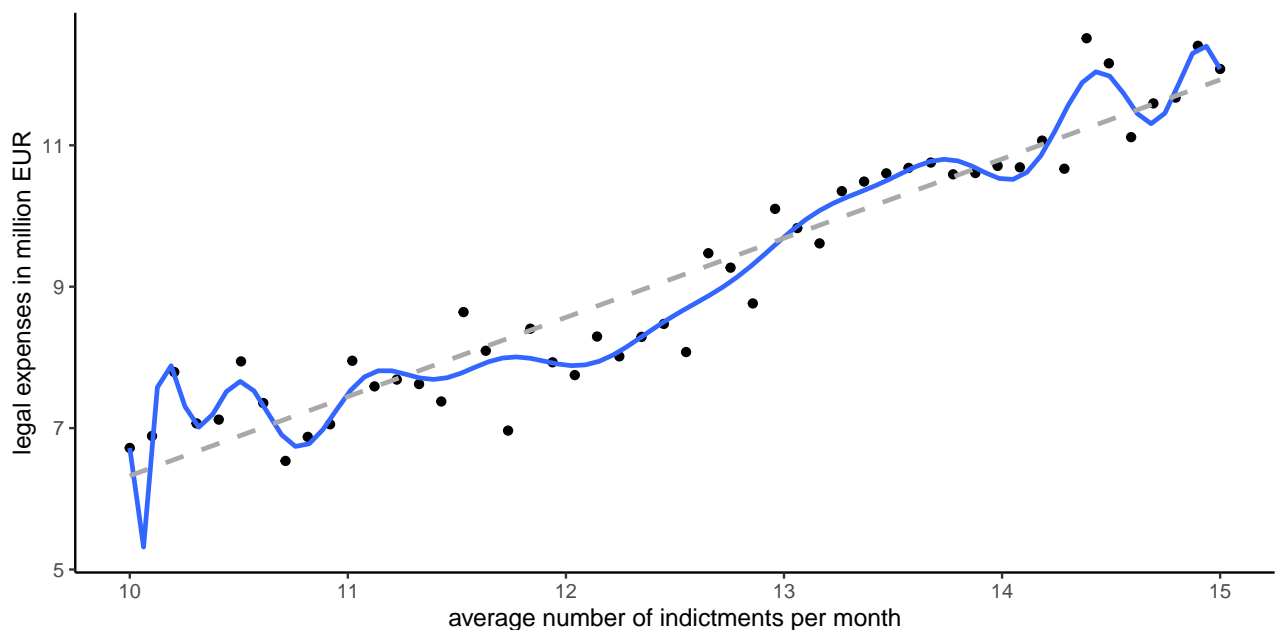


Exercise 1: Evaluating regression learners

Imagine you work for a data science start-up and sell turn-key statistical models. Based on a set of training data, you develop a regression model to predict a customer's legal expenses from the average monthly number of indictments brought against their firm.

- a) Due to the financial sensitivity of the situation, you opt for a very flexible learner that fits the customer's data ($n_{\text{train}} = 50$ observations) well, and end up with a degree-21 polynomial (blue, solid). Your colleague is skeptical and argues for a much simpler linear learner (gray, dashed). Which of the models will have a lower empirical risk if standard L_2 loss is used?



- b) Why might evaluation based on training error not be a good idea here?
- c) Evaluate both learners on the following test data ($n_{\text{test}} = 10$), using
- mean squared error (MSE), and
 - mean absolute error (MAE).

State your performance assessment and explain potential differences.

(Hint: use R if you don't feel like computing a degree-21 polynomial regression by hand.)

```
set.seed(123)
x_train <- seq(10, 15, length.out = 50)
y_train <- 10 + 3 * sin(0.15 * pi * x_train) + rnorm(length(x_train), sd = 0.5)
data_train <- data.frame(x = x_train, y = y_train)

set.seed(321)
x_test <- seq(10, 15, length.out = 10)
y_test <- 10 + 3 * sin(0.15 * pi * x_test) + rnorm(length(x_test), sd = 0.5)
data_test <- data.frame(x = x_test, y = y_test)
```

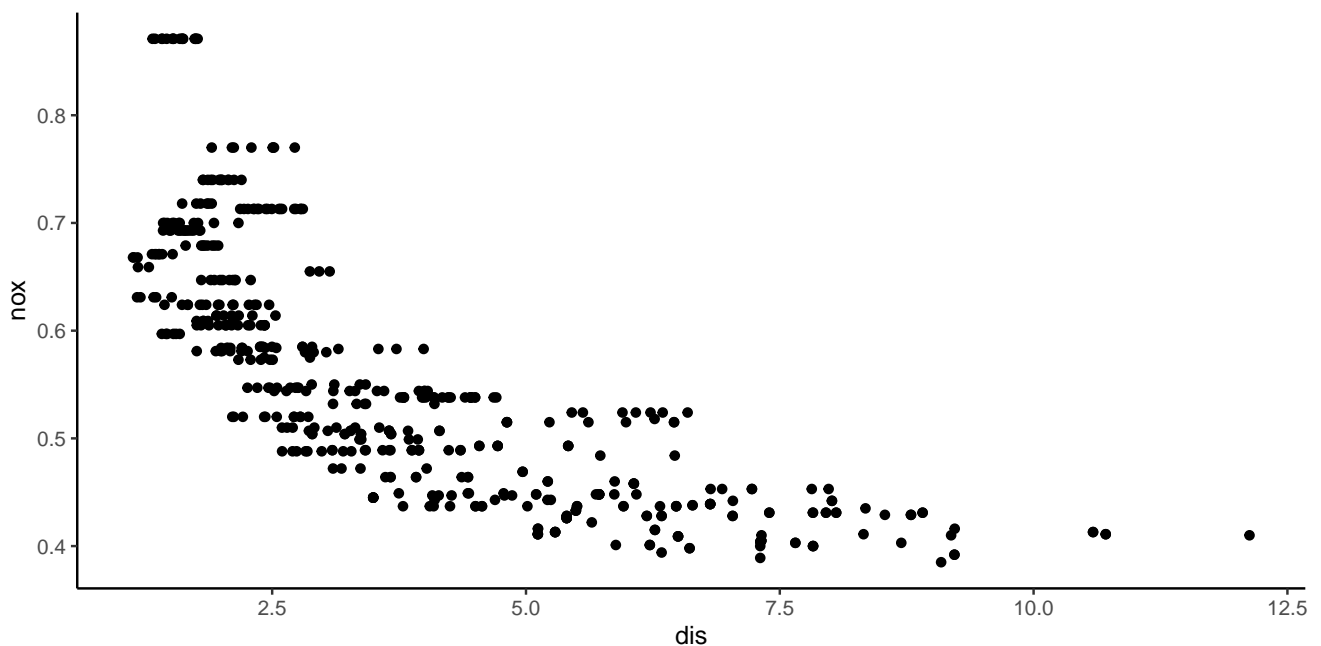
Exercise 2: Importance of train-test split

We consider the `BostonHousing` data for which we would like to predict the nitric oxides concentration (`nox`) from the distance to a number of firms (`dis`).

```
library(mlbench)
data(BostonHousing)
data_pollution <- data.frame(dis = BostonHousing$dis, nox = BostonHousing$nox)
data_pollution <- data_pollution[order(data_pollution$dis), ]
head(data_pollution)

##      dis    nox
## 373 1.1296 0.668
## 375 1.1370 0.668
## 372 1.1691 0.631
## 374 1.1742 0.668
## 407 1.1781 0.659
## 371 1.2024 0.631

ggplot2::ggplot(data_pollution, ggplot2::aes(x = dis, y = nox)) +
  ggplot2::geom_point() +
  ggplot2::theme_classic()
```



- Use the first ten observations as training data to compute a linear model with `mlr3` and evaluate the performance of your learner on the remaining data using MSE.
- What might be disadvantageous about the train-test split in a)?
- Now, sample your training observations from the data set at random. Use a share of 0.1 through 0.9, in 0.1 steps, of observations for training and repeat this procedure ten times. Afterwards, plot the resulting test errors (in terms of MSE) in a suitable manner.
(Hint: `rsmp` is a convenient function for splitting data – you will want to choose the "holdout" strategy. Afterwards, `resample` can be used to repeatedly fit the learner.)
- Interpret the findings from c).