Day 4 Cheatsheet

Data Manipulation

Major concepts:

- `Wide` data - multiple columns per individual, values spread across multiple columns
- `Long` data - multiple rows per observation, a single column contains the values

Functions

Library/Package	Piece of code	Example of usage	What it does
dplyr	separate()	`df %>% separate(x, c("A", "B"))`	Separate a character column into multiple columns with a regular expression or numeric locations
dplyr	unite()	`df %>% unite("z", x:y, remove = FALSE)`	Unite multiple columns together into one column
tidyr	pivot_longer()	`df %>% pivot_longer(!column_to_not_touch, names_to = "new_col_with_labels", values_to = "new_col_with_values")`	Lengthens a data frame by increasing the number of rows and decreasing the number of columns.
tidyr	pivot_wider()	`df %>% pivot_wider(names_from = "col_with_names", values_from = "col_with_values")`	Widens a data frame by decreasing the number of rows and increasing the number of columns.
dplyr	?_join()	`inner_join(x, y)`	Joins data from two data frames. **inner_join** - only rows that match for x and y are kept. **full_join** - all rows of x and y are kept. **left_join** - all rows of x are kept even if not merged with y. **right_join** - all rows of y are kept even if not merged with x. **anti_join** - all rows from x not in y keeping just columns from x.
Base `R`	duplicated()	`duplicated(x)`	Determines and removes duplicate elements from `x`.
Base `R`	t()	Transpose	Returns the transpose of a matrix or data frame. If given a data frame, returns a matrix.