6.7900: Machine Learning
Lecture 17

Lecture start: Tues/Thurs 2:35pm
Who's speaking today? Prof. Tamara Broderick
Course website: gradml.mit.edu
Questions? Ask here or on piazza.com/mit/fall2024/67900/
Materials: Slides, video, etc linked from gradml.mit.edu after the lecture (but there is no livestream)

Last Times

I. A common motif in spatiotemporal (and similar) data
II. GPs for regression: model and inference

Today

I. Gaussian processes: fitting hyperparameters, observation noise, high-dimensional inputs
II. Missing data

Recap: Gaussian processes (GPs)

Recap: Gaussian processes (GPs)

- Definition: "A *Gaussian process* is a collection of random variables, any finite number of which have a joint Gaussian distribution." [Rasmussen and Williams 2006; a much much older idea!]

Recap: Gaussian processes (GPs)

- Definition: "A *Gaussian process* is a collection of random variables, any finite number of which have a joint Gaussian distribution." [Rasmussen and Williams 2006; a much much older idea!]
  [Fun fact: an infinitely wide (deep) neural net is a GP: Neal 1994, Lee et al 2018]

Recap: Gaussian processes (GPs)

- Definition: "A *Gaussian process* is a collection of random variables, any finite number of which have a joint Gaussian distribution." [Rasmussen and Williams 2006; a much much older idea!]
  [Fun fact: an infinitely wide (deep) neural net is a GP: Neal 1994, Lee et al 2018]
  - E.g. the function $f(x)$ is a collection indexed by input x

Recap: Gaussian processes (GPs)

- Definition: "A *Gaussian process* is a collection of random variables, any finite number of which have a joint Gaussian distribution." [Rasmussen and Williams 2006; a much much older idea!]
  [Fun fact: an infinitely wide (deep) neural net is a GP: Neal 1994, Lee et al 2018]
  - E.g. the function $f(x)$ is a collection indexed by input x
- We write: $f \sim \mathcal{GP}(m, k)$, where

Recap: Gaussian processes (GPs)

- Definition: "A *Gaussian process* is a collection of random variables, any finite number of which have a joint Gaussian distribution." [Rasmussen and Williams 2006; a much much older idea!]
  [Fun fact: an infinitely wide (deep) neural net is a GP: Neal 1994, Lee et al 2018]
  - E.g. the function $f(x)$ is a collection indexed by input x
- We write: $f \sim \mathcal{GP}(m, k)$, where
  - Mean function $m(x) = \mathbb{E}[f(x)]$

Recap: Gaussian processes (GPs)

- Definition: "A *Gaussian process* is a collection of random variables, any finite number of which have a joint Gaussian distribution." [Rasmussen and Williams 2006; a much much older idea!]
  [Fun fact: an infinitely wide (deep) neural net is a GP: Neal 1994, Lee et al 2018]
  - E.g. the function $f(x)$ is a collection indexed by input x
- We write: $f \sim \mathcal{GP}(m, k)$, where
  - Mean function $m(x) = \mathbb{E}[f(x)]$
  - Covariance function (a.k.a. *kernel*)
    $k(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))]$

Recap: Gaussian processes (GPs)

- Definition: "A *Gaussian process* is a collection of random variables, any finite number of which have a joint Gaussian distribution." [Rasmussen and Williams 2006; a much much older idea!]
  [Fun fact: an infinitely wide (deep) neural net is a GP: Neal 1994, Lee et al 2018]
  - E.g. the function $f(x)$ is a collection indexed by input x
- We write: $f \sim \mathcal{GP}(m, k)$, where
  - Mean function $m(x) = \mathbb{E}[f(x)]$
  - Covariance function (a.k.a. *kernel*)

$$k(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))]$$

we'll assume x is a real vector

Recap: Gaussian processes (GPs)

- Definition: "A *Gaussian process* is a collection of random variables, any finite number of which have a joint Gaussian distribution." [Rasmussen and Williams 2006; a much much older idea!]
  [Fun fact: an infinitely wide (deep) neural net is a GP: Neal 1994, Lee et al 2018]
  - E.g. the function $f(x)$ is a collection indexed by input x
- We write: $f \sim \mathcal{GP}(m, k)$, where
  - Mean function $m(x) = \mathbb{E}[f(x)]$
  - Covariance function (a.k.a. *kernel*)

we'll assume x is a real vector

$$k(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))]$$

  - Common choices: $m(x) = 0$ and *squared exponential* kernel

Recap: Gaussian processes (GPs)

- Definition: "A *Gaussian process* is a collection of random variables, any finite number of which have a joint Gaussian distribution." [Rasmussen and Williams 2006; a much much older idea!]
  [Fun fact: an infinitely wide (deep) neural net is a GP: Neal 1994, Lee et al 2018]
  - E.g. the function $f(x)$ is a collection indexed by input x
- We write: $f \sim \mathcal{GP}(m, k)$, where
  - Mean function $m(x) = \mathbb{E}[f(x)]$
  - Covariance function (a.k.a. *kernel*)

we'll assume x is a real vector

$$k(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))]$$

- Common choices: $m(x) = 0$ and *squared exponential* kernel

$$k(x, x') = \sigma^2 \exp(-\frac{1}{2} \sum_{d=1}^{D} \frac{(x_d - x'_d)^2}{\ell_d^2})$$

signal variance lengthscales

Recap: Gaussian processes (GPs)

- Definition: "A *Gaussian process* is a collection of random variables, any finite number of which have a joint Gaussian distribution." [Rasmussen and Williams 2006; a much much older idea!]
  [Fun fact: an infinitely wide (deep) neural net is a GP: Neal 1994, Lee et al 2018]
  - E.g. the function $f(x)$ is a collection indexed by input x
- We write: $f \sim \mathcal{GP}(m, k)$, where
  - Mean function $m(x) = \mathbb{E}[f(x)]$
  - Covariance function (a.k.a. *kernel*)

$$k(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))]$$

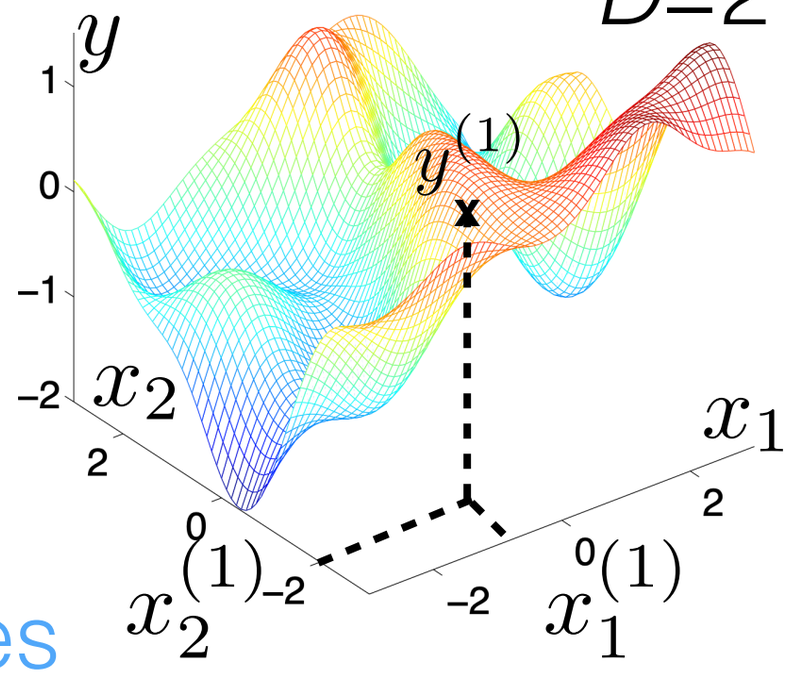
- Common choices: $m(x) = 0$ and *squared exponential* kernel

$$k(x, x') = \sigma^2 \exp(-\frac{1}{2} \sum_{d=1}^{D} \frac{(x_d - x'_d)^2}{\ell_d^2})$$

signal variance lengthscales

we'll assume x is a real vector

D=2



[adapted from Rasmussen & Williams 2006]

Recap: Gaussian processes (GPs)

- Definition: "A *Gaussian process* is a collection of random variables, any finite number of which have a joint Gaussian distribution." [Rasmussen and Williams 2006; a much much older idea!]
  [Fun fact: an infinitely wide (deep) neural net is a GP: Neal 1994, Lee et al 2018]
  - E.g. the function $f(x)$ is a collection indexed by input x
- We write: $f \sim \mathcal{GP}(m, k)$, where
  - Mean function $m(x) = \mathbb{E}[f(x)]$
  - Covariance function (a.k.a. *kernel*)

$$k(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))]$$

- Common choices: $m(x) = 0$ and *squared exponential* kernel

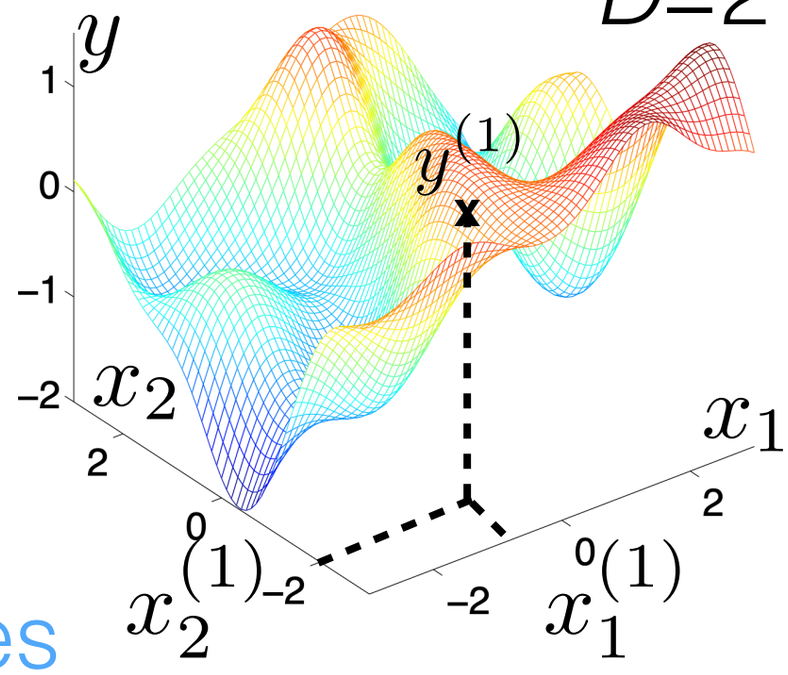
$$k(x, x') = \sigma^2 \exp\left(-\frac{1}{2} \sum_{d=1}^{D} \frac{(x_d - x'_d)^2}{\ell_d^2}\right)$$

signal variance lengthscales

An algorithm: • Fit the hyperparameters with the data

we'll assume x is a real vector

D=2



[adapted from Rasmussen & Williams 2006]

Recap: Gaussian processes (GPs)

- Definition: "A *Gaussian process* is a collection of random variables, any finite number of which have a joint Gaussian distribution." [Rasmussen and Williams 2006; a much much older idea!]
  [Fun fact: an infinitely wide (deep) neural net is a GP: Neal 1994, Lee et al 2018]
  - E.g. the function $f(x)$ is a collection indexed by input x
- We write: $f \sim \mathcal{GP}(m, k)$, where
  - Mean function $m(x) = \mathbb{E}[f(x)]$
  - Covariance function (a.k.a. *kernel*)

$$k(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))]$$

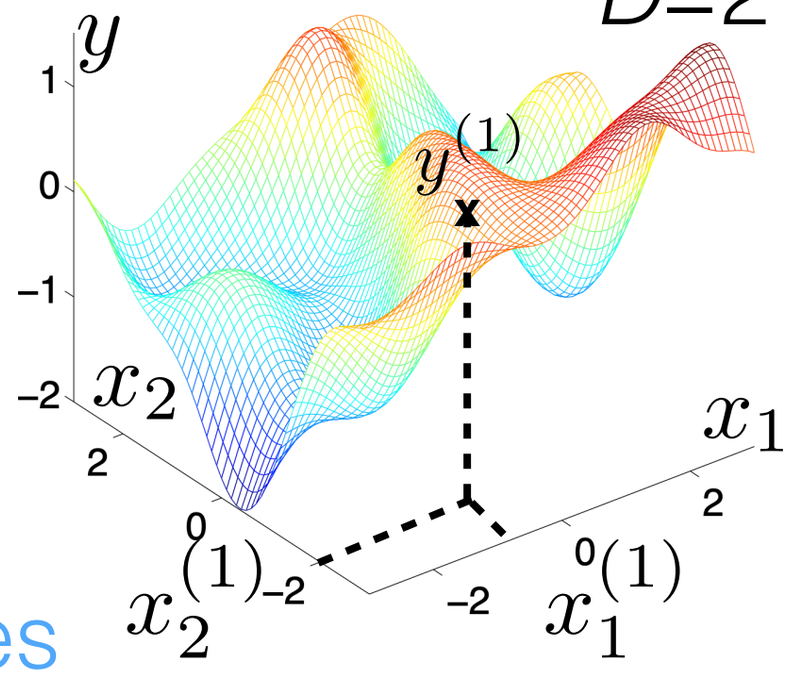
- Common choices: $m(x) = 0$ and *squared exponential* kernel

$$k(x, x') = \sigma^2 \exp\left(-\frac{1}{2}\sum_{d=1}^{D}\frac{(x_d - x'_d)^2}{\ell_d^2}\right)$$

signal variance; lengthscales

we'll assume x is a real vector

D=2



[adapted from Rasmussen & Williams 2006]

An algorithm: • Fit the hyperparameters with the data

- Given those values, now compute and report the posterior mean and uncertainty intervals

Recap: Gaussian processes (GPs)

- Definition: "A *Gaussian process* is a collection of random variables, any finite number of which have a joint Gaussian distribution." [Rasmussen and Williams 2006; a much much older idea!]
  [Fun fact: an infinitely wide (deep) neural net is a GP: Neal 1994, Lee et al 2018]
  - E.g. the function $f(x)$ is a collection indexed by input x
- We write: $f \sim \mathcal{GP}(m, k)$, where
  - Mean function $m(x) = \mathbb{E}[f(x)]$
  - Covariance function (a.k.a. *kernel*)

$$k(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))]$$

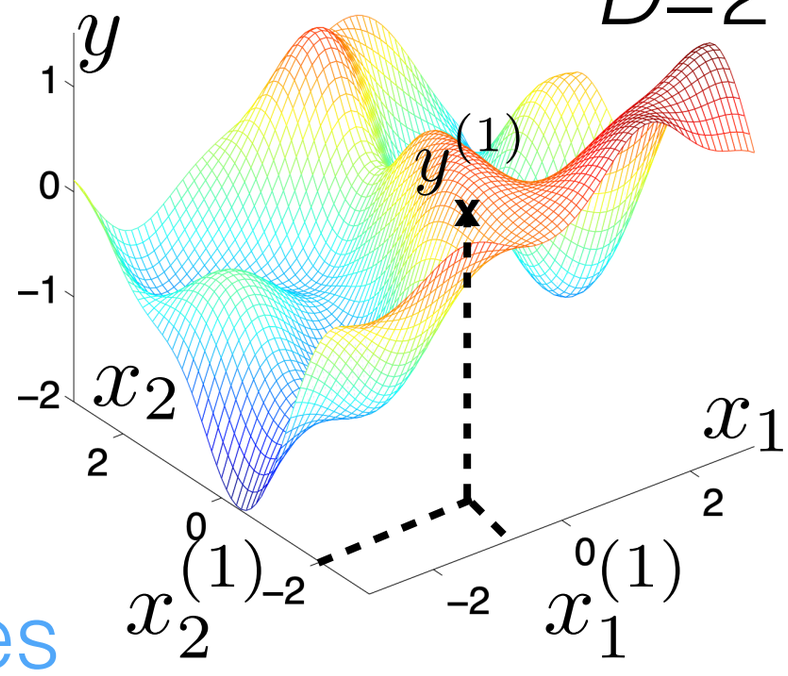
- Common choices: $m(x) = 0$ and *squared exponential* kernel

$$k(x, x') = \sigma^2 \exp\left(-\frac{1}{2}\sum_{d=1}^{D}\frac{(x_d - x'_d)^2}{\ell_d^2}\right)$$

signal variance; lengthscales

we'll assume x is a real vector

D=2



[adapted from Rasmussen & Williams 2006]

An algorithm: • Fit the hyperparameters with the data

- Given those values, now compute and report the posterior mean and uncertainty intervals

[demo1,2,3]

Recap: Gaussian processes (GPs)

- Definition: "A *Gaussian process* is a collection of random variables, any finite number of which have a joint Gaussian distribution." [Rasmussen and Williams 2006; a much much older idea!]
  [Fun fact: an infinitely wide (deep) neural net is a GP: Neal 1994, Lee et al 2018]
  - E.g. the function $f(x)$ is a collection indexed by input x
- We write: $f \sim \mathcal{GP}(m, k)$, where
  - Mean function $m(x) = \mathbb{E}[f(x)]$
  - Covariance function (a.k.a. *kernel*)

$$k(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))]$$

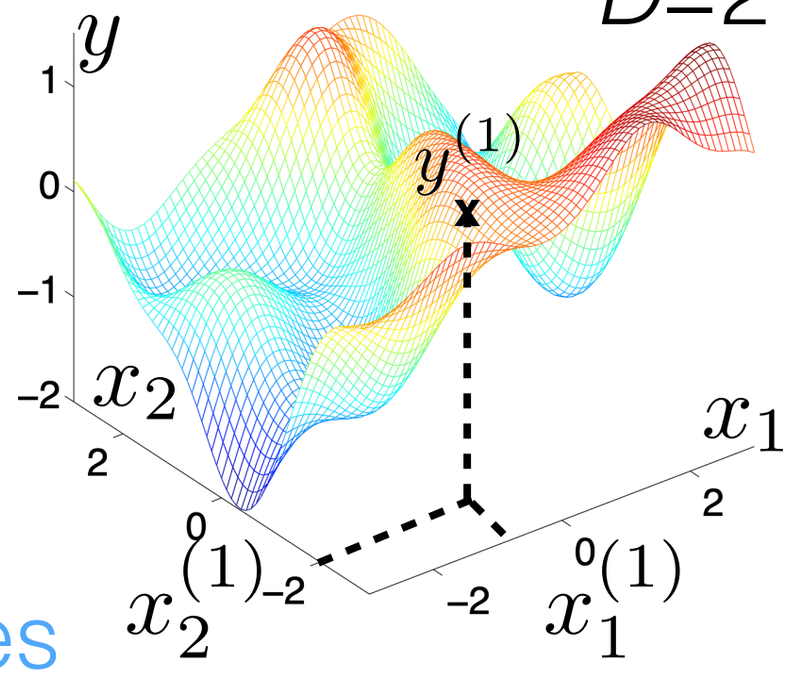
- Common choices: $m(x) = 0$ and *squared exponential* kernel

$$k(x, x') = \sigma^2 \exp\left(-\frac{1}{2}\sum_{d=1}^{D}\frac{(x_d - x'_d)^2}{\ell_d^2}\right)$$

signal variance lengthscales

we'll assume x is a real vector

D=2



[adapted from Rasmussen & Williams 2006]

An algorithm: • Fit the hyperparameters with the data

- Given those values, now compute and report the posterior mean and uncertainty intervals
- Automation / ease of use

[demo1,2,3]

Observation noise

Observation noise

- So far we've been assuming that we observed $f(x)$ directly

Observation noise

- So far we've been assuming that we observed $f(x)$ directly
- But often the actual observation y has additional noise:

Observation noise

- So far we've been assuming that we observed *f*(*x*) directly
- But often the actual observation *y* has additional noise:

$$f \sim \mathcal{GP}(m, k), y^{(n)} \sim f(x^{(n)}) + \epsilon^{(n)}, \epsilon^{(n)} \overset{iid}{\sim} \mathcal{N}(0, \tau^2)$$

Observation noise

- So far we've been assuming that we observed *f*(*x*) directly
- But often the actual observation *y* has additional noise:

$$f \sim \mathcal{GP}(m, k), y^{(n)} \sim f(x^{(n)}) + \epsilon^{(n)}, \epsilon^{(n)} \overset{iid}{\sim} \mathcal{N}(0, \tau^2)$$

  - We observe $\{(x^{(n)}, y^{(n)})\}_{n=1}^{N}$ and want to learn the latent *f*

Observation noise

- So far we've been assuming that we observed *f*(*x*) directly
- But often the actual observation *y* has additional noise:

$$f \sim \mathcal{GP}(m, k), y^{(n)} \sim f(x^{(n)}) + \epsilon^{(n)}, \epsilon^{(n)} \overset{iid}{\sim} \mathcal{N}(0, \tau^2)$$

  - We observe $\{(x^{(n)}, y^{(n)})\}_{n=1}^{N}$ and want to learn the latent *f*

[demo1]

Observation noise

- So far we've been assuming that we observed *f*(*x*) directly
- But often the actual observation *y* has additional noise:

$$f \sim \mathcal{GP}(m, k), y^{(n)} \sim f(x^{(n)}) + \epsilon^{(n)}, \epsilon^{(n)} \overset{iid}{\sim} \mathcal{N}(0, \tau^2)$$

  - We observe $\{(x^{(n)}, y^{(n)})\}_{n=1}^{N}$ and want to learn the latent *f*
  - The *y*'s are multivariate-Gaussian-distributed [demo1]

Why?

Observation noise

- So far we've been assuming that we observed *f*(*x*) directly
- But often the actual observation *y* has additional noise:

$$f \sim \mathcal{GP}(m, k), y^{(n)} \sim f(x^{(n)}) + \epsilon^{(n)}, \epsilon^{(n)} \overset{iid}{\sim} \mathcal{N}(0, \tau^2)$$

  - We observe $\{(x^{(n)}, y^{(n)})\}_{n=1}^{N}$ and want to learn the latent *f*
  - The *y*'s are multivariate-Gaussian-distributed [demo1]
    - Note: the sum of independent Gaussians is a Gaussian with means summed and covariances summed

Observation noise

- So far we've been assuming that we observed *f*(*x*) directly
- But often the actual observation *y* has additional noise:

$$f \sim \mathcal{GP}(m, k), y^{(n)} \sim f(x^{(n)}) + \epsilon^{(n)}, \epsilon^{(n)} \overset{iid}{\sim} \mathcal{N}(0, \tau^2)$$

  - We observe $\{(x^{(n)}, y^{(n)})\}_{n=1}^{N}$ and want to learn the latent *f*
  - The *y*'s are multivariate-Gaussian-distributed [demo1]
    - Note: the sum of independent Gaussians is a Gaussian with means summed and covariances summed
    - So the mean of $y^{(n)}$ is ?

Observation noise

- So far we've been assuming that we observed *f*(*x*) directly
- But often the actual observation *y* has additional noise:

$$f \sim \mathcal{GP}(m, k), y^{(n)} \sim f(x^{(n)}) + \epsilon^{(n)}, \epsilon^{(n)} \overset{iid}{\sim} \mathcal{N}(0, \tau^2)$$

  - We observe $\{(x^{(n)}, y^{(n)})\}_{n=1}^{N}$ and want to learn the latent *f*
  - The *y*'s are multivariate-Gaussian-distributed [demo1]
    - Note: the sum of independent Gaussians is a Gaussian with means summed and covariances summed
    - So the mean of $y^{(n)}$ is $m(x^{(n)})$ and

Observation noise

- So far we've been assuming that we observed *f*(*x*) directly
- But often the actual observation *y* has additional noise:

$$f \sim \mathcal{GP}(m, k), y^{(n)} \sim f(x^{(n)}) + \epsilon^{(n)}, \epsilon^{(n)} \overset{iid}{\sim} \mathcal{N}(0, \tau^2)$$

  - We observe $\{(x^{(n)}, y^{(n)})\}_{n=1}^{N}$ and want to learn the latent *f*
  - The *y*'s are multivariate-Gaussian-distributed [demo1]
    - Note: the sum of independent Gaussians is a Gaussian with means summed and covariances summed
    - So the mean of $y^{(n)}$ is $m(x^{(n)})$ and

$\mathrm{Cov}(y^{(n)}, y^{(n')}) =$?

Observation noise

- So far we've been assuming that we observed *f*(*x*) directly
- But often the actual observation *y* has additional noise:

$$f \sim \mathcal{GP}(m, k), y^{(n)} \sim f(x^{(n)}) + \epsilon^{(n)}, \epsilon^{(n)} \overset{iid}{\sim} \mathcal{N}(0, \tau^2)$$

  - We observe $\{(x^{(n)}, y^{(n)})\}_{n=1}^{N}$ and want to learn the latent *f*
  - The *y*'s are multivariate-Gaussian-distributed [demo1]
    - Note: the sum of independent Gaussians is a Gaussian with means summed and covariances summed
      - So the mean of $y^{(n)}$ is $m(x^{(n)})$ and

$$\mathrm{Cov}(y^{(n)}, y^{(n')}) = k(x^{(n)}, x^{(n')}) + \tau^2 \mathbf{1}\{n = n'\}$$

Observation noise

- So far we've been assuming that we observed *f*(*x*) directly
- But often the actual observation *y* has additional noise:

$$f \sim \mathcal{GP}(m, k), y^{(n)} \sim f(x^{(n)}) + \epsilon^{(n)}, \epsilon^{(n)} \overset{iid}{\sim} \mathcal{N}(0, \tau^2)$$

  - We observe $\{(x^{(n)}, y^{(n)})\}_{n=1}^{N}$ and want to learn the latent *f*
  - The *y*'s are multivariate-Gaussian-distributed [demo1]
    - Note: the sum of independent Gaussians is a Gaussian with means summed and covariances summed
      - So the mean of $y^{(n)}$ is $m(x^{(n)})$ and $\mathrm{Cov}(y^{(n)}, y^{(n')}) = k(x^{(n)}, x^{(n')}) + \tau^2 \mathbf{1}\{n = n'\}$

Why compare indices, not *x*'s?

Observation noise

- So far we've been assuming that we observed *f*(*x*) directly
- But often the actual observation *y* has additional noise:

$$f \sim \mathcal{GP}(m, k), y^{(n)} \sim f(x^{(n)}) + \epsilon^{(n)}, \epsilon^{(n)} \overset{iid}{\sim} \mathcal{N}(0, \tau^2)$$

  - We observe $\{(x^{(n)}, y^{(n)})\}_{n=1}^{N}$ and want to learn the latent *f*
  - The *y*'s are multivariate-Gaussian-distributed [demo1]
    - Note: the sum of independent Gaussians is a Gaussian with means summed and covariances summed
      - So the mean of $y^{(n)}$ is $m(x^{(n)})$ and $\mathrm{Cov}(y^{(n)}, y^{(n')}) = k(x^{(n)}, x^{(n')}) + \tau^2 \mathbf{1}\{n = n'\}$ Why compare indices, not *x*'s?
      - Before: $\begin{bmatrix} f(X) \\ f(X') \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(X,X) & K(X,X') \\ K(X',X) & K(X',X') \end{bmatrix}\right)$

Observation noise

- So far we've been assuming that we observed *f*(*x*) directly
- But often the actual observation *y* has additional noise:

$$f \sim \mathcal{GP}(m, k), y^{(n)} \sim f(x^{(n)}) + \epsilon^{(n)}, \epsilon^{(n)} \overset{iid}{\sim} \mathcal{N}(0, \tau^2)$$

  - We observe $\{(x^{(n)}, y^{(n)})\}_{n=1}^{N}$ and want to learn the latent *f*
  - The *y*'s are multivariate-Gaussian-distributed [demo1]
    - Note: the sum of independent Gaussians is a Gaussian with means summed and covariances summed
      - So the mean of $y^{(n)}$ is $m(x^{(n)})$ and $\mathrm{Cov}(y^{(n)}, y^{(n')}) = k(x^{(n)}, x^{(n')}) + \tau^2 \mathbf{1}\{n = n'\}$ Why compare indices, not *x*'s?
      - Before: $\begin{bmatrix} f(X) \\ f(X') \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(X,X) & K(X,X') \\ K(X',X) & K(X',X') \end{bmatrix}\right)$
      - Now: $\begin{bmatrix} y^{(1:N)} \\ f(X') \end{bmatrix}$

Observation noise

- So far we've been assuming that we observed *f*(*x*) directly
- But often the actual observation *y* has additional noise:

$$f \sim \mathcal{GP}(m, k), y^{(n)} \sim f(x^{(n)}) + \epsilon^{(n)}, \epsilon^{(n)} \overset{iid}{\sim} \mathcal{N}(0, \tau^2)$$

  - We observe $\{(x^{(n)}, y^{(n)})\}_{n=1}^{N}$ and want to learn the latent *f*
  - The *y*'s are multivariate-Gaussian-distributed [demo1]
    - Note: the sum of independent Gaussians is a Gaussian with means summed and covariances summed
      - So the mean of $y^{(n)}$ is $m(x^{(n)})$ and $\mathrm{Cov}(y^{(n)}, y^{(n')}) = k(x^{(n)}, x^{(n')}) + \tau^2 \mathbf{1}\{n = n'\}$ Why compare indices, not *x*'s?
      - Before: $\begin{bmatrix} f(X) \\ f(X') \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(X,X) & K(X,X') \\ K(X',X) & K(X',X') \end{bmatrix}\right)$
      - Now: $\begin{bmatrix} y^{(1:N)} \\ f(X') \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(X,X) + \tau^2 I & K(X,X') \\ K(X',X) & K(X',X') \end{bmatrix}\right)$

Observation noise

- So far we've been assuming that we observed *f*(*x*) directly
- But often the actual observation *y* has additional noise:

$$f \sim \mathcal{GP}(m, k), y^{(n)} \sim f(x^{(n)}) + \epsilon^{(n)}, \epsilon^{(n)} \overset{iid}{\sim} \mathcal{N}(0, \tau^2)$$

  - We observe $\{(x^{(n)}, y^{(n)})\}_{n=1}^{N}$ and want to learn the latent *f*
  - The *y*'s are multivariate-Gaussian-distributed [demo1]
    - Note: the sum of independent Gaussians is a Gaussian with means summed and covariances summed
      - So the mean of $y^{(n)}$ is $m(x^{(n)})$ and $\mathrm{Cov}(y^{(n)}, y^{(n')}) = k(x^{(n)}, x^{(n')}) + \tau^2 \mathbf{1}\{n = n'\}$ Why compare indices, not *x*'s?
      - Before: $\begin{bmatrix} f(X) \\ f(X') \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(X,X) & K(X,X') \\ K(X',X) & K(X',X') \end{bmatrix}\right)$
      - Now: $\begin{bmatrix} y^{(1:N)} \\ f(X') \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(X,X) + \tau^2 I & K(X,X') \\ K(X',X) & K(X',X') \end{bmatrix}\right)$

Observation noise

- So far we've been assuming that we observed *f*(*x*) directly
- But often the actual observation *y* has additional noise:

$$f \sim \mathcal{GP}(m, k), y^{(n)} \sim f(x^{(n)}) + \epsilon^{(n)}, \epsilon^{(n)} \overset{iid}{\sim} \mathcal{N}(0, \tau^2)$$

  - We observe $\{(x^{(n)}, y^{(n)})\}_{n=1}^{N}$ and want to learn the latent *f*
  - The *y*'s are multivariate-Gaussian-distributed [demo1]
    - Note: the sum of independent Gaussians is a Gaussian with means summed and covariances summed
      - So the mean of $y^{(n)}$ is $m(x^{(n)})$ and $\mathrm{Cov}(y^{(n)}, y^{(n')}) = k(x^{(n)}, x^{(n')}) + \tau^2 \mathbf{1}\{n = n'\}$ Why compare indices, not *x*'s?
      - Before: $\begin{bmatrix} f(X) \\ f(X') \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(X,X) & K(X,X') \\ K(X',X) & K(X',X') \end{bmatrix}\right)$
      - Now: $\begin{bmatrix} y^{(1:N)} \\ f(X') \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(X,X) + \tau^2 I & K(X,X') \\ K(X',X) & K(X',X') \end{bmatrix}\right)$

Observation noise

- So far we've been assuming that we observed *f*(*x*) directly
- But often the actual observation *y* has additional noise:

$$f \sim \mathcal{GP}(m, k), y^{(n)} \sim f(x^{(n)}) + \epsilon^{(n)}, \epsilon^{(n)} \overset{iid}{\sim} \mathcal{N}(0, \tau^2)$$

  - We observe $\{(x^{(n)}, y^{(n)})\}_{n=1}^{N}$ and want to learn the latent *f*
  - The *y*'s are multivariate-Gaussian-distributed [demo1]
    - Note: the sum of independent Gaussians is a Gaussian with means summed and covariances summed
      - So the mean of $y^{(n)}$ is $m(x^{(n)})$ and $\mathrm{Cov}(y^{(n)}, y^{(n')}) = k(x^{(n)}, x^{(n')}) + \tau^2 \mathbf{1}\{n = n'\}$ Why compare indices, not *x*'s?
      - Before: $\begin{bmatrix} f(X) \\ f(X') \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(X,X) & K(X,X') \\ K(X',X) & K(X',X') \end{bmatrix}\right)$
      - Now: $\begin{bmatrix} y^{(1:N)} \\ f(X') \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(X,X) + \tau^2 I & K(X,X') \\ K(X',X) & K(X',X') \end{bmatrix}\right)$

What if we put *y* here instead?

Observation noise

- So far we've been assuming that we observed *f*(*x*) directly
- But often the actual observation *y* has additional noise:

$$f \sim \mathcal{GP}(m, k), y^{(n)} \sim f(x^{(n)}) + \epsilon^{(n)}, \epsilon^{(n)} \overset{iid}{\sim} \mathcal{N}(0, \tau^2)$$

  - We observe $\{(x^{(n)}, y^{(n)})\}_{n=1}^{N}$ and want to learn the latent *f*
  - The *y*'s are multivariate-Gaussian-distributed [demo1]
    - Note: the sum of independent Gaussians is a Gaussian with means summed and covariances summed
      - So the mean of $y^{(n)}$ is $m(x^{(n)})$ and $\mathrm{Cov}(y^{(n)}, y^{(n')}) = k(x^{(n)}, x^{(n')}) + \tau^2 \mathbf{1}\{n = n'\}$ Why compare indices, not *x*'s?
      - Before: $\begin{bmatrix} f(X) \\ f(X') \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(X,X) & K(X,X') \\ K(X',X) & K(X',X') \end{bmatrix}\right)$
      - Now: $\begin{bmatrix} y^{(1:N)} \\ f(X') \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(X,X) + \tau^2 I & K(X,X') \\ K(X',X) & K(X',X') \end{bmatrix}\right)$

What if we put *y* here instead? [demo2, demo3]

Observation noise

- So far we've been assuming that we observed *f*(*x*) directly
- But often the actual observation *y* has additional noise:

$$f \sim \mathcal{GP}(m, k), y^{(n)} \sim f(x^{(n)}) + \epsilon^{(n)}, \epsilon^{(n)} \overset{iid}{\sim} \mathcal{N}(0, \tau^2)$$

  - We observe $\{(x^{(n)}, y^{(n)})\}_{n=1}^{N}$ and want to learn the latent *f*
  - The *y*'s are multivariate-Gaussian-distributed [demo1]
    - Note: the sum of independent Gaussians is a Gaussian with means summed and covariances summed
      - So the mean of $y^{(n)}$ is $m(x^{(n)})$ and $\mathrm{Cov}(y^{(n)}, y^{(n')}) = k(x^{(n)}, x^{(n')}) + \tau^2 \mathbf{1}\{n = n'\}$ Why compare indices, not *x*'s?
      - Before: $\begin{bmatrix} f(X) \\ f(X') \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(X,X) & K(X,X') \\ K(X',X) & K(X',X') \end{bmatrix}\right)$
      - Now: $\begin{bmatrix} y^{(1:N)} \\ f(X') \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(X,X) + \tau^2 I & K(X,X') \\ K(X',X) & K(X',X') \end{bmatrix}\right)$

Can you state a non-trivial lower bound on the marginal variance of a test $y^{(m)}$? [demo2, demo3]

Observation noise

Even when observations are "perfect," use a (very small) *nugget* for numerical reasons

- So far we've been assuming that we observed $\pi(x)$ directly
- But often the actual observation y has additional noise:

$$f \sim \mathcal{GP}(m, k), y^{(n)} \sim f(x^{(n)}) + \epsilon^{(n)}, \epsilon^{(n)} \overset{iid}{\sim} \mathcal{N}(0, \tau^2)$$

  - We observe $\{(x^{(n)}, y^{(n)})\}_{n=1}^{N}$ and want to learn the latent f
  - The y's are multivariate-Gaussian-distributed [demo1]
    - Note: the sum of independent Gaussians is a Gaussian with means summed and covariances summed
      - So the mean of $y^{(n)}$ is $m(x^{(n)})$ and $\mathrm{Cov}(y^{(n)}, y^{(n')}) = k(x^{(n)}, x^{(n')}) + \tau^2 \mathbf{1}\{n = n'\}$

Why compare indices, not x's?

- Before: $\begin{bmatrix} f(X) \\ f(X') \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(X,X) & K(X,X') \\ K(X',X) & K(X',X') \end{bmatrix}\right)$
- Now: $\begin{bmatrix} y^{(1:N)} \\ f(X') \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(X,X) + \tau^2 I & K(X,X') \\ K(X',X) & K(X',X') \end{bmatrix}\right)$

Can you state a non-trivial lower bound on the marginal variance of a test $y^{(m)}$? [demo2, demo3]

Extrapolation

- *Extrapolation*: Prediction beyond the observed data

Extrapolation

- *Extrapolation*: Prediction beyond the observed data
  - Compare to *interpolation*: Prediction within the observed data (e.g. within convex hull)

Extrapolation

- *Extrapolation*: Prediction beyond the observed data
  - Compare to *interpolation*: Prediction within the observed data (e.g. within convex hull) [demo1,2]

Extrapolation

- *Extrapolation*: Prediction beyond the observed data
  - Compare to *interpolation*: Prediction within the observed data (e.g. within convex hull) [demo1,2]
- When using GPs with a squared exponential kernel:
  - Data points that are more than a handful of length scales from other data points will revert to prior behavior

Extrapolation

- *Extrapolation*: Prediction beyond the observed data
  - Compare to *interpolation*: Prediction within the observed data (e.g. within convex hull) [demo1,2]
- When using GPs with a squared exponential kernel:
  - Data points that are more than a handful of length scales from other data points will revert to prior behavior
- Recall our discussion from Lecture 07 on extrapolation

Extrapolation

- *Extrapolation*: Prediction beyond the observed data
  - Compare to *interpolation*: Prediction within the observed data (e.g. within convex hull) [demo1,2]
- When using GPs with a squared exponential kernel:
  - Data points that are more than a handful of length scales from other data points will revert to prior behavior
- Recall our discussion from Lecture 07 on extrapolation
  - What behavior can we expect from any machine learning algorithm when extrapolating from a single set of data?

Extrapolation

- *Extrapolation*: Prediction beyond the observed data
  - Compare to *interpolation*: Prediction within the observed data (e.g. within convex hull) [demo1,2]
- When using GPs with a squared exponential kernel:
  - Data points that are more than a handful of length scales from other data points will revert to prior behavior
- Recall our discussion from Lecture 07 on extrapolation
  - What behavior can we expect from any machine learning algorithm when extrapolating from a single set of data?
- Other sources of information for prediction

Extrapolation

- *Extrapolation*: Prediction beyond the observed data
  - Compare to *interpolation*: Prediction within the observed data (e.g. within convex hull) [demo1,2]
- When using GPs with a squared exponential kernel:
  - Data points that are more than a handful of length scales from other data points will revert to prior behavior
- Recall our discussion from Lecture 07 on extrapolation
  - What behavior can we expect from any machine learning algorithm when extrapolating from a single set of data?
- Other sources of information for prediction, e.g. mechanistic

Extrapolation

- *Extrapolation*: Prediction beyond the observed data
  - Compare to *interpolation*: Prediction within the observed data (e.g. within convex hull) [demo1,2]
- When using GPs with a squared exponential kernel:
  - Data points that are more than a handful of length scales from other data points will revert to prior behavior
- Recall our discussion from Lecture 07 on extrapolation
  - What behavior can we expect from any machine learning algorithm when extrapolating from a single set of data?
- Other sources of information for prediction, e.g. mechanistic
  - Having multiple "datasets" is different than having a single dataset

Extrapolation

- *Extrapolation*: Prediction beyond the observed data
  - Compare to *interpolation*: Prediction within the observed data (e.g. within convex hull) [demo1,2]
- When using GPs with a squared exponential kernel:
  - Data points that are more than a handful of length scales from other data points will revert to prior behavior
- Recall our discussion from Lecture 07 on extrapolation
  - What behavior can we expect from any machine learning algorithm when extrapolating from a single set of data?
- Other sources of information for prediction, e.g. mechanistic
  - Having multiple "datasets" is different than having a single dataset
    - Contrast: population prediction vs. vital sign records of many patients, purchase behavior for many books, life cycles of many cells

Extrapolation

- *Extrapolation*: Prediction beyond the observed data
  - Compare to *interpolation*: Prediction within the observed data (e.g. within convex hull) [demo1,2]
- When using GPs with a squared exponential kernel:
  - Data points that are more than a handful of length scales from other data points will revert to prior behavior
- Recall our discussion from Lecture 07 on extrapolation
  - What behavior can we expect from any machine learning algorithm when extrapolating from a single set of data?
- Other sources of information for prediction, e.g. mechanistic
  - Having multiple "datasets" is different than having a single dataset (when is it still extrapolation?)
    - Contrast: population prediction vs. vital sign records of many patients, purchase behavior for many books, life cycles of many cells

More than one input

More than one input

- Our illustrations have almost all been for one input so far

More than one input

- Our illustrations have almost all been for one input so far
- But in real life, it's typical to have more than one input

More than one input

- Our illustrations have almost all been for one input so far
- But in real life, it's typical to have more than one input
- What could go wrong?

More than one input

- Our illustrations have almost all been for one input so far
- But in real life, it's typical to have more than one input
- What could go wrong?
  - It's easy to be misled by plots (always in 2 dimensions)

More than one input

- Our illustrations have almost all been for one input so far
- But in real life, it's typical to have more than one input
- What could go wrong?
  - It's easy to be misled by plots (always in 2 dimensions)
  - Regression in high dimensions is a fundamentally hard problem (*without additional assumptions*)

More than one input

- Our illustrations have almost all been for one input so far
- But in real life, it's typical to have more than one input
- What could go wrong?
  - It's easy to be misled by plots (always in 2 dimensions)
  - Regression in high dimensions is a fundamentally hard problem (*without additional assumptions*)
- All points are "far away" in high dimensions. Illustration:

More than one input

- Our illustrations have almost all been for one input so far
- But in real life, it's typical to have more than one input
- What could go wrong?
  - It's easy to be misled by plots (always in 2 dimensions)
  - Regression in high dimensions is a fundamentally hard problem (*without additional assumptions*)
- All points are "far away" in high dimensions. Illustration:
  - Uniformly randomly sample 10,000 points on $[0,1]^D$

[Binois, Wycoff 2022]

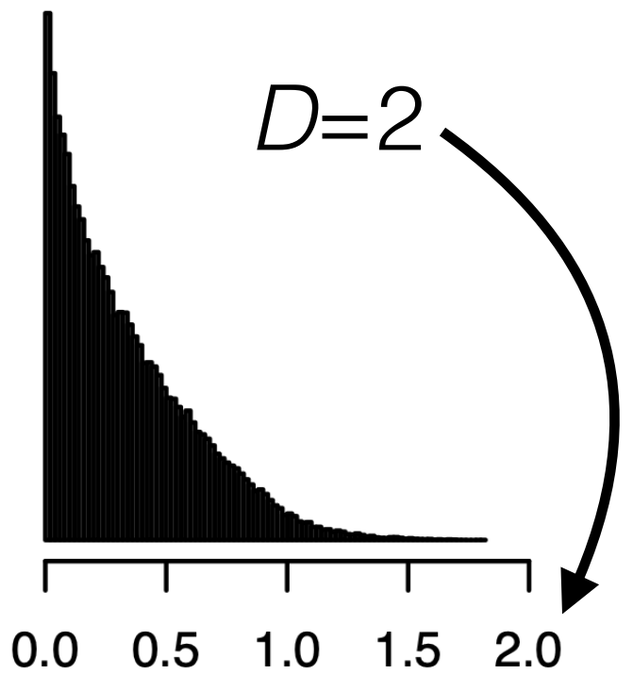
More than one input

- Our illustrations have almost all been for one input so far
- But in real life, it's typical to have more than one input
- What could go wrong?
  - It's easy to be misled by plots (always in 2 dimensions)
  - Regression in high dimensions is a fundamentally hard problem (*without additional assumptions*)
- All points are "far away" in high dimensions. Illustration:
  - Uniformly randomly sample 10,000 points on $[0,1]^D$
  - Make a histogram of squared inter-point distances

[Binois, Wycoff 2022]

More than one input

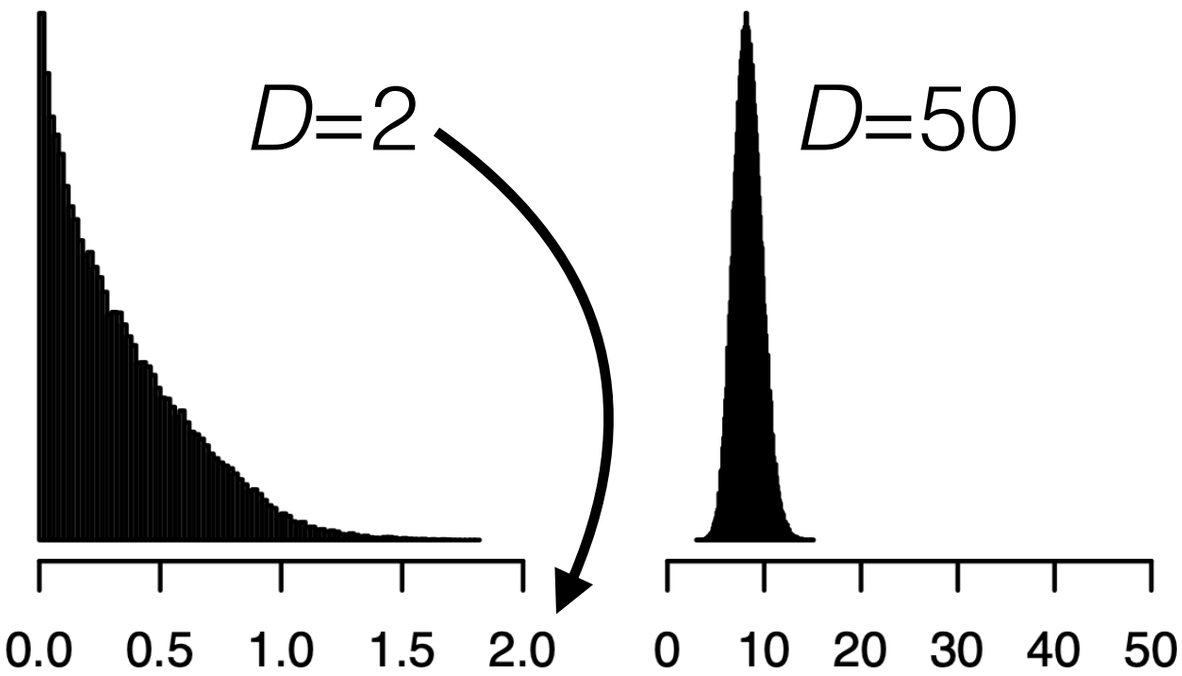
- Our illustrations have almost all been for one input so far
- But in real life, it's typical to have more than one input
- What could go wrong?
  - It's easy to be misled by plots (always in 2 dimensions)
  - Regression in high dimensions is a fundamentally hard problem (*without additional assumptions*)
- All points are "far away" in high dimensions. Illustration:
  - Uniformly randomly sample 10,000 points on $[0,1]^D$
  - Make a histogram of squared inter-point distances



[Binois, Wycoff 2022]

More than one input

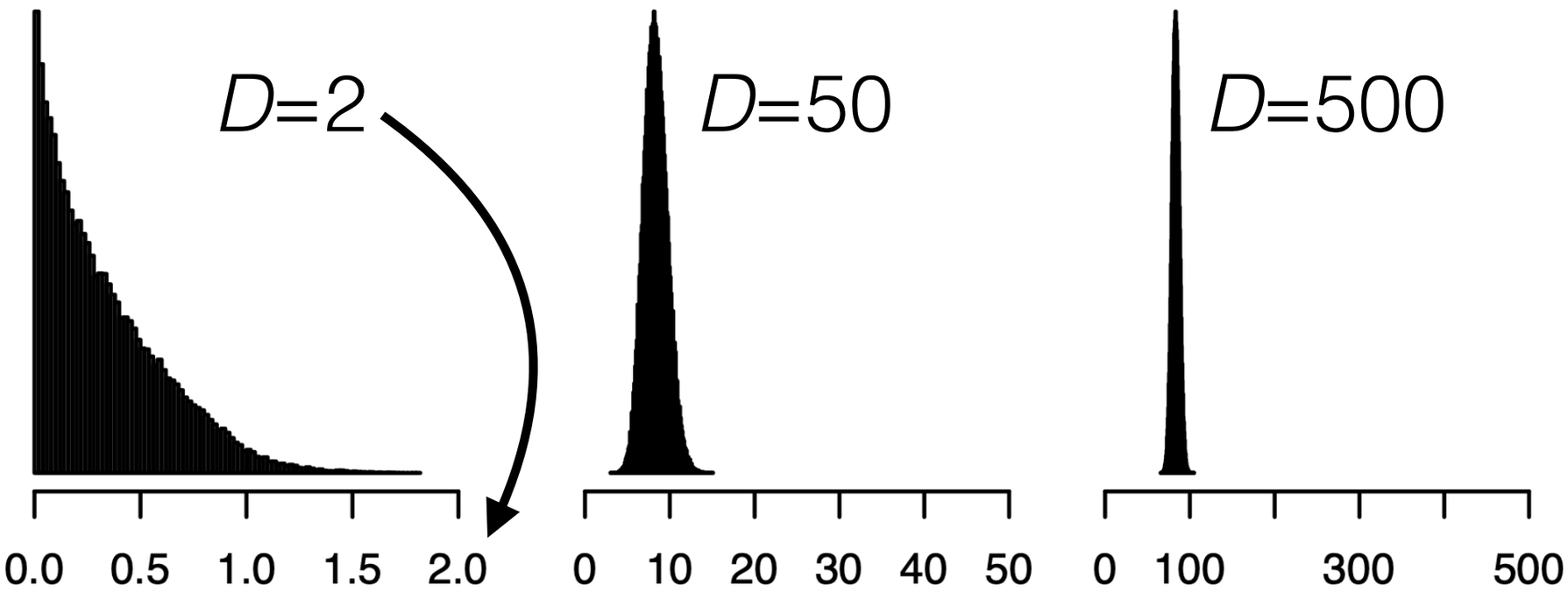
- Our illustrations have almost all been for one input so far
- But in real life, it's typical to have more than one input
- What could go wrong?
  - It's easy to be misled by plots (always in 2 dimensions)
  - Regression in high dimensions is a fundamentally hard problem (*without additional assumptions*)
- All points are "far away" in high dimensions. Illustration:
  - Uniformly randomly sample 10,000 points on $[0,1]^D$
  - Make a histogram of squared inter-point distances



[Binois, Wycoff 2022]

More than one input

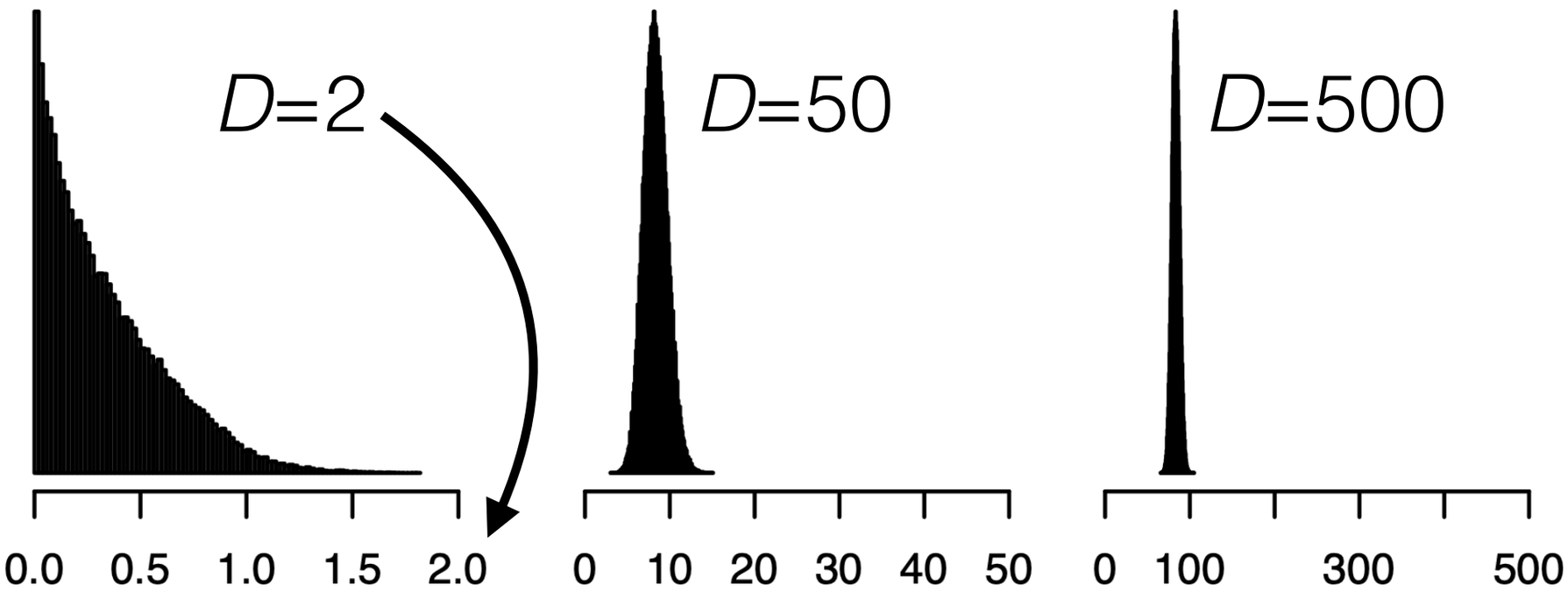
- Our illustrations have almost all been for one input so far
- But in real life, it's typical to have more than one input
- What could go wrong?
  - It's easy to be misled by plots (always in 2 dimensions)
  - Regression in high dimensions is a fundamentally hard problem (*without additional assumptions*)
- All points are "far away" in high dimensions. Illustration:
  - Uniformly randomly sample 10,000 points on $[0,1]^D$
  - Make a histogram of squared inter-point distances



[Binois, Wycoff 2022]

More than one input

- Our illustrations have almost all been for one input so far
- But in real life, it's typical to have more than one input
- What could go wrong?
  - It's easy to be misled by plots (always in 2 dimensions)
  - Regression in high dimensions is a fundamentally hard problem (*without additional assumptions*)
- All points are "far away" in high dimensions. Illustration:
  - Uniformly randomly sample 10,000 points on $[0,1]^D$
  - Make a histogram of squared inter-point distances



[Binois, Wycoff 2022]

- Recall: points "far" from data default to the prior mean and variance

Back to noise in the data

Back to noise in the data

- Some common noise models within ML models:

Back to noise in the data

- Some common noise models within ML models:
  - Standard parametric noise model in labels: e.g. Gaussian noise, Bernoulli draw given probability of a class.

Back to noise in the data

- Some common noise models within ML models:
  - Standard parametric noise model in labels: e.g. Gaussian noise, Bernoulli draw given probability of a class.
  - Adversarial noise in inputs/features or labels

Back to noise in the data

- Some common noise models within ML models:
  - Standard parametric noise model in labels: e.g. Gaussian noise, Bernoulli draw given probability of a class.
  - Adversarial noise in inputs/features or labels
- A very common real-life form of “noise”: missing data

Back to noise in the data

- Some common noise models within ML models:
  - Standard parametric noise model in labels: e.g. Gaussian noise, Bernoulli draw given probability of a class.
  - Adversarial noise in inputs/features or labels
- A very common real-life form of “noise”: missing data
  - “Data that you wish you had, but you do not”

[with thanks to Lucia Petito’s slides]

Back to noise in the data

- Some common noise models within ML models:
  - Standard parametric noise model in labels: e.g. Gaussian noise, Bernoulli draw given probability of a class.
  - Adversarial noise in inputs/features or labels
- A very common real-life form of "noise": missing data
  - "Data that you wish you had, but you do not"
  - Many forms! We'll see examples shortly.

[with thanks to Lucia Petito's slides]

Back to noise in the data

- Some common noise models within ML models:
  - Standard parametric noise model in labels: e.g. Gaussian noise, Bernoulli draw given probability of a class.
  - Adversarial noise in inputs/features or labels
- A very common real-life form of "noise": missing data
  - "Data that you wish you had, but you do not"
  - Many forms! We'll see examples shortly.
- Two concerns:

[with thanks to Lucia Petito's slides]

Back to noise in the data

- Some common noise models within ML models:
  - Standard parametric noise model in labels: e.g. Gaussian noise, Bernoulli draw given probability of a class.
  - Adversarial noise in inputs/features or labels
- A very common real-life form of "noise": missing data
  - "Data that you wish you had, but you do not"
  - Many forms! We'll see examples shortly.

[with thanks to Lucia Petito's slides]

- Two concerns:
  1. Typical supervised learning algorithms require having a full feature matrix *X* and label vector *Y*

Back to noise in the data

- Some common noise models within ML models:
  - Standard parametric noise model in labels: e.g. Gaussian noise, Bernoulli draw given probability of a class.
  - Adversarial noise in inputs/features or labels
- A very common real-life form of "noise": missing data
  - "Data that you wish you had, but you do not"
  - Many forms! We'll see examples shortly.

[with thanks to Lucia Petito's slides]

- Two concerns:
  1. Typical supervised learning algorithms require having a full feature matrix X and label vector Y
     - If some elements of X or Y are missing, we can't run our algorithms without additional action

Back to noise in the data

- Some common noise models within ML models:
  - Standard parametric noise model in labels: e.g. Gaussian noise, Bernoulli draw given probability of a class.
  - Adversarial noise in inputs/features or labels
- A very common real-life form of "noise": missing data
  - "Data that you wish you had, but you do not"
  - Many forms! We'll see examples shortly.

[with thanks to Lucia Petito's slides]

- Two concerns:
  1. Typical supervised learning algorithms require having a full feature matrix X and label vector Y
     - If some elements of X or Y are missing, we can't run our algorithms without additional action
  2. Quality of any procedure that deals with missing data (including ignoring it)

Back to noise in the data

- Some common noise models within ML models:
  - Standard parametric noise model in labels: e.g. Gaussian noise, Bernoulli draw given probability of a class.
  - Adversarial noise in inputs/features or labels
- A very common real-life form of "noise": missing data
  - "Data that you wish you had, but you do not"
  - Many forms! We'll see examples shortly.

[with thanks to Lucia Petito's slides]

- Two concerns:
  1. Typical supervised learning algorithms require having a full feature matrix *X* and label vector *Y*
     - If some elements of *X* or *Y* are missing, we can't run our algorithms without additional action
  2. Quality of any procedure that deals with missing data (including ignoring it)
- What are our options?
- It depends on *how* data are missing

Some examples of missing data

Some examples of missing data

- Example: we'd like to predict life outcomes (e.g. high school graduation) from family demographics [Salganik et al 2020a,b]

Some examples of missing data

- Example: we'd like to predict life outcomes (e.g. high school graduation) from family demographics [Salganik et al 2020a,b]
  - Suppose we obtain family demographics from surveys

Some examples of missing data

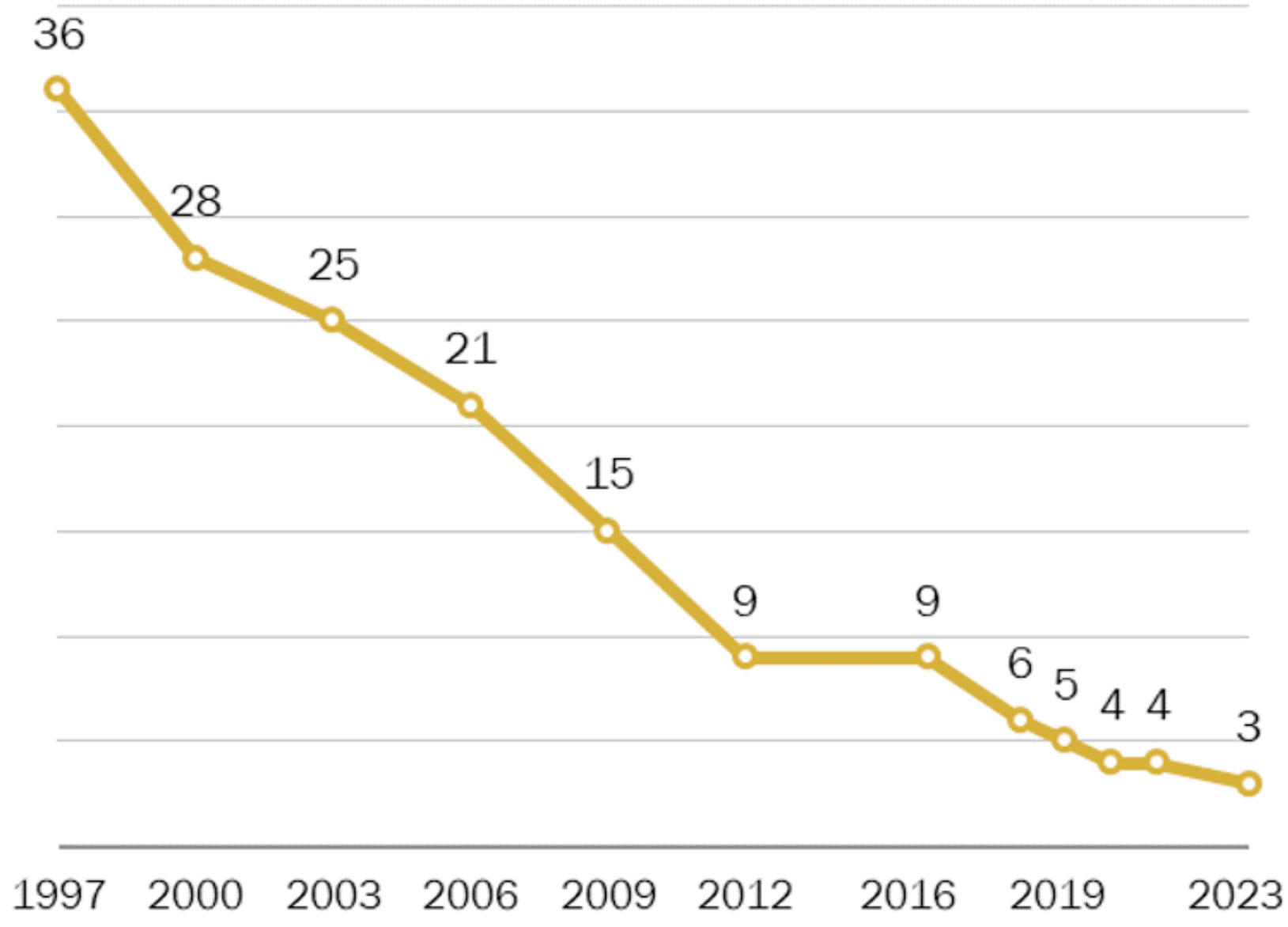
- Example: we'd like to predict life outcomes (e.g. high school graduation) from family demographics [Salganik et al 2020a,b]
  - Suppose we obtain family demographics from surveys
  - Survey non-response [Gelman, Hill 2007, Ch 25.1]

Some examples of missing data

- Example: we'd like to predict life outcomes (e.g. high school graduation) from family demographics [Salganik et al 2020a,b]
  - Suppose we obtain family demographics from surveys
  - Survey non-response [Gelman, Hill 2007, Ch 25.1]
    - Could miss whole person who does not respond

Telephone survey response rates, 1997-2023

% responding, based on AAPOR RR3



Annual averages of Pew Research Center telephone survey response rates

PEW RESEARCH CENTER

Some examples of missing data

- Example: we'd like to predict life outcomes (e.g. high school graduation) from family demographics [Salganik et al 2020a,b]
  - Suppose we obtain family demographics from surveys
  - Survey non-response [Gelman, Hill 2007, Ch 25.1]
    - Could miss whole person who does not respond

Some examples of missing data

- Example: we'd like to predict life outcomes (e.g. high school graduation) from family demographics [Salganik et al 2020a,b]
  - Suppose we obtain family demographics from surveys
  - Survey non-response [Gelman, Hill 2007, Ch 25.1]
    - Could miss whole person who does not respond
    - Could miss a question: E.g. "What is your income?"

Some examples of missing data

- Example: we'd like to predict life outcomes (e.g. high school graduation) from family demographics [Salganik et al 2020a,b]
  - Suppose we obtain family demographics from surveys
  - Survey non-response [Gelman, Hill 2007, Ch 25.1]
    - Could miss whole person who does not respond
    - Could miss a question: E.g. "What is your income?"
- Example: Collecting video of meteors, but some have very short visible paths due to arrival angle and might be difficult to distinguish from stars

Some examples of missing data

- Example: we'd like to predict life outcomes (e.g. high school graduation) from family demographics [Salganik et al 2020a,b]
  - Suppose we obtain family demographics from surveys
  - Survey non-response [Gelman, Hill 2007, Ch 25.1]
    - Could miss whole person who does not respond
    - Could miss a question: E.g. "What is your income?"
- Example: Collecting video of meteors, but some have very short visible paths due to arrival angle and might be difficult to distinguish from stars
- Example: combining datasets/changing features

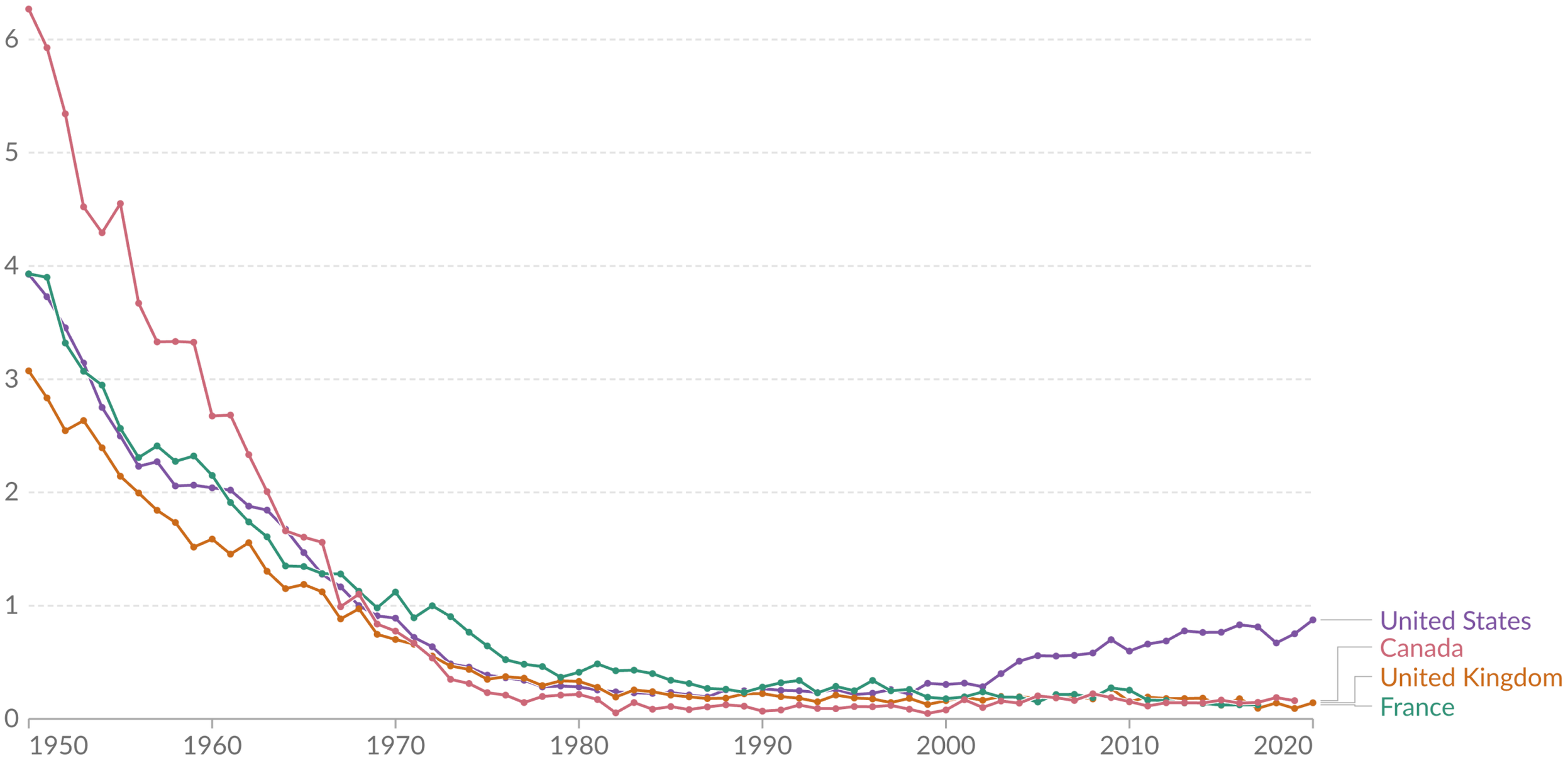
Some examples of missing data

- Example: we'd like to predict life outcomes (e.g. high school graduation) from family demographics [Salganik et al 2020a,b]
  - Suppose we obtain family demographics from surveys
  - Survey non-response [Gelman, Hill 2007, Ch 25.1]
    - Could miss whole person who does not respond
    - Could miss a question: E.g. "What is your income?"
- Example: Collecting video of meteors, but some have very short visible paths due to arrival angle and might be difficult to distinguish from stars
- Example: combining datasets/changing features
  - Maternal mortality [Dattani 2024]

Reported maternal mortality rate

Our World in Data

Reported annual death rate from maternal conditions per 100,000 women and girls, based on official statistics from each country. This includes late maternal deaths that occur up to 1 year after the end of pregnancy. Due to limited reporting, figures are lower than the true number of maternal deaths.



Data source: WHO Mortality Database (2022)

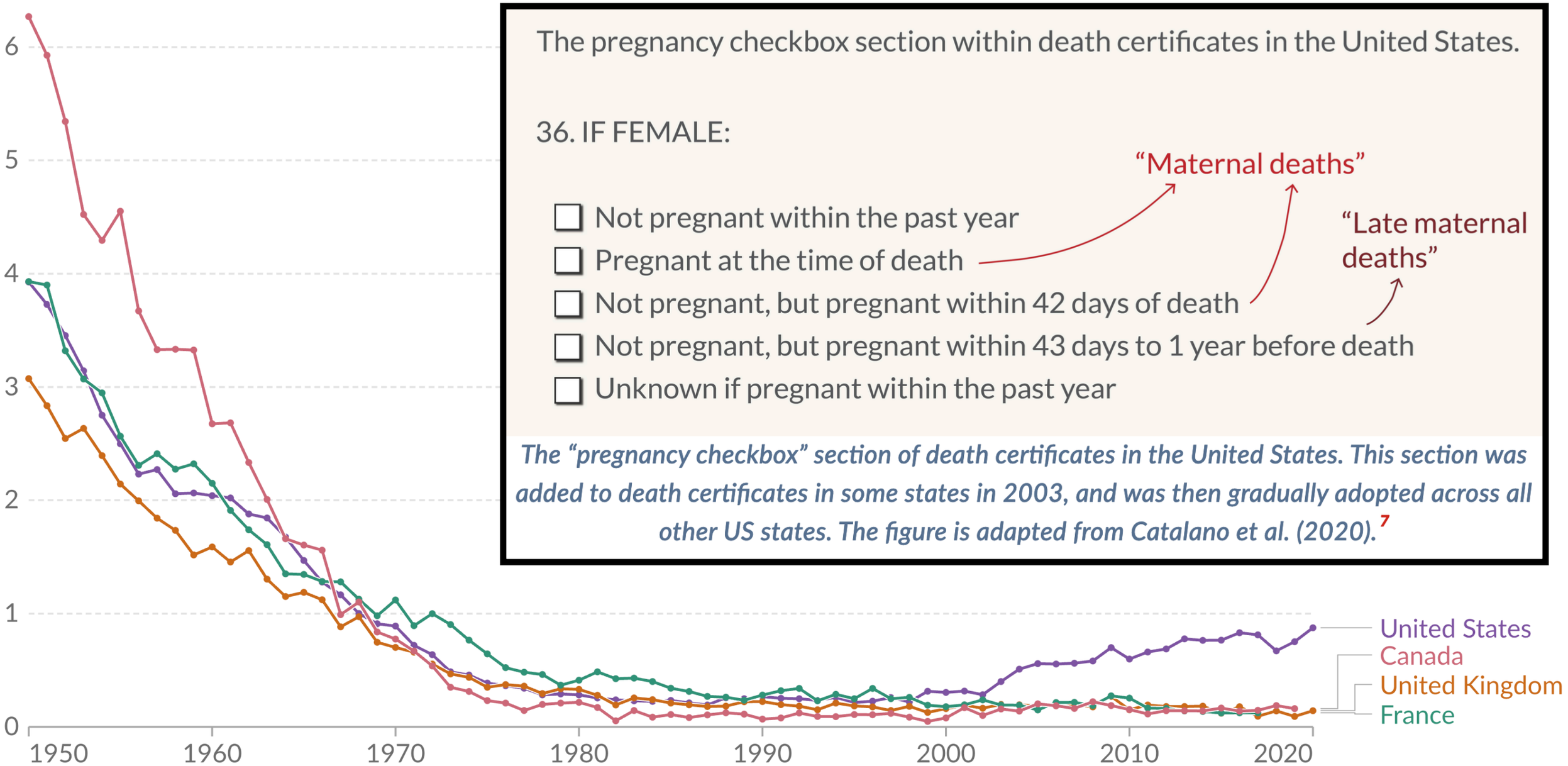
OurWorldInData.org/causes-of-death | CC BY

Note: To allow for comparisons between countries and over time, this metric is age-standardized. All deaths in a country may not have been registered with a cause of death.

Reported maternal mortality rate

Our World in Data

Reported annual death rate from maternal conditions per 100,000 women and girls, based on official statistics from each country. This includes late maternal deaths that occur up to 1 year after the end of pregnancy. Due to limited reporting, figures are lower than the true number of maternal deaths.



The pregnancy checkbox section within death certificates in the United States.

36. IF FEMALE:

"Maternal deaths"

"Late maternal deaths"

- ☐ Not pregnant within the past year
- ☐ Pregnant at the time of death
- ☐ Not pregnant, but pregnant within 42 days of death
- ☐ Not pregnant, but pregnant within 43 days to 1 year before death
- ☐ Unknown if pregnant within the past year

The "pregnancy checkbox" section of death certificates in the United States. This section was added to death certificates in some states in 2003, and was then gradually adopted across all other US states. The figure is adapted from Catalano et al. (2020).[7]

Data source: WHO Mortality Database (2022)

OurWorldInData.org/causes-of-death | CC BY

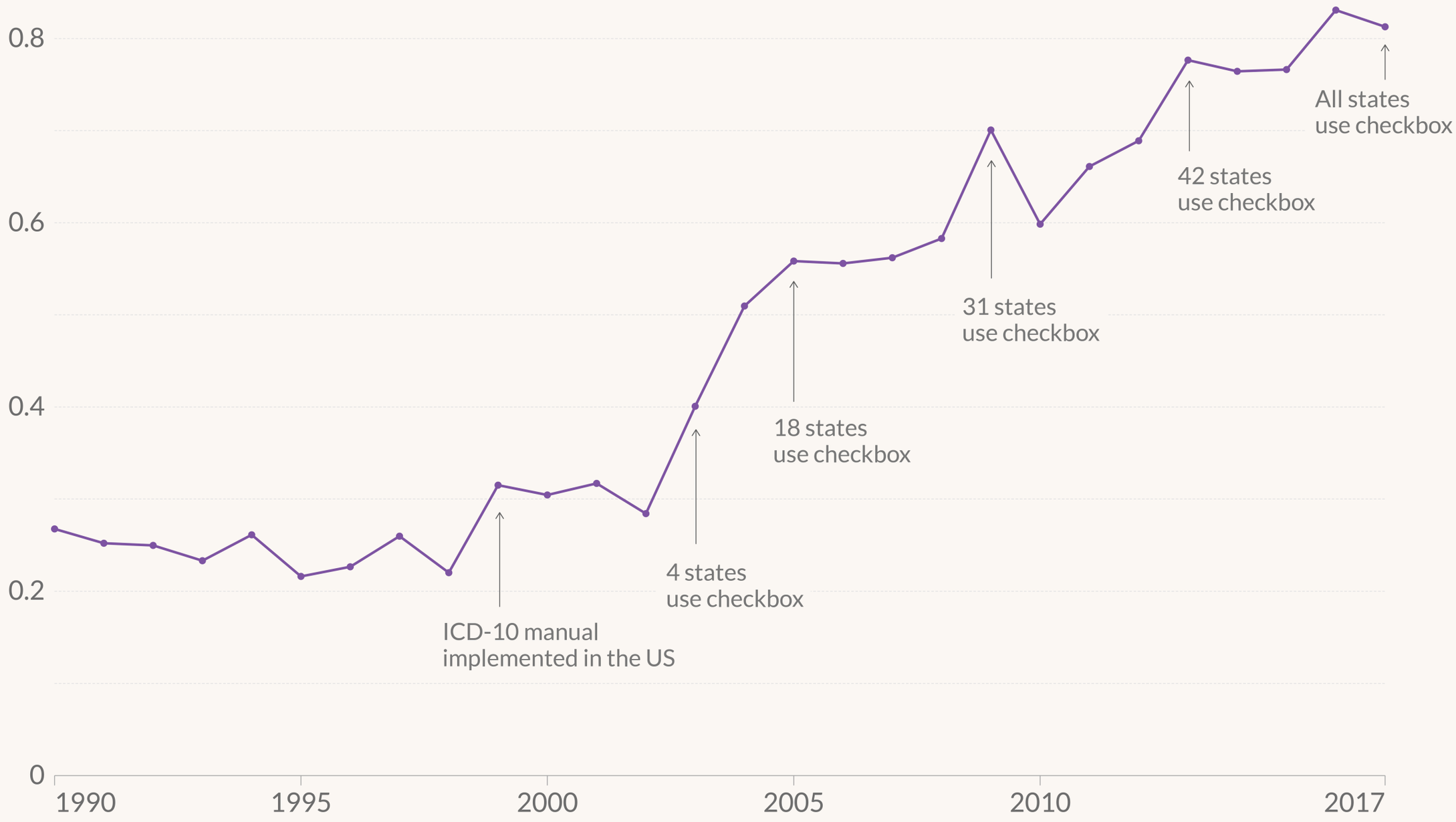
Note: To allow for comparisons between countries and over time, this metric is age-standardized. All deaths in a country may not have been registered with a cause of death.

The US maternal mortality rate rose as more states adopted the “pregnancy checkbox”

Our World in Data

As more states in the US adopted the “pregnancy checkbox” on death certificates — which asked if the deceased had been pregnant or recently pregnant — the reported maternal mortality rate rose.

Maternal mortality rate, per 100,000 females



Source: WHO Mortality Database (2022). Adapted from KS Joseph et al. (2021) Maternal mortality in the United States.
Data includes “late maternal deaths”, which occur up to 1 year after the end of pregnancy.

Some examples of missing data

- Example: we'd like to predict life outcomes (e.g. high school graduation) from family demographics [Salganik et al 2020a,b]
  - Suppose we obtain family demographics from surveys
  - Survey non-response [Gelman, Hill 2007, Ch 25.1]
    - Could miss whole person who does not respond
    - Could miss a question: E.g. "What is your income?"
- Example: Collecting video of meteors, but some have very short visible paths due to arrival angle and might be difficult to distinguish from stars
- Example: combining datasets/changing features
  - Maternal mortality [Dattani 2024]

Some examples of missing data

- Example: we'd like to predict life outcomes (e.g. high school graduation) from family demographics [Salganik et al 2020a,b]
  - Suppose we obtain family demographics from surveys
  - Survey non-response [Gelman, Hill 2007, Ch 25.1]
    - Could miss whole person who does not respond
    - Could miss a question: E.g. "What is your income?"
- Example: Collecting video of meteors, but some have very short visible paths due to arrival angle and might be difficult to distinguish from stars
- Example: combining datasets/changing features
  - Maternal mortality: missing true cause(s) of death [Dattani 2024]

Some examples of missing data

- Example: we'd like to predict life outcomes (e.g. high school graduation) from family demographics [Salganik et al 2020a,b]
  - Suppose we obtain family demographics from surveys
  - Survey non-response [Gelman, Hill 2007, Ch 25.1]
    - Could miss whole person who does not respond
    - Could miss a question: E.g. "What is your income?"
- Example: Collecting video of meteors, but some have very short visible paths due to arrival angle and might be difficult to distinguish from stars
- Example: combining datasets/changing features
  - Maternal mortality: missing true cause(s) of death [Dattani 2024]
  - For a particular way of counting maternal mortality, missing (1) full US time series & (2) across-country data

Some examples of missing data

- Example: we'd like to predict life outcomes (e.g. high school graduation) from family demographics [Salganik et al 2020a,b]
  - Suppose we obtain family demographics from surveys
  - Survey non-response [Gelman, Hill 2007, Ch 25.1]
    - Could miss whole person who does not respond
    - Could miss a question: E.g. "What is your income?"
- Example: Collecting video of meteors, but some have very short visible paths due to arrival angle and might be difficult to distinguish from stars
- Example: combining datasets/changing features
  - Maternal mortality: missing true cause(s) of death [Dattani 2024]
  - For a particular way of counting maternal mortality, missing (1) full US time series & (2) across-country data
  - Depending on what you're trying to do, might think of as feature presence/absence

Some examples of missing data

- Example: we'd like to predict life outcomes (e.g. high school graduation) from family demographics [Salganik et al 2020a,b]
  - Suppose we obtain family demographics from surveys
  - Survey non-response [Gelman, Hill 2007, Ch 25.1]
    - Could miss whole person who does not respond
    - Could miss a question: E.g. "What is your income?"
- Example: Collecting video of meteors, but some have very short visible paths due to arrival angle and might be difficult to distinguish from stars
- Example: combining datasets/changing features
  - Maternal mortality: missing true cause(s) of death [Dattani 2024]
  - For a particular way of counting maternal mortality, missing (1) full US time series & (2) across-country data
  - Depending on what you're trying to do, might think of as feature presence/absence
  - Aside: worth digging into data provenance

Some Types of Missingness

Some Types of Missingness

- Let $M_{nd} = 1$ if feature d in data point n is missing, else 0

Some Types of Missingness

- Let $M_{nd} = 1$ if feature d in data point n is missing, else 0
- Assume Y and X_{obs} observed; X_{mis} missing

Some Types of Missingness

- Let $M_{nd} = 1$ if feature d in data point n is missing, else 0
- Assume Y and X_{obs} observed; X_{mis} missing
- Missing Completely at Random (MCAR):

Some Types of Missingness

- Let M_{nd} = 1 if feature d in data point n is missing, else 0
- Assume Y and X_{obs} observed; X_{mis} missing
- Missing Completely at Random (MCAR):
  - Missingness doesn't depend on latent or observed X, Y

$$p(M|X_{\text{obs}}, X_{\text{mis}}, Y) = p(M)$$

Some Types of Missingness

- Let M_{nd} = 1 if feature d in data point n is missing, else 0
- Assume Y and X_{obs} observed; X_{mis} missing
- Missing Completely at Random (MCAR):
  - Missingness doesn't depend on latent or observed X, Y

$$p(M|X_{\text{obs}}, X_{\text{mis}}, Y) = p(M)$$

  - E.g. some data entries were corrupted

Some Types of Missingness

- Let M_{nd} = 1 if feature d in data point n is missing, else 0
- Assume Y and X_{obs} observed; X_{mis} missing
- Missing Completely at Random (MCAR):
  - Missingness doesn't depend on latent or observed X, Y

$$p(M|X_{\text{obs}}, X_{\text{mis}}, Y) = p(M)$$

  - E.g. some data entries were corrupted
  - Super convenient but mostly unrealistic

Some Types of Missingness

- Let M_{nd} = 1 if feature d in data point n is missing, else 0
- Assume Y and X_{obs} observed; X_{mis} missing
- Missing Completely at Random (MCAR):
  - Missingness doesn't depend on latent or observed X, Y

$$p(M|X_{\text{obs}}, X_{\text{mis}}, Y) = p(M)$$

  - E.g. some data entries were corrupted
  - Super convenient but mostly unrealistic
- Missing at Random (MAR):

Some Types of Missingness

- Let M_{nd} = 1 if feature d in data point n is missing, else 0
- Assume Y and X_{obs} observed; X_{mis} missing
- Missing Completely at Random (MCAR):
  - Missingness doesn't depend on latent or observed X, Y

$$p(M|X_{\text{obs}}, X_{\text{mis}}, Y) = p(M)$$

  - E.g. some data entries were corrupted
  - Super convenient but mostly unrealistic
- Missing at Random (MAR):
  - Missingness depends only on observed (not latent) data

$$p(M|X_{\text{obs}}, X_{\text{mis}}, Y) = p(M|X_{\text{obs}}, Y)$$

Some Types of Missingness

- Let M_{nd} = 1 if feature d in data point n is missing, else 0
- Assume Y and X_{obs} observed; X_{mis} missing
- Missing Completely at Random (MCAR):
  - Missingness doesn't depend on latent or observed X, Y

$$p(M|X_{\text{obs}}, X_{\text{mis}}, Y) = p(M)$$

  - E.g. some data entries were corrupted
  - Super convenient but mostly unrealistic
- Missing at Random (MAR):
  - Missingness depends only on observed (not latent) data

$$p(M|X_{\text{obs}}, X_{\text{mis}}, Y) = p(M|X_{\text{obs}}, Y)$$

  - E.g. probability of income response depends entirely on age, which is fully observed (and not on latent income)

Some Types of Missingness

- Let M_{nd} = 1 if feature d in data point n is missing, else 0
- Assume Y and X_{obs} observed; X_{mis} missing
- Missing Completely at Random (MCAR):
  - Missingness doesn't depend on latent or observed X, Y

$$p(M|X_{\text{obs}}, X_{\text{mis}}, Y) = p(M)$$

  - E.g. some data entries were corrupted
  - Super convenient but mostly unrealistic
- Missing at Random (MAR):
  - Missingness depends only on observed (not latent) data

$$p(M|X_{\text{obs}}, X_{\text{mis}}, Y) = p(M|X_{\text{obs}}, Y)$$

  - E.g. probability of income response depends entirely on age, which is fully observed (and not on latent income)
  - A bit more realistic but misses important cases (see next slide)

Some Types of Missingness

- Let M_{nd} = 1 if feature d in data point n is missing, else 0
- Assume Y and X_{obs} observed; X_{mis} missing
- Missing Completely at Random (MCAR):
  - Missingness doesn't depend on latent or observed X, Y

$$p(M|X_{\text{obs}}, X_{\text{mis}}, Y) = p(M)$$

  - E.g. some data entries were corrupted
  - Super convenient but mostly unrealistic
- Missing at Random (MAR):
  - Missingness depends only on observed (not latent) data

$$p(M|X_{\text{obs}}, X_{\text{mis}}, Y) = p(M|X_{\text{obs}}, Y)$$

  - E.g. probability of income response depends entirely on age, which is fully observed (and not on latent income)
  - A bit more realistic but misses important cases (see next slide)
  - Still feasible that we might take practical actions

Some Types of Missingness

Some Types of Missingness

- Not Missing at Random (NMAR or MNAR)

Some Types of Missingness

- Not Missing at Random (NMAR or MNAR)
  - Missingness can depend on something besides the observed data

Some Types of Missingness

- Not Missing at Random (NMAR or MNAR)
  - Missingness can depend on something besides the observed data
  - E.g. Probability of missingness is direct function of the unobserved and unknown value

Some Types of Missingness

- Not Missing at Random (NMAR or MNAR)
  - Missingness can depend on something besides the observed data
  - E.g. Probability of missingness is direct function of the unobserved and unknown value
    - E.g. Probability of responding to the income question is a function of (unknown) income

Some Types of Missingness

- Not Missing at Random (NMAR or MNAR)
  - Missingness can depend on something besides the observed data
  - E.g. Probability of missingness is direct function of the unobserved and unknown value
    - E.g. Probability of responding to the income question is a function of (unknown) income
    - E.g. Probability of observing the meteor depends on (unknown) meteor path

Some Types of Missingness

- Not Missing at Random (NMAR or MNAR)
  - Missingness can depend on something besides the observed data
  - E.g. Probability of missingness is direct function of the unobserved and unknown value
    - E.g. Probability of responding to the income question is a function of (unknown) income
    - E.g. Probability of observing the meteor depends on (unknown) meteor path
  - E.g. There are confounders

Some Types of Missingness

- Not Missing at Random (NMAR or MNAR)
  - Missingness can depend on something besides the observed data
  - E.g. Probability of missingness is direct function of the unobserved and unknown value
    - E.g. Probability of responding to the income question is a function of (unknown) income
    - E.g. Probability of observing the meteor depends on (unknown) meteor path
  - E.g. There are confounders
    - E.g. Suppose we don't collect data on education (or there is some nonresponse), but education predicts both (1) income and (2) probability of income response

Some Types of Missingness

- Not Missing at Random (NMAR or MNAR)
  - Missingness can depend on something besides the observed data
  - E.g. Probability of missingness is direct function of the unobserved and unknown value
    - E.g. Probability of responding to the income question is a function of (unknown) income
    - E.g. Probability of observing the meteor depends on (unknown) meteor path
  - E.g. There are confounders
    - E.g. Suppose we don't collect data on education (or there is some nonresponse), but education predicts both (1) income and (2) probability of income response
  - Most realistic in many cases, but need strong modeling assumptions to make practical progress

Some Types of Missingness

- Not Missing at Random (NMAR or MNAR)
  - Missingness can depend on something besides the observed data
  - E.g. Probability of missingness is direct function of the unobserved and unknown value
    - E.g. Probability of responding to the income question is a function of (unknown) income
    - E.g. Probability of observing the meteor depends on (unknown) meteor path
  - E.g. There are confounders
    - E.g. Suppose we don't collect data on education (or there is some nonresponse), but education predicts both (1) income and (2) probability of income response
  - Most realistic in many cases, but need strong modeling assumptions to make practical progress
    - Cf. our discussion on extrapolation

References (1/2)

Binois, M., & Wycoff, N. (2022). A survey on high-dimensional Gaussian process modeling with application to Bayesian optimization. ACM Transactions on Evolutionary Learning and Optimization, 2(2), 1-26.

Dattani, S. (2024). The rise in reported maternal mortality rates in the US is largely due to a change in measurement. https://ourworldindata.org/rise-us-maternal-mortality-rates-measurement

Gelman, A. (2007). Data analysis using regression and multilevel/hierarchical models. Cambridge university press.

Lee, J., Bahri, Y., Novak, R., Schoenholz, S. S., Pennington, J., & Sohl-Dickstein, J. (2018). Deep neural networks as Gaussian processes. *ICLR*.

Neal, Radford M. (1994). Priors for infinite networks (tech. rep. no. crg-tr-94-1). University of Toronto.

Petito, L. (2023). Unseen Worlds: How Missing Data Impact: Statistical Analyses https://www.feinberg.northwestern.edu/sites/bcc/docs/2022-2023-lectures/petito-slides-unseen-worlds.pdf

Pew Research Center (2024). Methodology: The American Trends Panel survey methodology. https://www.pewresearch.org/politics/2024/07/11/election-2024-july-methodology/

Salganik, M. J., et al. (2020). Measuring the predictability of life outcomes with a scientific mass collaboration. Proceedings of the National Academy of Sciences, 117(15), 8398-8403.

Salganik, M., Maffeo, L., & Rudin, C. (2020). Prediction, machine learning, and individual lives: An interview with Matthew Salganik. Harvard Data Science Review.

References (2/2)

Tierney, N. (2004). Gallery of Missing Data Visualisations. https://cran.r-project.org/web/packages/naniar/vignettes/naniar-visualisation.html

Williams, C. K., & Rasmussen, C. E. (2006). *Gaussian Processes for Machine Learning*. MIT Press.