Confouding adjustment

Michael Cork

1/20/2022

Confounding

We now look to generate the exposure data using different specifications of the relationship between confounding and exposure based on work by Xiao. We generate E using six different specifications that rely on the function $\gamma(\mathbf{C}) = -0.8 +$ $(0.1, 0.1, -0.1, 0.2, 0.1, 0.1)$ C. Specifically,

1. $E = 9 \times \gamma(\mathbf{C}) + 17 + N(0, 5)$;
2. $E = 15 \times \gamma(\mathbf{C}) + 22 + T(2)$;
3. $E = 9 \times \gamma(\mathbf{C}) + 3/2C_3^2 + 15 + N(0, 5)$
4. $E = 49 \times \frac{\exp(\gamma(\mathbf{C}))}{1+\exp(\gamma(\mathbf{C}))} - 6 + N(0, 5)$;
5. $E = 42 \times \frac{1}{1+\exp(\gamma(\mathbf{C})\})} - 18 + N(0, 5)$;
6. $E = 7 \times \log(\gamma(\mathbf{C})) + 13 + N(0, 4)$;

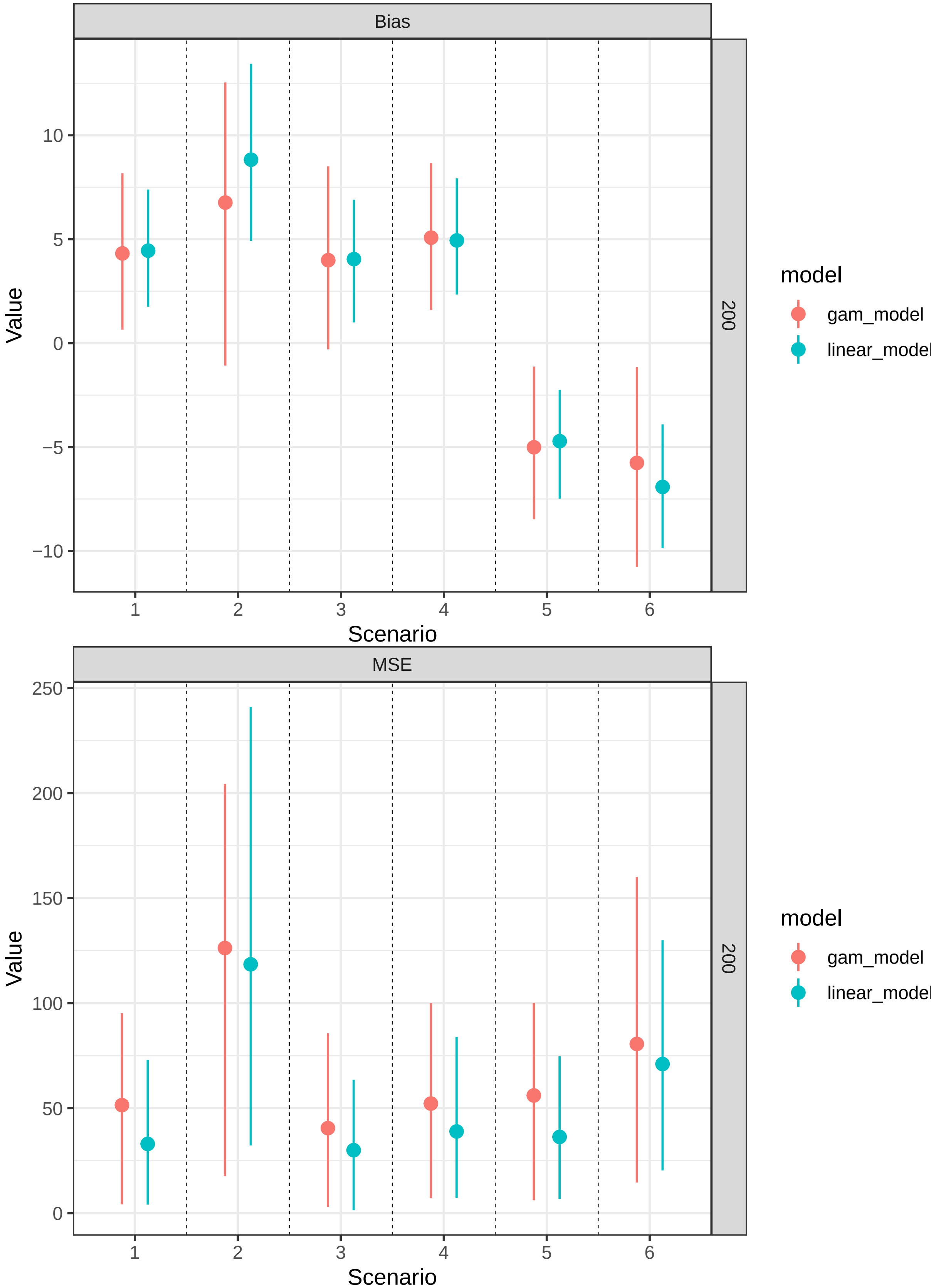
One thing to note is that for 6), in actuality it is using $W = 7 \times \log(|\gamma(\mathbf{C})|) + 13 + N(0, 4)$. I then ran through three different iterations of the sample size (N = 200, 1000, 5000) and then fit the following outcome model:

$$Y|E, C \sim N(\mu(E, C), 10^2)$$
$$\mu(E, C) = 20 + 0.1 * E - (2, 2, 3, -1, 2, 2) * C$$

Bias without linear adjustment for confounders

I first ran through each model to see the bias without adjustment for confounders to see the baseline of how biased the results might be without proper adjustment.

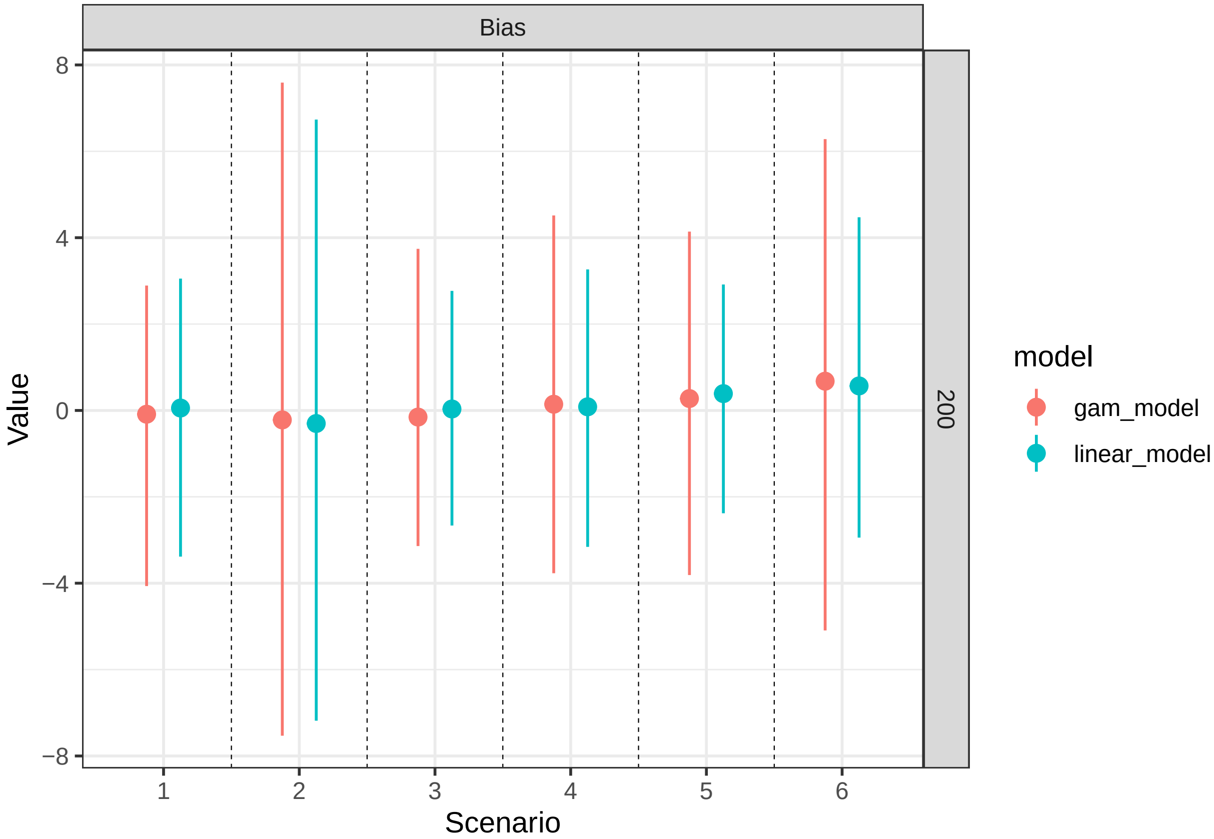
```
## `summarise()` has grouped output by 'model', 'gps_mod', 'sample_size'. You can override using the `.g
```

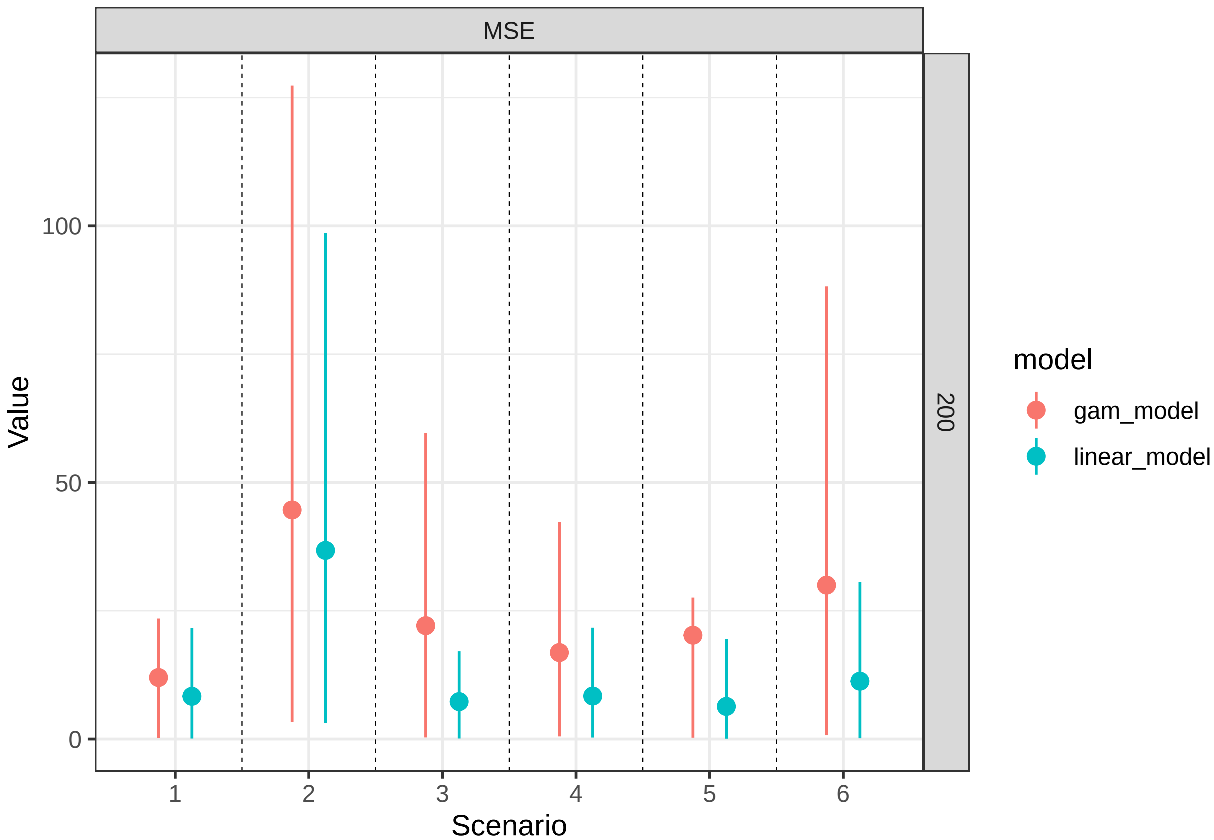

Bias
MSE
Value
Scenario
200
model
gam_model
linear_model
10
5
0
−5
−10
250
200
150
100
50
0
1
2
3
4
5
6

With adjustement for confounders

Now I linearly adjust for confounders in the additive and linear model and should see improvements, especially in scenario 1, 2.

```
## `summarise()` has grouped output by 'model', 'gps_mod', 'sample_size'. You can override using the `.g
```

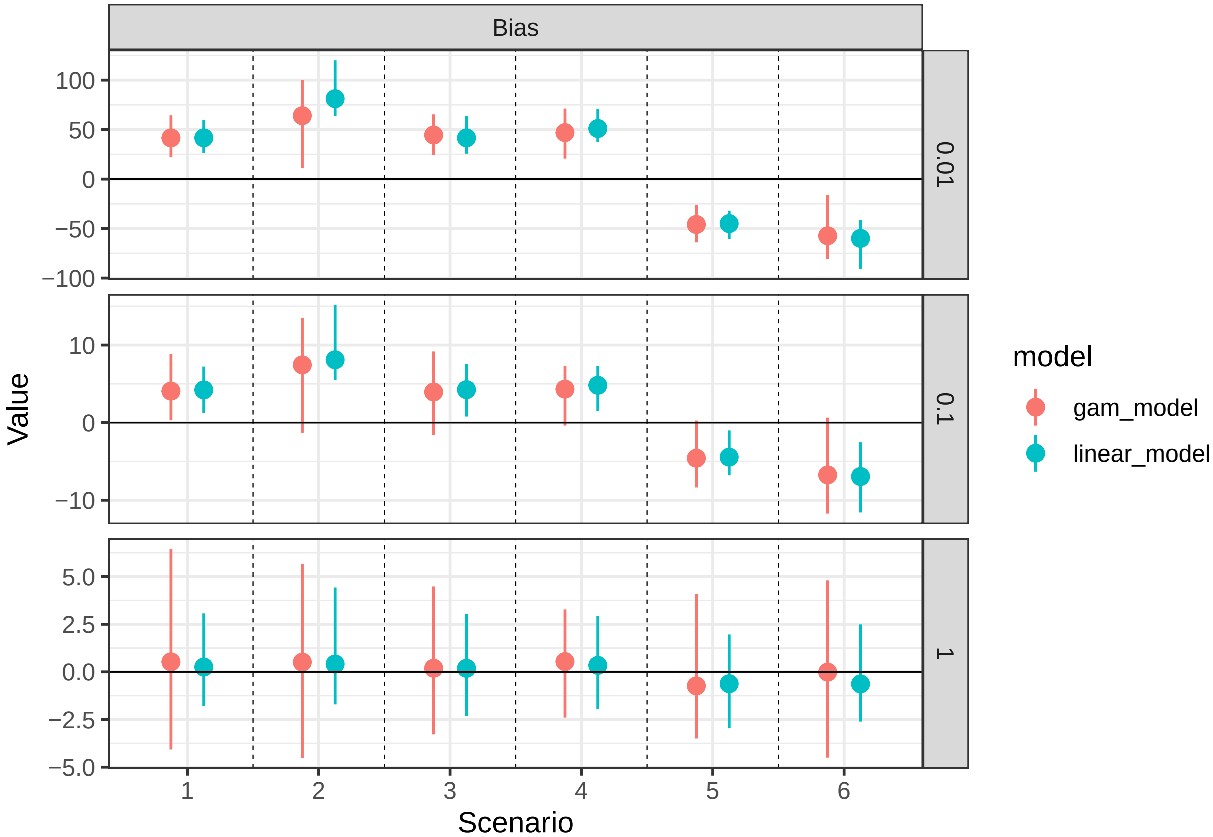


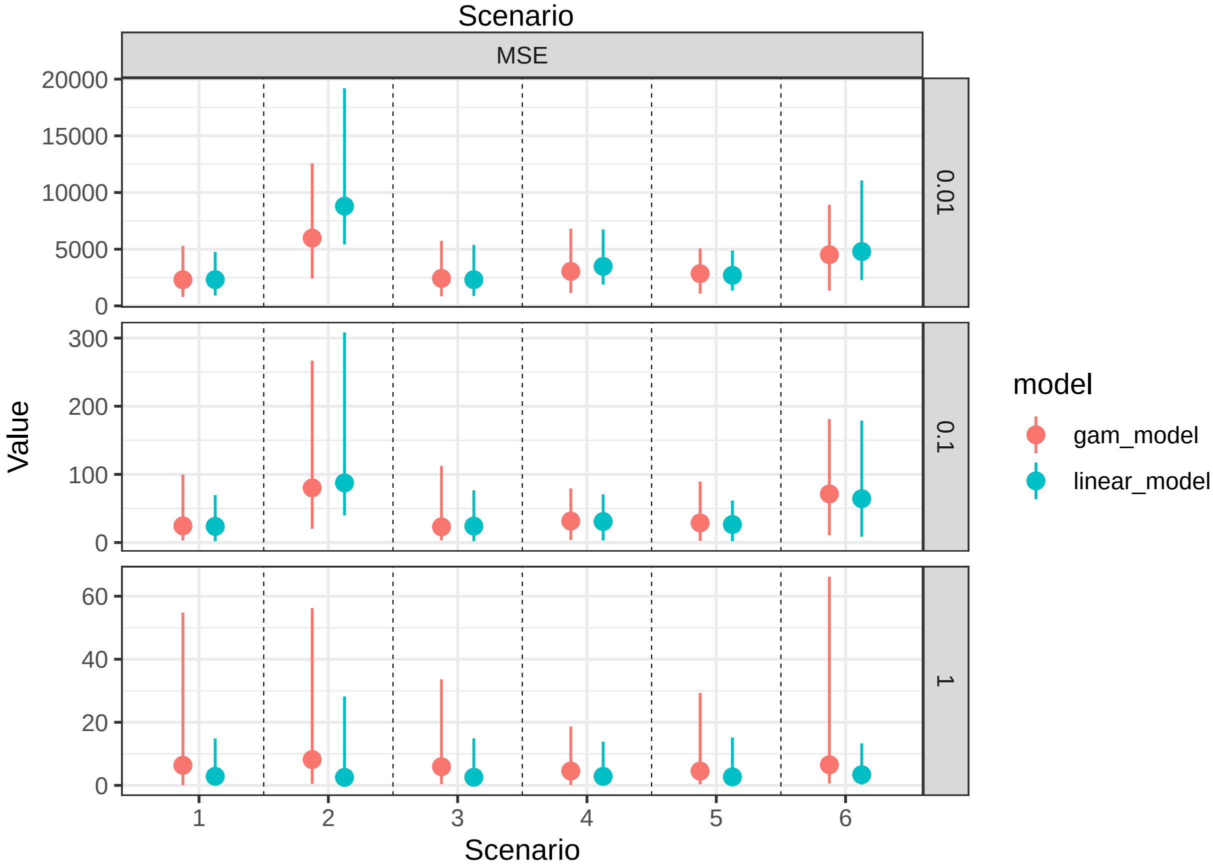


Increase effect of confounding

Now we adjust for the ratio between the standard deviation of the exposure and the standard deviation of the confounding, keeping the exposure relationship constant at 0.1. Here we don't adjust for confounders at all. Smaller values for this ratio indicate that the standard deviation of the confounding is much larger than the effect of the exposure.

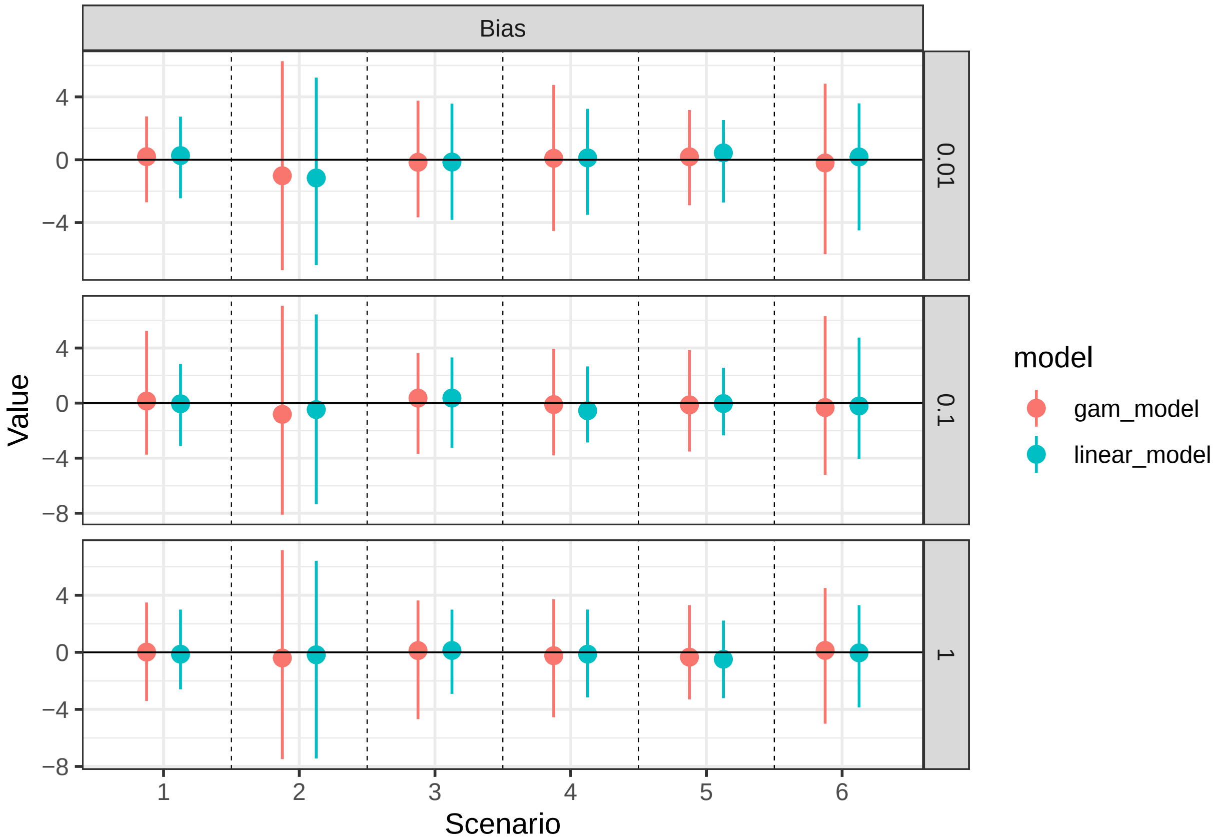
```
## `summarise()` has grouped output by 'model', 'gps_mod', 'sample_size', 'exp_conf_ratio'. You can ove
```

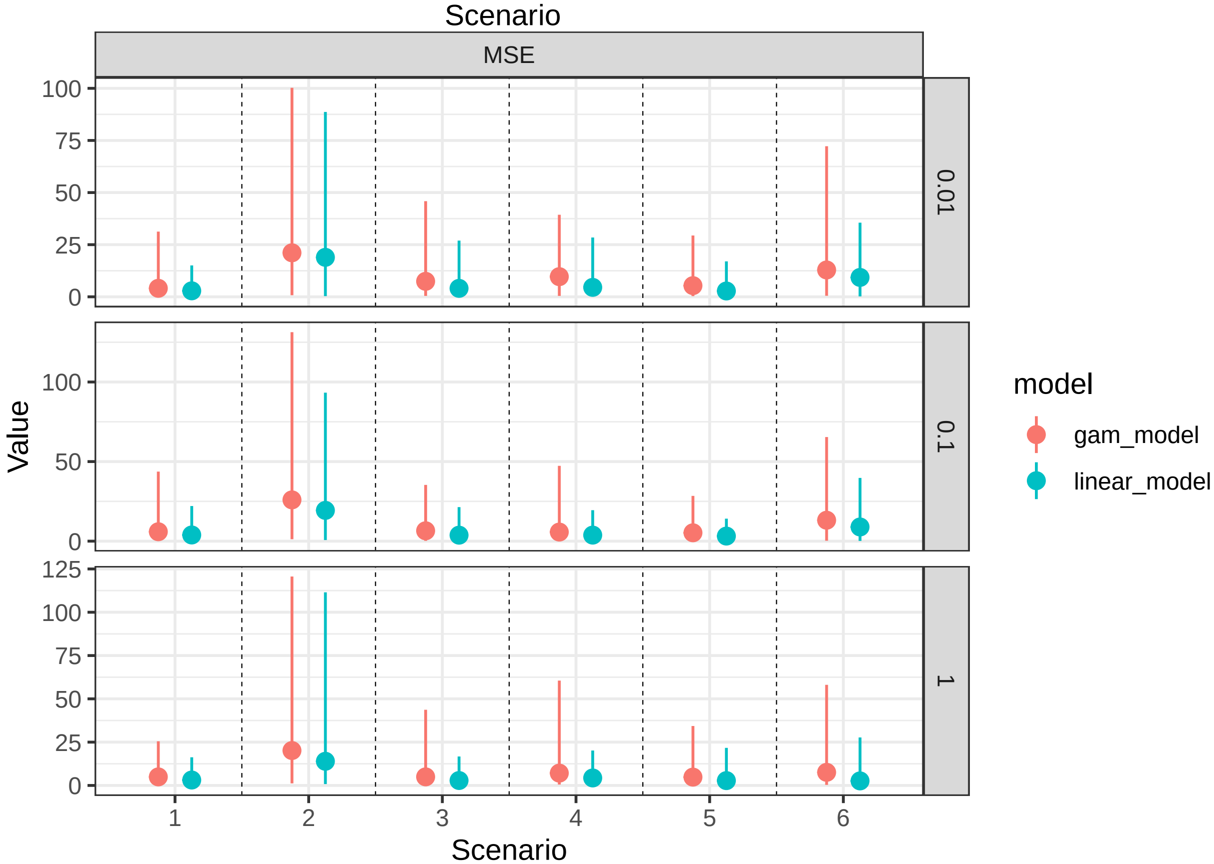




then repeat with adjustment for linear confounding

```
## `summarise()` has grouped output by 'model', 'gps_mod', 'sample_size', 'exp_conf_ratio'. You can ove
```





- Recode how you calculate the absolute bias and recode how you are varying the confoudning each time. Then run throught the simulations again and have a better description of what is going on. Dan is really good at explaining, so maybe check in with him
- Reach out to Dan on a good time to meet weekly
- Email esolinga@hsph.harvard.edu, sluvisi@hsph.harvard.edu, francesca for keys to building

Run this sample like 50 or 100 times to see if its doing the right things. For now rerun in a reaonable amount of time. Take the confounding effect, the vector times C and multiply that by 10 and 50 to see how confounding is adjusting. You can also do sd(e) / sd(C) and then find x such that sd(e) / sd(c) is .05, .01 or .005.

scale by sd(e) / sd(x*c) different noise settings. Inject more random noise. Should have bigger sample and more overalp and a little bit lower bias. Bias is going to remain to an extent. The model wont be consistent becuase the model is wrong, remain pretty unbiased. When it is not linearly adjusted.

Take .1 * E and just distribute its

Could include log of gamma c and have that as a term in the model… would be correct. You could make a truth model

sd(0.1 * exposure) / sd(c(2, 2, 3, -1, -2, -2) %*% t(confounders))

0.1*W=0.19gamma(C)+0.1*17+0.1*N(0,5) 20+0.19gamma(C)+0.1*17-vec*C 0.1*N(0.5)