Total: 9.6 points %---------------------------------------------------------------

Exercise 1: Some issues with the in-sample error calculation: Note that # Import libraries should read # Load packages Note that if (!require("ElemStatLearn")) install.packages("C:/Users/avalder/Downloads/ElemStatLearn\_2015.6.26 (1).tar.gz", repos = NULL, type = "source") uses an absolute path. You could insert the URL "https://cran.r-project.org/src/contrib/Archive/ElemStatLearn/ElemStatLearn\_2015.6.26.2.tar.gz" to ensure that the code runs in general. The test data is not generated correctly. You use y\_new <- fitted\_values[i] + errors # generate new responses for y\_j = f(x\_i) + error\_j, j = 1,...,N\_error which uses for the mean X \hat{\beta}. However, one should draw from the true data generation process with X\beta. You only use each x value once. To obtain a less volatile estimate for the in-sample error, one could have drawn several new y\_new datasets for the given x values and averaged over them. %---------------------------------------------------------------

Exercise 2: Good, except: For the empirical bootstrap you use the same data points for fitting and for evaluation. In the bootstrap case one could either use all data for evaluation or only those which have not been included in the bootstrap samples to obtain a more realistic estimate of the prediction error. %---------------------------------------------------------------

Exercise 3: Good. I personally prefer to use the base R pipe |> instead of the magrittr pipe %>%. Note that set.seed(123) foldid <- sample(1:10, size = nrow(X), replace = TRUE) # Example: 10-fold CV introduces some randomness into the size of the folds. This could be avoided for example using set.seed(123) foldid <- sample(rep(1:10, length.out = nrow(X))) %---------------------------------------------------------------

Exercise 4: Good. %--------------------------------------------------------------- Exercise 5: Good. Note that you could have also used dbinom() to determine the log-likelihood: > mean(dbinom(phontrain$phonclass, size = 1, prob = phonprob.train, log = TRUE)) [1] -0.1927175 %---------------------------------------------------------------