This is a summary of our current matching methods for creating a weight-based matching algorithm for factorial designs.

## First step:

Originally we'd be estimating optimal weights $w_i$ for each observation $i$. These weights encode balance of both treatment assignment and covariate values, but are continuous. We want to create a subsample of these weights for matching, so we instead solve a different optimization problem with the same balancing constraints but this time changing the objective function to a maximization of the sample size:

Obj: $\max \sum w_i$

as opposed to

Obj: $\min \sum w_i^2$

However, it may be infeasible to exactly solve the original optimization problem's balancing constraints if $w_i \in{0,1}$. It cannot pick a strict subset that exactly reproduces the entire sample's sums.

Therefore, we allow a tolerance on the SMD of the lhs and rhs for every balancing constraints. We calculate the SMD of the lhs and rhs of every balancing constraints and let them smaller than this tolerance value.

## Second step:

We propose a method for conducting matching after getting the initial subsample of 'weighted' observations.

The main idea is we perform cluster based matching, aiming to group observations into matched sets that are as similar as possible in terms of covariate values (minimizing mahalanobis distance) while maximizing the number of matched sets.

The objective function for this is:

$$ \text{minimize} \quad \sum_{(i,j)} d_{(i,j)} \cdot x_{(i,j)} - \gamma \sum_i y_i + \sum_{(i,k)} \omega_k \cdot z_{(i,k)} $$

Summing the Mahalanobis distance measure between covariates, $d_{(i,j)}$, over all pairs of observations $(i,j)$, with the penalty term $\gamma \sum_i y_i + \sum_{(i,k)} \omega_k \cdot z_{(i,k)}$ creating and maximizing the number of matched sets. The optimization problem maximizes the number of clusters / matched samples while minimizing the distance in covariate space within each matched set.

Some notes about the variables used:

- $\gamma$ hyperparameter (higher = less matched sets, lower = more matched sets)
    - The idea being, that more matched sets = higher precision/closer distance within that matched set, so we have better guarantees on the actual covariate balance within each matched set, as opposed to one single matched set where covariate balance can vary
    - However, tradeoff is that we will have smaller samples within each matched set, and we may also not have observations within a matched set that are representative of the whole treatment combination space (i.e, you will have multiple observations with 1-1-0 treatment and only one of 0-0-1, which isn't reliable for making a causal estimate).
- Imbalance penalty term $\omega_k$

    - (In Zubizarreta, uses higher moments such as variance, skew, etc, to additionally balance based on distribution and other statistical properties)
    - We can probably toss this out
- $z_{(i,k)}$ is the absolute deviation of the cluster mean for covariate $k$ from the overall mean of covariate k. Think of it as the term that controls for the size of each cluster.

Additional modifications:

Adds some additional balancing constraints / terms to objective function that:

- Ensure a minimum representation of treatment combinations within each matched set
- Ensure uniformity of matched set cluster sizes

Mathematically they are represented as:

$$ C_{(i, t)} \geq m_t \cdot y_i \forall i, t$$

$$ S_i \leq s_{max} + M \cdot (1-y_i) \forall i$$

$$ S_i \geq s_{min} - M \cdot (1-y_i) \forall i$$

$$ x_{(i, j)}, y_i \in {0, 1} $$

$$ z_{(i, k)}, S_i, T_{(i, k)} \geq 0 \forall i, k$$

$$ C_{(i, t)} \geq \Z_+ \forall i, t$$

$C_{(i,t)}$ counts the number of units with treatment $t$ in cluster $i$

$m_t$ is the minimum required count for treatment combination $t$ in each cluster

$s_max$ and $s_min$ are the maximum and minimum matched set / cluster sizes smin are the maximum and minimum cluster sizes

$\delta$ is the hyperparameter constant for cluster size balance penalty

However, this would obviously come with the tradeoff that this reduces covariate balance within each matched set, which is why we're including these terms of hyperparameters to be tuned. This will also obviously raise the overall computational complexity of the optimization problem, which we have to consider given this is design-stage methodology.

By controlling for representation, i.e imposing some minimum of treatment combinations within each matched set, we are also implicitly controlling for the size of each matched set. This is a small but noteworthy simplification of the optimization problem.

What would we do to calculate treatment effects after peforming matching?

- Within each matched set, fit a regression model and derive treatment effect estimates $\hat{\tau}$ from the fitted coefficients$.