```
  # Min distance between points
  min_distance = map_dbl(colors, ~ min_distance(.x, "hunterlab")),
  min_distance_deutan = map_dbl(colors, ~ min_distance(deutan(.x), "hunterlab")),
  min_distance_protan = map_dbl(colors, ~ min_distance(protan(.x), "hunterlab")),
  min_distance_tritan = map_dbl(colors, ~ min_distance(tritan(.x), "hunterlab")),
  # Max distance between points
  max_distance = map_dbl(colors, ~ max_distance(.x, "hunterlab")),
  max_distance_deutan = map_dbl(colors, ~ max_distance(deutan(.x), "hunterlab")),
  max_distance_protan = map_dbl(colors, ~ max_distance(protan(.x), "hunterlab")),
  max_distance_tritan = map_dbl(colors, ~ max_distance(tritan(.x), "hunterlab")),
  # IQR distance between points
  iqr_distance = map_dbl(colors, ~ iqr_distance(.x, "hunterlab")),
  iqr_distance_deutan = map_dbl(colors, ~ iqr_distance(deutan(.x), "hunterlab")),
  iqr_distance_protan = map_dbl(colors, ~ iqr_distance(protan(.x), "hunterlab")),
  iqr_distance_tritan = map_dbl(colors, ~ iqr_distance(tritan(.x), "hunterlab"))
  )
```

SCALING

```
data_meta <- data_num %>%
  mutate(id = as.character(str_glue("{package}-{palette}-{group}")))

data_prescaled <- data_meta %>%
  select(-package, -palette, -colors) %>%
  mutate_at(vars(type, group), funs(as.factor))

library(recipes)

re <- recipe(id ~ ., data = data_prescaled) %>%
  step_dummy(type, group) %>%
  step_center(all_predictors()) %>%
  step_scale(all_predictors()) %>%
  prep()

data_scaled <- bake(re, newdata = data_prescaled)
```