Example with Two Houses (Citizen and Token)

Preconditions:

- The voting results in the file can only be FOR or AGAINST.
- The column name representing proposal results in `voting_res_filter` must be `proposal_choice`.

Filtering Data and Adding Proposal Results

```
prop_token_filter_t = proposal_filter(prop_token_t, 5)
prop_cit_filter_t = proposal_filter(prop_cit_t, 5)

voting_res_filter_final_t = final_voting_result(voting_res_filter)

prop_token_filter_add_t = proposal_result_merge(voting_res_filter_final_t, prop_token_filter_t)
prop_cit_filter_add_t = proposal_result_merge(voting_res_filter_final_t, prop_cit_filter_t)
```

Calculating Similarity Matrix and Network Clustering

```
M_token_tranc_t = Similar_matriix(prop_token_filter_add_t, 3)
M_cit_tranc_t = Similar_matriix(prop_cit_filter_add_t, 3)
set.seed(1)
community_token_t = Community_detection(M_token_tranc_t, 0.1, 3)
community_cit_t = Community_detection(M_cit_tranc_t, 0.1, 3)

Louvain_member_token_t = community_token_t$membership
Louvain_member_cit_t = community_cit_t$membership

Group_token_t = data.frame(address = unique(prop_token_filter_add_t$address),
group = Louvain_member_token_t)  # Display each address and its corresponding group
```

Logistic Regression

```
log_eff_token_t = Logistic_reg_single(prop_token_filter_add_t, Louvain_member_token_t, F)
ad_log_eff_token_t = Logistic_reg_single(prop_token_filter_add_t, Louvain_member_token_t, T)
hist(log_eff_token_t)
hist(ad_log_eff_token_t)

log_eff_cit_t = Logistic_reg_single(prop_cit_filter_add_t, Louvain_member_cit_t, F)
ad_log_eff_cit_t = Logistic_reg_single(prop_cit_filter_add_t, Louvain_member_cit_t, T)
hist(log_eff_cit_t)
hist(ad_log_eff_cit_t)
# var(log_eff_token_t[!is.na(log_eff_token_t)])

log_eff_both_t = Logistic_reg_multiple(prop_cit_filter_add_t, prop_token_filter_add_t)
hist(log_eff_both_t)
```

Counterfactual Logistic Regression

```
log_eff_token_rev_t = CT_Logistic_reg_single(prop_token_filter_t, Louvain_member_token_t, F)
hist(log_eff_token_rev_t)
ad_log_eff_token_rev_t = CT_Logistic_reg_single(prop_token_filter_t, Louvain_member_token_t, T)
hist(ad_log_eff_token_rev_t)  # Not the same

log_eff_cit_rev_t = CT_Logistic_reg_single(prop_cit_filter_t, Louvain_member_cit_t, F)
hist(log_eff_cit_rev_t)
ad_log_eff_cit_rev_t = CT_Logistic_reg_single(prop_cit_filter_t, Louvain_member_cit_t, T)
hist(ad_log_eff_cit_rev_t)

log_eff_both_rev_t = CT_Logistic_reg_multiple(prop_token_filter_t, prop_cit_filter_t)
hist(log_eff_both_rev_t)
```

Centralization Metrics

```
statistics_table <- Centrality_statistics(log_eff_both_rev_t)
print(statistics_table)
```