

Federated Learning with Incomplete Data: A Weighted Approach with Variance Correction

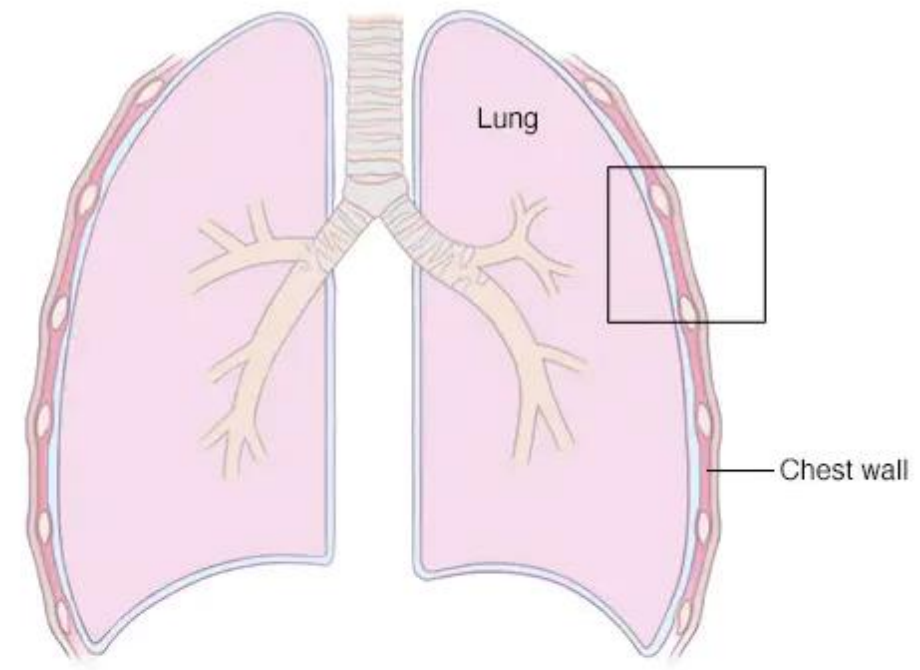
CIDI T32 Meeting
March 10, 2026

Jesus E. Vazquez, PhD | JHU Provost Postdoctoral Fellow

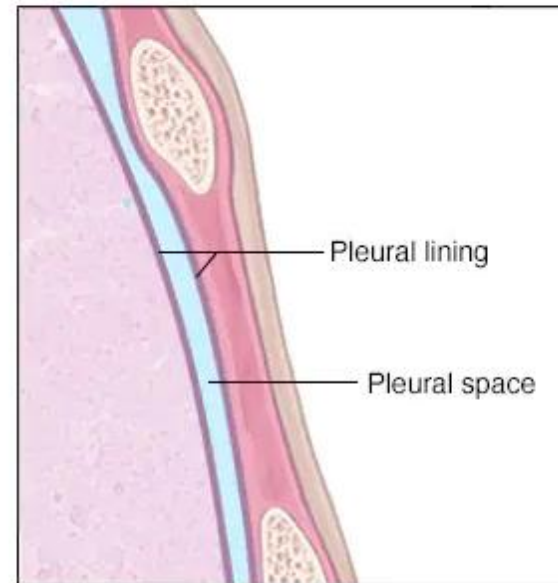


Pleural infections

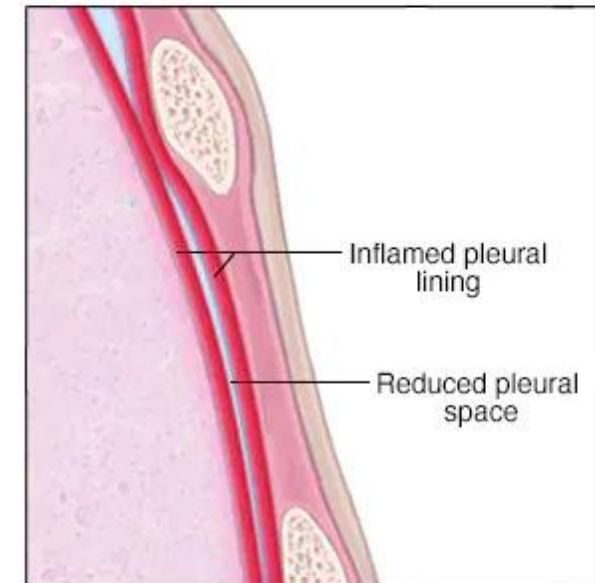
- **Remains a serious clinical problem**, with substantial morbidity and mortality despite modern antibiotics and drainage strategies¹.



Typical pleura

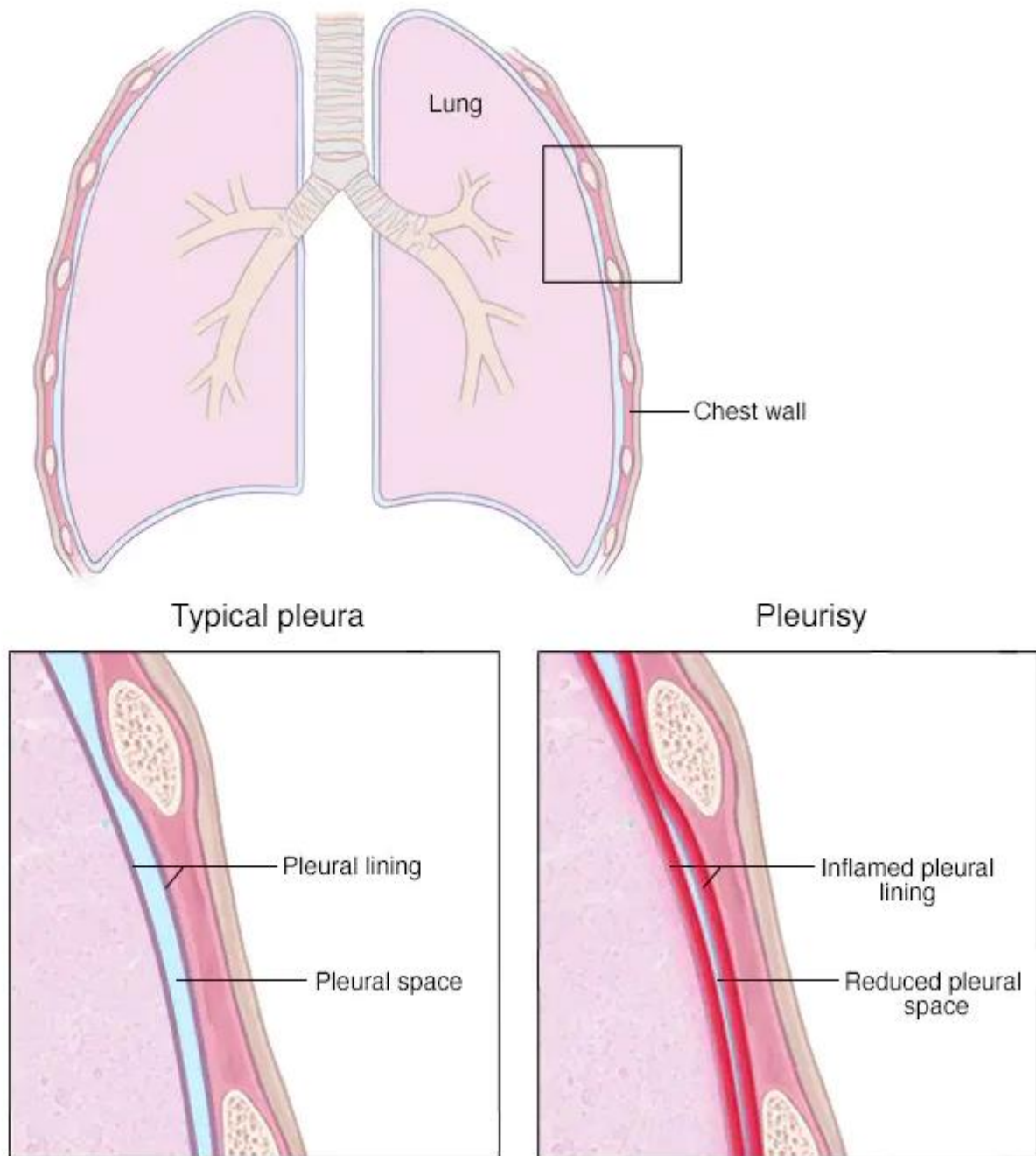


Pleurisy



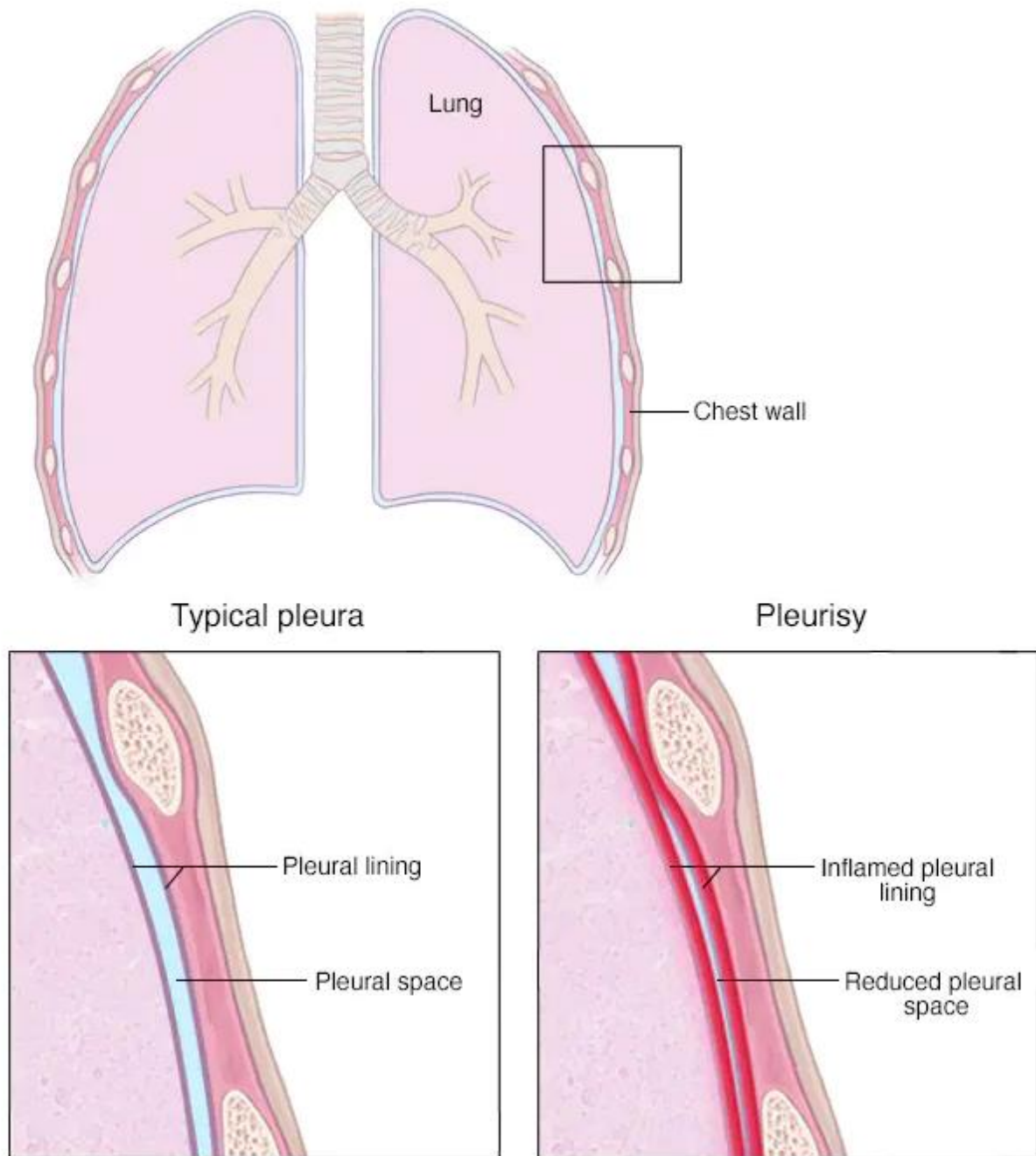
Pleural infections

- **Remains a serious clinical problem**, with substantial morbidity and mortality despite modern antibiotics and drainage strategies¹.
- **Intrapleural enzyme therapy (IET)** using tissue plasminogen activator and DNase has improved outcomes and reduced the need for surgery, **but treatment response remains highly variable across patients**.



Pleural infections

- **Remains a serious clinical problem**, with substantial morbidity and mortality despite modern antibiotics and drainage strategies¹.
- **Intrapleural enzyme therapy (IET)** using tissue plasminogen activator and DNase has improved outcomes and reduced the need for surgery, **but treatment response remains highly variable across patients**.
- **Understanding which patients are at highest risk of poor outcomes is important.**



Data

Two sources: Hopkins (5 Hospitals) and UNC (23 Centers, USA and UK)

	Johns Hopkins (n=454)	UNC (n=1,494)
90-day mortality		
Alive	371 (81.7%)	1279 (85.6%)
Dead	83 (18.3%)	215 (14.4%)
Age $\geq$ 50 years		
No	110 (24.2%)	448 (30.0%)
Yes	344 (75.8%)	1046 (70.0%)
BUN $\geq$ 14		
No	274 (60.4%)	994 (66.5%)
Yes	180 (39.6%)	500 (33.5%)
Albumin $<$ 2.7		
No	191 (43.0%)	635 (50.1%)
Yes	253 (56.9%)	631 (49.9%)
Missing	10 (2.2%)	228 (15.3%)

Data

Two sources: Hopkins (5 Hospitals) and UNC (23 Centers, USA and UK)

Covariate Missingness:
Albumin, more pronounced in the UNC Study

	Johns Hopkins (n=454)	UNC (n=1,494)
90-day mortality		
Alive	371 (81.7%)	1279 (85.6%)
Dead	83 (18.3%)	215 (14.4%)
Age ≥ 50 years		
No	110 (24.2%)	448 (30.0%)
Yes	344 (75.8%)	1046 (70.0%)
BUN ≥ 14		
No	274 (60.4%)	994 (66.5%)
Yes	180 (39.6%)	500 (33.5%)
Albumin < 2.7		
No	191 (43.0%)	635 (50.1%)
Yes	253 (56.9%)	631 (49.9%)
Missing	10 (2.2%)	228 (15.3%)

Data

Two sources: Hopkins (5 Hospitals) and UNC (23 Centers, USA and UK)

Covariate Missingness:
Albumin, more pronounced in the UNC Study

Q: How can I combine data sources and adjust for the missing data?

	Johns Hopkins (n=454)	UNC (n=1,494)
90-day mortality		
Alive	371 (81.7%)	1279 (85.6%)
Dead	83 (18.3%)	215 (14.4%)
Age ≥ 50 years		
No	110 (24.2%)	448 (30.0%)
Yes	344 (75.8%)	1046 (70.0%)
BUN ≥ 14		
No	274 (60.4%)	994 (66.5%)
Yes	180 (39.6%)	500 (33.5%)
Albumin < 2.7		
No	191 (43.0%)	635 (50.1%)
Yes	253 (56.9%)	631 (49.9%)
Missing	10 (2.2%)	228 (15.3%)

Distributed research networks (DRN)

How is the data stored?

- **Centralized:** stored at a single site.
- **Decentralized:** data stored across sites.



Decentralized DRN

- Federated learning enables estimation of regression and causal parameters when it is not possible to pool or “stack” patient-level data².

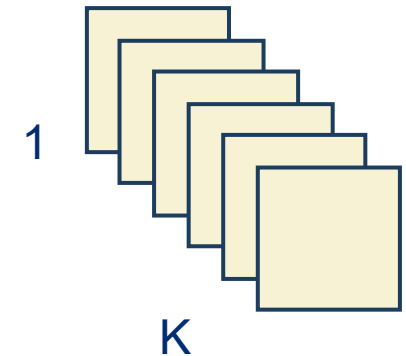
Distributed research networks (DRN)

How is the data stored?

- **Centralized:** stored at a single site.
- **Decentralized:** data stored across sites.

Decentralized DRN

- Federated learning enables estimation of regression and causal parameters when it is not possible to pool or “stack” patient-level data².



The three constraints in federated learning



Privacy

Only summary statistics or
limited information may
leave a site

The three constraints in federated learning



Privacy

Only summary statistics or limited information may leave a site



Communication efficiency

Ideally only few rounds of communication

The three constraints in federated learning



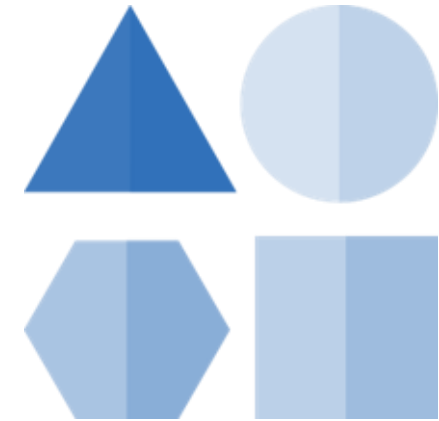
Privacy

Only summary statistics or limited information may leave a site



Communication efficiency

Ideally only few rounds of communication



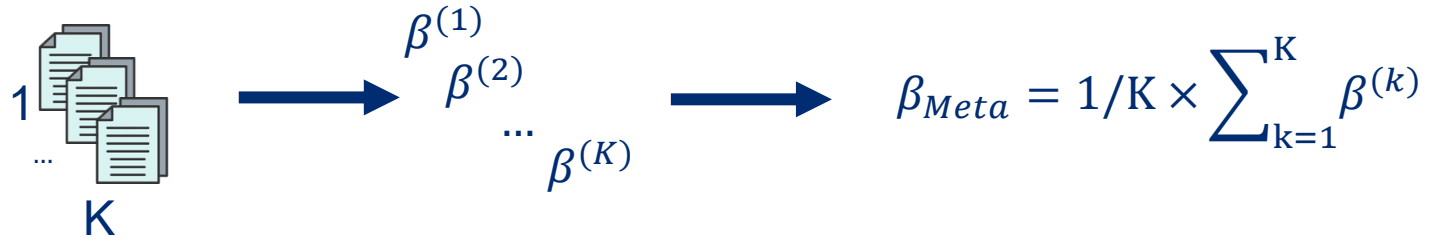
Heterogeneity

Site-specific distributions and effects

Different Approaches



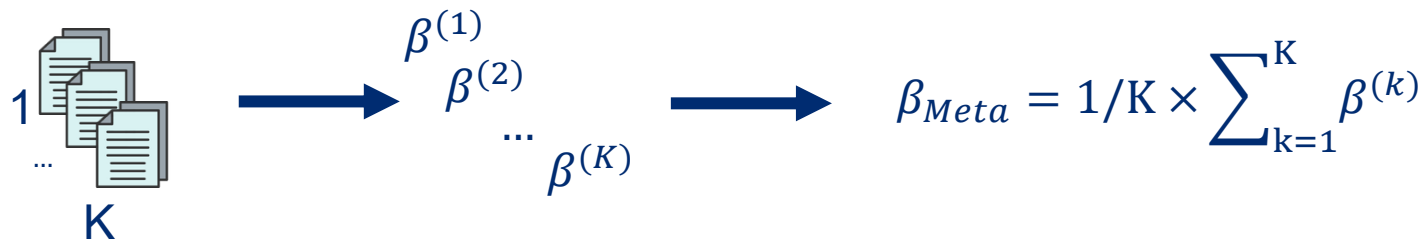
Meta-analysis



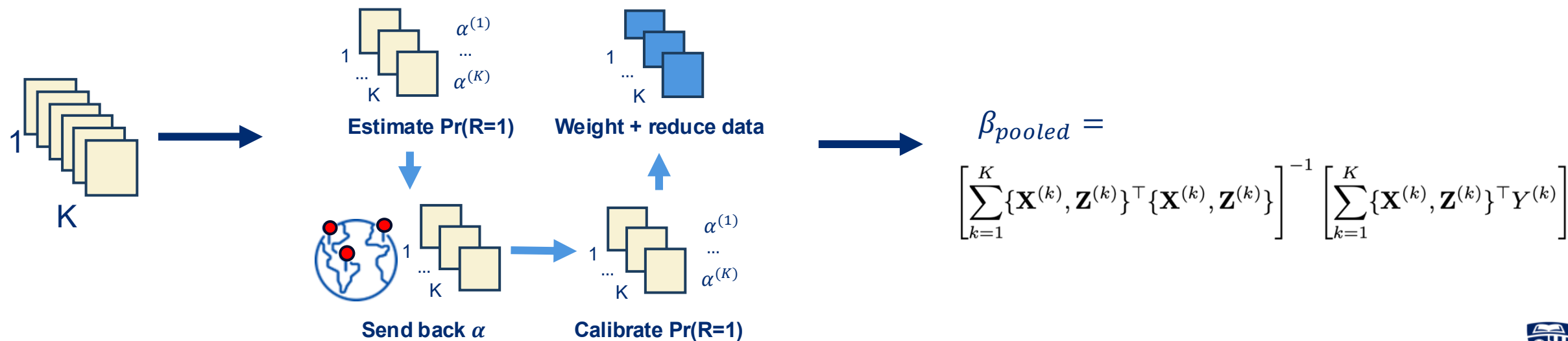
Different Approaches



Meta-analysis



Federated Learning



Privacy: What information can I share?

It depends!

- No individual-level data
- Aggregated statistics only
- Data user agreement governs sharing



Centers for Medicare & Medicaid Services (CMM)

- Operates under the US Department of Health and Human Services
- Small cell counts (≤ 10) are suppressed in accordance with CMS disclosure policies to reduce re-identification risk.

Examples: Small Cell Count Suppression

Table 1a. Age Distribution of Patients with Procedure A

Beneficiary age group	Procedure A (N=2,690), N (%)
60-69 years	1,900 (71)
70-74 years	400 (15)
75-79 years	290 (11)
80-84 years	94 (3.5)
85+ years	6 (0.22)



Examples: Small Cell Count Suppression

Ben

60-69

70-74

75-79

80-84

85+

Table 1b. Age Distribution of Patients with Procedure A

Beneficiary age group

Procedure A (N=2,690), N (%)

60-69 years

1,900 (71)

70-74 years

400 (15)

75-79 years

290 (11)

80+ years

100 (3.7)



Examples: Small Cell Count Suppression

Table 1c. Age distribution of patients who had procedure A

Beneficiary age group	Procedure A (N=2,690), N (%)
60-69 years	>1895 (>70.4)
70-74 years	400 (15)
75-79 years	290 (11)
80-84 years	94 (3.5)
85+ years	<11 (<0.4)

Examples: Small Cell Count Suppression

Table 1d. Age Distribution of Patients with Procedure A	
Beneficiary age group	Procedure A (N=2,690), N (%)
60-69 years	1,900 (71)
70-74 years	400 (15)
75-79 years	290 (11)
80-84 years	94 (3.5)
85+ years	**

What is the statistical challenge?

Previous research is limited or inconclusive

- Collaborative one-shot and lossless algorithms of generalized linear models **(COLA-GLM)** estimator assumes complete data².
- Most existing approaches in FL ignore missing data³ or are either *single- or multiple-imputation* based and generally assume missing at random (MAR)⁴.



What is the statistical challenge?

Previous research is limited or inconclusive

- Collaborative one-shot and lossless algorithms of generalized linear models (**COLA-GLM**) estimator assumes complete data².
- Most existing approaches in FL ignore missing data³ or are either *single- or multiple-imputation* based and generally assume missing at random (MAR)⁴.

Opportunity: Develop an inverse probability COLA-GLM for various types of missingness.

Impact: Provide statisticians with an additional estimator that allows for the consistent estimation of regression parameters.

<https://doi.org/10.1038/s41746-025-01781-1>

COLA-GLM: collaborative one-shot and lossless algorithms of generalized linear models for decentralized observational healthcare data



Check for updates

Qiong Wu^{1,2,3}, Jenna M. Repts^{4,5,6}, Lu Li^{3,7}, Bingyu Zhang^{3,7}, Yiwen Lu^{3,7}, Jiayi Tong^{2,3,8}, Dazheng Zhang^{2,3}, Thomas Lumley⁹, Milou T. Brand¹⁰, Mui Van Zandt^{4,10}, Thomas Falconer¹¹, Xing He^{12,13}, Yu Huang^{12,13}, Haoyang Li¹⁴, Chao Yan¹⁵, Guojun Tang¹⁶, Andrew E. Williams^{17,18}, Fei Wang¹⁴, Jiang Bian^{12,13}, Bradley Malin^{15,19,20}, George Hripcsak¹¹, Martijn J. Schuemie^{4,5,21}, Yun Lu²², Steve Drew¹⁶, Jiayu Zhou²³, David A. Asch^{24,25} & Yong Chen^{2,3,24,26,27}



Intro to COLA-GLM



Estimand vs. Estimator vs. Estimate

θ

Estimand

Estimand vs. Estimator vs. Estimate

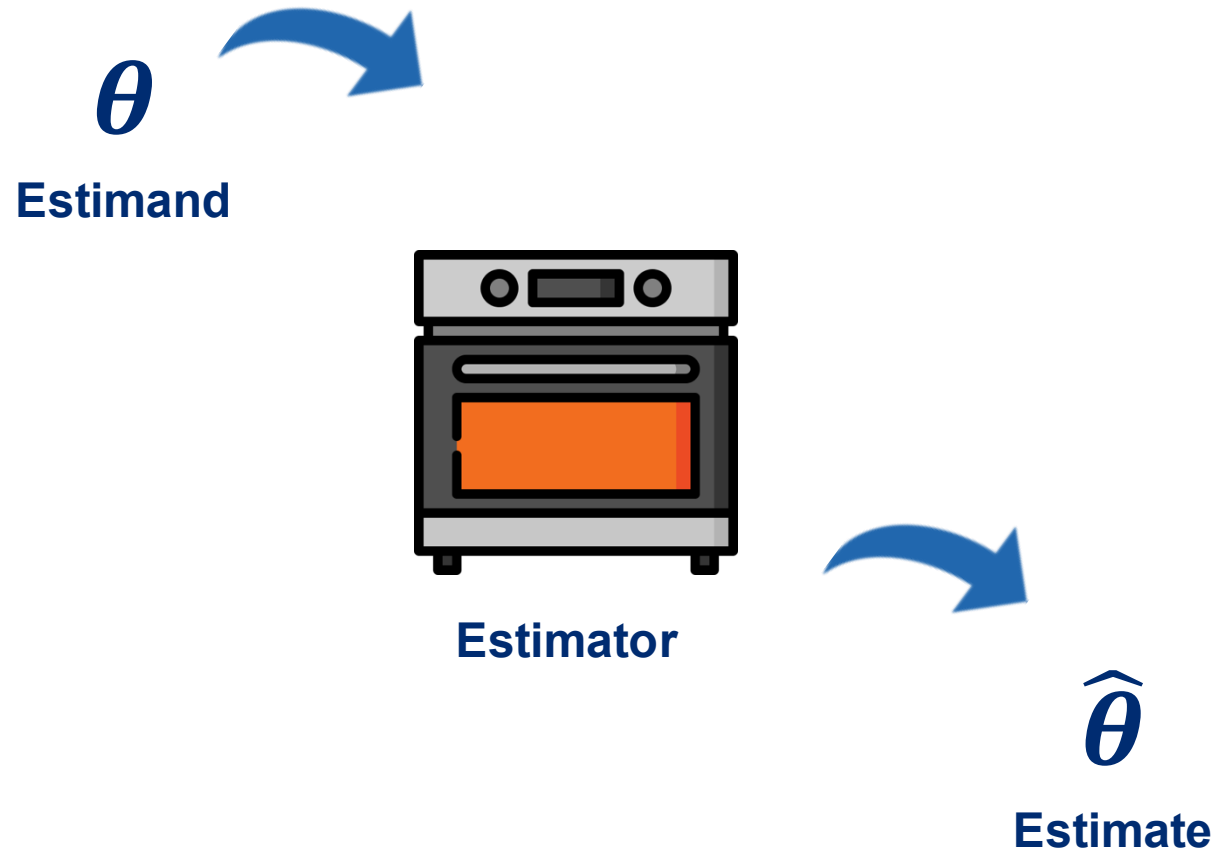
θ

Estimand



Estimator

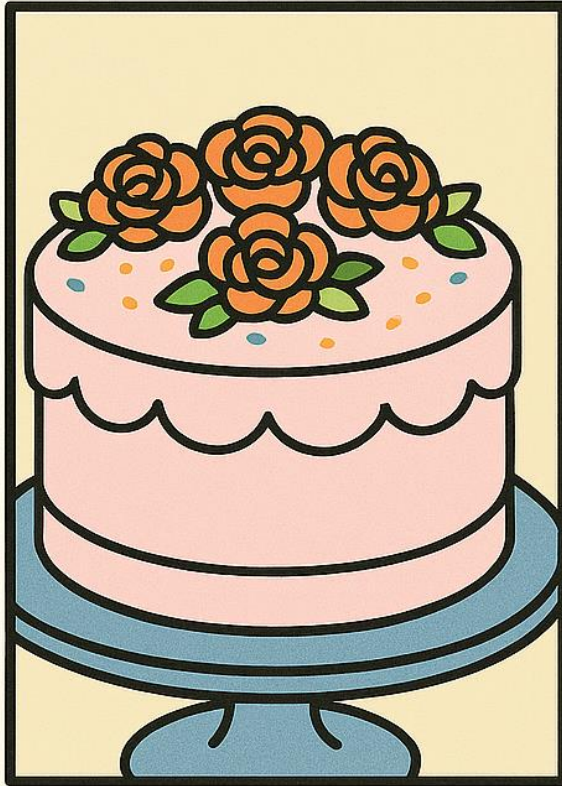
Estimand vs. Estimator vs. Estimate



Estimand vs. Estimator vs. Estimate

Estir

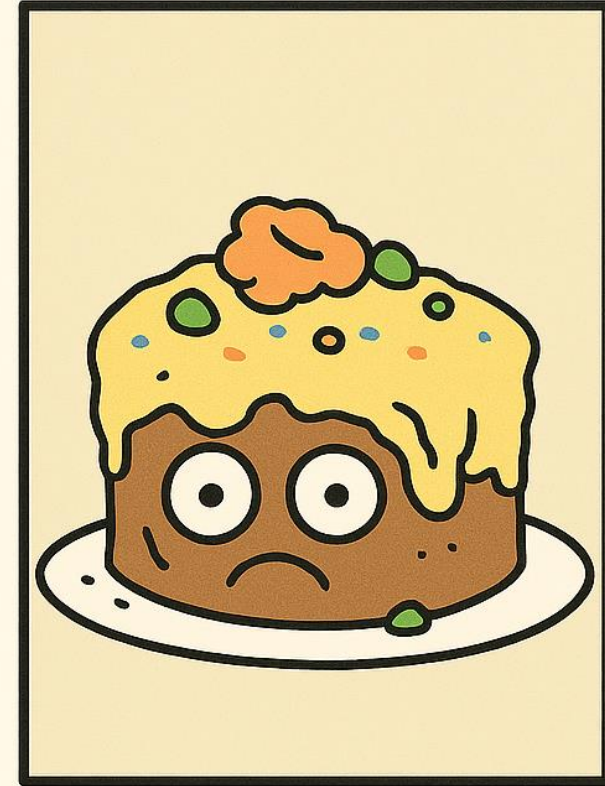
Estimand



Estimator



Estimate



<https://www.r-causal.org/chapters/10-estimands>

COLA-GLM | Example dataset

Extract number of combinations

- Outcome+covariates: 2 levels each
- Total number of combinations: $2^4=16$

id	Y	X1	X2	X3
1	1	1	1	1
2	0	0	1	1
3	1	0	0	2
4	1	1	0	0
		...		
998	1	1	0	1
999	0	0	1	0
1000	0	1	1	0

COLA-GLM | Example dataset

Extract number of combinations

- Outcome+covariates: 2 levels each
- Total number of combinations: $2^4=16$

id	Y	X1	X2	X3
1	1	1	1	1
2	0	0	1	1
3	1	0	0	2
4	1	1	0	0
		...		
998	1	1	0	1
999	0	0	1	0
1000	0	1	1	0

Generate summary statistics table

- 16 rows, representing each combination

id	Y	X1	X2	X3	n
1	1	1	1	1	11
2	1	1	1	0	40
3	1	1	0	0	100
4	1	0	0	0	200
		...			
14	0	1	1	0	150
15	0	0	1	0	45
16	0	0	0	0	12



COLA-GLM | Simulation Example (n=1,000)

Simulate Data

```
> dat = mydata(K=1)
> dat %>%
+   dplyr::select(Y, X1, X2, X3)
  Y X1 X2 X3
1  1  1  1  0
2  1  0  1  1
3  1  0  1  0
4  1  1  0  0
5  1  0  0  1
```

Create Summary Table

```
> dat %>%
+   dplyr::select(Y, X1, X2, X3) %>%
+   group_by(Y,X1,X2,X3) %>%
+   summarise(count = n())
```

```
# A tibble: 16 × 5
# Groups:   Y, X1, X2 [8]
   Y     X1     X2     X3 count
  <int> <int> <int> <int> <int>
1     0     0     0     0    28
2     0     0     0     1    15
3     0     0     1     0    17
4     0     0     1     1     6
5     0     1     0     0    20
6     0     1     0     1     4
7     0     1     1     0     8
8     0     1     1     1     1
9     1     0     0     0    81
10    1     0     0     1   119
11    1     0     1     0    98
12    1     0     1     1   147
13    1     1     0     0   114
14    1     1     0     1   115
15    1     1     1     0   116
16    1     1     1     1   111
```



COLA-GLM | Simulation Example (n=1,000)

Simulate Data

```
> dat = mydata(K=1)
> dat %>%
+   dplyr::select(Y, X1, X2, X3)
  Y X1 X2 X3
1  1  1  1  0
2  1  0  1  1
3  1  0  1  0
4  1  1  0  0
5  1  0  0  1
```

Create Summary Table

```
> dat %>%
+   dplyr::select(Y, X1, X2, X3) %>%
+   group_by(Y,X1,X2,X3) %>%
+   summarise(count = n())
```

```
# A tibble: 16 × 5
# Groups:   Y, X1, X2 [8]
   Y     X1     X2     X3 count
  <int> <int> <int> <int> <int>
1     0     0     0     0    28
2     0     0     0     1    15
3     0     0     1     0    17
4     0     0     1     1     6
5     0     1     0     0    20
6     0     1     0     1     4
7     0     1     1     0     8
8     0     1     1     1     1
9     1     0     0     0    81
10    1     0     0     1   119
11    1     0     1     0    98
12    1     0     1     1   147
13    1     1     0     0   114
14    1     1     0     1   115
15    1     1     1     0   116
16    1     1     1     1   111
```

Based on regression model

$$\text{logit}\{\Pr(Y = 1)\} = \beta_0 + \beta_1 X1 + \beta_2 X2 + \beta_3 X3$$

$$(\beta_0, \beta_1, \beta_2, \beta_3) = (1, 1, 1, 1)$$

$$\text{logit}\{\Pr(R_X = 1)\} \sim Y + X2 + X3$$



COLA-GLM | Simulation Example (n=1,000)

Simulate Data

```
> dat = mydata(K=1)
> dat %>%
+   dplyr::select(Y, X1, X2, X3)
  Y X1 X2 X3
1  1  1  1  0
2  1  0  1  1
3  1  0  1  0
4  1  1  0  0
5  1  0  0  1
```

Create Summary Table

```
> dat %>%
+   dplyr::select(Y, X1, X2, X3) %>%
+   group_by(Y,X1,X2,X3) %>%
+   summarise(count = n())
```

```
# A tibble: 16 × 5
# Groups:   Y, X1, X2 [8]
   Y     X1     X2     X3 count
<int> <int> <int> <int> <int>
1     0     0     0     0    28
2     0     0     0     1    15
3     0     0     1     0    17
4     0     0     1     1     6
5     0     1     0     0    20
6     0     1     0     1     4
7     0     1     1     0     8
8     0     1     1     1     1
9     1     0     0     0    81
10    1     0     0     1   119
11    1     0     1     0    98
12    1     0     1     1   147
13    1     1     0     0   114
14    1     1     0     1   115
15    1     1     1     0   116
16    1     1     1     1   111
```

Based on regression model

$$\text{logit}\{\Pr(Y = 1)\} = \beta_0 + \beta_1 X1 + \beta_2 X2 + \beta_3 X3$$

$$(\beta_0, \beta_1, \beta_2, \beta_3) = (1, 1, 1, 1)$$

$$\text{logit}\{\Pr(R_X = 1)\} \sim Y + X2 + X3$$



COLA-GLM | Simulation Example (n=1,000)

Pooled Data

```
```{R}
dat = mydata(K=1)
dat %>% dplyr::select(Y, X1, X2, X3)
glm(Y ~ X1+X2+X3, data = dat, family = "binomial") %>% summary()
```
```

Call:

```
glm(formula = Y ~ X1 + X2 + X3, family = "binomial", data = dat)
```

Coefficients:

| | Estimate | Std. Error | z value | Pr(> z) | |
|-------------|----------|------------|---------|----------|-----|
| (Intercept) | 0.8509 | 0.1736 | 4.902 | 9.51e-07 | *** |
| X1 | 0.8001 | 0.2233 | 3.584 | 0.000338 | *** |
| X2 | 1.3956 | 0.2467 | 5.658 | 1.53e-08 | *** |
| X3 | 1.0340 | 0.2322 | 4.453 | 8.46e-06 | *** |

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Aggregated Data

COLA-GLM | Simulation Example (n=1,000)

Pooled Data

```
```{R}
dat = mydata(K=1)
dat %>% dplyr::select(Y, X1, X2, X3)
glm(Y ~ X1+X2+X3, data = dat, family = "binomial") %>% summary()
```
```

Call:

```
glm(formula = Y ~ X1 + X2 + X3, family = "binomial", data = dat)
```

Coefficients:

| | Estimate | Std. Error | z value | Pr(> z) | |
|-------------|----------|------------|---------|--------------|--|
| (Intercept) | 0.8509 | 0.1736 | 4.902 | 9.51e-07 *** | |
| X1 | 0.8001 | 0.2233 | 3.584 | 0.000338 *** | |
| X2 | 1.3956 | 0.2467 | 5.658 | 1.53e-08 *** | |
| X3 | 1.0340 | 0.2322 | 4.453 | 8.46e-06 *** | |

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Aggregated Data

```
```{R}
datCola = dat %>%
 dplyr::select(Y, X1, X2, X3) %>%
 group_by(Y,X1,X2,X3) %>%
 summarise(count = n())

glm(Y ~ X1+X2+X3, data = datCola, weights = count, family = "binomial") %>% summary()
```
```

Call:

```
glm(formula = Y ~ X1 + X2 + X3, family = "binomial", data = datCola,
     weights = count)
```

Coefficients:

| | Estimate | Std. Error | z value | Pr(> z) | |
|-------------|----------|------------|---------|--------------|--|
| (Intercept) | 0.8509 | 0.1736 | 4.902 | 9.51e-07 *** | |
| X1 | 0.8001 | 0.2232 | 3.584 | 0.000338 *** | |
| X2 | 1.3956 | 0.2466 | 5.658 | 1.53e-08 *** | |
| X3 | 1.0340 | 0.2322 | 4.453 | 8.46e-06 *** | |

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Can replicate the COLA-GLM based on the aggregated and weighting by the count!

COLA-GLM | Simulation Example (n=1,000)

Pooled Data

```
```{R}
dat = mydata(K=1)
dat %>% dplyr::select(Y, X1, X2, X3)
glm(Y ~ X1+X2+X3, data = dat, family = "binomial") %>% summary()
```
```

Call:

```
glm(formula = Y ~ X1 + X2 + X3, family = "binomial", data = dat)
```

Coefficients:

| | Estimate | Std. Error | z value | Pr(> z) | |
|-------------|----------|------------|---------|--------------|--|
| (Intercept) | 0.8509 | 0.1736 | 4.902 | 9.51e-07 *** | |
| X1 | 0.8001 | 0.2233 | 3.584 | 0.000338 *** | |
| X2 | 1.3956 | 0.2467 | 5.658 | 1.53e-08 *** | |
| X3 | 1.0340 | 0.2322 | 4.453 | 8.46e-06 *** | |

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Aggregated Data

```
```{R}
datCola = dat %>%
 dplyr::select(Y, X1, X2, X3) %>%
 group_by(Y,X1,X2,X3) %>%
 summarise(count = n())

glm(Y ~ X1+X2+X3, data = datCola, weights = count, family = "binomial") %>% summary()
```
```

Call:

```
glm(formula = Y ~ X1 + X2 + X3, family = "binomial", data = datCola,
     weights = count)
```

Coefficients:

| | Estimate | Std. Error | z value | Pr(> z) | |
|-------------|----------|------------|---------|--------------|--|
| (Intercept) | 0.8509 | 0.1736 | 4.902 | 9.51e-07 *** | |
| X1 | 0.8001 | 0.2232 | 3.584 | 0.000338 *** | |
| X2 | 1.3956 | 0.2466 | 5.658 | 1.53e-08 *** | |
| X3 | 1.0340 | 0.2322 | 4.453 | 8.46e-06 *** | |

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

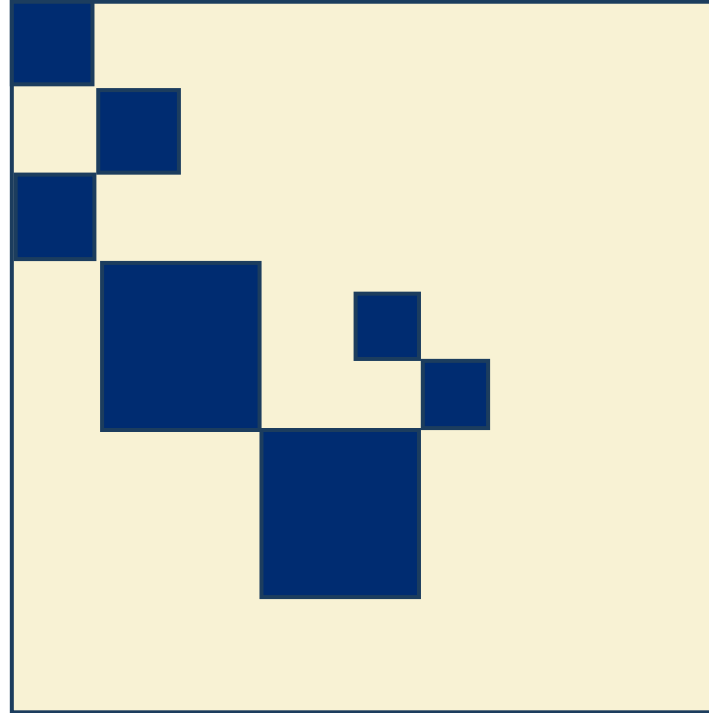
Can replicate the COLA-GLM based on the aggregated and weighting by the count!

COLA-GLM with missing data



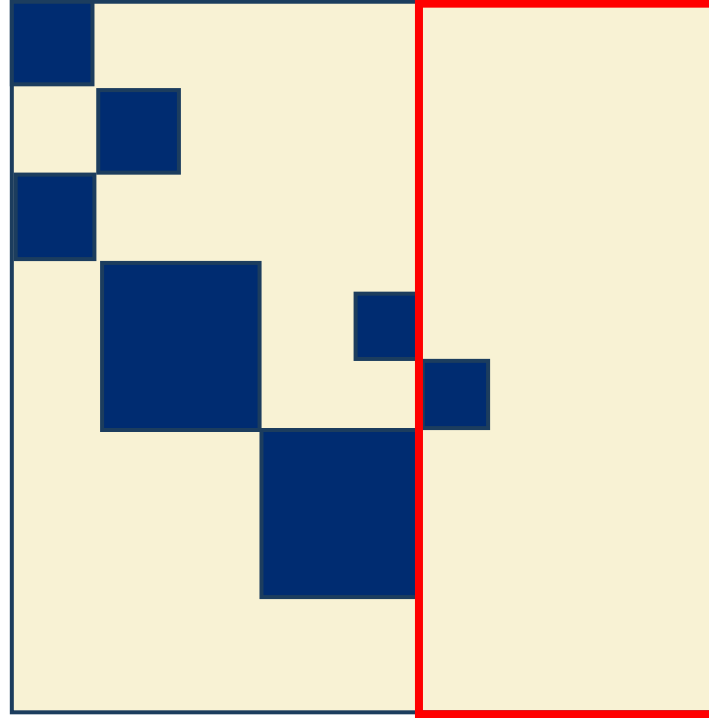
Complete case estimator

The easiest estimator to use:
one restricts the data to only the
observations that have complete
data before applying the model
of interest may only require one
communication between sites.



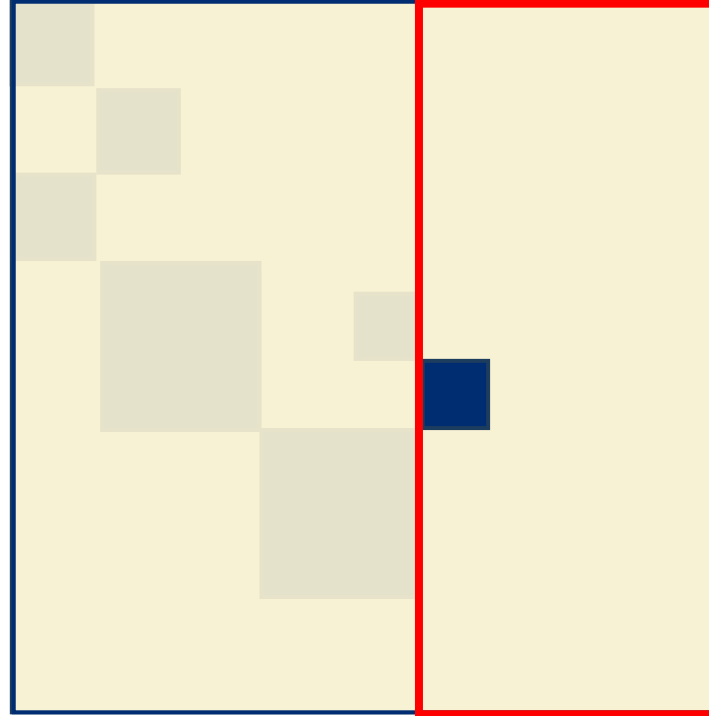
Complete case estimator

The easiest estimator to use:
one restricts the data to only the
observations that have complete
data before applying the model
of interest may only require one
communication between sites.



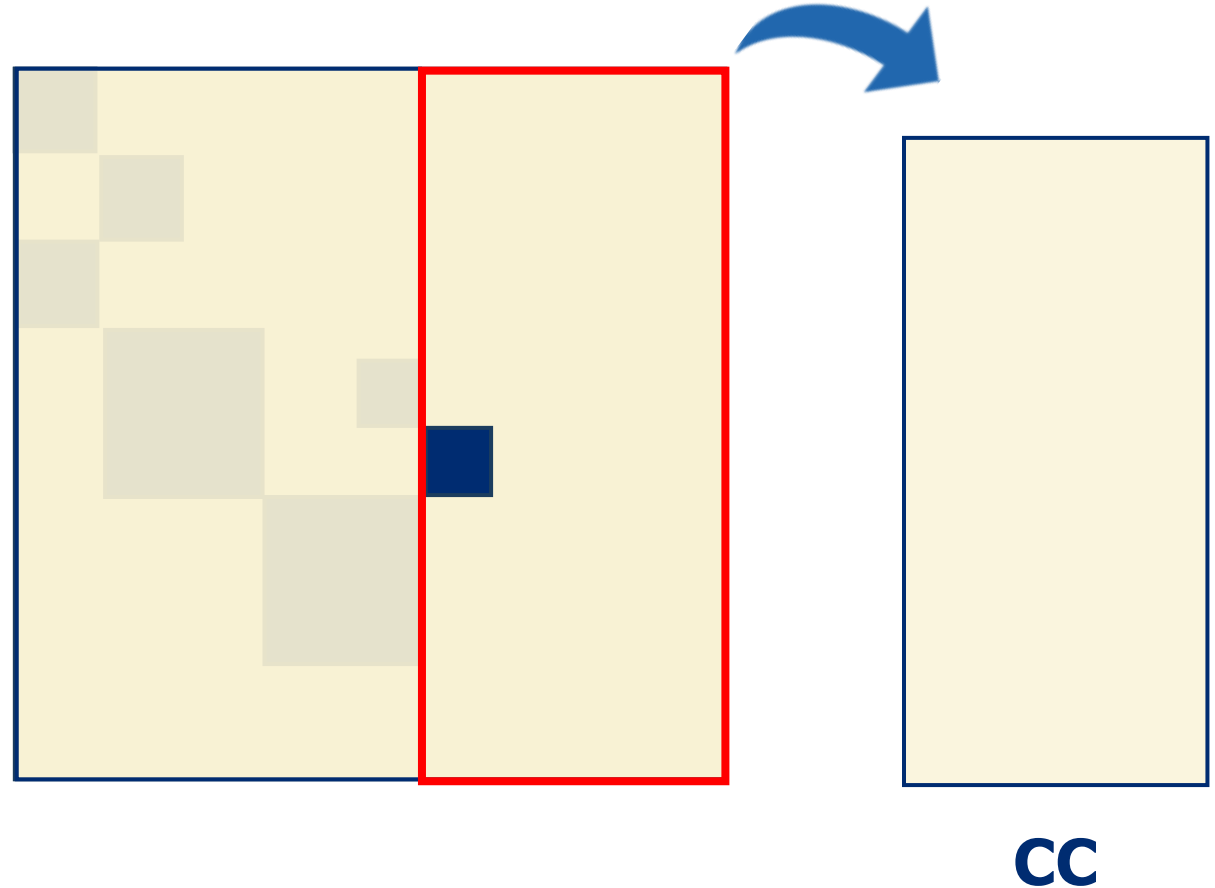
Complete case estimator

The easiest estimator to use:
one restricts the data to only the
observations that have complete
data before applying the model
of interest may only require one
communication between sites.



Complete case estimator

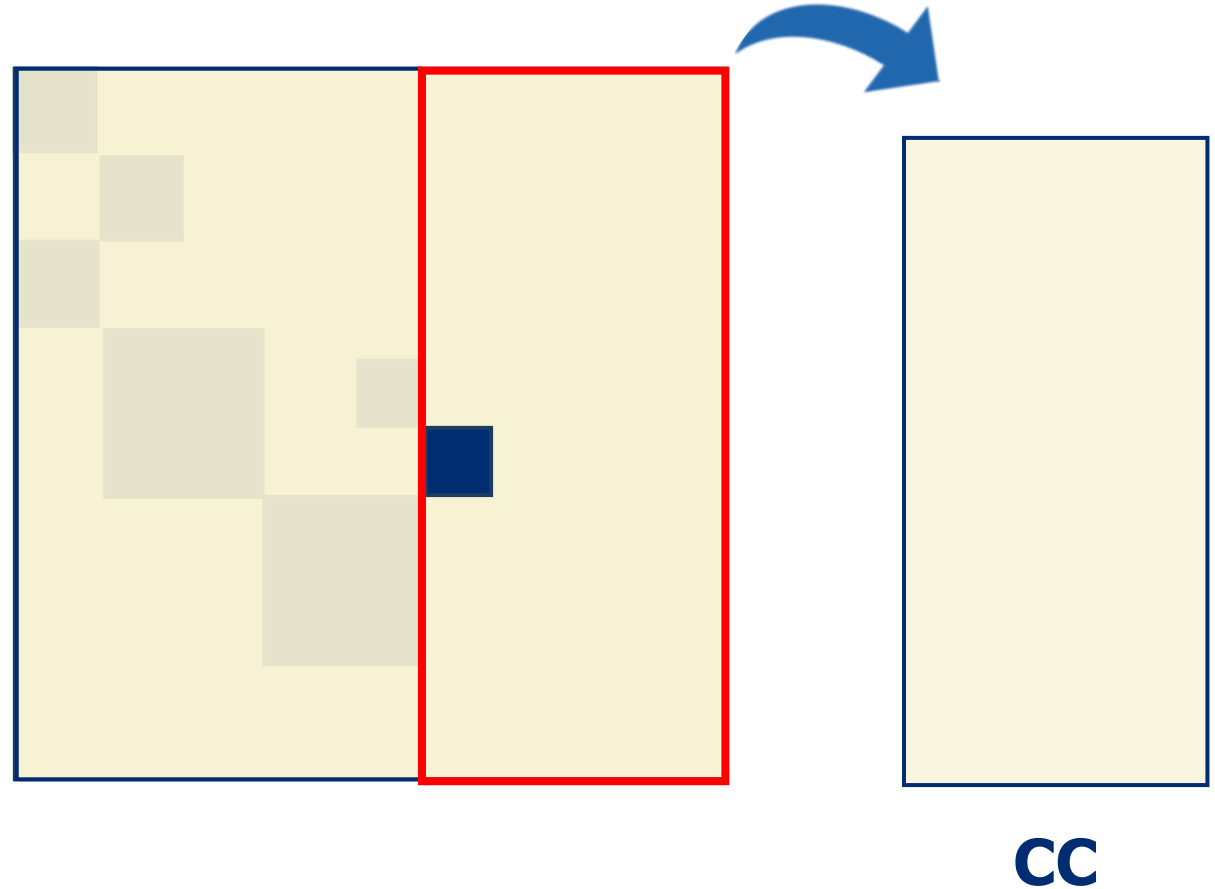
The easiest estimator to use:
one restricts the data to only the
observations that have complete
data before applying the model
of interest may only require one
communication between sites.



Complete case estimator

The easiest estimator to use:
one restricts the data to only the
observations that have complete
data before applying the model
of interest may only require one
communication between sites.

**Bias depends on the missingness
mechanism and % of missing
observations**



Complete Case COLA-GLM

Step 1: Reduce data to complete data only

Step 2: Create summary statistics table



Complete Case COLA-GLM

Step 1: Reduce data to complete data only

Step 2: Create summary statistics table

Step 3: Run COLA-GLM regularly

Note that if we sum the count variable, we will get the total number of complete observations (n=616)

```
```{R}
datCola = dat %>%
 dplyr::filter(R_MAR == 1) %>%
 dplyr::select(Y, X1, X2, X3) %>%
 group_by(Y, X1, X2, X3) %>%
 summarise(count = n())

glm(Y ~ X1+X2+X3, data = datCola, weights = count,
 family = "binomial") %>% summary()
```
```

Call:

```
glm(formula = Y ~ X1 + X2 + X3, family = "binomial", data = datCola,
    weights = count)
```

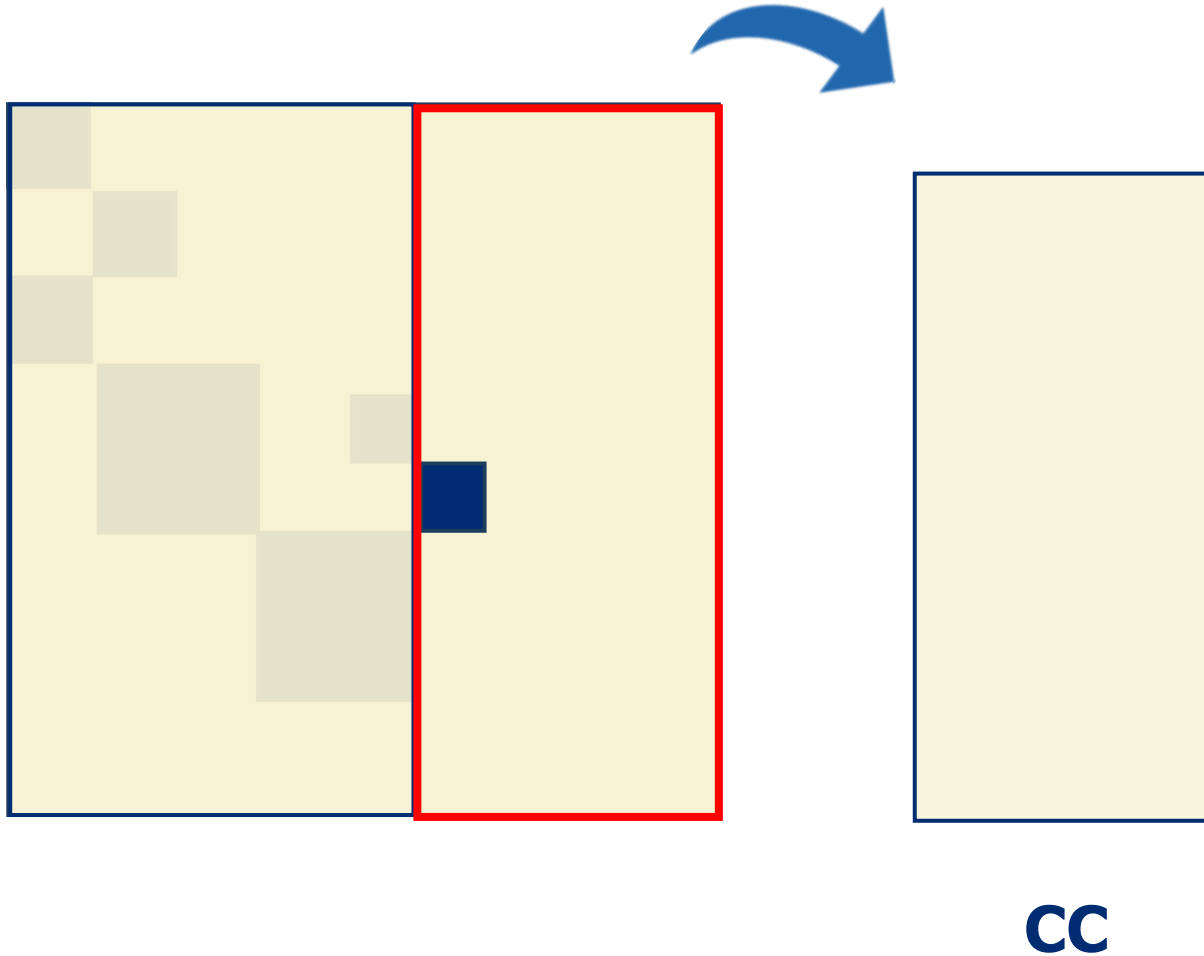
Coefficients:

| | Estimate | Std. Error | z value | Pr(> z) | |
|-------------|----------|------------|---------|----------|-----|
| (Intercept) | 0.9667 | 0.2336 | 4.139 | 3.49e-05 | *** |
| X1 | 0.5786 | 0.2818 | 2.054 | 0.040022 | * |
| X2 | 1.2675 | 0.3140 | 4.036 | 5.43e-05 | *** |
| X3 | 1.1197 | 0.2985 | 3.751 | 0.000176 | *** |

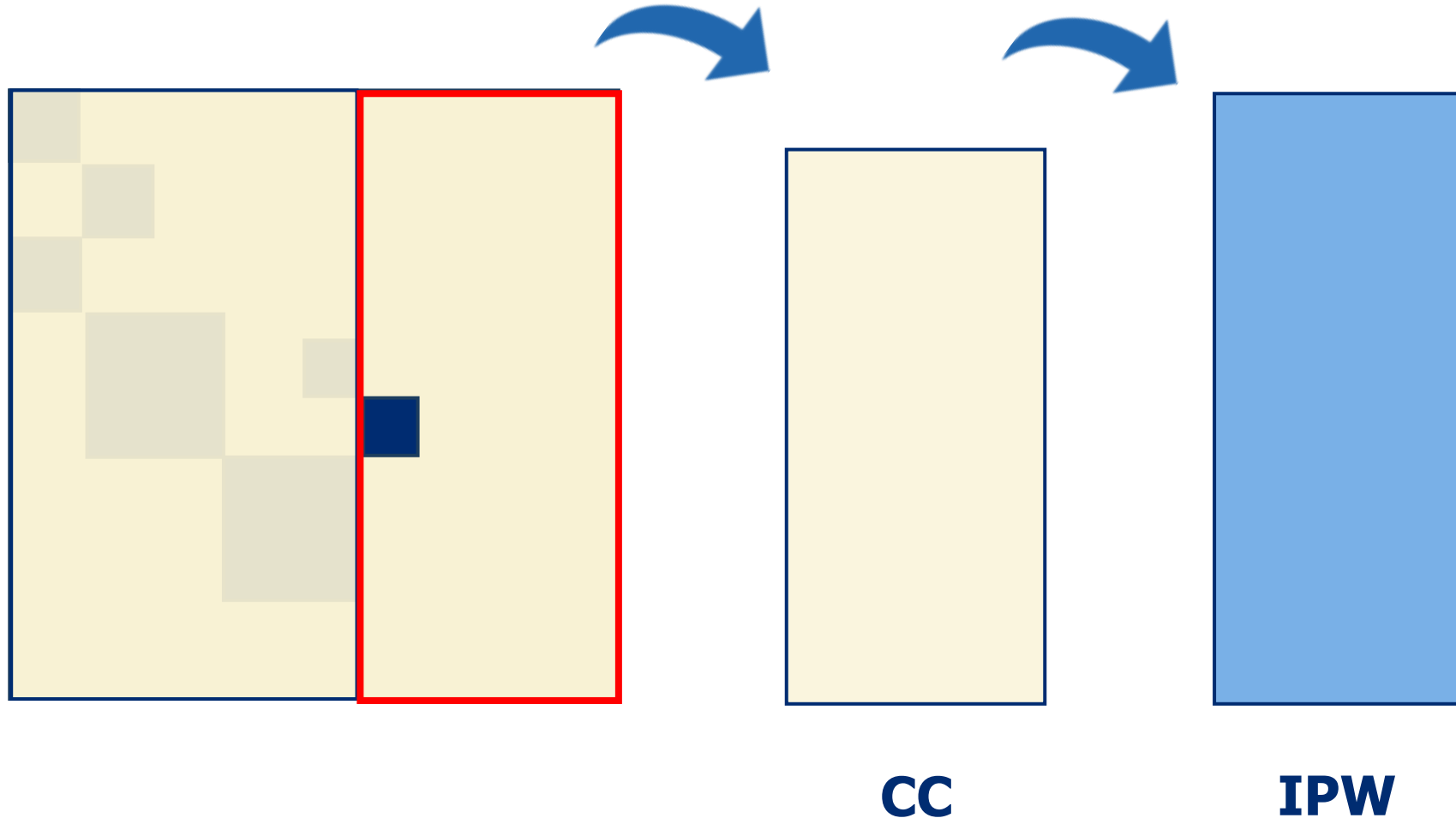
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1



Inverse probability weighting estimator



Inverse probability weighting estimator



Strategy: Weight observations to create a pseudo-sample that approximates the original full data.

Weighted COLA-GLM

Step 0: Calculate Pr of complete observation

Step 1: Reduce data to complete data only

Step 2: Create summary statistics table

```
> datCola = dat %>%
+   dplyr::filter(R_MAR == 1) %>%
+   dplyr::select(Y, X1, X2, X3, myp) %>%
+   group_by(Y,X1,X2,X3) %>%
+   summarise(count = n(),
+             myp = unique(myp))
`summarise()` has grouped output by 'Y',
> datCola
# A tibble: 16 x 6
# Groups:   Y, X1, X2 [8]
      Y     X1     X2     X3 count  myp
  <int> <int> <int> <int> <int> <dbl>
1     0     0     0     0    18 0.550
2     0     0     0     1     8 0.574
3     0     0     1     0     6 0.574
4     0     0     1     1     3 0.599
5     0     1     0     0    14 0.550
6     0     1     0     1     6 0.574
7     0     1     1     0     5 0.574
8     0     1     1     1     1 0.599
9     1     0     0     0    47 0.599
10    1     0     0     1    63 0.622
11    1     0     1     0    61 0.622
12    1     0     1     1    79 0.646
13    1     1     0     0    69 0.599
14    1     1     0     1    81 0.622
15    1     1     1     0    66 0.622
16    1     1     1     1    89 0.646
```



Weighted COLA-GLM

Step 0: Calculate Pr of complete observation

Step 1: Reduce data to complete data only

Step 2: Create summary statistics table

```
> datCola = dat %>%  
+   dplyr::filter(R_MAR == 1) %>%  
+   dplyr::select(Y, X1, X2, X3, myp) %>%  
+   group_by(Y, X1, X2, X3) %>%  
+   summarise(count = n(),  
+             myp = unique(myp))  
`summarise()` has grouped output by 'Y',  
> datCola  
# A tibble: 16 × 6  
# Groups:   Y, X1, X2 [8]
```

| | Y | X1 | X2 | X3 | count | myp |
|----|-------|-------|-------|-------|-------|-------|
| | <int> | <int> | <int> | <int> | <int> | <dbl> |
| 1 | 0 | 0 | 0 | 0 | 18 | 0.550 |
| 2 | 0 | 0 | 0 | 1 | 8 | 0.574 |
| 3 | 0 | 0 | 1 | 0 | 6 | 0.574 |
| 4 | 0 | 0 | 1 | 1 | 3 | 0.599 |
| 5 | 0 | 1 | 0 | 0 | 14 | 0.550 |
| 6 | 0 | 1 | 0 | 1 | 6 | 0.574 |
| 7 | 0 | 1 | 1 | 0 | 5 | 0.574 |
| 8 | 0 | 1 | 1 | 1 | 1 | 0.599 |
| 9 | 1 | 0 | 0 | 0 | 47 | 0.599 |
| 10 | 1 | 0 | 0 | 1 | 63 | 0.622 |
| 11 | 1 | 0 | 1 | 0 | 61 | 0.622 |
| 12 | 1 | 0 | 1 | 1 | 79 | 0.646 |
| 13 | 1 | 1 | 0 | 0 | 69 | 0.599 |
| 14 | 1 | 1 | 0 | 1 | 81 | 0.622 |
| 15 | 1 | 1 | 1 | 0 | 66 | 0.622 |
| 16 | 1 | 1 | 1 | 1 | 89 | 0.646 |



Weighted COLA-GLM

Step 0: Calculate Pr of complete observation

Step 1: Reduce data to complete data only

Step 2: Create summary statistics table

Step 3: Run COLA-GLM regularly using count as the weight

Note that if we sum the count/my_p, we will get the total number of complete obs ($n \approx 1,000$)

```
```{R}
datCola = dat %>%
 dplyr::filter(R_MAR == 1) %>%
 dplyr::select(Y, X1, X2, X3, myp_est) %>%
 group_by(Y,X1,X2,X3) %>%
 summarise(count = n(),
 myp = unique(myp_est))

glm(Y ~ X1+X2+X3, data = datCola, weights = count/myp,|
 family = "binomial") %>% summary()
```
```

`summarise()` has grouped output by 'Y', 'X1', 'X2'. You can override this argument.

Call:

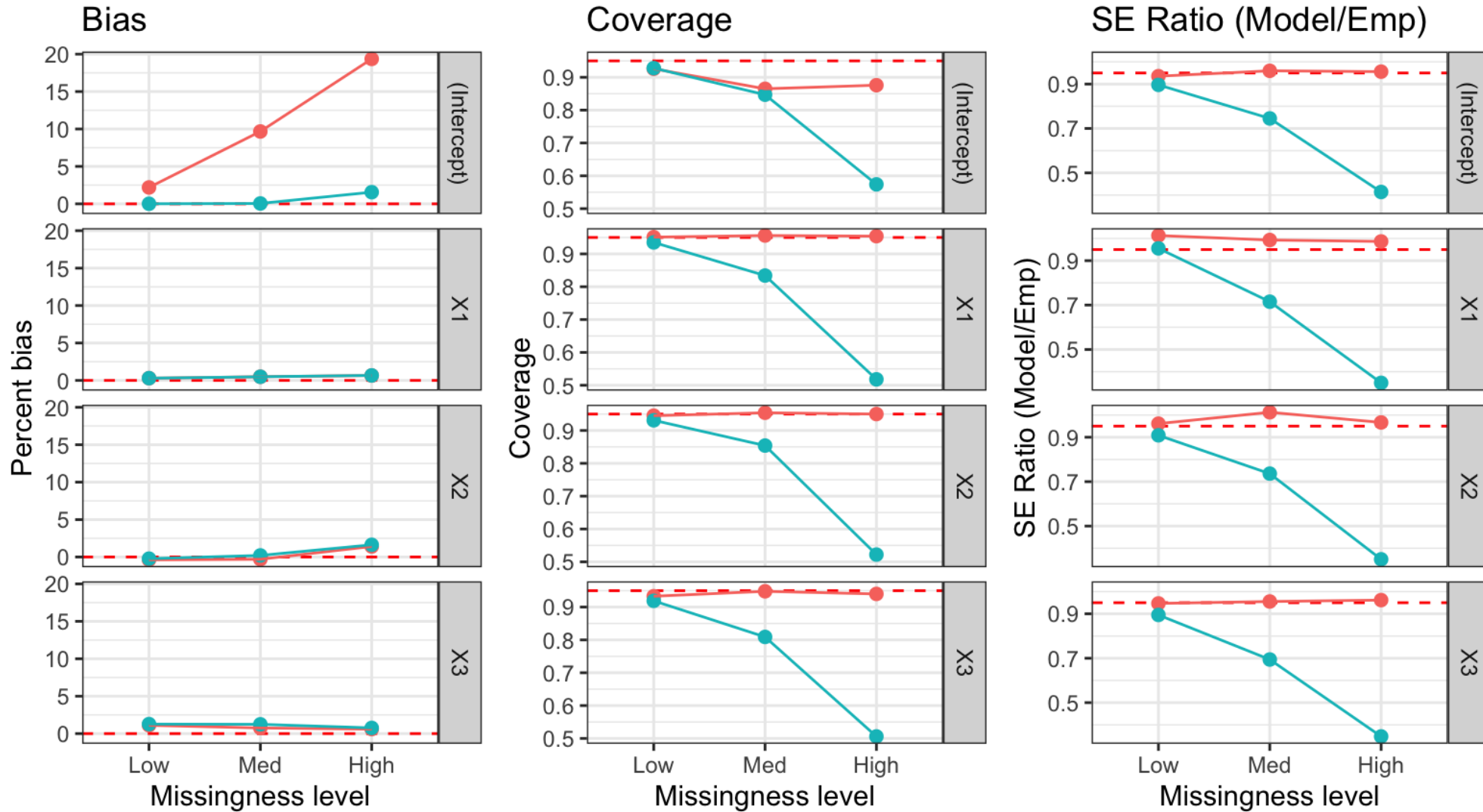
```
glm(formula = Y ~ X1 + X2 + X3, family = "binomial", data = datCola,
     weights = count/myp)
```

Coefficients:

| | Estimate | Std. Error | z value | Pr(> z) | |
|-------------|----------|------------|---------|----------|-----|
| (Intercept) | 0.9351 | 0.1796 | 5.206 | 1.93e-07 | *** |
| X1 | 0.5746 | 0.2173 | 2.645 | 0.00817 | ** |
| X2 | 1.2612 | 0.2398 | 5.259 | 1.44e-07 | *** |
| X3 | 1.1255 | 0.2339 | 4.812 | 1.49e-06 | *** |

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Small simulation study



Estimator ● CC ● IPW



Gaps remain

- **Underestimation of the standard errors,** which can lead to overly narrow confidence intervals and incorrect statistical inference.



Gaps remain

- **Underestimation of the standard errors**, which can lead to overly narrow confidence intervals and incorrect statistical inference.
- **Weight estimation is performed locally**, meaning the method does not fully leverage information across sites when modeling the probability of a complete observation.

Gaps remain

- **Underestimation of the standard errors**, which can lead to overly narrow confidence intervals and incorrect statistical inference.
- **Weight estimation is performed locally**, meaning the method does not fully leverage information across sites when modeling the probability of a complete observation.
- **Limited model flexibility**, as extensions beyond binary outcomes (e.g., count or continuous outcomes) require different regression frameworks.



Gaps remain

- **Underestimation of the standard errors**, which can lead to overly narrow confidence intervals and incorrect statistical inference.
- **Weight estimation is performed locally**, meaning the method does not fully leverage information across sites when modeling the probability of a complete observation.
- **Limited model flexibility**, as extensions beyond binary outcomes (e.g., count or continuous outcomes) require different regression frameworks.



Beyond COLA-GLM



A more general approach

Consider the following regression model

$$E_{Y|X,Z}\{g(Y)\} = m(X, Z; \theta)$$

A more general approach

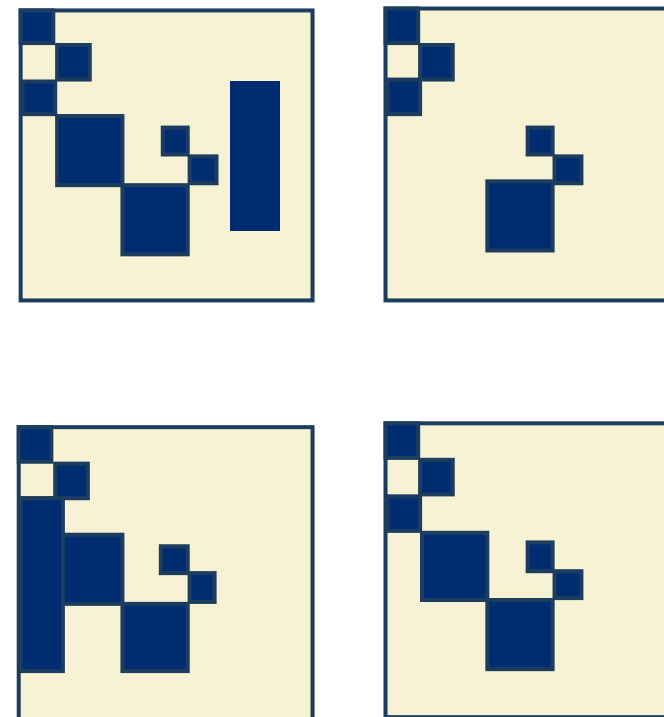
Consider the following regression model

$$E_{Y|X,Z}\{g(Y)\} = m(X, Z; \theta)$$

We are interested in estimating θ and var θ under two main restrictions.

- Full but unobserved data: for $k \in (1, \dots, K)$, let $O^{(k)} = \{Y^{(k)}, X^{(k)}, Z^{(k)}\}$
- Observed data: $O^{(k)} = \{Y^{(k)}, R^{(k)}, R^{(k)}X^{(k)}, Z^{(k)}\}$

$K = 4$



A more general approach

Consider the following regression model

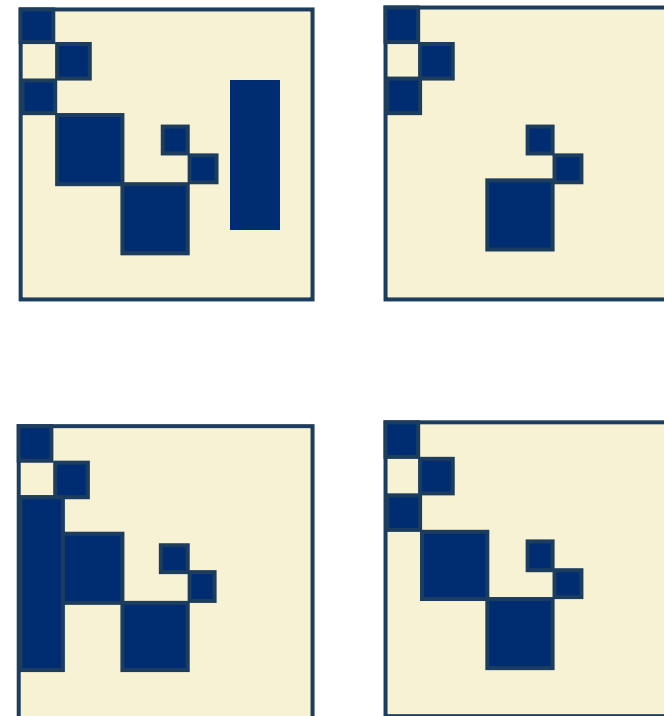
$$E_{Y|X,Z}\{g(Y)\} = m(X, Z; \theta)$$

We are interested in estimating θ and $\text{var } \theta$ under two main restrictions.

- Full but unobserved data: for $k \in (1, \dots, K)$, let $O^{(k)} = \{Y^{(k)}, X^{(k)}, Z^{(k)}\}$
- Observed data: $O^{(k)} = \{Y^{(k)}, R^{(k)}, R^{(k)}X^{(k)}, Z^{(k)}\}$

M-estimating equation approach

$K = 4$



Proposed solution

| Estimator | Estimate of θ | Estimate of Variance |
|--------------------------------|----------------------|----------------------|
| Complete case | One | One |
| Inverse probability weighting | | |
| <i>Local weights</i> | One | Three |
| <i>Multiply-Robust weights</i> | Five | Five |

Table 1: **Estimators and required communication rounds.** The number of communication rounds required to estimate θ and its variance differs across estimators depending on how the weights are estimated for the IPW estimator.

More simulation studies

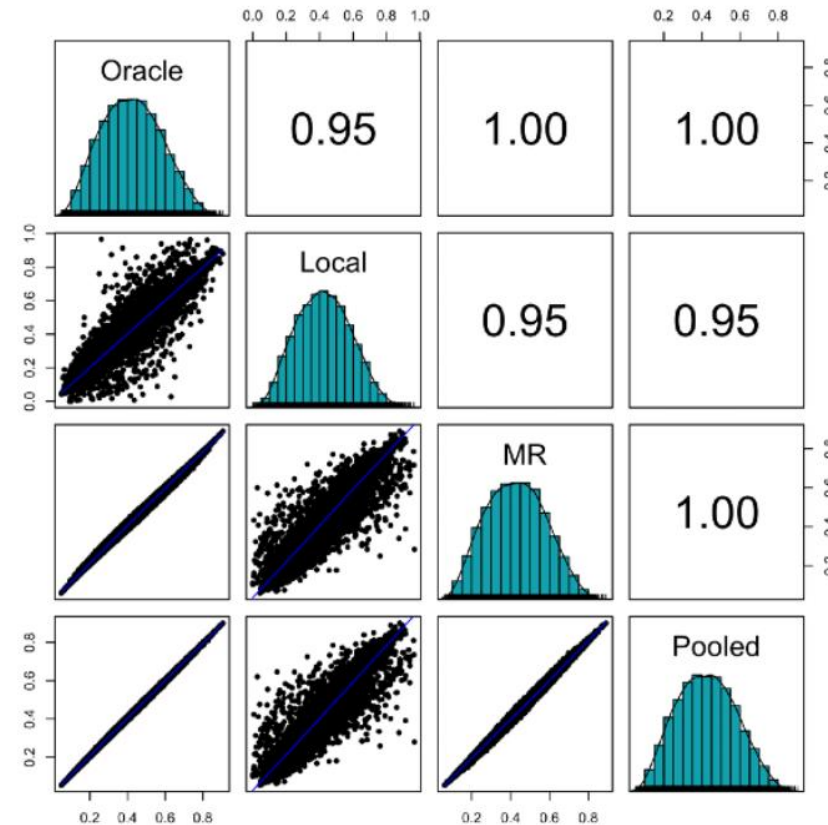


Fig. 3: Scatter plots of Pearson correlation estimates of oracle and estimated probabilities of a complete observation. *Oracle* denotes the true probability of complete observation; *Local* denotes site-specific estimates; *MR* denotes multiply robust estimates obtained by sharing nuisance parameters across sites; and *Pooled* denotes estimates obtained using pooled data. Results shown are for the case of data MAR.

More simulation studies

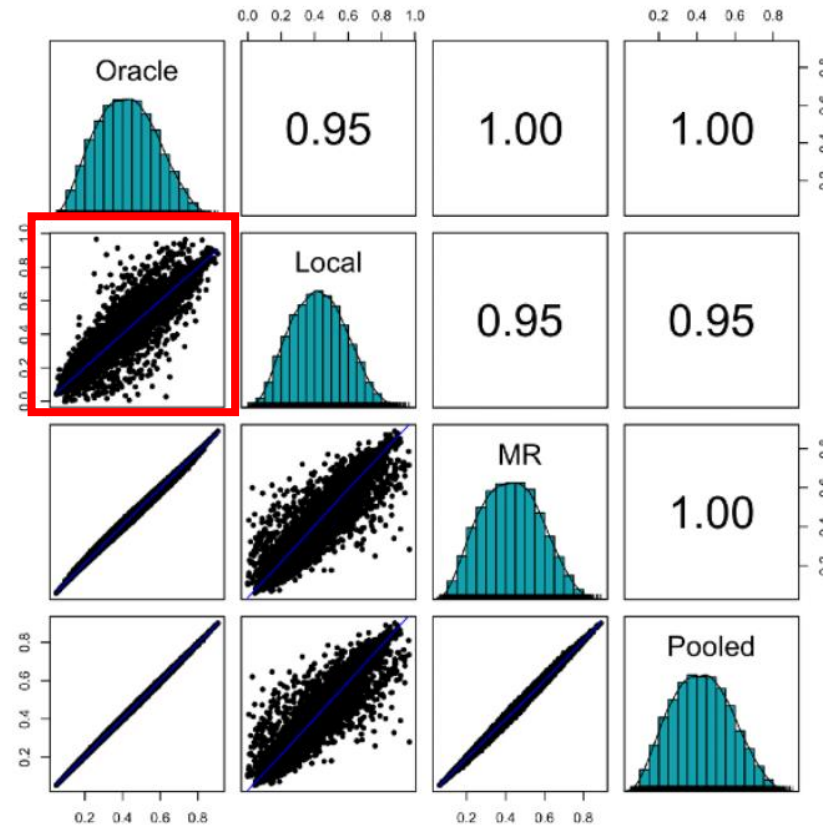


Fig. 3: Scatter plots of Pearson correlation estimates of oracle and estimated probabilities of a complete observation. *Oracle* denotes the true probability of complete observation; *Local* denotes site-specific estimates; *MR* denotes multiply robust estimates obtained by sharing nuisance parameters across sites; and *Pooled* denotes estimates obtained using pooled data. Results shown are for the case of data MAR.

More simulation studies

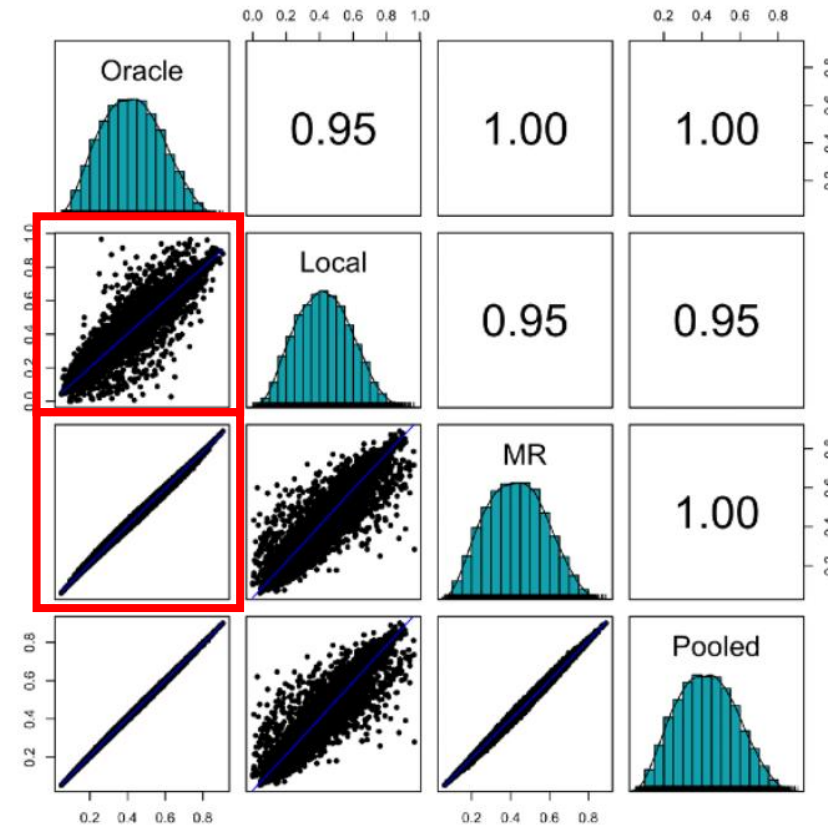


Fig. 3: Scatter plots of Pearson correlation estimates of oracle and estimated probabilities of a complete observation. *Oracle* denotes the true probability of complete observation; *Local* denotes site-specific estimates; *MR* denotes multiply robust estimates obtained by sharing nuisance parameters across sites; and *Pooled* denotes estimates obtained using pooled data. Results shown are for the case of data MAR.

More simulation studies

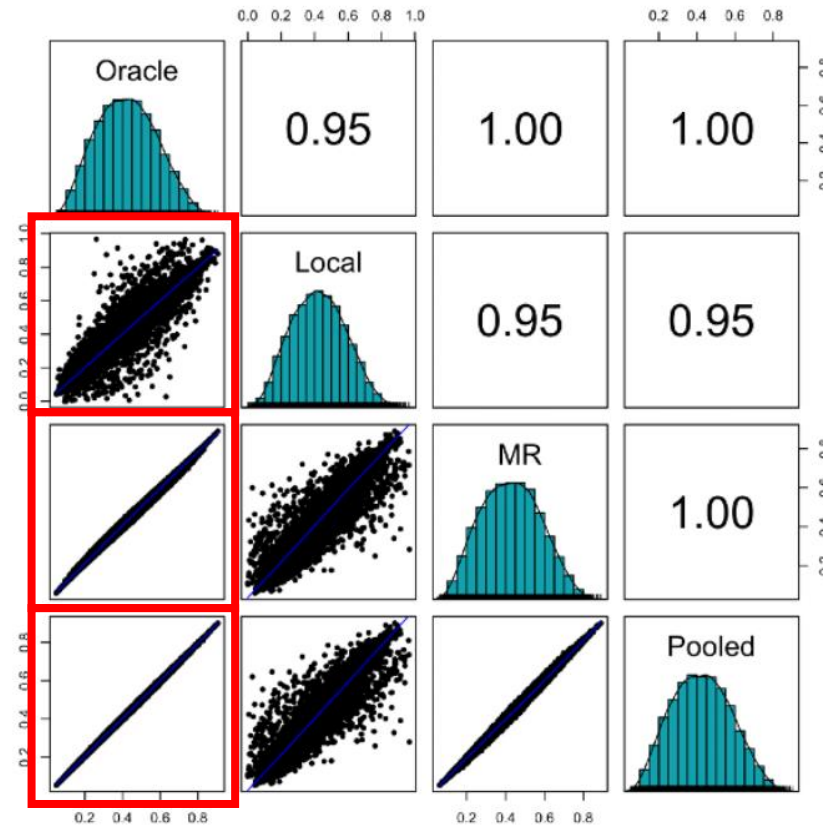
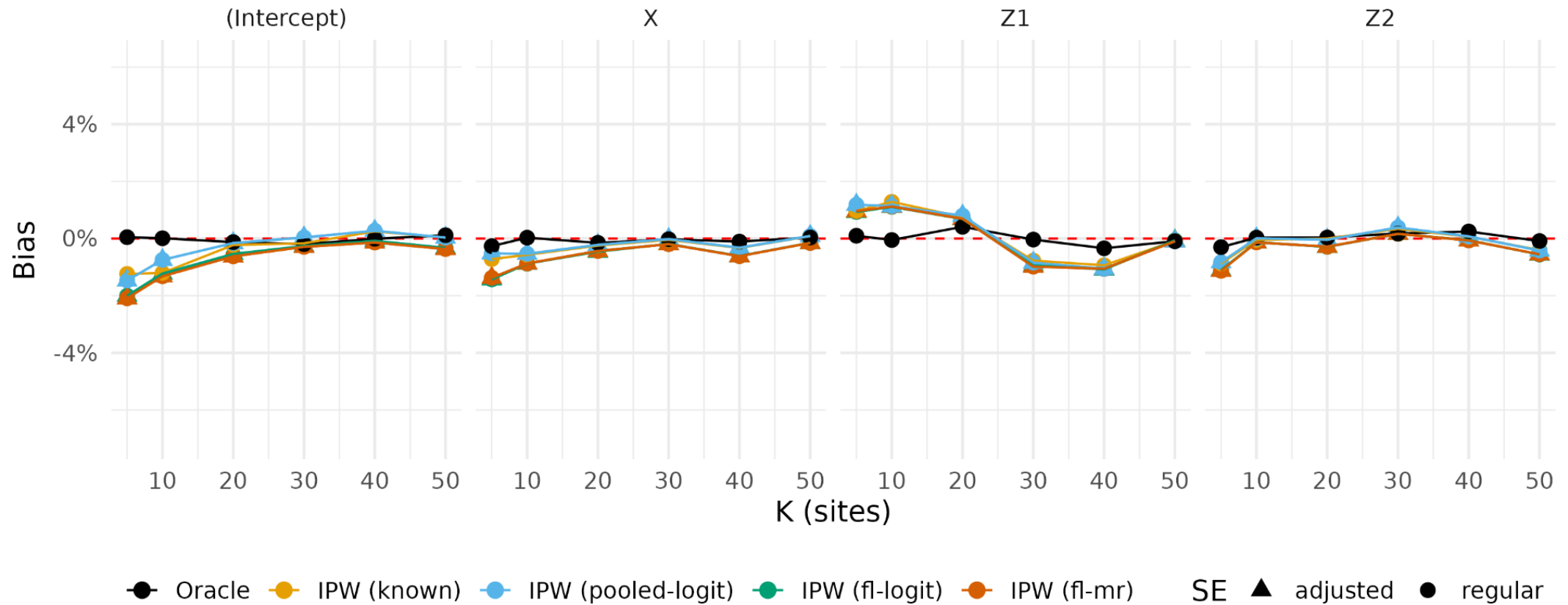
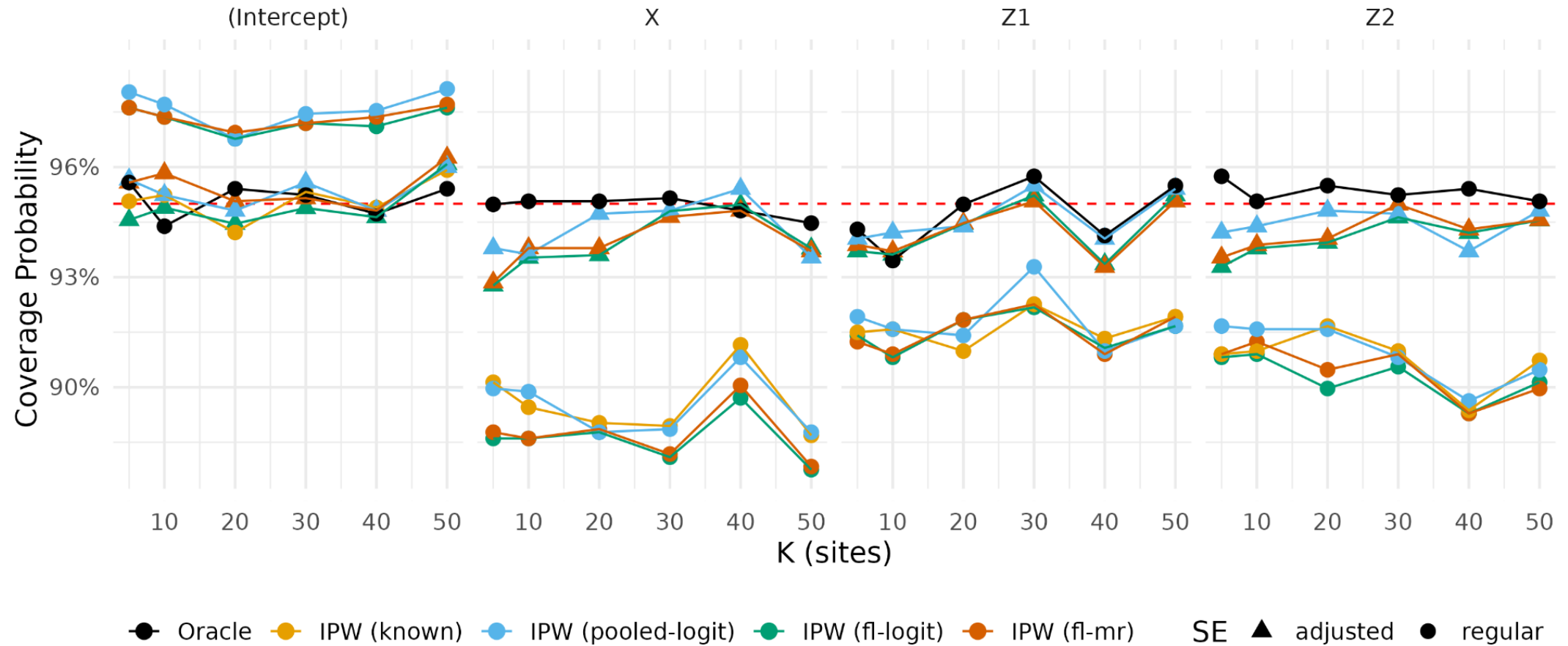


Fig. 3: Scatter plots of Pearson correlation estimates of oracle and estimated probabilities of a complete observation. *Oracle* denotes the true probability of complete observation; *Local* denotes site-specific estimates; *MR* denotes multiply robust estimates obtained by sharing nuisance parameters across sites; and *Pooled* denotes estimates obtained using pooled data. Results shown are for the case of data MAR.

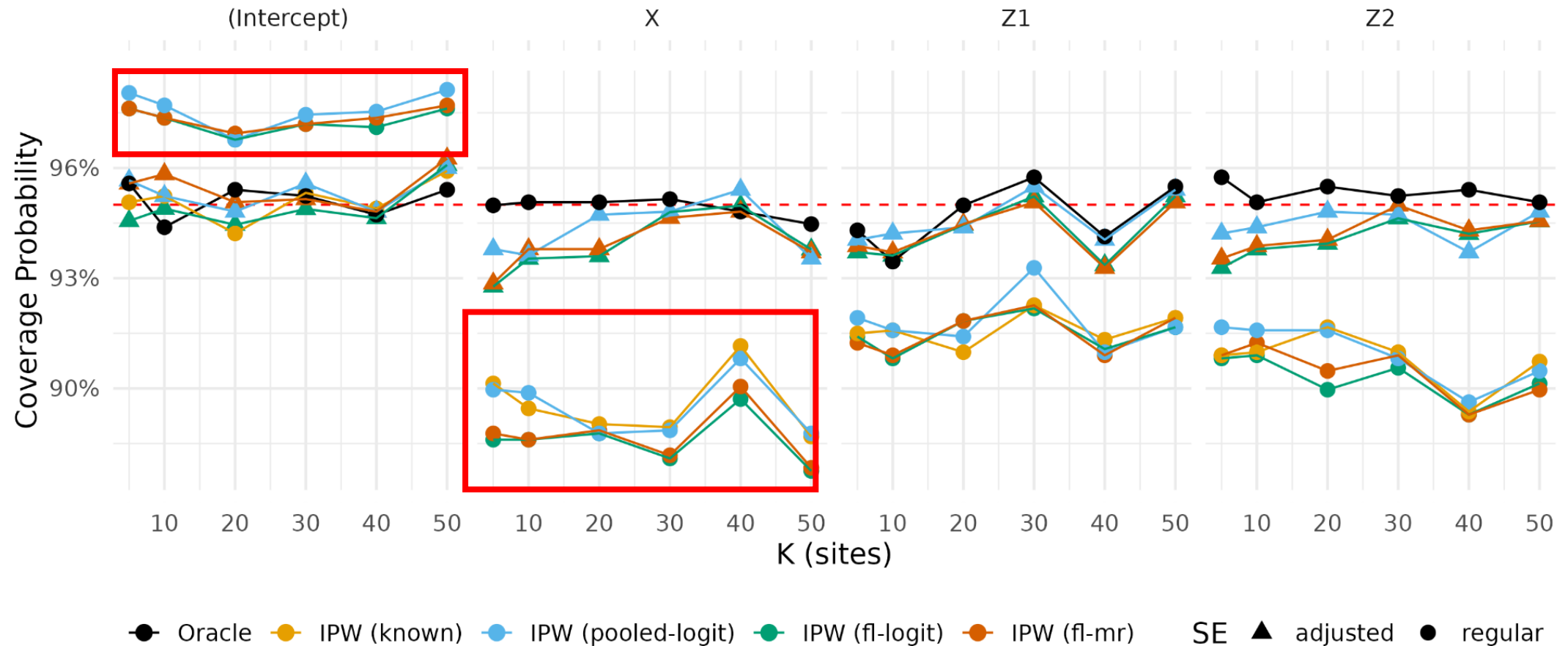
More simulation studies: Bias



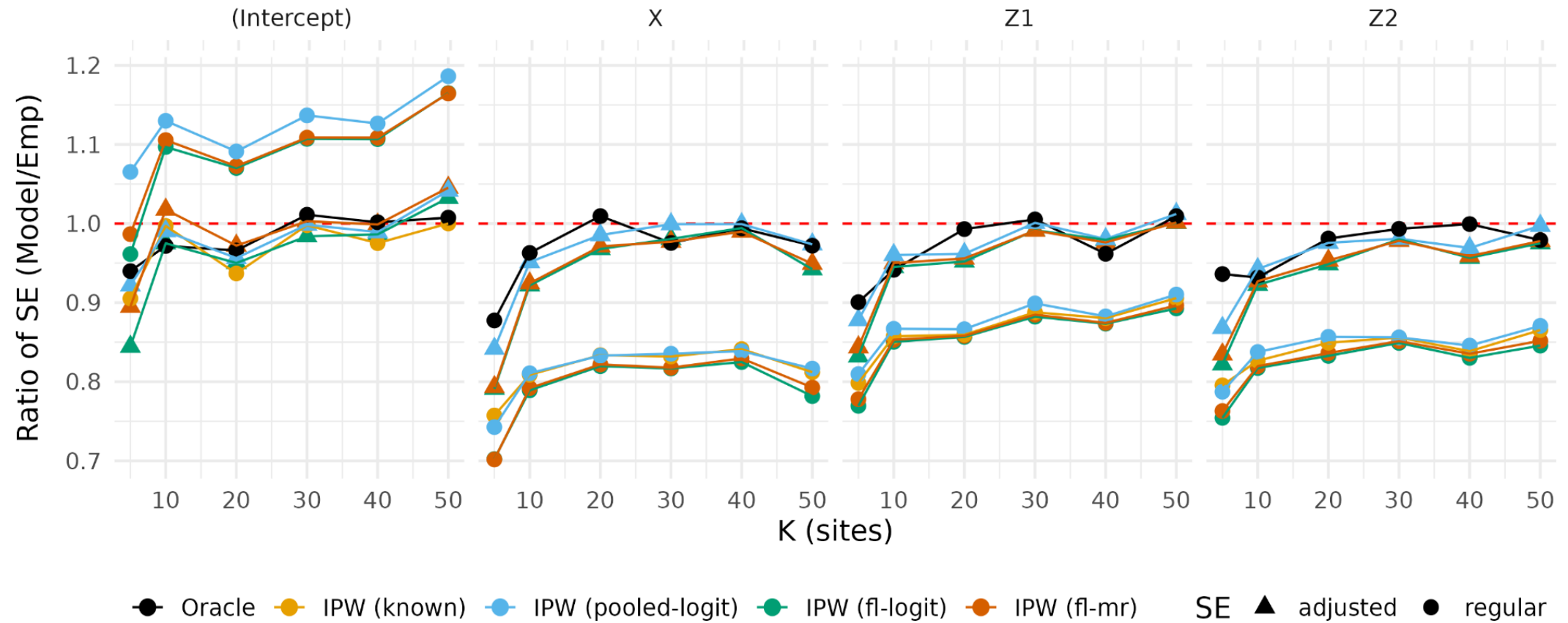
More simulation studies: Coverage



More simulation studies: Coverage



More simulation studies: SE ratio



Thank You!

jvazqu18@jh.edu

Jesus E. Vazquez, PhD
Provost's Postdoctoral Fellow



Works cited

1. White HD, Henry C, Stock EM, Arroliga AC, Ghamande S. Predicting Long-Term Outcomes in Pleural Infections. RAPID Score for Risk Stratification. Ann Am Thorac Soc. 2015 Sep;12(9):1310-6. doi: 10.1513/AnnalsATS.201505-272OC. PMID: 26193196.
2. Wu, Qiong, et al. "COLA-GLM: collaborative one-shot and lossless algorithms of generalized linear models for decentralized observational healthcare data." npj Digital Medicine 8.1 (2025): 442.
3. Kairouz, P., McMahan, H.B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A.N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R. and D'Oliveira, R.G., 2021. Advances and open problems in federated learning. Foundations and trends® in machine learning, 14(1–2), pp.1-210.
4. Chang, Changgee, et al. "Multiple imputation for analysis of incomplete data in distributed health data networks." Nature communications 11.1 (2020): 5467.