The *Add Rows* Family of Functions

There are two functions that will add rows to a **gt** table: `summary_rows()` and `grand_summary_rows()`. These are useful for adding groupwise and grand summary rows.

`summary_rows()`: Add groupwise summary rows using aggregation functions

Add summary rows to one or more row groups by using the table data and any suitable aggregation functions.

EXAMPLE

Use `sp500` to create a **gt** table with row groups. Create summary rows (`min`, `max`, `avg`) by row group, where each each row group is a week number.

```
sp500 %>%
  dplyr::filter(
    date >= "2015-01-05" &
      date <="2015-01-16"
  ) %>%
  dplyr::arrange(date) %>%
  dplyr::mutate(
    week = paste0(
      "W", strftime(date, format = "%V"))
  ) %>%
  dplyr::select(-adj_close, -volume) %>%
  gt(rowname_col = "date", groupname_col = "week") %>%
  summary_rows(
    groups = TRUE,
    columns = c(open, high, low, close),
    fns = list(
      min = ~min(.),
      max = ~max(.),
      avg = ~mean(.)),
    formatter = fmt_number,
    use_seps = FALSE
  )
```

	open	high	low	close
W02				
2015-01-05	2054.44	2054.44	2017.34	2020.58
2015-01-06	2022.15	2030.25	1992.44	2002.61
2015-01-07	2005.55	2029.61	2005.55	2025.90
2015-01-08	2030.61	2064.08	2030.61	2062.14
2015-01-09	2063.45	2064.43	2038.33	2044.81
min	2005.55	2029.61	1992.44	2002.61

	open	high	low	close
max	2063.45	2064.43	2038.33	2062.14
avg	2035.24	2048.56	2016.85	2031.21
W03				
2015-01-12	2046.13	2049.30	2022.58	2028.26
2015-01-13	2031.58	2056.93	2008.25	2023.03
2015-01-14	2018.40	2018.40	1988.44	2011.27
2015-01-15	2013.75	2021.35	1991.47	1992.67
2015-01-16	1992.25	2020.46	1988.12	2019.42
min	1992.25	2018.40	1988.12	1992.67
max	2046.13	2056.93	2022.58	2028.26
avg	2020.42	2033.29	1999.77	2014.93

`grand_summary_rows()` : Add grand summary rows using aggregation functions

Add grand summary rows to the **gt** table by using applying aggregation functions to the table data. The summary rows incorporate all of the available data, regardless of whether some of the data are part of row groups.

EXAMPLE

Use `sp500` to create a gt table with row groups. Create grand summary rows (`min` , `max` , `avg`) for the table.

```
sp500 %>%
  dplyr::filter(
    date >= "2015-01-05" &
      date <="2015-01-16"
  ) %>%
  dplyr::arrange(date) %>%
  dplyr::mutate(
    week = paste0(
      "W", strftime(date, format = "%V"))
  ) %>%
  dplyr::select(-adj_close, -volume) %>%
  gt(
    rowname_col = "date",
    groupname_col = "week"
  ) %>%
  grand_summary_rows(
    columns = c(open, high, low, close),
    fns = list(
      min = ~min(.),
      max = ~max(.),
      avg = ~mean(.)),
    formatter = fmt_number,
    use_seps = FALSE
  )
```

	open	high	low	close
W02				
2015-01-05	2054.44	2054.44	2017.34	2020.58
2015-01-06	2022.15	2030.25	1992.44	2002.61
2015-01-07	2005.55	2029.61	2005.55	2025.90
2015-01-08	2030.61	2064.08	2030.61	2062.14
2015-01-09	2063.45	2064.43	2038.33	2044.81
W03				
2015-01-12	2046.13	2049.30	2022.58	2028.26
2015-01-13	2031.58	2056.93	2008.25	2023.03
2015-01-14	2018.40	2018.40	1988.44	2011.27
2015-01-15	2013.75	2021.35	1991.47	1992.67
2015-01-16	1992.25	2020.46	1988.12	2019.42
min	1992.25	2018.40	1988.12	1992.67

	open	high	low	close
max	2063.45	2064.43	2038.33	2062.14
avg	2027.83	2040.92	2008.31	2023.07