Simulation to assess varying coefficients and guideline

This runs a series of small simulations where we examine varying beta coefficients over time. In these simulations, we generate data and estimate the guidelines based on that data, and then compare these estimated recommendations to the oracle truth (using our general theorem) to see how our guideline works when it is technically misspecified.

In particular, this document produces Figure 4 and Table 2 in Appendix B.

Run simulation across range of sigma_pre

Our initial simulation has varying coefficients for both X and Z, along with θ. The misspecification gives a reduced recommendation to match as shown on the figure to the right.

```
# Number of simulation replicates per scenario
K = 100 #1000


sigma_pre_tests = seq( 0.3, 1.6, by=0.15 )

names(sigma_pre_tests) = sigma_pre_tests
sim_res <- map_df( sigma_pre_tests,
                   ~ run_scenario( sigma_pre = .,
                                   beta_theta_0 = c( 0.5, 1.0 ),
                                   beta_theta_1 = 1.5,
                                   beta_x_0 = c( 0.6, 1.1),
                                   beta_x_1 = 1.3,
                                   beta_z_0 = c( 0.3, 0.7 ),
                                   beta_z_1 = 1.0,
                                   cor_XZ = 0.5,
                                   K = K ),
                   .id = "sigma_pre" ) %>%
    mutate( sigma_pre = as.numeric(sigma_pre) )

saveRDS(sim_res, file = "~/Downloads/plot_df.rds")

sim_res
```

```
##   sigma_pre per_match     a_tau_xy   SE_tau_xy a_est_beta0 a_est_beta1
## 1      0.30      1.00  0.266621216 0.009181200   0.7507159    1.502332
## 2      0.45      1.00  0.200837815 0.008213436   0.7509806    1.501105
## 3      0.60      1.00  0.128404926 0.006431776   0.7482817    1.497770
## 4      0.75      1.00  0.057476403 0.005514885   0.7500856    1.501285
## 5      0.90      0.06 -0.008760078 0.005700840   0.7511143    1.501573
## 6      1.05      0.00 -0.064287938 0.007259206   0.7490938    1.498755
## 7      1.20      0.00 -0.113698434 0.008741283   0.7510503    1.501202
## 8      1.35      0.00 -0.151809898 0.010493378   0.7489427    1.500200
## 9      1.50      0.00 -0.185777608 0.014613596   0.7514319    1.498185
```

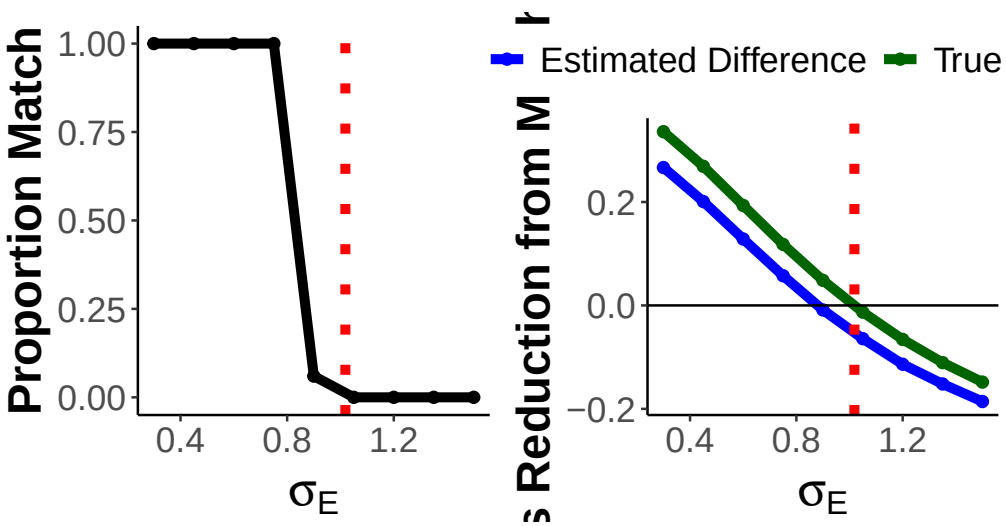
```
##   a_est_delta_theta a_est_sig_pre_sq        R2 bias_X    bias_XY match_X
## 1         0.9990255        0.1930239 0.7826691    0.4 0.06409496    TRUE
## 2         1.0008490        0.3056016 0.7516786    0.4 0.13108564    TRUE
## 3         0.9970248        0.4645840 0.7116574    0.4 0.20669856    TRUE
## 4         0.9990477        0.6658909 0.6708075    0.4 0.28198433    TRUE
## 5         0.9991613        0.9136137 0.6263417    0.4 0.35153707    TRUE
## 6         0.9997398        1.2029844 0.5839190    0.4 0.41295357    TRUE
## 7         0.9994045        1.5422719 0.5421425    0.4 0.46576819    TRUE
## 8         1.0028049        1.9258658 0.5025059    0.4 0.51053399    TRUE
## 9         1.0009102        2.3548697 0.4665626    0.4 0.54822335    TRUE
##   match_XY reduce_X   reduce_XY
## 1     TRUE      1.3  0.33590504
## 2     TRUE      1.3  0.26891436
## 3     TRUE      1.3  0.19330144
## 4     TRUE      1.3  0.11801567
## 5     TRUE      1.3  0.04846293
## 6    FALSE      1.3 -0.01295357
## 7    FALSE      1.3 -0.06576819
## 8    FALSE      1.3 -0.11053399
## 9    FALSE      1.3 -0.14822335
```

Our table shows, for different residual variation, the proportion of the trials that say "match!", the average estimated reduction in bias, the standard deviation of the estimates across simulation (which is the true SE), and whether the oracle says to match and how much bias would be reduced. The last column is the R^2 for a regression of outcome onto the two observed covariates to get a sense of how much variation is explained by what we can match on.

```
sim_res %>%
    dplyr::select( sigma_pre, per_match, a_tau_xy, SE_tau_xy, match_XY, reduce_XY, R2 ) %>%
    knitr::kable( digits = 3 )
```

sigma_pre	per_match	a_tau_xy	SE_tau_xy	match_XY	reduce_XY	R2
0.30	1.00	0.267	0.009	TRUE	0.336	0.783
0.45	1.00	0.201	0.008	TRUE	0.269	0.752
0.60	1.00	0.128	0.006	TRUE	0.193	0.712
0.75	1.00	0.057	0.006	TRUE	0.118	0.671
0.90	0.06	-0.009	0.006	TRUE	0.048	0.626
1.05	0.00	-0.064	0.007	FALSE	-0.013	0.584
1.20	0.00	-0.114	0.009	FALSE	-0.066	0.542
1.35	0.00	-0.152	0.010	FALSE	-0.111	0.503
1.50	0.00	-0.186	0.015	FALSE	-0.148	0.467

```
plt <- make_result_plot( sim_res )
plt
```


Proportion Match
1.00
0.75
0.50
0.25
0.00
0.4
0.8
1.2
σE
Estimated Difference
True
s Reduction from M
0.2
0.0
−0.2
0.4
0.8
1.2
σE

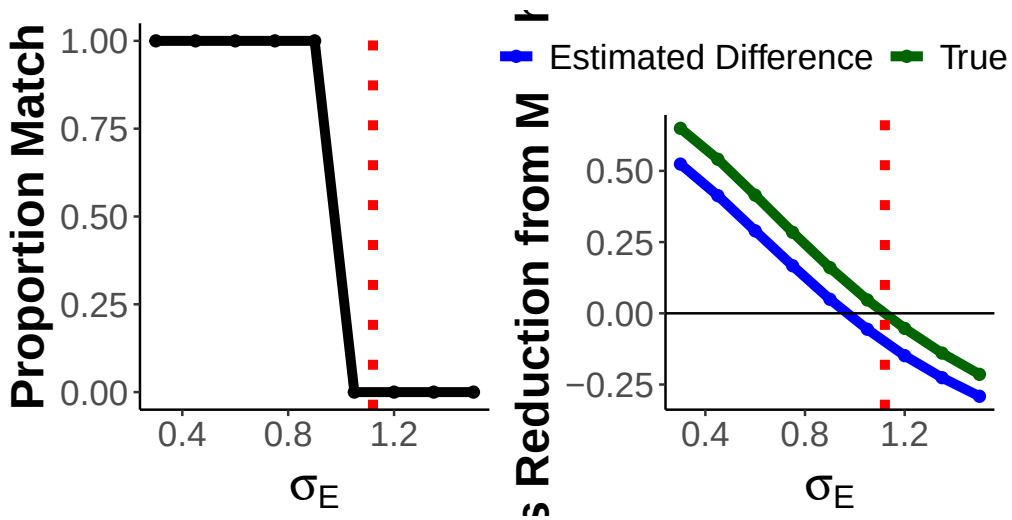
Independent covariates

This is our initial simulation, except the covariates are no independent. The independence does not help us?

```
sim_resB <- map_df( sigma_pre_tests,
                    ~ run_scenario( sigma_pre = .,
                                    beta_theta_0 = c( 0.5, 1.0 ),
                                    beta_theta_1 = 1.5,
                                    beta_x_0 = c( 0.6, 1.1),
                                    beta_x_1 = 1.3,
                                    beta_z_0 = c( 0.3, 0.7 ),
                                    beta_z_1 = 1.0,
                                    cor_XZ = 0.0,
                                    cor_Xtheta = c(0,0),
                                    K = K ),
                    .id = "sigma_pre" ) %>%
    mutate( sigma_pre = as.numeric(sigma_pre) )

saveRDS(sim_resB, file = "~/Downloads/plotB_df.rds")

plt <- make_result_plot( sim_resB )
plt
```



Alternate scenario: correct specification

If we have parallel trends for all three covariates (two observed, one latent) then our guideline works as expected.

Also note that due to large sample size our match recommendation is very precisely estimated because our bias reduction is also very precisely estimated (see the SE column).

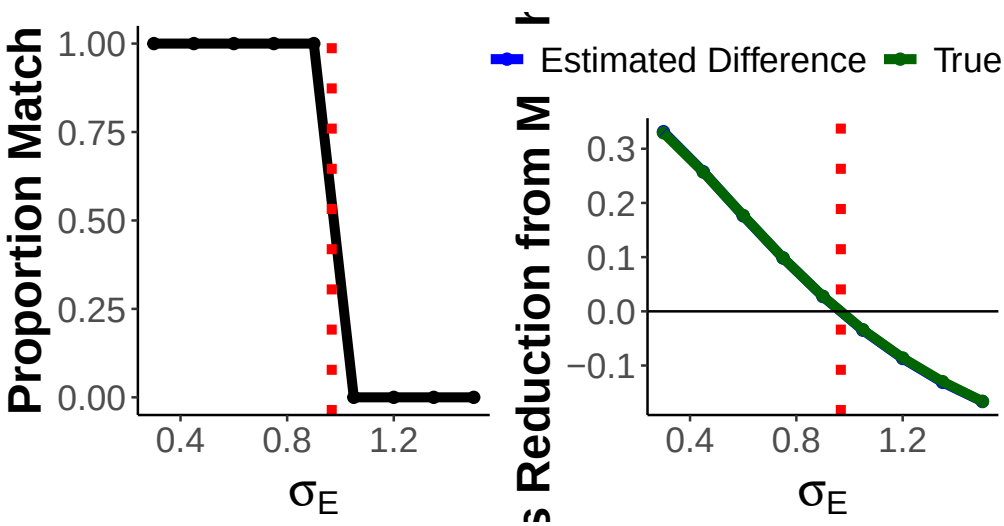
```
sim_res2 <- map_df( sigma_pre_tests,
                    ~ run_scenario( sigma_pre = .,
                                    K = K,
                                    beta_theta_0 = c( 0.75, 0.75 ),
                                    beta_theta_1 = 1.5,
                                    beta_x_0 = c( 0.85, 0.85),
                                    beta_x_1 = 1.3,
                                    beta_z_0 = c( 0.5, 0.5 ),
                                    beta_z_1 = 1.0,
                                    cor_XZ = 0.5 ),
    mutate( sigma_pre = as.numeric(sigma_pre) )

saveRDS(sim_res2, file = "~/Downloads/plot2_df.rds")
sim_res2 = readRDS( "~/Downloads/plot2_df.rds" )

sim_res2 %>%
    dplyr::select( sigma_pre, per_match, a_tau_xy, SE_tau_xy, match_XY, reduce_XY, R2 ) %>%
    knitr::kable( digits = 3 )
```

sigma_pre	per_match	a_tau_xy	SE_tau_xy	match_XY	reduce_XY	R2
0.30	1	0.331	0.011	TRUE	0.329	0.788
0.45	1	0.257	0.010	TRUE	0.257	0.764
0.60	1	0.176	0.007	TRUE	0.177	0.733
0.75	1	0.098	0.007	TRUE	0.098	0.697
0.90	1	0.027	0.006	TRUE	0.028	0.657
1.05	0	-0.035	0.006	FALSE	-0.034	0.616
1.20	0	-0.087	0.008	FALSE	-0.086	0.575
1.35	0	-0.131	0.009	FALSE	-0.130	0.536
1.50	0	-0.166	0.012	FALSE	-0.166	0.497

```
plt <- make_result_plot( sim_res2 )
plt
```


Proportion Match
1.00
0.75
0.50
0.25
0.00
0.4
0.8
1.2
σE
Estimated Difference
True
s Reduction from M
0.3
0.2
0.1
0.0
−0.1
0.4
0.8
1.2
σE

Alternate scenario: theta parallel, covariates not

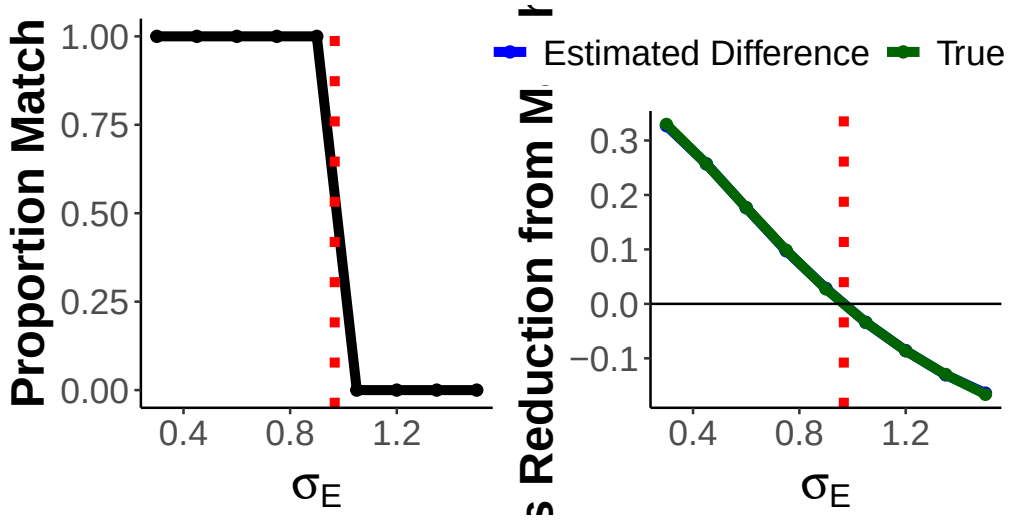
Here we have X and Z correlated, but theta is parallel.

```
sim_res3 <- map_df( sigma_pre_tests,
                    ~ run_scenario( sigma_pre = .,
                                    K = K,
                                    beta_theta_0 = c( 0.75, 0.75 ),
                                    beta_theta_1 = 1.5,
                                    beta_x_0 = c( 0.6, 1.1),
                                    beta_x_1 = 1.3,
                                    beta_z_0 = c( 0.3, 0.7 ),
                                    beta_z_1 = 1.0,
                                    cor_XZ = 0.5 ),
                    .id = "sigma_pre" ) %>%
    mutate( sigma_pre = as.numeric(sigma_pre) )

saveRDS(sim_res3, file = "~/Downloads/plot3_df.rds")
sim_res3 = readRDS("~/Downloads/plot3_df.rds" )
sim_res3 %>%
    dplyr::select( sigma_pre, per_match, a_tau_xy, SE_tau_xy, match_XY, reduce_XY, R2 ) %>%
    knitr::kable( digits = 3 )
```

sigma_pre	per_match	a_tau_xy	SE_tau_xy	match_XY	reduce_XY	R2
0.30	1	0.327	0.011	TRUE	0.329	0.770
0.45	1	0.257	0.009	TRUE	0.257	0.744
0.60	1	0.177	0.008	TRUE	0.177	0.711
0.75	1	0.098	0.006	TRUE	0.098	0.676
0.90	1	0.028	0.006	TRUE	0.028	0.634
1.05	0	-0.034	0.006	FALSE	-0.034	0.593
1.20	0	-0.086	0.007	FALSE	-0.086	0.554
1.35	0	-0.130	0.011	FALSE	-0.130	0.515
1.50	0	-0.164	0.011	FALSE	-0.166	0.479

```
plt <- make_result_plot( sim_res3 )
plt
```



Alternate scenario: less predictive covariates, more time periods

If we make covariates less predictive, but have more pre-treatment time periods? (We have also increased variation in theta to cause more trouble.)

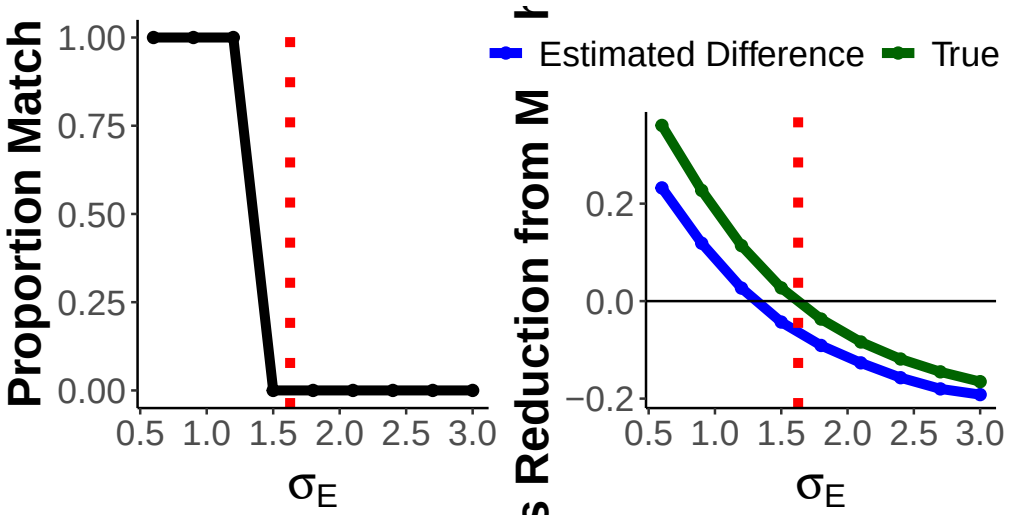
```
sigmas_larger = sigma_pre_tests * 2
names(sigmas_larger) = sigmas_larger
sim_res3 <- map_df( sigmas_larger,
                    ~ run_scenario( sigma_pre = .,
                                    beta_theta_0 = c( 0, 0.3, 0.7, 1.0 ),
                                    beta_theta_1 = 1.5,
                                    beta_x_0 = c( 0.6, 1.1, 1.1, 0.6 ),
                                    beta_x_1 = 1.3,
                                    beta_z_0 = c( 0.7, 0.7, 0.3, 0.3 ),
                                    beta_z_1 = 1.0,
                                    cor_XZ = 0.5,
                                    K = K ),
                    .id = "sigma_pre" ) %>%
    mutate( sigma_pre = as.numeric(sigma_pre) )

saveRDS(sim_res3, file = "~/Downloads/plot3_df.rds")

sim_res3 %>%
    dplyr::select( sigma_pre, per_match, a_tau_xy, SE_tau_xy, match_XY, reduce_XY, R2 ) %>%
    knitr::kable( digits = 3 )
```

sigma_pre	per_match	a_tau_xy	SE_tau_xy	match_XY	reduce_XY	R2
0.6	1	0.232	0.009	TRUE	0.360	0.728
0.9	1	0.119	0.006	TRUE	0.227	0.630
1.2	1	0.027	0.007	TRUE	0.114	0.533
1.5	0	-0.043	0.009	TRUE	0.027	0.448
1.8	0	-0.091	0.014	FALSE	-0.037	0.376
2.1	0	-0.127	0.014	FALSE	-0.084	0.317
2.4	0	-0.157	0.021	FALSE	-0.119	0.269
2.7	0	-0.180	0.024	FALSE	-0.145	0.231
3.0	0	-0.192	0.027	FALSE	-0.165	0.200

```
plt <- make_result_plot( sim_res3 )
plt
```


Proportion Match
1.00
0.75
0.50
0.25
0.00
0.5 1.0 1.5 2.0 2.5 3.0
σE
Estimated Difference
True
0.2
0.0
−0.2
0.5 1.0 1.5 2.0 2.5 3.0
σE

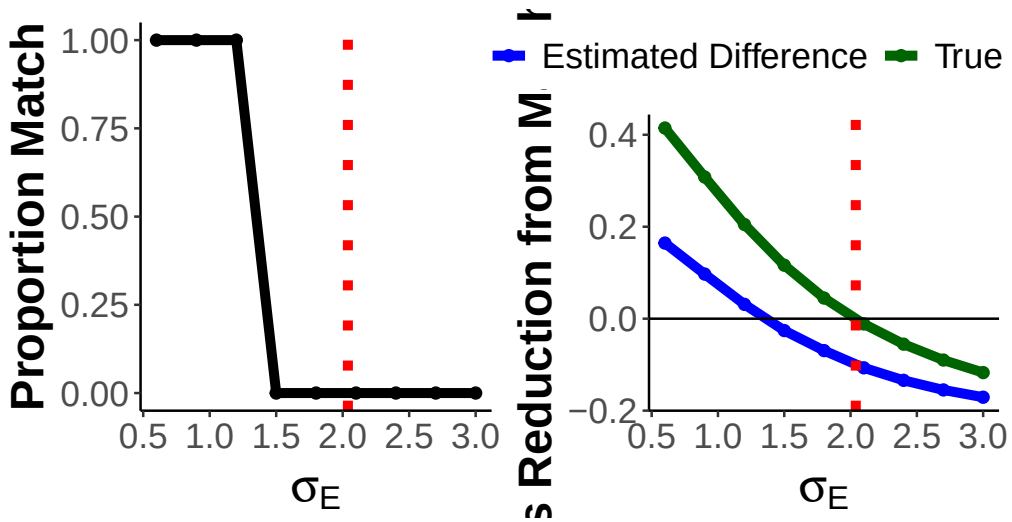
Alternate scenario: narrow theta assumption only

If all we get is parallel theta in the final two periods, but theta is not parallel before that?

```
sim_res4 <- map_df( sigmas_larger,
                    ~ run_scenario( sigma_pre = .,
                                    beta_theta_0 = c( -0.5, 0.5, 1.0, 1.0 ),
                                    beta_theta_1 = 1.5,
                                    beta_x_0 = c( 0.6, 1.1, 1.1, 0.6 ),
                                    beta_x_1 = 1.3,
                                    beta_z_0 = c( 0.7, 0.7, 0.3, 0.3 ),
                                    beta_z_1 = 1.0,
                                    cor_XZ = 0.5,
                                    K = K ),
                    .id = "sigma_pre" ) %>%
    mutate( sigma_pre = as.numeric(sigma_pre) )

saveRDS(sim_res4, file = "~/Downloads/plot4_df.rds")

plt <- make_result_plot( sim_res4 )
plt
```



Alternate scenario: small sample size

If we reduce sample size, estimation error should flatten our curve. It seems like estimation error is very small, which is surprising given all the residualization?

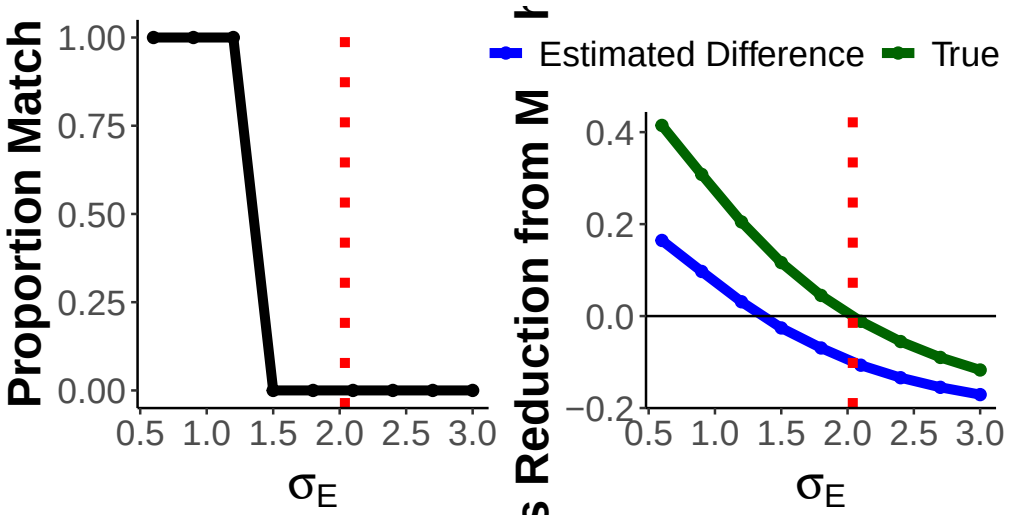
```
sim_res5 <- map_df( sigma_pre_tests,
                    ~ run_scenario( sigma_pre = .,
                                    beta_theta_0 = c( 0.5, 1.0 ),
                                    beta_theta_1 = 1.5,
                                    beta_x_0 = c( 0.6, 1.1),
                                    beta_x_1 = 1.3,
                                    beta_z_0 = c( 0.3, 0.7 ),
                                    beta_z_1 = 1.0,
                                    cor_XZ = 0.5,
                                     N = 3000 ),
                    .id = "sigma_pre" ) %>%
    mutate( sigma_pre = as.numeric(sigma_pre) )

sim_res5 %>%
    dplyr::select( sigma_pre, per_match, a_tau_xy, SE_tau_xy, match_XY, reduce_XY, R2 ) %>%
    knitr::kable( digits = 3 )
```

sigma_pre	per_match	a_tau_xy	SE_tau_xy	match_XY	reduce_XY	R2
0.30	1	0.275	0.027	TRUE	0.336	0.782
0.45	1	0.197	0.018	TRUE	0.269	0.752
0.60	1	0.129	0.019	TRUE	0.193	0.713
0.75	1	0.053	0.013	TRUE	0.118	0.673
0.90	0	-0.017	0.013	TRUE	0.048	0.624
1.05	0	-0.058	0.020	FALSE	-0.013	0.588
1.20	0	-0.106	0.019	FALSE	-0.066	0.543
1.35	0	-0.164	0.027	FALSE	-0.111	0.506
1.50	0	-0.184	0.039	FALSE	-0.148	0.468

```
saveRDS(sim_res5, file = "~/Downloads/plot5_df.rds")

plt <- make_result_plot( sim_res4 )
plt
```


Proportion Match
1.00
0.75
0.50
0.25
0.00
0.5 1.0 1.5 2.0 2.5 3.0
σE
Estimated Difference
True
0.4
0.2
0.0
−0.2
0.5 1.0 1.5 2.0 2.5 3.0
σE

Alternate scenario: theta independent

Here we have our theta stable in the final two time periods, and no correlation between any of our three. covariates.

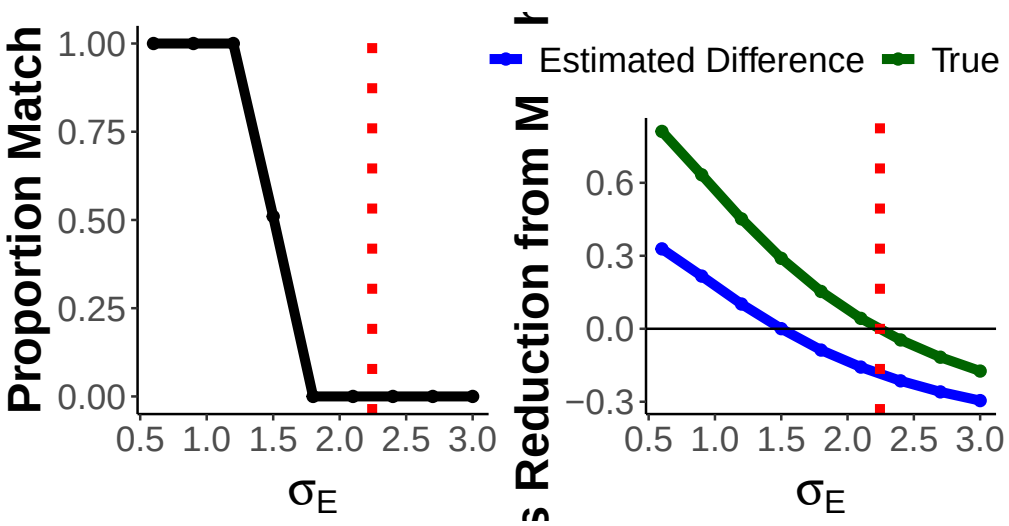
```
sim_res6 <- map_df( sigmas_larger,
                    ~ run_scenario( sigma_pre = .,
                                    beta_theta_0 = c( -0.5, 0.5, 1.0, 1.0 ),
                                    beta_theta_1 = 1.5,
                                    beta_x_0 = c( 0.6, 1.1, 1.1, 0.6 ),
                                    beta_x_1 = 1.3,
                                    beta_z_0 = c( 0.7, 0.7, 0.3, 0.3 ),
                                    beta_z_1 = 1.0,
                                    cor_XZ = 0,
                                    cor_Xtheta = c( 0, 0 ),
                                    K = K ),
                    .id = "sigma_pre" ) %>%
    mutate( sigma_pre = as.numeric(sigma_pre) )

saveRDS(sim_res6, file = "~/Downloads/plot6_df.rds")

sim_res6 %>%
    dplyr::select( sigma_pre, per_match, a_tau_xy, SE_tau_xy, match_XY, reduce_XY, R2 ) %>%
    knitr::kable( digits = 3 )
```

sigma_pre	per_match	a_tau_xy	SE_tau_xy	match_XY	reduce_XY	R2
0.6	1.00	0.328	0.009	TRUE	0.811	0.467
0.9	1.00	0.216	0.008	TRUE	0.633	0.390
1.2	1.00	0.102	0.008	TRUE	0.452	0.317
1.5	0.51	0.000	0.008	TRUE	0.289	0.258
1.8	0.00	-0.088	0.011	TRUE	0.153	0.212
2.1	0.00	-0.158	0.018	TRUE	0.043	0.175
2.4	0.00	-0.214	0.018	FALSE	-0.046	0.145
2.7	0.00	-0.260	0.025	FALSE	-0.117	0.123
3.0	0.00	-0.295	0.029	FALSE	-0.174	0.105

```
plt <- make_result_plot( sim_res6 )
plt
```


Proportion Match
1.00
0.75
0.50
0.25
0.00
0.5 1.0 1.5 2.0 2.5 3.0
σE
Estimated Difference
True
0.6
0.3
0.0
−0.3
0.5 1.0 1.5 2.0 2.5 3.0
σE