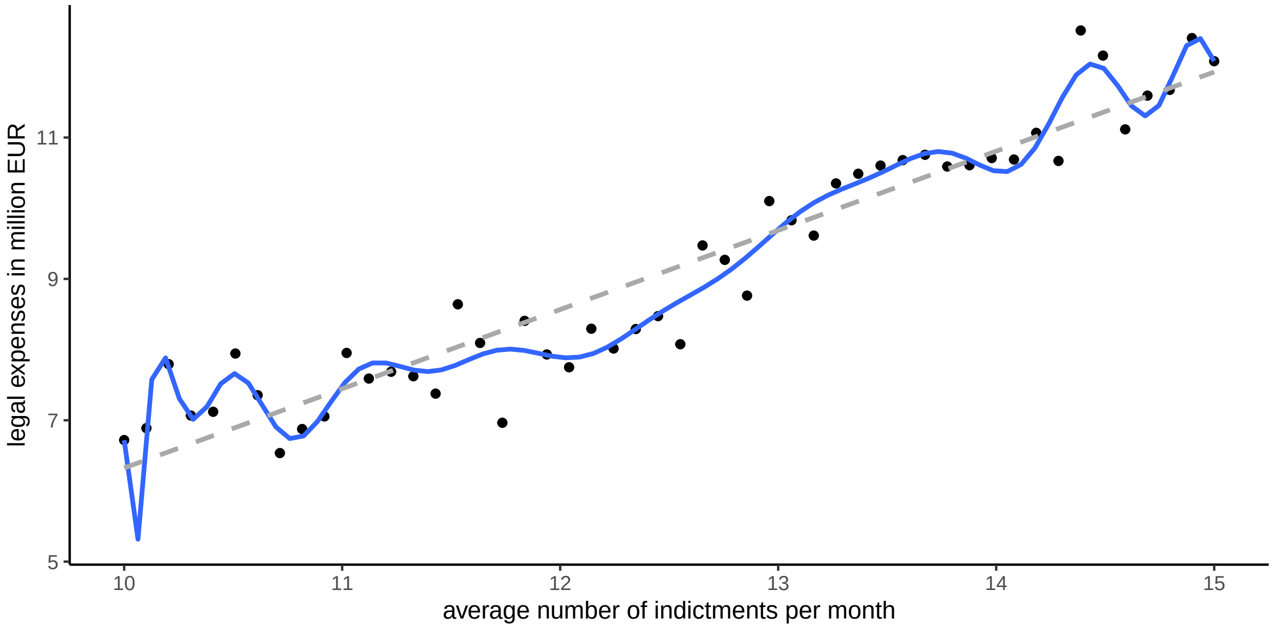
Exercise 1: Evaluating regression learners

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 $L2$ 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 ($n_{\text{test}} = 20$), using

   i) mean squared error (MSE), and

   ii) 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 = 19)
y_test <- 10 + 3 * sin(0.15 * pi * x_test) + rnorm(length(x_test), sd = 0.1)
data_test <- data.frame(x = c(x_test, 15), y = c(y_test, 20))

data_test
```

```
##           x         y
## 1  10.00000  7.170490
## 2  10.27778  6.954462
## 3  10.55556  7.074424
## 4  10.83333  7.216397
## 5  11.11111  7.389528
## 6  11.38889  7.646758
## 7  11.66667  7.951364
## 8  11.94444  8.197029
## 9  12.22222  8.533911
## 10 12.50000  8.796758
## 11 12.77778  9.258313
## 12 13.05556  9.756881
## 13 13.33333 10.018833
## 14 13.61111 10.635905
## 15 13.88889 10.661113
## 16 14.16667 11.067583
## 17 14.44444 11.545607
## 18 14.72222 11.868318
## 19 15.00000 12.179079
## 20 15.00000 20.000000
```

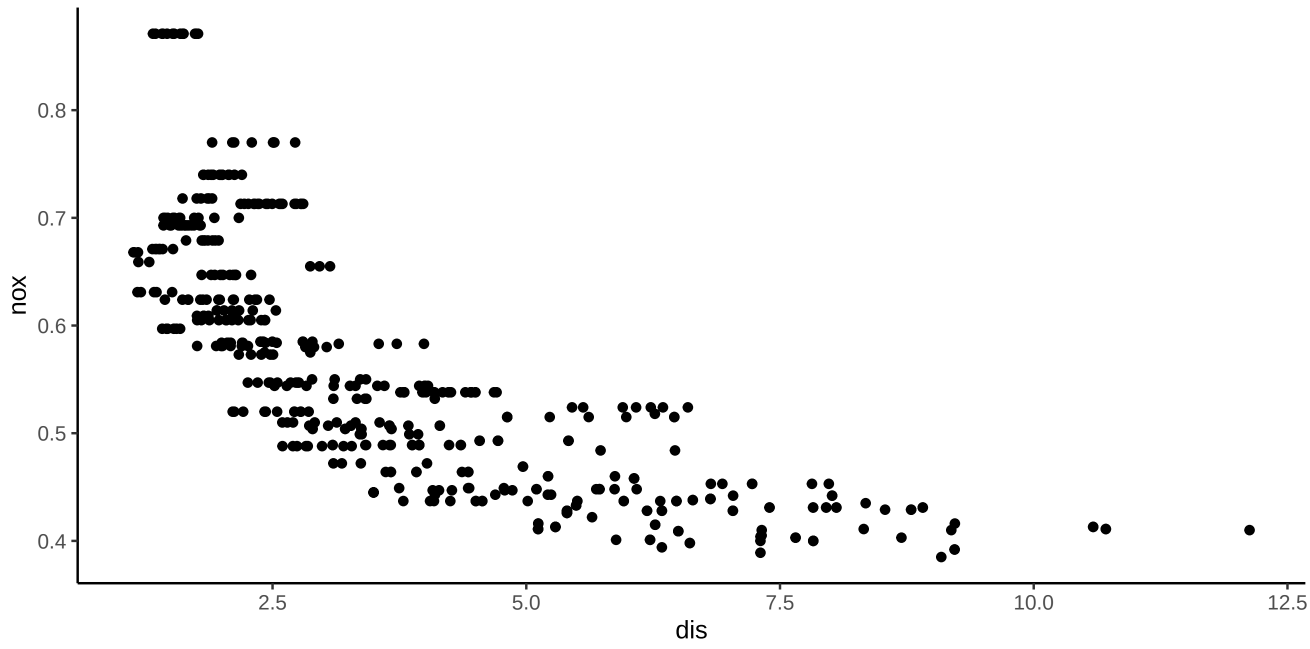
Exercise 2: Importance of train-test split

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`).

```r
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
```

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



a) 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.

b) What might be disadvantageous about the train-test split in a)?

c) 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.)

d) Interpret the findings from c).

e) After this empirical experiment we take a look at the mathematical background of the bias-variance trade-off in choosing the split ratio (with no specific assumption about the kind of learner used). Consider the expected quadratic error between predictions $\hat{y}$ and target values y, given the training data:

$$\mathbb{E}_{\mathbb{P}_{xy}}((\hat{y} - y)^2 \mid \mathbf{x})$$

and first show that

$$\mathbb{E}_{\mathbb{P}_{xy}}((\hat{y} - y)^2 \mid \mathbf{x}) = \left(\hat{y} - \mathbb{E}_{\mathbb{P}_{xy}}(y \mid \mathbf{x})\right)^2 + \mathsf{Var}_{\mathbb{P}_{xy}}(y \mid \mathbf{x}).$$

We then go one step further and treat our prediction $\hat{y}$ as a random variable whose value depends on how training and test observations are sampled. This sampling process we denote by $\mathcal{S}$. Building on the previous exercise, show that

$$\mathbb{E}_{\mathcal{S},\mathbb{P}_{xy}}((\hat{y} - y)^2 \mid \mathbf{x}) = \left(\mathbb{E}_{\mathcal{S}}(\hat{y}) - \mathbb{E}_{\mathbb{P}_{xy}}(y \mid \mathbf{x})\right)^2 + \mathsf{Var}_{\mathcal{S}}(\hat{y}) + \mathbb{E}_{\mathbb{P}_{xy}}(y \mid \mathbf{x})$$

and identify the components that represent bias and error, respectively. Can you imagine what the remaining term stands for?