Exploratory Plot

Ben Sherwood

2022-10-25

Getting data

```
allData <- NULL
for(n in c(100,500,1000,5000)){
  allData <- rbind(allData, readRDS(paste0("goodHeteroxyBig_n",n,"_2022.RDS"))$results)
}
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.1.1
```

```
library(reshape)
```

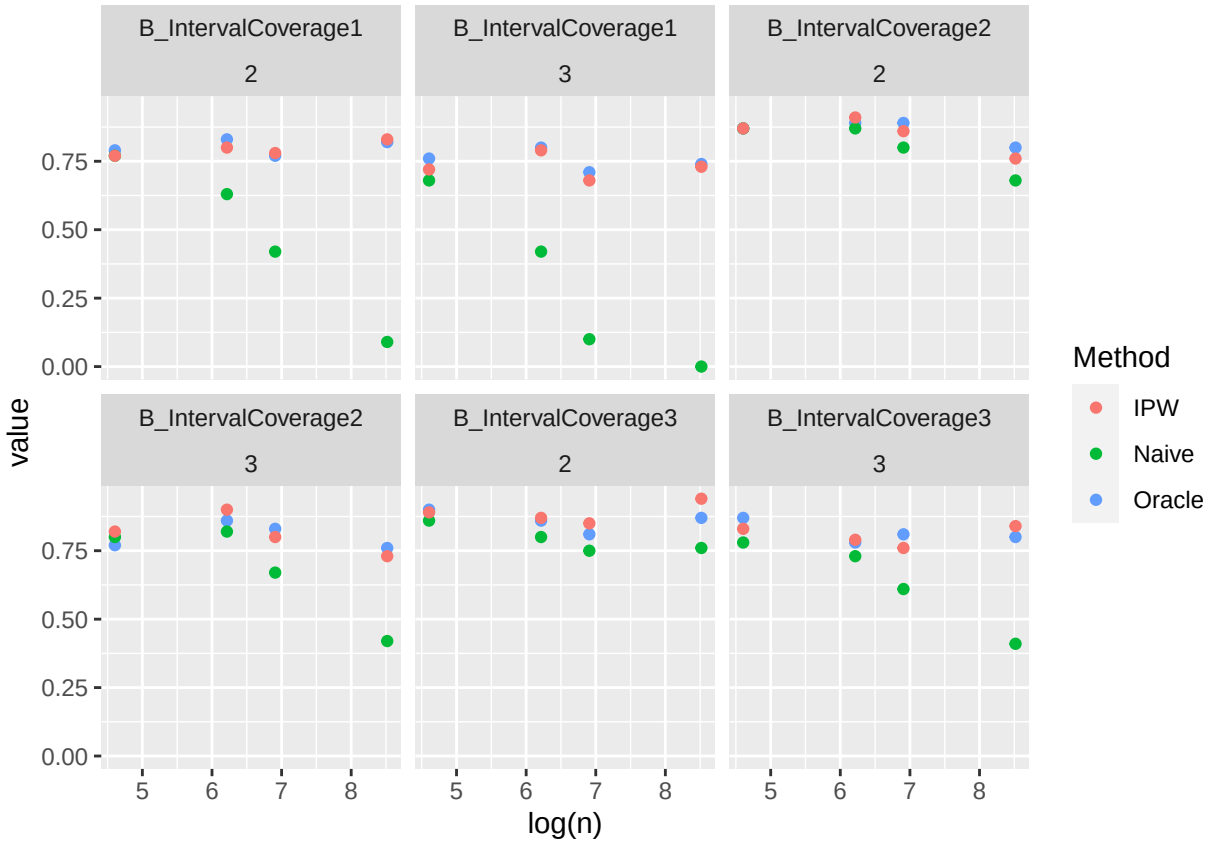
```
## Warning: package 'reshape' was built under R version 4.1.3
```

```
allData$Method <- paste(allData$Method,allData$x1Type)
allData <- allData[,-ncol(allData)]
```

$X_1 \sim U[0,1]$

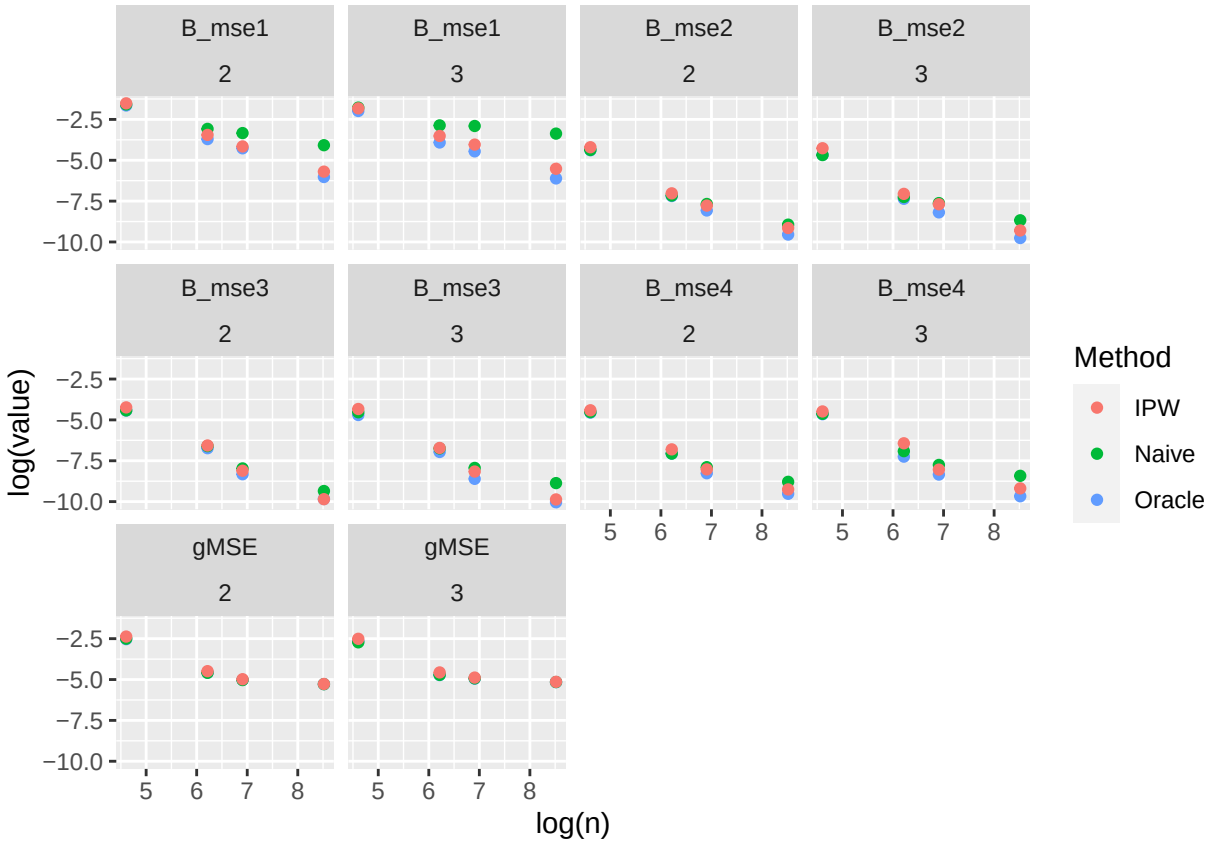
Coverage

```
diffMissInter <- allData

missPlotData <- melt(diffMissInter, id.vars=c("Method","tau","m","rho","n","ycoef"))


cData <- subset(missPlotData, variable %in% c("B_IntervalCoverage1","B_IntervalCoverage2","B_IntervalCo

qplot(x=log(n),y=value, data=cData, color=Method) + facet_wrap(~variable*m)
```



MSE Results

```
mseData <- subset(missPlotData, variable %in% c("B_mse1","B_mse2","B_mse3","B_mse4","gMSE"))
mseData$variable <- paste(mseData$variable)

qplot(x=log(n),y=log(value), data=mseData, color=Method) + facet_wrap(~variable*m)
```




```
mseTable <- allData[,c(1:3,5,11:15)]
for(i in 5:9){
  mseTable[,i] <- round(mseTable[,i],5)
}
mseTable
```

```
##      Method tau m    n  B_mse1  B_mse2  B_mse3  B_mse4    gMSE
## 1  Oracle  0.7 2  100 0.19241 0.01246 0.01194 0.01051 0.07970
## 2   Naive  0.7 2  100 0.20522 0.01265 0.01207 0.01086 0.08224
## 3     IPW  0.7 2  100 0.21915 0.01496 0.01448 0.01215 0.09336
## 4  Oracle  0.7 3  100 0.13753 0.00937 0.00908 0.00954 0.07079
## 5   Naive  0.7 3  100 0.16852 0.00926 0.01076 0.00978 0.06562
## 6     IPW  0.7 3  100 0.16143 0.01405 0.01313 0.01130 0.08144
## 7  Oracle  0.7 2  500 0.02462 0.00075 0.00119 0.00086 0.01020
## 8   Naive  0.7 2  500 0.04594 0.00078 0.00136 0.00086 0.01019
## 9     IPW  0.7 2  500 0.03177 0.00089 0.00141 0.00111 0.01126
## 10 Oracle  0.7 3  500 0.02000 0.00064 0.00095 0.00071 0.00885
## 11  Naive  0.7 3  500 0.05694 0.00071 0.00117 0.00100 0.00896
## 12    IPW  0.7 3  500 0.02984 0.00086 0.00121 0.00161 0.01040
## 13 Oracle  0.7 2 1000 0.01401 0.00031 0.00024 0.00026 0.00659
## 14  Naive  0.7 2 1000 0.03555 0.00046 0.00035 0.00037 0.00654
## 15    IPW  0.7 2 1000 0.01567 0.00042 0.00030 0.00033 0.00687
## 16 Oracle  0.7 3 1000 0.01164 0.00028 0.00018 0.00024 0.00712
## 17  Naive  0.7 3 1000 0.05492 0.00049 0.00036 0.00043 0.00724
## 18    IPW  0.7 3 1000 0.01758 0.00046 0.00029 0.00032 0.00760
## 19 Oracle  0.7 2 5000 0.00242 0.00007 0.00005 0.00007 0.00509
## 20  Naive  0.7 2 5000 0.01692 0.00013 0.00009 0.00015 0.00507
```

```
## 21    IPW  0.7 2 5000 0.00336 0.00011 0.00005 0.00010 0.00513
## 22 Oracle  0.7 3 5000 0.00222 0.00006 0.00004 0.00006 0.00576
## 23  Naive  0.7 3 5000 0.03442 0.00017 0.00014 0.00022 0.00577
## 24    IPW  0.7 3 5000 0.00401 0.00009 0.00005 0.00010 0.00585
```

```
library(xtable)
```

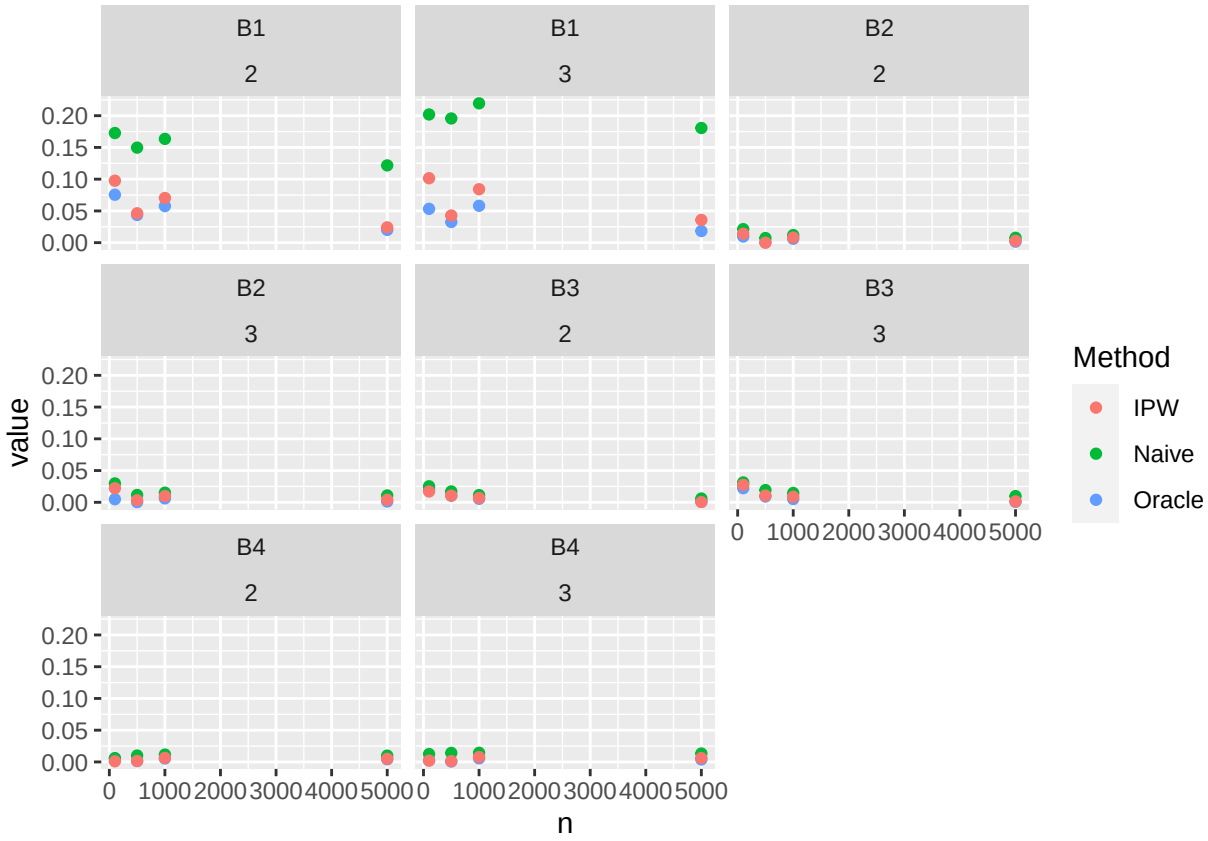
```
## Warning: package 'xtable' was built under R version 4.1.1
```

```
xmseHetero <- xtable(mseTable, digits=c(0,0,1,0,0,5,5,5,5,5))
print(xmseHetero, include.rownames=FALSE)
```

```
## % latex table generated in R 4.1.0 by xtable 1.8-4 package
## % Mon Dec 19 16:50:39 2022
## \begin{table}[ht]
## \centering
## \begin{tabular}{lrrrrrrrr}
##   \hline
## Method & tau & m & n & B\_mse1 & B\_mse2 & B\_mse3 & B\_mse4 & gMSE \\
##   \hline
## Oracle  & 0.7 & 2 & 100 & 0.19241 & 0.01246 & 0.01194 & 0.01051 & 0.07970 \\
##   Naive  & 0.7 & 2 & 100 & 0.20522 & 0.01265 & 0.01207 & 0.01086 & 0.08224 \\
##   IPW  & 0.7 & 2 & 100 & 0.21915 & 0.01496 & 0.01448 & 0.01215 & 0.09336 \\
##   Oracle  & 0.7 & 3 & 100 & 0.13753 & 0.00937 & 0.00908 & 0.00954 & 0.07079 \\
##   Naive  & 0.7 & 3 & 100 & 0.16852 & 0.00926 & 0.01076 & 0.00978 & 0.06562 \\
##   IPW  & 0.7 & 3 & 100 & 0.16143 & 0.01405 & 0.01313 & 0.01130 & 0.08144 \\
##   Oracle  & 0.7 & 2 & 500 & 0.02462 & 0.00075 & 0.00119 & 0.00086 & 0.01020 \\
##   Naive  & 0.7 & 2 & 500 & 0.04594 & 0.00078 & 0.00136 & 0.00086 & 0.01019 \\
##   IPW  & 0.7 & 2 & 500 & 0.03177 & 0.00089 & 0.00141 & 0.00111 & 0.01126 \\
##   Oracle  & 0.7 & 3 & 500 & 0.02000 & 0.00064 & 0.00095 & 0.00071 & 0.00885 \\
##   Naive  & 0.7 & 3 & 500 & 0.05694 & 0.00071 & 0.00117 & 0.00100 & 0.00896 \\
##   IPW  & 0.7 & 3 & 500 & 0.02984 & 0.00086 & 0.00121 & 0.00161 & 0.01040 \\
##   Oracle  & 0.7 & 2 & 1000 & 0.01401 & 0.00031 & 0.00024 & 0.00026 & 0.00659 \\
##   Naive  & 0.7 & 2 & 1000 & 0.03555 & 0.00046 & 0.00035 & 0.00037 & 0.00654 \\
##   IPW  & 0.7 & 2 & 1000 & 0.01567 & 0.00042 & 0.00030 & 0.00033 & 0.00687 \\
##   Oracle  & 0.7 & 3 & 1000 & 0.01164 & 0.00028 & 0.00018 & 0.00024 & 0.00712 \\
##   Naive  & 0.7 & 3 & 1000 & 0.05492 & 0.00049 & 0.00036 & 0.00043 & 0.00724 \\
##   IPW  & 0.7 & 3 & 1000 & 0.01758 & 0.00046 & 0.00029 & 0.00032 & 0.00760 \\
##   Oracle  & 0.7 & 2 & 5000 & 0.00242 & 0.00007 & 0.00005 & 0.00007 & 0.00509 \\
##   Naive  & 0.7 & 2 & 5000 & 0.01692 & 0.00013 & 0.00009 & 0.00015 & 0.00507 \\
##   IPW  & 0.7 & 2 & 5000 & 0.00336 & 0.00011 & 0.00005 & 0.00010 & 0.00513 \\
##   Oracle  & 0.7 & 3 & 5000 & 0.00222 & 0.00006 & 0.00004 & 0.00006 & 0.00576 \\
##   Naive  & 0.7 & 3 & 5000 & 0.03442 & 0.00017 & 0.00014 & 0.00022 & 0.00577 \\
##   IPW  & 0.7 & 3 & 5000 & 0.00401 & 0.00009 & 0.00005 & 0.00010 & 0.00585 \\
##    \hline
## \end{tabular}
## \end{table}
```

Bias

```
biasData <- subset(missPlotData, variable %in% c("B1","B2","B3","B4"))
biasData$variable <- paste(biasData$variable)

qplot(x=n,y=value, data=biasData, color=Method) + facet_wrap(~variable*m)
```


B1
2
B1
3
B2
2
B2
3
B3
2
B3
3
B4
2
B4
3
value
0.20
0.15
0.10
0.05
0.00
0
1000
2000
3000
4000
5000
n
Method
IPW
Naive
Oracle