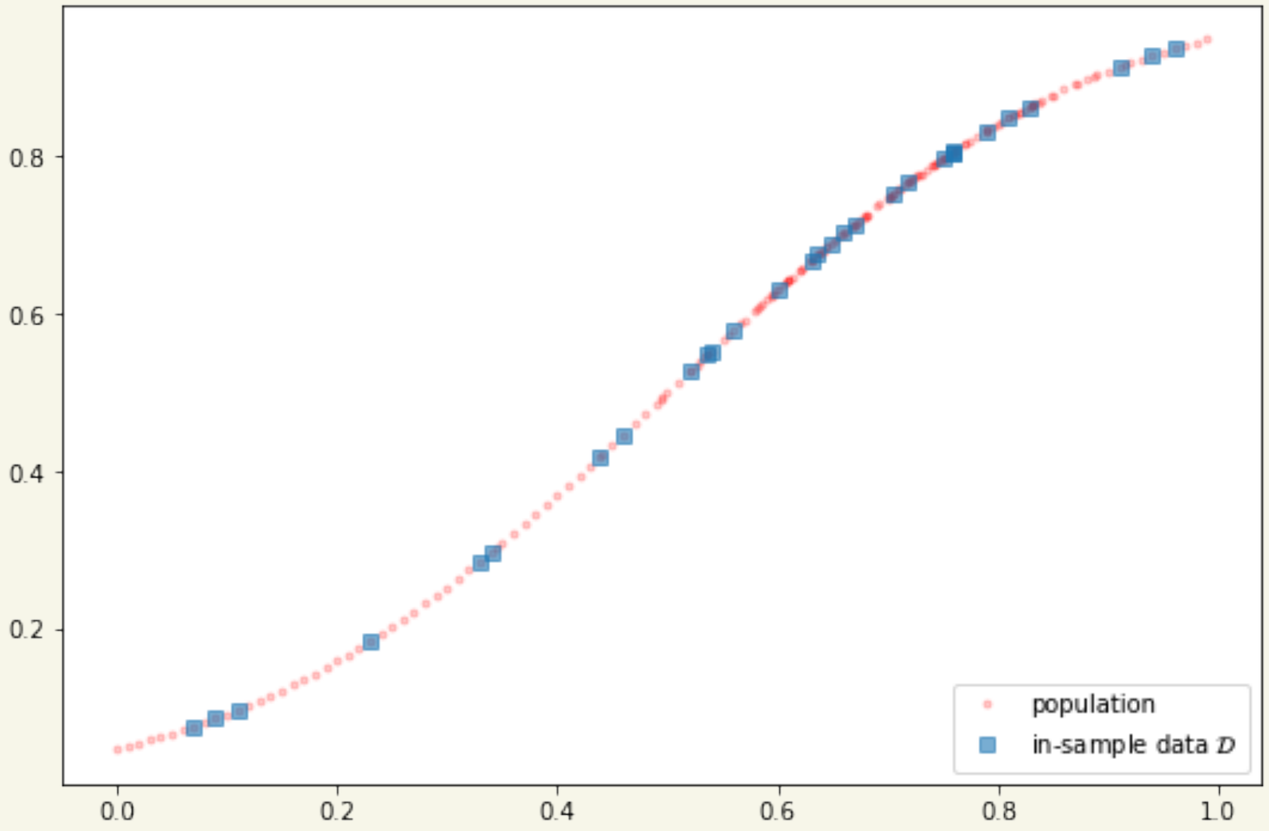
In other words y_i is data that pollsters have taken which tells us their estimate of the fraction of people voting for Romney and x_i is the fraction of religious people in county i.

Let us assume that we have a "population" of 200 counties.

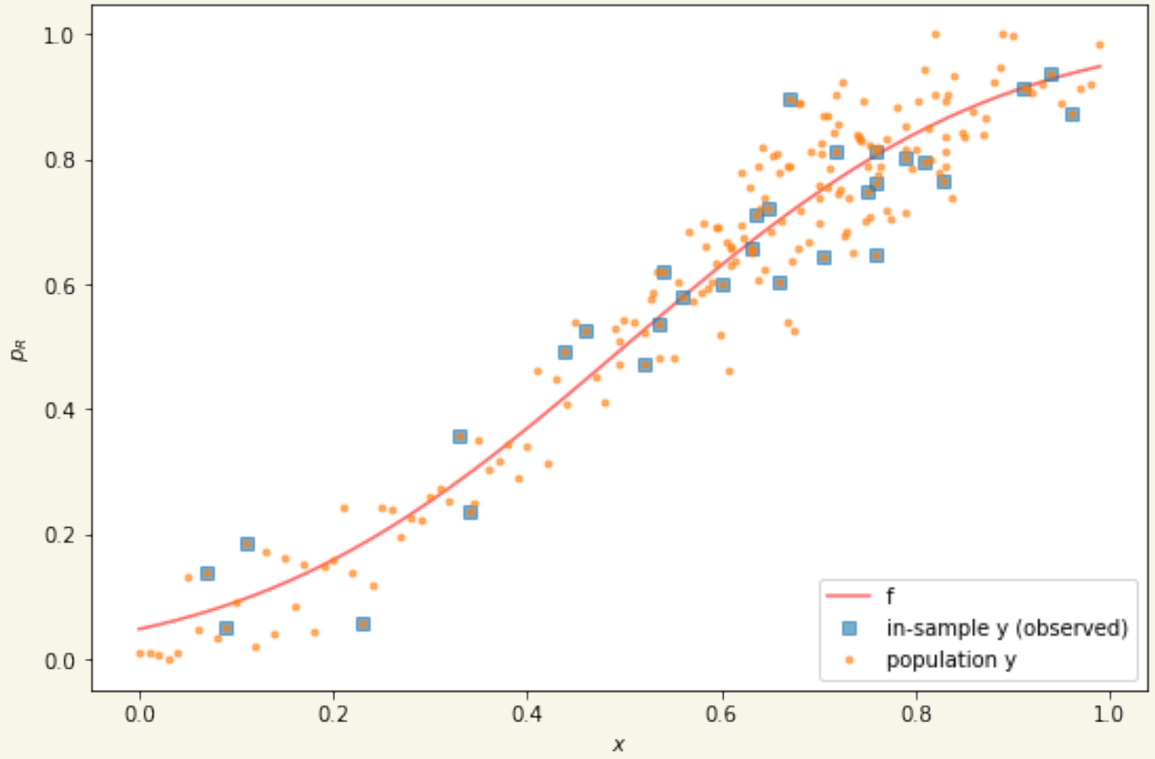
A sample from this population

Lets assume that out of this population of 200 points we are given a sample $\mathcal{D}$ of 30 data points. Such data is called **in-sample data**. Contrastingly, the entire population of data points is also called **out-of-sample data**.

Now pretend the red dots are taken away, and you are left with the blue squares. Our job look at different hypotheses and find the best one amongst them.



A noisy population (and thus sample)



Now let us assume there is some noise in our measurements..

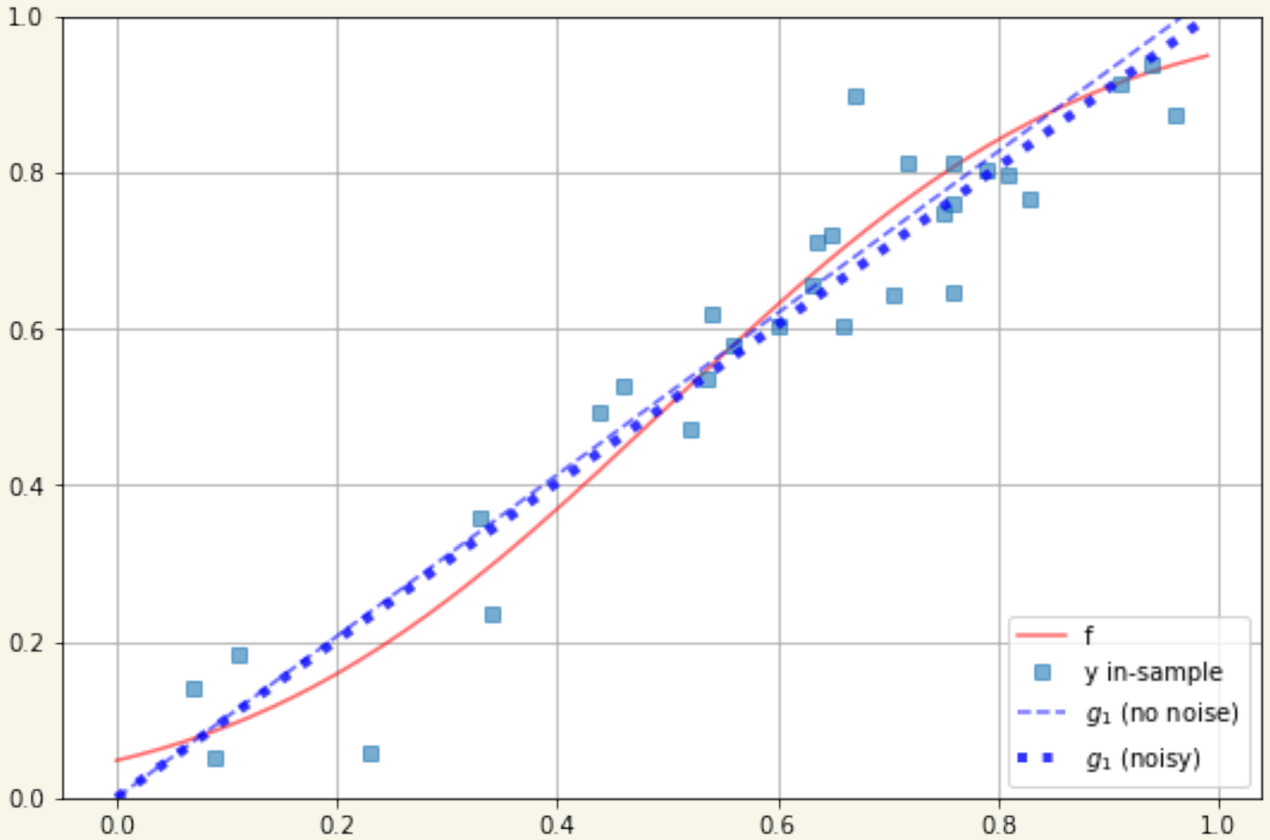
Where does the noise come from?

Consider for example two counties, one with $x = 0.8$ fraction of self-identified religious people in the county, and another with $x = 0.82$. Based on historical trends, if the first county was mostly white, the fraction of those claiming they would vote for Romney might be larger than in a second, mostly black county. Thus you might have two very y's next to each other on our graphs.

We wish to estimate a function $f(x)$ so that the values y_i come from the function f and include some noise. What we have done is introduced a noisy target y, so that

$$y = f(x) + \epsilon$$

Fitting using straight lines



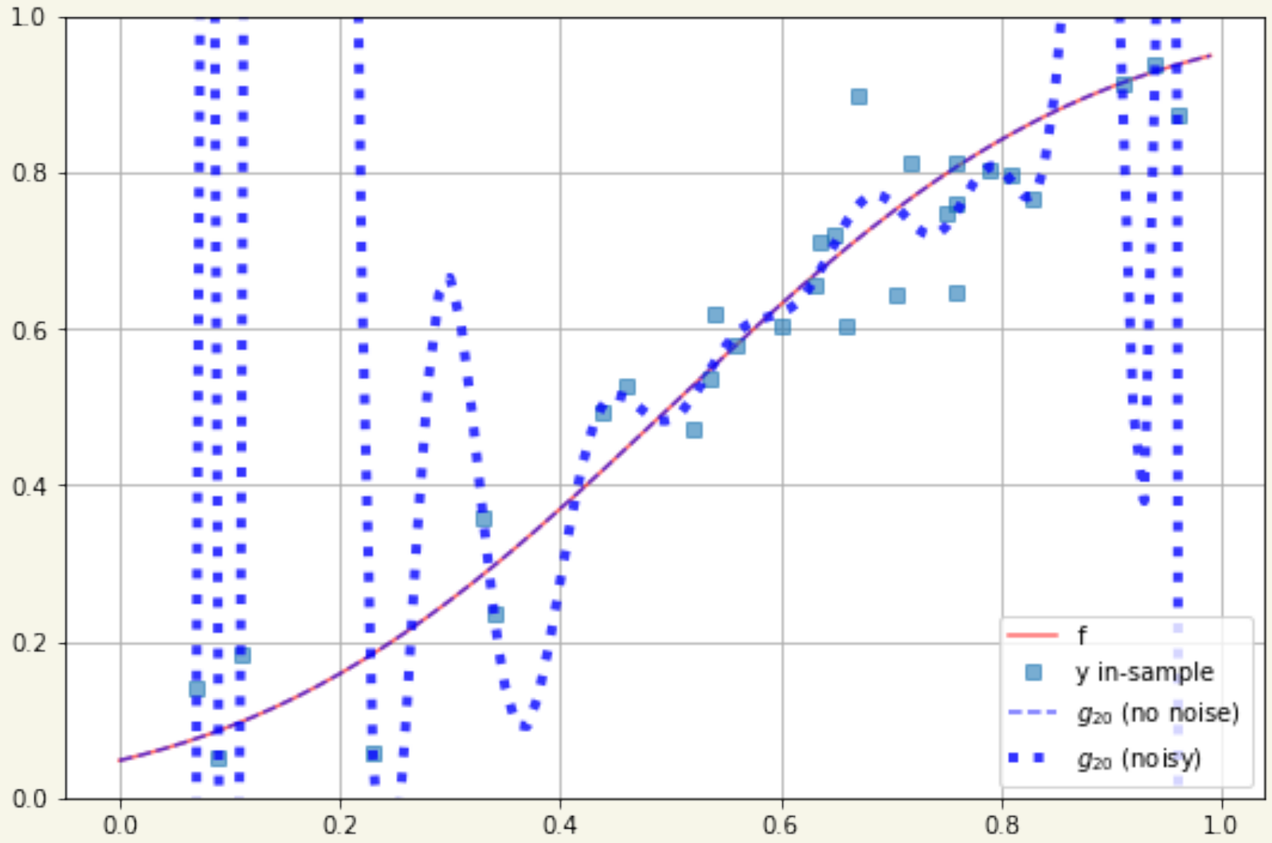
The noise changes the best fit line ($g_1(x)$) by a little but not by much. The best fit line still does a very poor job of capturing the variation in the data, after the target is noisy

Fitting using 20th order polynomials

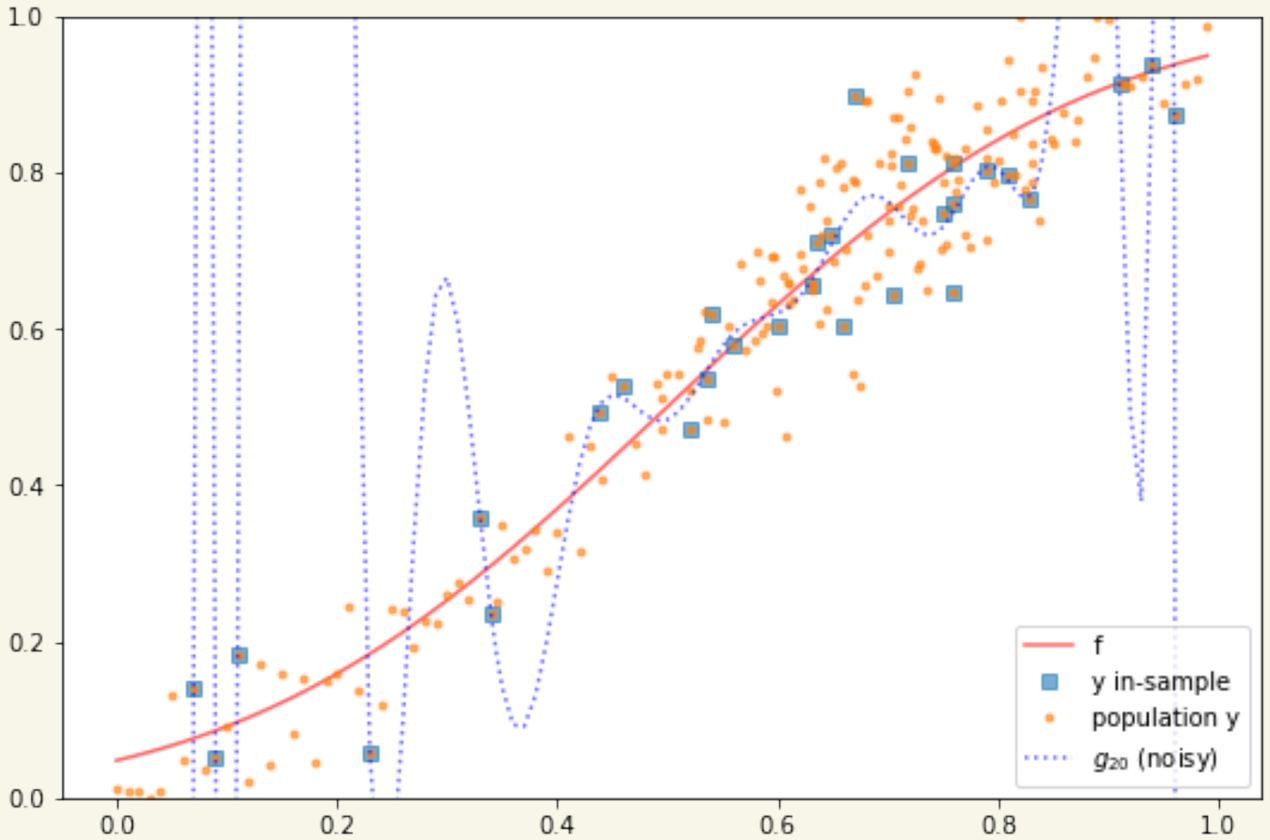
For the 20th order polynomial, the results are (to put it mildly) very interesting.

The best fit 20th order polynomial, g_{20}, tries to follow all the curves of the observations..in other words, **it tries to fit the noise**.

The curve goes through or near all the points in our sample. But how does it do on the poulation?



Overfitting



The best fit g_{20} has **overfit** to the sample. Look on the left of the figure where it is nowhere near the orange population points.

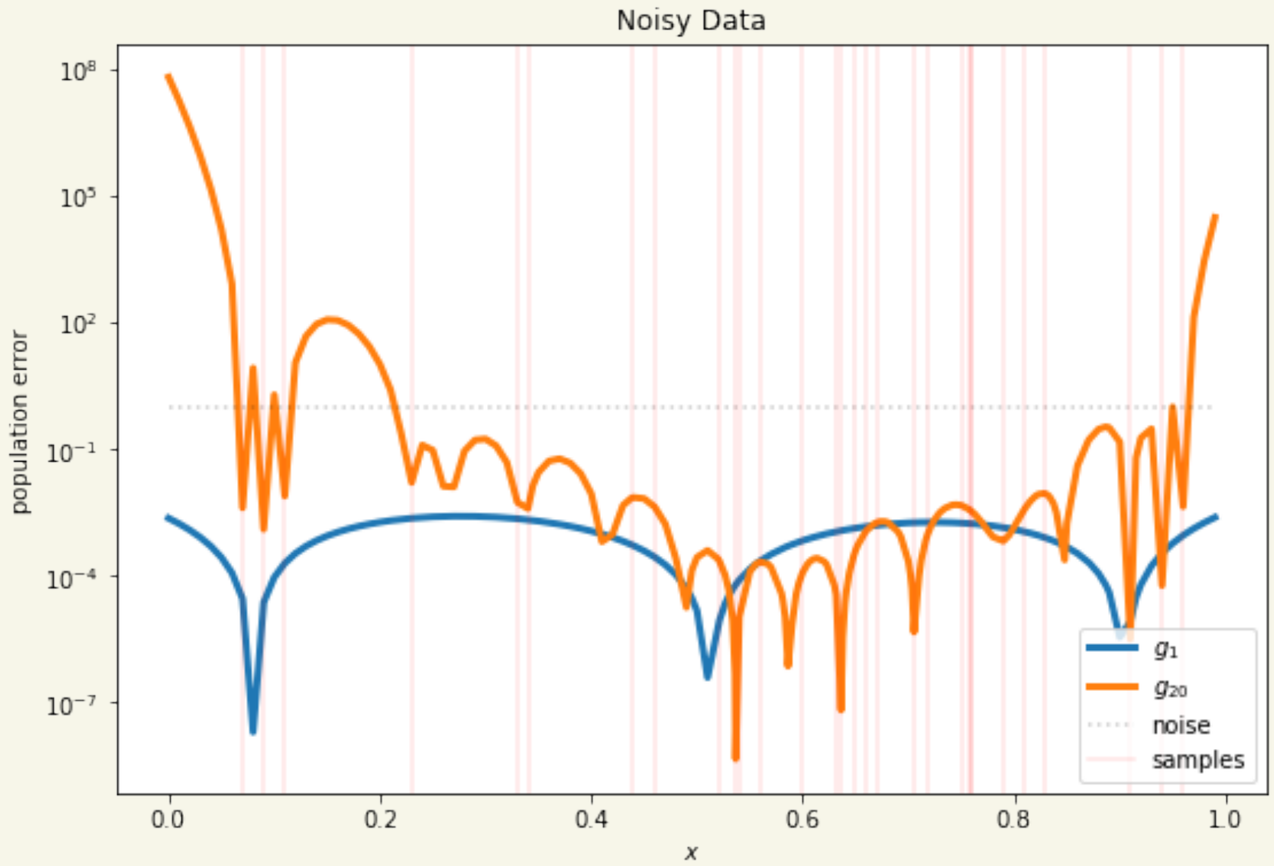
This fitting to the noise is a danger you will encounter again and again in learning. Its called **overfitting**. So, $\mathcal{H}_{20}$ which seemed to be such a good candidate hypothesis space in the absence of noise, ceases to be one. The take away lesson from this is that we must further ensure that our **model does not fit the noise**.

$\mathcal{H}_1$ is now a better hypothesis space

g_1 is an underfit model. Its straightness cannot capture the curves of the target function, with or without noise. Yet it is now a better model with less point-wise error on the population.

It atleast captures the zeitgeist of the target, as compared to the wildly hairy g_{20}.

How do we systematically decide what is underfit and what is overfit? How do we find a hypothesis set of the "correct" complexity?



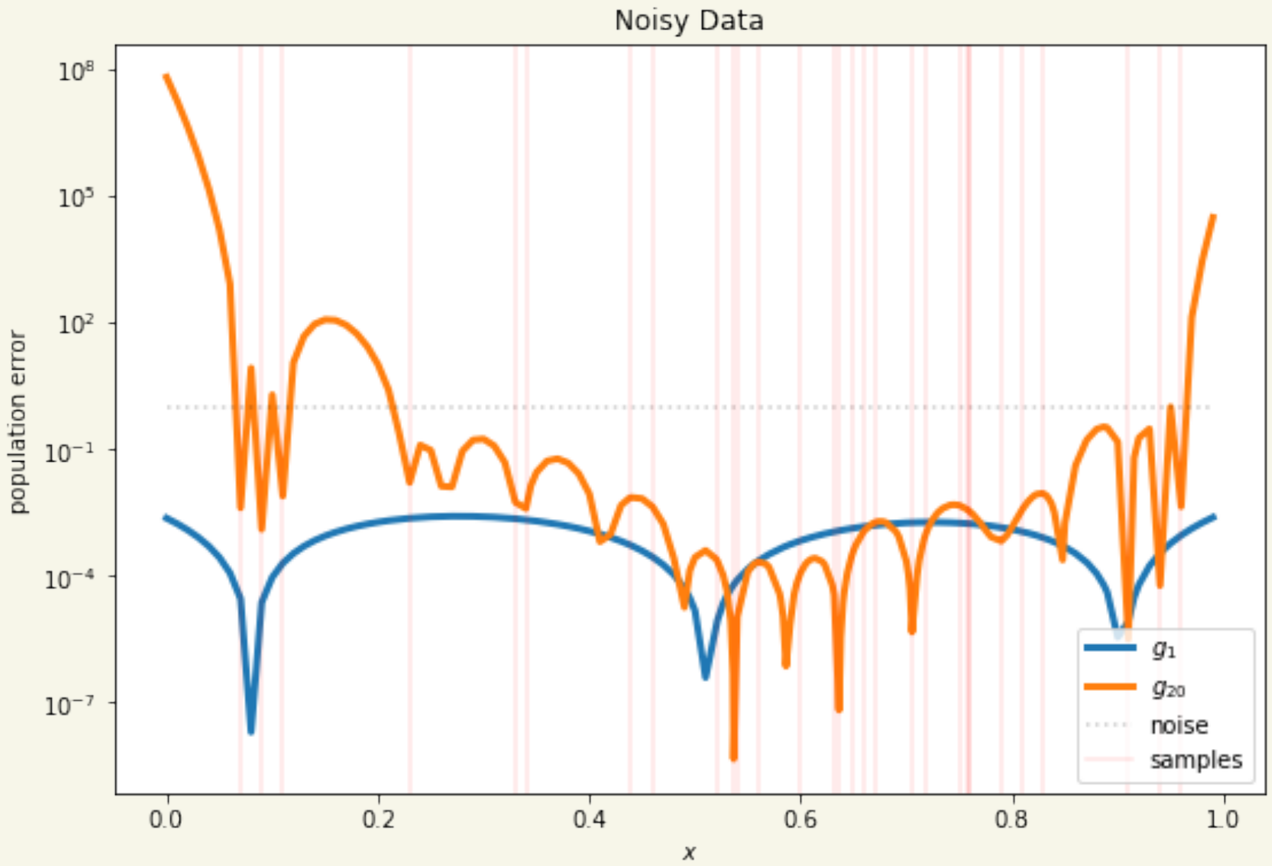
$\mathcal{H}_{20}$ is an bad hypothesis space

How do we systematically decide what is underfit and what is overfit? How do we find a hypothesis set of the "correct" complexity?

The plot provides a hint!

The error from g_{20} is high except at the points on the population which co-incide with those on the sample.

We'll formalize this soon. But, to make the case for overfitting stronger, lets fit on multiple samples..

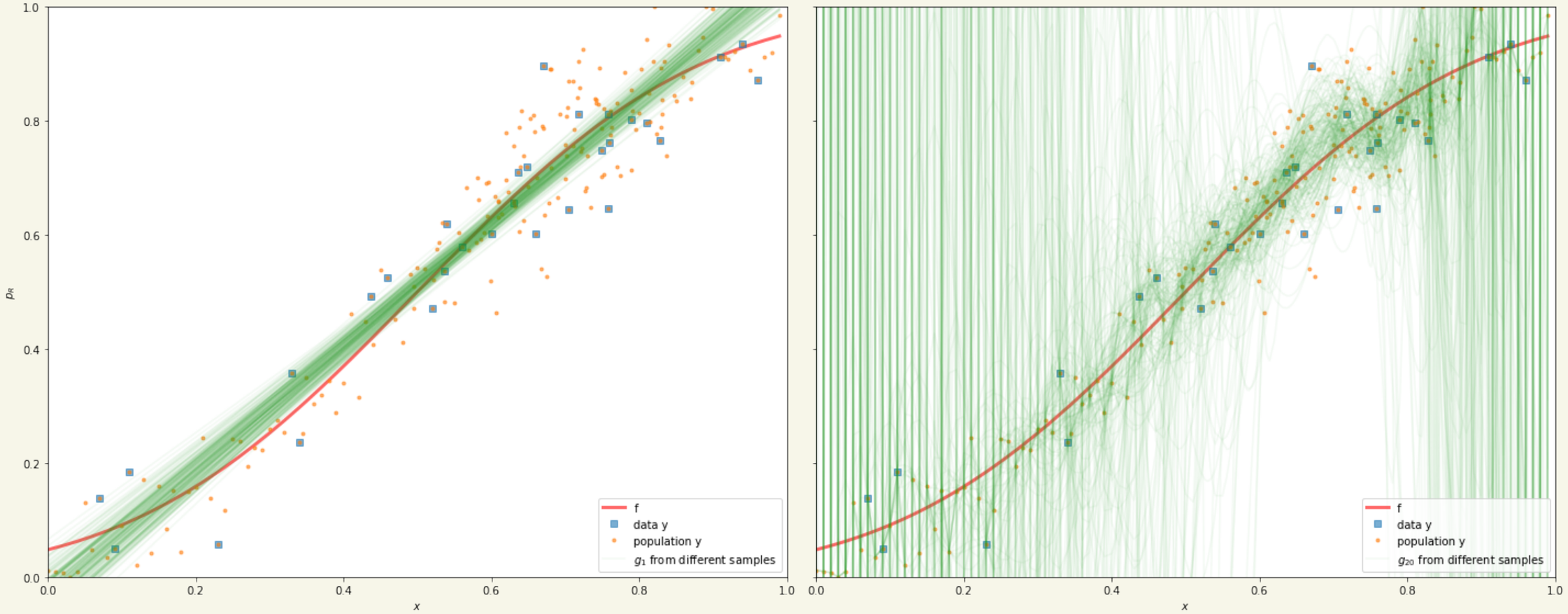


Fitting on multiple samples

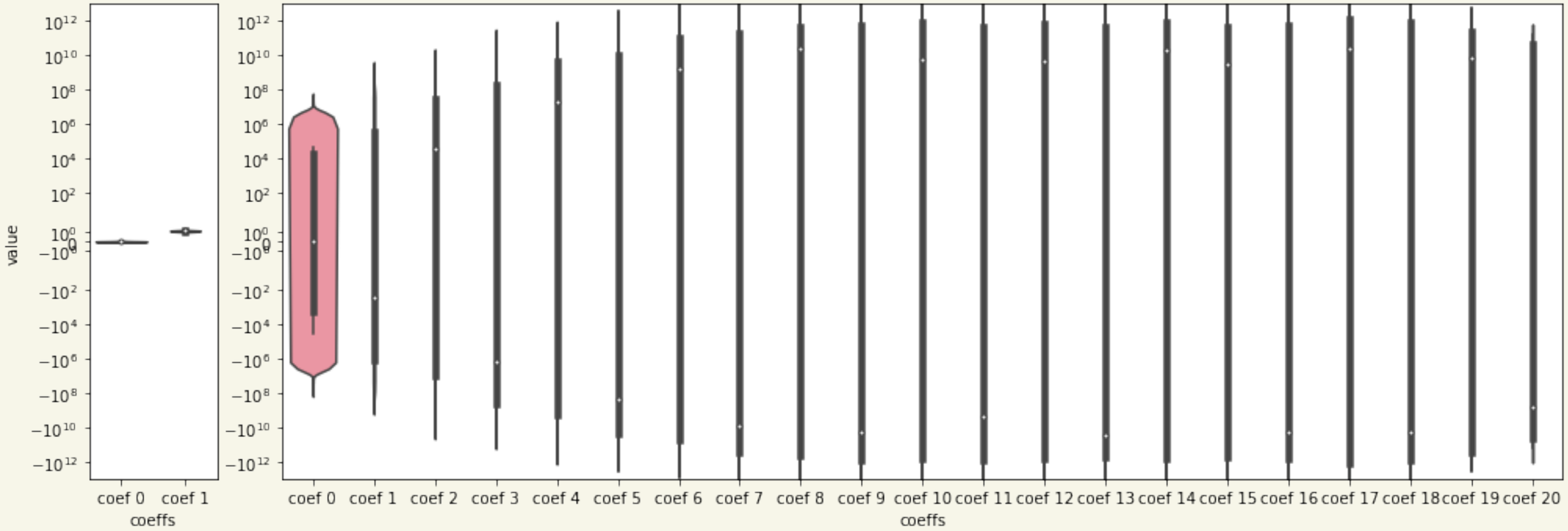
- look at fits on different "training sets $\mathcal{D}$"
- in other words, different samples
- in real life we are not so lucky, usually we get only one sample
- but lets pretend, shall we? After all we are in simulation mode.

We'll make 200 samples of size 30 from our population, and fit a straight line and a 20th order polynomial to each of these, and plot them all in one plot:

UNDERFITTING (Bias) vs OVERFITTING (Variance)



High **variance** in polynomial coefficients



Across samples, coefficients for g_2 are small and finite, while those for g_{20} vary wildly! This accounts for the hair in the previous plot.