Unit 2 Lecture 2: Bias-variance tradeoff

September 19, 2023

In this R demo, we will explore the bias-variance tradeoff in the context of natural spline fits. We'll need the following R packages:

```r
library(tidyverse)
library(stat471)  # for spline_simulation() and bias_variance_tradeoff()
```

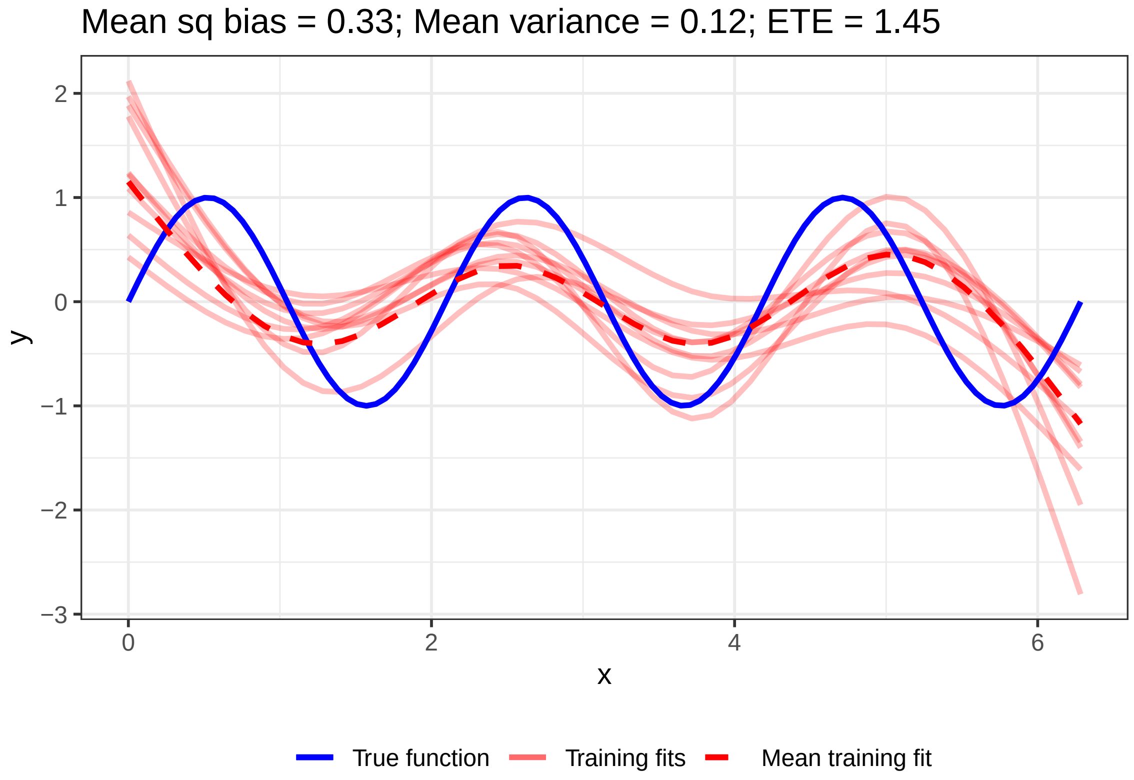
Simulating to compute mean squared bias, mean variance, and ETE

Let us set the following simulation parameters:

```r
f <- function(x) sin(3 * x) # the true trend
sigma <- 1                  # the noise level
n <- 50                     # the training sample size
df <- 6                     # the degrees of freedom for the spline fit
```

The idea for our numerical simulation is that we will repeatedly generate training data based on `f`, `sigma`, and `n`, and then fit a natural cubic spline with `df` degrees of freedom. We can then compute the mean squared bias, mean variance, and ETE. The function `spline_simulation()` from the `stat471` R package will do this for us:

```r
spline_simulation(f, sigma, n, df)
```

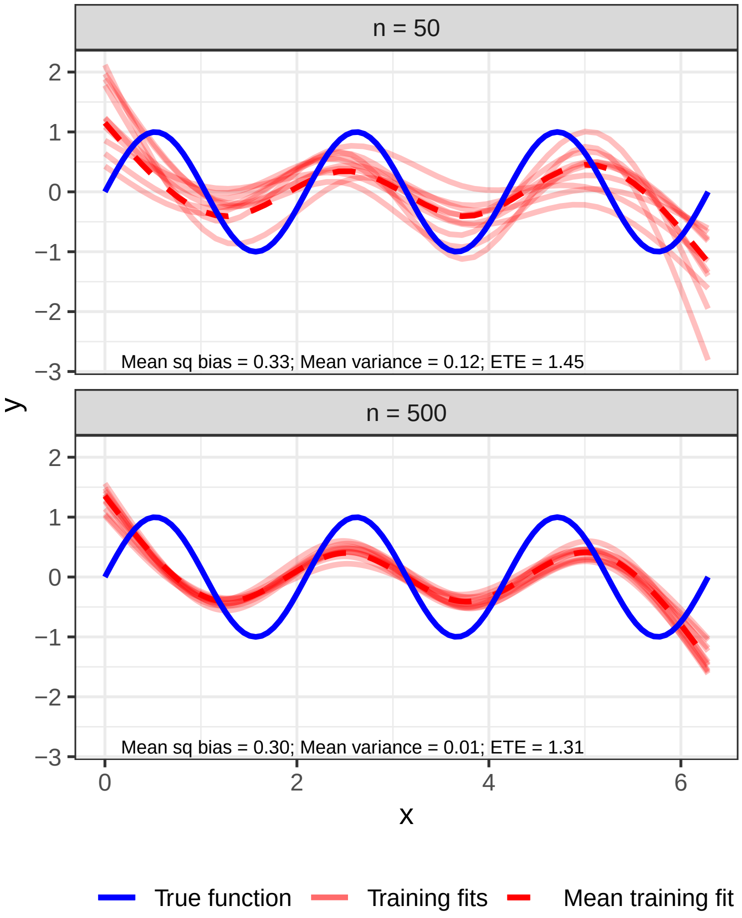


Questions:

- Based on the plot, describe how to visualize bias and variance at a given value of `x`. Approximately what is th bias at `x = 6`?
- Compute the mean variance based on the formula from the lecture slides ($\sigma^2 \cdot p/n$). Does it match what was computed in the simulation?
- Does the relationship among the computed ETE, mean squared bias, and mean variance match that presented in the lecture slides?

Let's see how the plot changes as we vary the sample size:

```r
f <- function(x) sin(3 * x) # the true trend
sigma <- 1                  # the noise level
n <- c(50, 500)             # two values of the training sample size
df <- 6                     # the degrees of freedom for the spline fit
spline_simulation(f, sigma, n, df)
```



Question: Among mean squared bias, mean variance, and ETE: Which quantities changed and why?

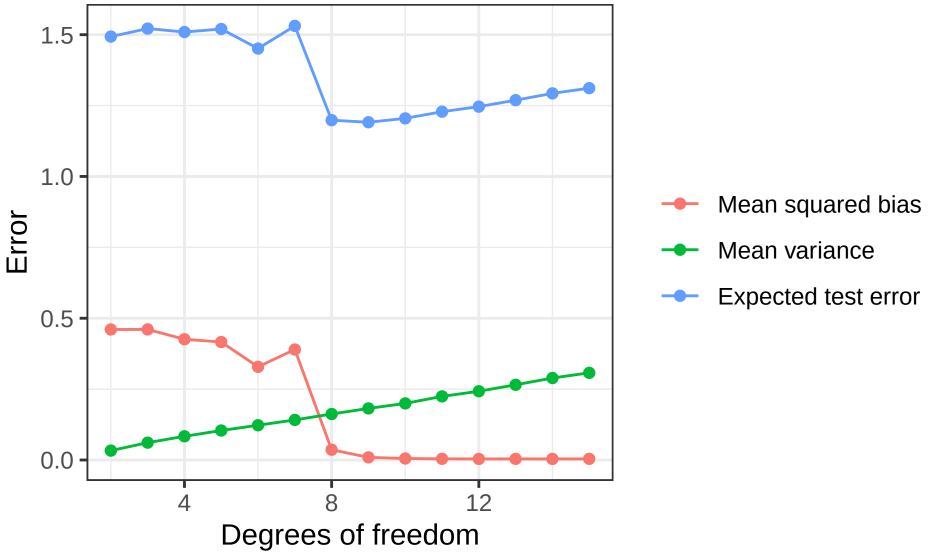
Exercises

- What happens to the picture as we change the complexity of the true trend `f`? Consider `f <- function(x) sin(k*x)` for different `k`. Higher `k` leads to more wiggles, i.e. increased complexity.
- What happens to the picture as we change the noise level `sigma`?
- What happens to the picture as we change the degrees of freedom `df`?

Varying the degrees of freedom

The one of these parameters under our control is `df`. We want to choose this parameter to optimally trade off bias and variance. We can visualize this the bias variance trade-off via the `bias_variance_tradeoff()` function from the `stat471` package:

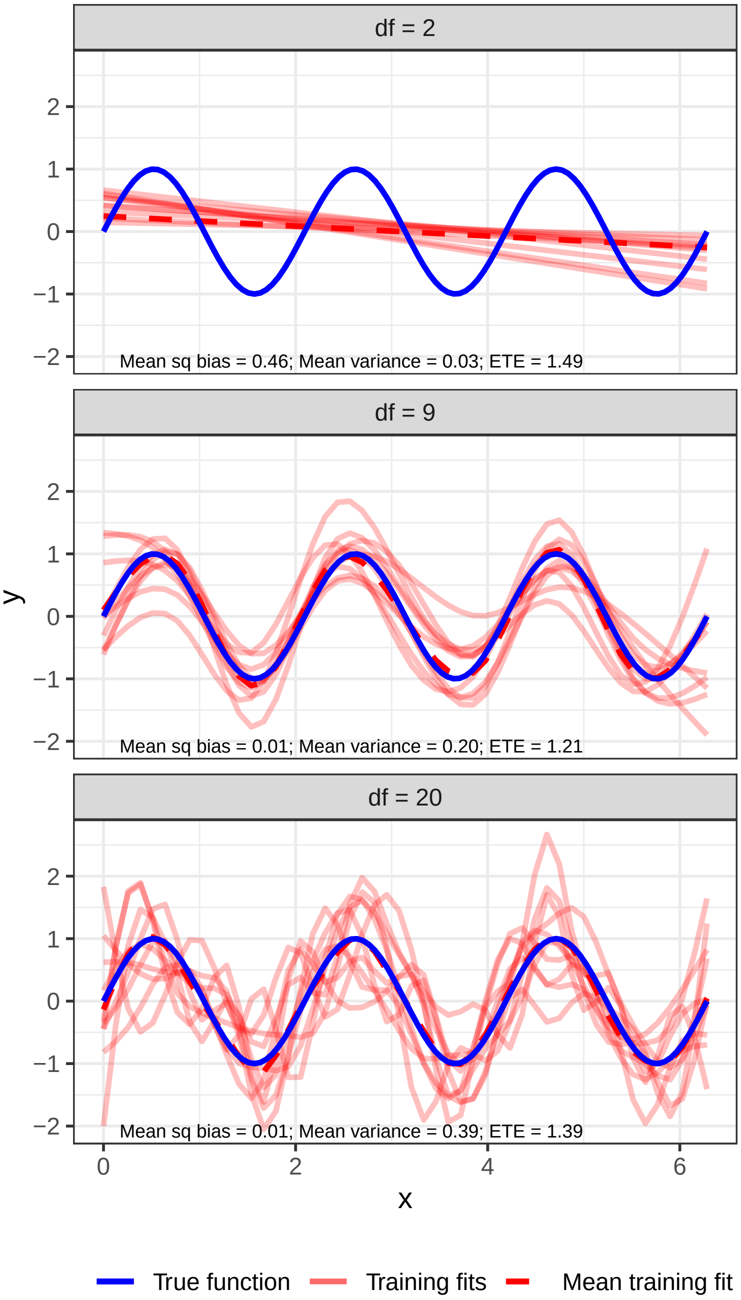
```
f <- function(x) sin(3 * x) # the true trend
sigma <- 1                  # the noise level
n <- 50                     # the training sample size
bias_variance_tradeoff(f, sigma, n)
```



Questions: What trends do we observe in this plot? What appears to be the best degrees of freedom? Why does the variance curve appear linear? What does it mean that the bias is approximately zero from `df` $= 9$ onwards?

Let's take a look at three values for `df`:

```
f <- function(x) sin(3 * x) # the true trend
sigma <- 1                  # the noise level
n <- 50                     # two values of the training sample size
df <- c(2, 9, 20)           # the degrees of freedom for the spline fit
spline_simulation(f, sigma, n, df)
```



What conclusions can we make about this plot, and how it relates to the previous plot above?

Exercises

Varying the noise level

What happens to the bias variance tradeoff plot if we increase the noise standard deviation? What happens if we decrease it? How does this change the best value of `df`? (Large changes will help you see the difference.)

Varying the sample size

What happens to the bias variance tradeoff plot if we increase the sample standard deviation? What happens if we decrease it? How does this change the best value of `df`? (Large changes will help you see the difference.)

Varying the complexity of the underlying function

What happens to the bias variance tradeoff plot if we increase the complexity of the underlying function `f`?? What happens if we decrease it? How does this change the best value of `df`? (Large changes will help you see the difference.) Consider `f <- function(x)(sin(k*x))` for different `k`. Higher `k` leads to more wiggles, i.e. increased complexity.