# Synthetic Differences in Differences Simulations

*Joshua Derenski*

## Simulations

### Parameters

- Number of Ls: 1
- Draws per L: 100
- Number of Units: 100
- Number of Control Units: 95
- Number of Times: 200
- Number of pre-treatment Times: 190
- Rank of L: 10
- Autocorrelation Parameter: 0
- True Effect Size for Constant Effect: 10
- Error Type: gaussian
- Error Variance (if Gaussian error): 16
- Degrees of freedom (if t-error): 5
- Exchangable: FALSE
- Penalized: TRUE
- Rank Estimation Method: threshold
- Scaling for $L$: 5
- Treatment Effect Type: constant
- Treatment Design: block_treatment
- Lag Structure (if using lagged adoption structure): random
- Average Treatment Length (if using lagged adoption structure, with random adoption): 6
- Maximum lag: 3

# Simulation Description

In this simulation we sample 1 signal matricies $L \in \mathbb{R}^{100\times200}$ of rank 10, and for each $L$ we sample 100 $Y s$ such that for each $Y$ $E(Y_{ij}) = L_{ij} + \tau W_{ij}$ where $W$ is known, and $\tau = 10$ (in the case of the t-distribution, we ensure the median of each cell is $L_{ij} + \tau W_{ij}$). We generate $L$ so that the rows and columns are not exchangable, and we estimate the weights in SDID using penalized regressions. For estimating the rank of $L$ we are using a threshold method. From each corresponding pair of $(L, Y)$ an estimate of $\tau$ is generated via method $i$, which we call $\hat{\tau}_{(L,Y),i}$. To evaluate the performance of method $i$ we calculate the following mse:

$$mse_i = \sqrt{\frac{1}{100} \sum_{(L,Y)} (\hat{\tau}_{(L,Y),i} - \tau)^2}$$

# Our Method vs Competitors, Fixed Parameters

## Results

### Signal to Noise Ratio

```
## [1] 21.79198
```

### mse for DID

```
##      mse
## 18.70946
```

### Se for mse for DID

```
##  se_mse
## 0.56006
```

### mse for SC

```
##      mse
## 7.527665
```

### Se for mse for SC

```
##    se_mse
## 0.2941494
```

### mse for our Method (Explicit Tau)

```
##      mse
## 3.130453
```

**Se for mse for our Method (Explicit Tau)**

```
##   se_mse
## 0.147327
```

**mse for SDID**

```
##      mse
## 7.210203
```

**Se for mse for SDID**

```
##    se_mse
## 0.2814516
```

**mse For Our Method (Not Explicit Tau)**

```
##      mse
## 3.130451
```

**Se for mse for Our Method (Not Explicit Tau)**

```
##    se_mse
## 0.1473274
```

**mse For Oracle (Perfect L)**

```
##      mse
## 2.279232
```

**mse For Oracle (Perfect L)**

```
##    se_mse
## 0.1095005
```

# Matrix Bias vs Reduction in Variance due to Averaging

For more general designs of $W$ (like the block design scheme considered here) we allow a block in the bottom right hand corner of $W$ to be non-zero. When implementing our method, we have two competing effects on estimation:

- The bias that's introduced by making more of the $Y_{ij}$s zero.
- The help we get with estimating $\tau$ by being able to average over cells (because we assume *tau*) is the same for all units and times.

It would appear that accurracy increases for estimating $\tau$ to a point, and then decreases when the bias introduced by replacement of cells with 0 in $Y$ becomes too great.

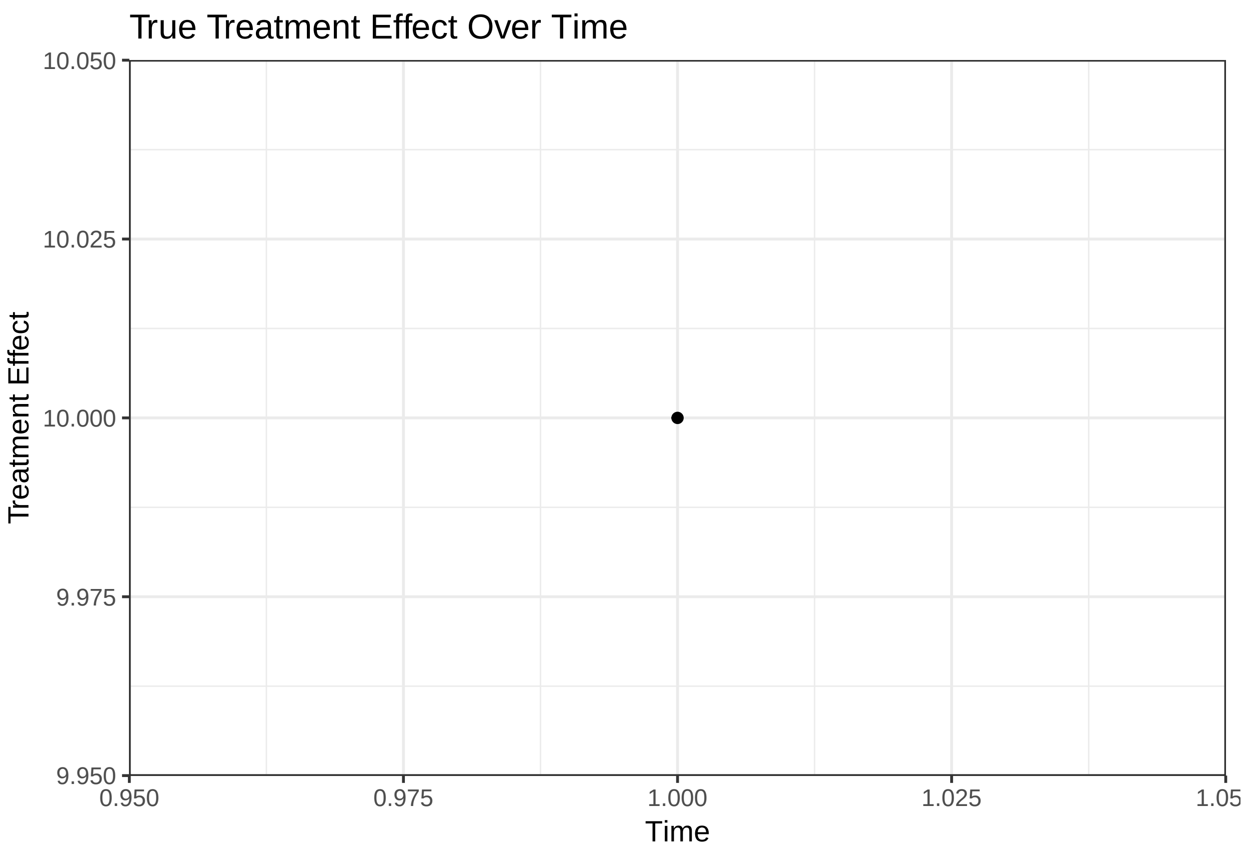
### Influence of $N_0/N$ on Performance

### Influence of $\rho$ on Performance

### Influence of $\tau$ on Performance

### Influence True Rank on Performance

### Influence of Rank Error on Performance




```
quiz_1_tues <- c(12,12.34,10,12,12,12.34,10,12,12,9,13,6,12,9.66,9,13,12,12,11.34,11.34,13,13,12.34,13,
            13,13,6.32,6.32,2,11.68,12,11.66,9.66,12.34,12,2.02,13,11,12,12.34,10.32,12,13)

quiz_1_thurs <- c(11,13,12,13,13,13,0,10.64,
      13,12,5.02,11.34,12,13,8,12,13,13,13,13,10,10.02,12,10.34,13,11.34,13,9.68,
            8,12,3.66,12,11.32,10.32,9.98,12.34,13,12.34,12.34,13,11,9.98,12.34,
      10.34,4,12,12)
```