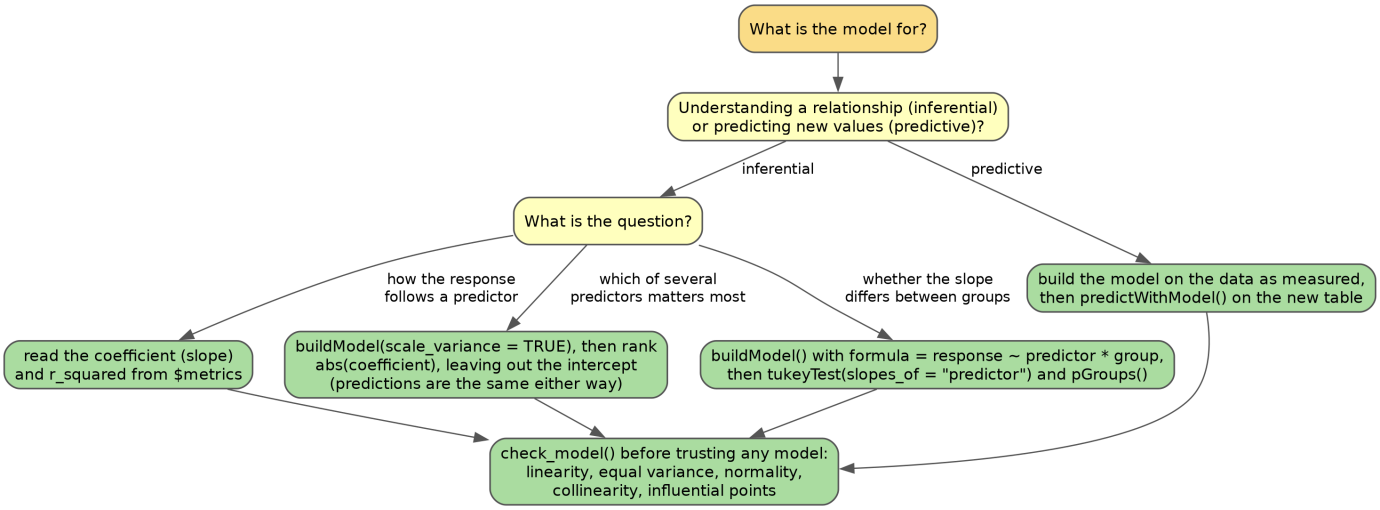
regression models: summary

Integrated Bioanalytical Chemistry · thebustalab.github.io/integrated_bioanalytics



Building a regression model starts with what it is for. An **inferential** use reads the coefficients, their p-values and r-squared to describe how the response relates to the predictors; a **predictive** use applies the model to a table whose response was never measured. Four decisions follow. **How many predictors:** more can explain more of the variation, as r-squared shows. **Whether to scale:** build with `scale_variance = TRUE` before ranking coefficients by size, and leave it off to read them in real units. The fit and the predictions are the same either way. **Whether slopes differ between groups:** join the predictor and the group with `*`, not `+` (one shared slope), then compare the slopes with `tukeyTest(slopes_of = )`. **Whether to trust it:** check the residuals, the predictors and any influential points with `check_model()`.

Function or step	What it is for	What we set or vary
`Hmisc::impute()`	filling in missing values, because `buildModel()` stops on any	the fill value (here each column's median)
`buildModel()`	fitting a model	`data`, `model_type = "linear_regression"`, `predictors` (one or more), `response`; or a `formula`, `response ~ predictor * group`
`$model`, `$metrics`	the fitted model, and a table of its r-squared, coefficients and p-values	`filter(type == "coefficient")`, `filter(type == "statistic")`, `filter(variable == "r_squared")`
`predictWithModel()`	predictions for any table with the same predictor columns, returned as that table with a new `predicted_<response>` column	`data`, and `model` (the whole `buildModel()` output)
`buildModel(scale_variance = TRUE)`	giving the numeric predictors a mean of 0 and a standard deviation of 1 before comparing coefficients, with `predictWithModel()` scaling new data to match	TRUE to rank predictors, FALSE (the default) for coefficients in real units
`arrange(desc(abs(value)))`	ranking coefficients by size	drop the intercept first with `variable != "(Intercept)"`
`tukeyTest(slopes_of = )`, `pGroups()`	which groups' slopes differ in a model built with `*`; letters and each group's slope	`slopes_of`, the numeric predictor whose slope is compared
`check_model()`	diagnostic panels: posterior predictive check, linearity, equal variance, normal residuals, collinearity, influential points	given `model$model`