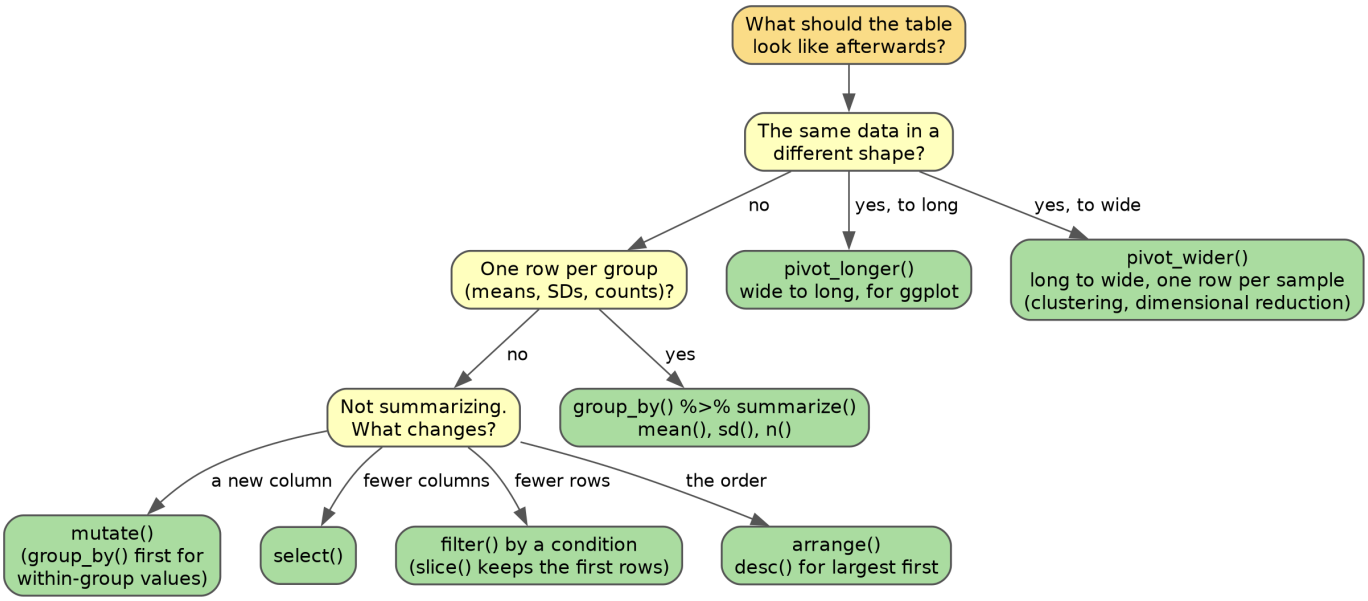
wrangling and summaries: summary

Integrated Bioanalytical Chemistry · thebustalab.github.io/integrated_bioanalytics



Wrangling comes down to deciding what the table should look like after each step. Pivoting changes the shape without changing the data: long tables suit ggplot, and wide tables with one row per sample suit the matrix analyses later in the book. `group_by()` with `summarize()` collapses each group to a single row, while `mutate()` keeps every row and adds a column. The pipe (`%>%`) chains these steps together, and `view()` on the result is the quickest way to check that each step did what was intended.

Function	What it does	What to set
`view()`	opens a whole table as a scrollable grid, with each column's type	the table, plus `n = 5000` to see past the first 1000 rows (`View()` in RStudio)
`select()`	keeps only the named columns	the column names
`filter()`	keeps only the rows that meet a condition	the condition, such as `park == "BELA"`
`pivot_longer()`	wide to long: column names and values each become a new column	`cols`, `names_to`, `values_to`
`pivot_wider()`	long to wide, the exact inverse of `pivot_longer()`	`names_from`, `values_from`
`%>%`	passes the result on the left to the function on the right as its first argument	
`group_by()` + `summarize()`	one row per group, holding the statistics asked for	the grouping columns, with `mean()`, `sd()`, `n()` inside `summarize()`
`arrange()`	sorts rows, smallest first	the column, wrapped in `desc()` for largest first
`slice()`	keeps the first n rows (of each group, after `group_by()`)	n, as in `slice(1)`
`mutate()`	adds a new column and keeps every row, and works within groups after `group_by()`	the new column, as in `percent_S = mg_per_L/sum(mg_per_L)*100`
`geom_pointrange()`	plots a summary point with a range around it	`x` and `y`, plus `ymin` and `ymax` (or `xmin` and `xmax`)