# Comments

Total: 10 points Very nice, detailed comments below.

Exercise 1: Good. correlation between tch and hdl -> absolute correlation between tch and hdl # Import libraries -> # Import packages The pairs plot is not optimal. You already changed the pch value to something more suitable. I would then also suggest to use alpha shading, e.g., pairs(diabetes, pch = 19, col = rgb(0, 0, 0, 0.1))

Exercise 2: Good. Note that you do not need to use wage$ in the mutate statement. I.e., Wage <- Wage %>% mutate("education" = factor(Wage$education, ordered = TRUE)) can be replaced with Wage <- Wage %>% mutate("education" = factor(education, ordered = TRUE)) because dplyr uses non-standard evaluation. You could have included the poly() in the formula as well as included the contrasts in the lm() call: lm(wage ~ poly(age, degree = 10, raw = TRUE) + education, data = Wage, contrasts = list(education = "contr.poly")) Using poly() in the formula is important for predicting new observations. Regarding contrasts for ordinal variables, using polynomial contrasts might be hard to interpret. But using treatment contrasts is usually not that interesting in case one already has an effect from the first to the second level. I personally also like MASS::contr.sdif which looks at the additional effect by the next level.

Exercise 3: Good. You assume that \beta\_0 is known and not estimated. The code beta\_hat <- solve(crossprod(X\_train))%\*%crossprod(X\_train,y\_train) is not computationally efficient. Note that solve() has two arguments so that one can solve the linear equation system by already providing the right hand side. Alternatively lm.fit() in R uses QR decomposition to obtain the OLS estimates. Cholesky decomposition would be a further option.

Exercise 4: Good. Instead of using the result for the \chi^2 distribution, one could also make use of the trick E(e^\top X e) = E[tr(e^\top X e)] = E[tr(Xee^\top)] = tr(XE(ee^\top)) with E(ee^\top) being the variance-covariance matrix of e.

Exercise 5: Good. You use y <- rowSums(X[,1:10]) + rnorm(100, mean=0,sd=0.1) But according to instructions one should have used a variance of 0.1.