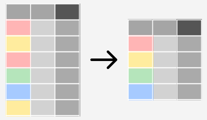
Group Data



Group by values in column named "col", returning a GroupBy object

```
df.groupby("groups")
```

All of the aggregation functions from above can be applied to a group as well

```
df.groupby(by="groups").agg(
  [
      # Sum values
      pl.sum("random").alias("sum"),

      # Minimum value
      pl.min("random").alias("min"),

      # Maximum value
      pl.max("random").alias("max"),
      # or
      pl.col("random").max().alias("other_max")

      # Standard deviation
      pl.std("random").alias("std_dev"),

      # Variance
      pl.var("random").alias("variance"),

      # Median
      pl.median("random").alias("median"),

      # Mean
      pl.mean("random").alias("mean"),

      # Quantile
      pl.quantile("random", 0.75) \
        .alias("quantile_0.75"),
      # or
      pl.col("random").quantile(0.75) \
        .alias("other_quantile_0.75"),

      # First value
      pl.first("random").alias("first"),
  ]
)
```

Additional GroupBy functions

```
df.groupby(by="groups").agg(
  [
    # Count the number of values in each group
    pl.count("random").alias("size"),

    # Sample one element in each group
    pl.col("names").apply(
      lambda group_df: group_df.sample(1)
    ),
  ]
)
```