Replication Document for the Main Results in "Precise Unbiased Estimation in Randomized Experiments using Auxiliary Observational Data"

1 Preliminaries

This document will reproduce all of the tables and figures from the manuscript. The tables and figures will appear in the compiled version of this PDF, as well as in stand-alone files to be incorporated into the main manuscript.

This analysis in this document starts *after* the deep learning prediction model has already been fit to the remnant data and predicted outcomes for RCT subjects have already been generated. For code to replicate that part of the process, see `https://github.com/adamSales/rebarLoop`. The deep learning models involve a random component, so each time they are fit they return slightly different results; unfortunately, when we performed the analysis that gave rise to the results in the paper, we did not set a random seed, so the model predictions for the results in the paper are not exactly replicable. However, the subsequent analysis is exaclty replicable—starting the same set of model predictions that we had, the following document will generate the precise results reported in the paper. To do so, specify the following variable:

```
exactReplication <- FALSE
```

If you wish to re-run the deep learning models using the posted replication code—and generate slightly different results from what's in the paper (they shouldn't differ *too* much)—then change the above code to 'exactReplication ¡- FALSE'.

```
set.seed(365)
```

```
library(scales)
#library(tidyverse)
library(dplyr)
library(ggplot2)
library(tibble)
library(purrr)
library(tidyr)
```

```r
library(loop.estimator)
library(kableExtra)
library(xtable)
library(knitr)
library(tikzDevice)
library(estimatr)
library(forcats)


## specialized versions of the LOOP estimator
source('code/loop_ols.R')
source('code/loop_ext.R')
## functions for estimating effects
source('code/analysisFunctions.r')
```

Names of covariates for within-sample covariate adjustment:

```r
covNames <- c(
    "Prior.Problem.Count",
    "Prior.Percent.Correct",
    "Prior.Assignments.Assigned",
    "Prior.Percent.Completion",
    "Prior.Class.Percent.Completion",
    "Prior.Homework.Assigned",
    "Prior.Homework.Percent.Completion",
    "Prior.Class.Homework.Percent.Completion",
    "male",
    "unknownGender")#)
```

2 Data

Here we load in the data for estimating effects and standard errors using several different methods discussed in the manuscript. Note that the predictions from the model fit in the remnant are already part of the datasets (which are themselves part of the GitHub repository) under the column name `p_complete`.

Load in and clean the data:

```
source('code/dataPrep.r')
```

Replicating Table 1 from the manuscript:

```
source('code/covTable.r')
print(covTable, add.to.row=ATR)
```

	Mean	SD	% Missing
Problem Count	601.13	784.45	2
Percent Correct	0.68	0.13	2
Assignments Assigned	104.25	413.94	13
Percent Completion	0.89	0.21	13
Class Percent Completion	0.90	0.13	22
Homework Assigned	25.97	29.90	50
Homework Percent Completion	0.93	0.16	59
Class Homework Percent Completion	0.93	0.09	56
Guessed Gender	Male: 36%	Female: 36%	Unknown: 28%

Table 1: Pooled summary statistics for aggregate prior ASSISTments performance used as within-sample covariates.

2.1 Imputing Missing Covariates

To impute missing covariate values, when possible we imputed the classroom mean covariate value for students working on that skill builder. When there were no other available values for a covariate for students in the same classroom working on the same skill builder, we imputed with the global mean of students working on that skill builder. Since covariates are all pre-treatment and the imputation did not depend on treatment status, the imputed covariates are themselves covariates, measured for all subjects. Therefore, we need not correct for the imputation scheme in our treatment effect estimation.

```
### first fill in with class/problem_set mean
### if that doesn't work, fill in with problem_set mean
dat <- dat%>%
  group_by(Class.ID,problem_set)%>%
  mutate(
    across(all_of(covNames),~ifelse(is.finite(.),.,mean(.,na.rm=TRUE)))
  )%>%
  group_by(problem_set)%>%
  mutate(
```

```
    across(all_of(covNames),~ifelse(is.finite(.),.,mean(.,na.rm=TRUE)))
  )%>%
  ungroup()

stopifnot(all(sapply(covNames,function(x) mean(is.finite(dat[[x]])))==1))
```

3 Estimate Effects

Here we estimate effects of treatment for each of the 33 skill builders in the dataset. The functions for estimating effects are all found in the file `code/analysisFunctions.r`. This includes the function `full()` which estimates all five treatment effects discussed in the paper.

```
fullres <- sapply(levels(dat$problem_set),full,dat=dat,
                  covNames=covNames,simplify=FALSE)

### name the problem sets based on the
### SE from the simple difference estimator
rnk <- rank(sapply(fullres,function(x) x['simpDiff','se']))
names(fullres) <- as.character(rnk)

for(i in 1:length(fullres))
    attr(fullres[[i]],'psid') <- levels(dat$problem_set)[i]

save(fullres,file='results/fullres.RData')

dat$ps <- rnk[as.character(dat$problem_set)]
```

Replicate Table 2. The numbering of the experiments derives from the estimated standard errors, so this comes after effect estimation.

```
source('code/psTable.r')

 kbl(tab,

           booktabs=FALSE,
           col.names=rep(c("",rep(c("Trt","Ctl"),2)),2),
           caption="Sample sizes and \\% homework completion by treatment
           group in each of the 33 A/B tests.",
           label="info")%>%
```

Table 2: Sample sizes and % homework completion by treatment group in each of the 33 A/B tests.

Experiment	n		% Complete		Experiment	n		% Complete	
	Trt	Ctl	Trt	Ctl		Trt	Ctl	Trt	Ctl
1	956	961	94	93	18	165	170	92	89
2	329	363	98	96	19	259	246	82	85
3	649	610	86	88	20	199	213	85	88
4	201	228	97	95	21	258	276	82	80
5	910	887	73	72	22	188	193	89	85
6	931	900	61	64	23	242	266	81	76
7	360	344	88	88	24	279	235	72	69
8	492	463	79	81	25	269	288	65	59
9	215	211	93	92	26	225	232	73	74
10	231	197	92	91	27	267	256	63	62
11	607	578	68	63	28	228	244	68	64
12	370	384	83	82	29	239	258	54	48
13	338	289	88	84	30	74	92	91	84
14	478	476	76	73	31	69	67	91	87
15	193	209	89	93	32	76	81	62	70
16	404	451	73	69	33	15	11	73	55
17	264	274	84	85	NA	NA	NA	NA	NA

```
    kable_styling()%>%
    column_spec(5,border_right=TRUE)%>%
    add_header_above(rep(c("Experiment"=1,"n"=2,"% Complete"=2),2))
```

4 Figures

The following code creates a dataset called `comparisons` that includes the sampling variance ratios comparing each method to the others, for each problem set. It also produces a table (which is not in the manuscript) giving the estimated standard error for each method and each experiment.

```
source('code/figurePrep.r')

pwidePrint <- pwide
names(pwidePrint)[-1] <- paste0('$',methodName[names(pwidePrint)[-1]],'$')
```

```
kable(pwidePrint,row.names=FALSE,
    caption="Estimated standard error for the ATE
     in each skill builder, using each method
     discussed in the manuscript",
     label="tab:SEs",digits=3,escape=FALSE)
```

Figure 1, comparing $\hat{\tau}^{\mathrm{DM}}$, $\hat{\tau}^{\mathrm{RE}}$, and $\hat{\tau}^{\mathrm{SS}}[x^r, \mathrm{LS}]$:

```
p <- comparisons%>%
    filter(method1%in%c('ReLOOP','Rebar'),
    method2%in%c('ReLOOPEN','Rebar','SimpleDifference'))%>%
    ggplot(aes(ssMult))+#,fill=exGroup))+
    geom_dotplot( method="histodot", binwidth = .05 )  +
    labs( x = "Relative Ratio of Sample Variances", y="" ) +
    geom_vline( xintercept = 1, col="red" ) +
    facet_wrap(~comp,nrow=1)+
    theme(legend.position = "none",
        panel.grid = element_blank(),
        axis.title.y = element_blank(),
        axis.text.y= element_blank(),
        axis.ticks.y = element_blank(),
        text=element_text(size=12),
        strip.text=element_text(size=12,lineheight=0.5))

tikz('figure/fig4.tex',width=6.4,height=2,standAlone=FALSE)
print(p)
dev.off()

## tikz output
##           2
```

Figure 2, comparing $\hat{\tau}^{\mathrm{DM}}$, $\hat{\tau}^{\mathrm{SS}}[x^r, \mathrm{LS}]$, $\hat{\tau}^{\mathrm{SS}}[\boldsymbol{x}, \mathrm{RF}]$, and $\hat{\tau}^{\mathrm{SS}}[\tilde{\boldsymbol{x}}, \mathrm{EN}]$:

```
p <- comparisons%>%
    filter(method1%in%c('ReLOOPEN'),
    method2%in%c('Loop','ReLOOP','SimpleDifference'))%>%
    mutate(comp=factor(comp,levels=unique(as.character(comp))))%>%
ggplot(aes(ssMult))+#,fill=exGroup))+
    geom_dotplot( method="histodot", binwidth = .05 )  +
    labs( x = "Relative Ratio of Sample Variances", y="" ) +
    geom_vline( xintercept = 1, col="red" ) +
```

Table 3: Estimated standard error for the ATE in each skill builder, using each method discussed in the manuscript

experiment	$\hat{\tau}^{\mathrm{SS}}[\tilde{\boldsymbol{x}}, \mathrm{EN}]$	$\hat{\tau}^{\mathrm{SS}}[x^r, \mathrm{LS}]$	$\hat{\tau}^{\mathrm{SS}}[\boldsymbol{x}, \mathrm{RF}]$	$\hat{\tau}^{\mathrm{RE}}$	$\hat{\tau}^{\mathrm{DM}}$
1	0.010	0.011	0.011	0.011	0.011
10	0.026	0.026	0.028	0.027	0.027
11	0.020	0.024	0.020	0.024	0.028
12	0.025	0.027	0.025	0.028	0.028
13	0.026	0.027	0.026	0.027	0.028
14	0.022	0.026	0.022	0.026	0.028
15	0.026	0.026	0.028	0.027	0.028
16	0.028	0.029	0.030	0.029	0.031
17	0.028	0.029	0.029	0.029	0.031
18	0.031	0.031	0.031	0.031	0.032
19	0.031	0.032	0.031	0.032	0.033
2	0.012	0.012	0.012	0.017	0.012
20	0.032	0.032	0.033	0.032	0.034
21	0.032	0.033	0.032	0.033	0.034
22	0.032	0.032	0.033	0.033	0.034
23	0.034	0.034	0.035	0.034	0.036
24	0.033	0.038	0.035	0.038	0.040
25	0.030	0.040	0.029	0.041	0.041
26	0.038	0.040	0.038	0.040	0.041
27	0.031	0.034	0.031	0.035	0.043
28	0.038	0.040	0.038	0.040	0.044
29	0.039	0.042	0.039	0.042	0.045
3	0.017	0.018	0.017	0.018	0.019
30	0.050	0.050	0.054	0.050	0.052
31	0.049	0.049	0.050	0.051	0.054
32	0.063	0.068	0.060	0.067	0.076
33	0.129	0.136	0.153	0.145	0.197
4	0.019	0.019	0.019	0.022	0.019
5	0.017	0.019	0.017	0.019	0.021
6	0.019	0.019	0.019	0.019	0.023
7	0.019	0.022	0.019	0.022	0.025
8	0.019	0.022	0.019	0.022	0.026
9	0.024	0.025	0.024	0.026	0.026

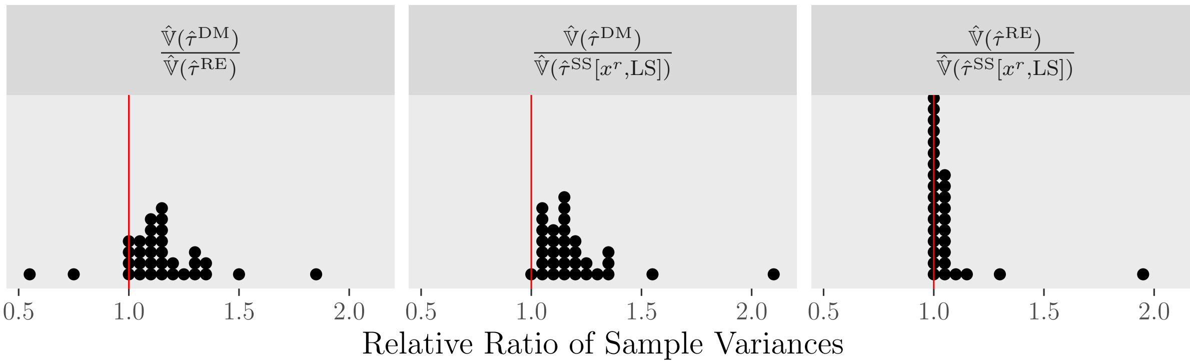


Figure 1: A dotplot showing sample size multipliers (i.e. sampling variance ratios) comparing $\hat{\tau}^{\mathrm{DM}}$, $\hat{\tau}^{\mathrm{RE}}$, and $\hat{\tau}^{\mathrm{SS}}[x^r, \mathrm{LS}]$ on the 33 ASSISTments TestBed experiments.

```
    facet_wrap(~comp,nrow=1)+
    theme(legend.position = "none",
        panel.grid = element_blank(),
        axis.title.y = element_blank(),
        axis.text.y= element_blank(),
        axis.ticks.y = element_blank(),
        text=element_text(size=12),
        strip.text=element_text(size=12,lineheight=0.5))
#print(p)

tikz('figure/fig5alt.tex',width=6.4,height=2,standAlone=FALSE)
print(p)

dev.off()

## tikz output
##           2
```

The following code reproduces some of the numbers in the manuscript text describing the results:

```
compTab <- comparisons%>%group_by(method1,method2)%>%
  summarize(
            worse=sum(ssMult<0.975),
            equal=sum(abs(ssMult-1)<0.025),
            better=sum(ssMult>1.025),
```

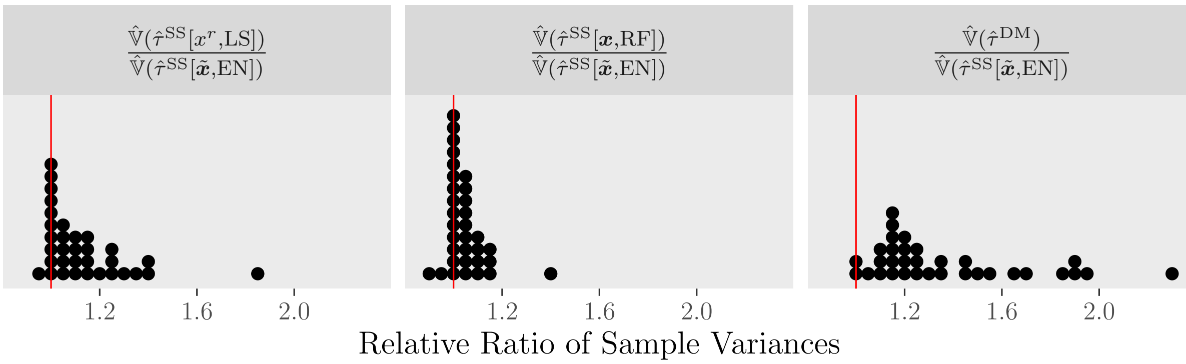


Figure 2: A dotplot showing sample size multipliers (i.e. sampling variance ratios) comparing $\hat{\tau}^{SS}[\tilde{\boldsymbol{x}}, \text{EN}]$ to $\hat{\tau}^{SS}[x^r, \text{LS}]$ $\hat{\tau}^{SS}[\boldsymbol{x}; \text{RF}]$, and $\hat{\tau}^{DM}$, respectively, on the 33 ASSISTments TestBed experiments.

```
            best=max(ssMult),
            bestPS=experiment[which.max(ssMult)],
            best2=sort(ssMult,decreasing=TRUE)[2],
            best2ps=experiment[rank(ssMult)==32],
            worst=min(ssMult),
            worstPS=experiment[which.min(ssMult)]
            )%>%ungroup()%>%
    mutate(across(starts_with('method'),
        ~paste0('$',methodName[as.character(.)],'$')))

compTab%>%select(method1:bestPS)%>%kable(escape = FALSE)
```

method1	method2	worse	equal	better	best	bestPS
$\hat{\tau}^{SS}[\tilde{\boldsymbol{x}}, \text{EN}]$	$\hat{\tau}^{SS}[x^r, \text{LS}]$	1	10	22	1.844856	25
$\hat{\tau}^{SS}[\tilde{\boldsymbol{x}}, \text{EN}]$	$\hat{\tau}^{SS}[\boldsymbol{x}, \text{RF}]$	2	14	17	1.400882	33
$\hat{\tau}^{SS}[\tilde{\boldsymbol{x}}, \text{EN}]$	$\hat{\tau}^{RE}$	1	3	29	1.905476	25
$\hat{\tau}^{SS}[\tilde{\boldsymbol{x}}, \text{EN}]$	$\hat{\tau}^{DM}$	0	2	31	2.314750	33
$\hat{\tau}^{SS}[x^r, \text{LS}]$	$\hat{\tau}^{SS}[\boldsymbol{x}, \text{RF}]$	17	5	11	1.266307	33
$\hat{\tau}^{SS}[x^r, \text{LS}]$	$\hat{\tau}^{RE}$	0	19	14	1.944435	2
$\hat{\tau}^{SS}[x^r, \text{LS}]$	$\hat{\tau}^{DM}$	0	1	32	2.092385	33
$\hat{\tau}^{SS}[\boldsymbol{x}, \text{RF}]$	$\hat{\tau}^{RE}$	7	4	22	1.926845	25
$\hat{\tau}^{SS}[\boldsymbol{x}, \text{RF}]$	$\hat{\tau}^{DM}$	3	0	30	1.949483	25
$\hat{\tau}^{RE}$	$\hat{\tau}^{DM}$	2	4	27	1.848485	33

```
compTab%>%select(method1,method2,best2:worstPS)%>%kable(escape=FALSE)
```

method1	method2	best2	best2ps	worst	worstPS
$\hat{\tau}^{\text{SS}}[\tilde{\boldsymbol{x}}, \text{EN}]$	$\hat{\tau}^{\text{SS}}[x^r, \text{LS}]$	1.417204	14	0.9710151	2
$\hat{\tau}^{\text{SS}}[\tilde{\boldsymbol{x}}, \text{EN}]$	$\hat{\tau}^{\text{SS}}[\boldsymbol{x}, \text{RF}]$	1.142609	30	0.9088411	32
$\hat{\tau}^{\text{SS}}[\tilde{\boldsymbol{x}}, \text{EN}]$	$\hat{\tau}^{\text{RE}}$	1.888076	2	0.9692726	30
$\hat{\tau}^{\text{SS}}[\tilde{\boldsymbol{x}}, \text{EN}]$	$\hat{\tau}^{\text{DM}}$	1.927863	25	1.0132275	4
$\hat{\tau}^{\text{SS}}[x^r, \text{LS}]$	$\hat{\tau}^{\text{SS}}[\boldsymbol{x}, \text{RF}]$	1.150106	30	0.5360364	25
$\hat{\tau}^{\text{SS}}[x^r, \text{LS}]$	$\hat{\tau}^{\text{RE}}$	1.315815	4	0.9756321	30
$\hat{\tau}^{\text{SS}}[x^r, \text{LS}]$	$\hat{\tau}^{\text{DM}}$	1.551418	27	1.0111747	4
$\hat{\tau}^{\text{SS}}[\boldsymbol{x}, \text{RF}]$	$\hat{\tau}^{\text{RE}}$	1.914722	2	0.8482977	30
$\hat{\tau}^{\text{SS}}[\boldsymbol{x}, \text{RF}]$	$\hat{\tau}^{\text{DM}}$	1.921804	27	0.9200898	30
$\hat{\tau}^{\text{RE}}$	$\hat{\tau}^{\text{DM}}$	1.506491	27	0.5403895	2

4.1 Comparing Sample Splitting to ANCOVA Estimators

The following creates the figures in 4.3 (plus some others)

This analysis used results from an updated, fully-replicable run of the deep learning model in the remnant.

This estimates the effects and their SEs:

```
exactReplication <- FALSE
source('code/dataPrep.r')
dat <- dat%>%
  group_by(Class.ID,problem_set)%>%
  mutate(
    across(all_of(covNames),~ifelse(is.finite(.),.,mean(.,na.rm=TRUE)))
  )%>%
  group_by(problem_set)%>%
  mutate(
    across(all_of(covNames),~ifelse(is.finite(.),.,mean(.,na.rm=TRUE)))
  )%>%
  ungroup()




ols <- sapply(levels(dat$problem_set),full,dat=dat,
covNames=covNames,simplify=FALSE,
                  methods=c('reloopLin','reloopPoor','reloopPlusLin',
                  'reloopPlusPoor','lin','ancova'))

save(ols,file='results/ols.RData')
```

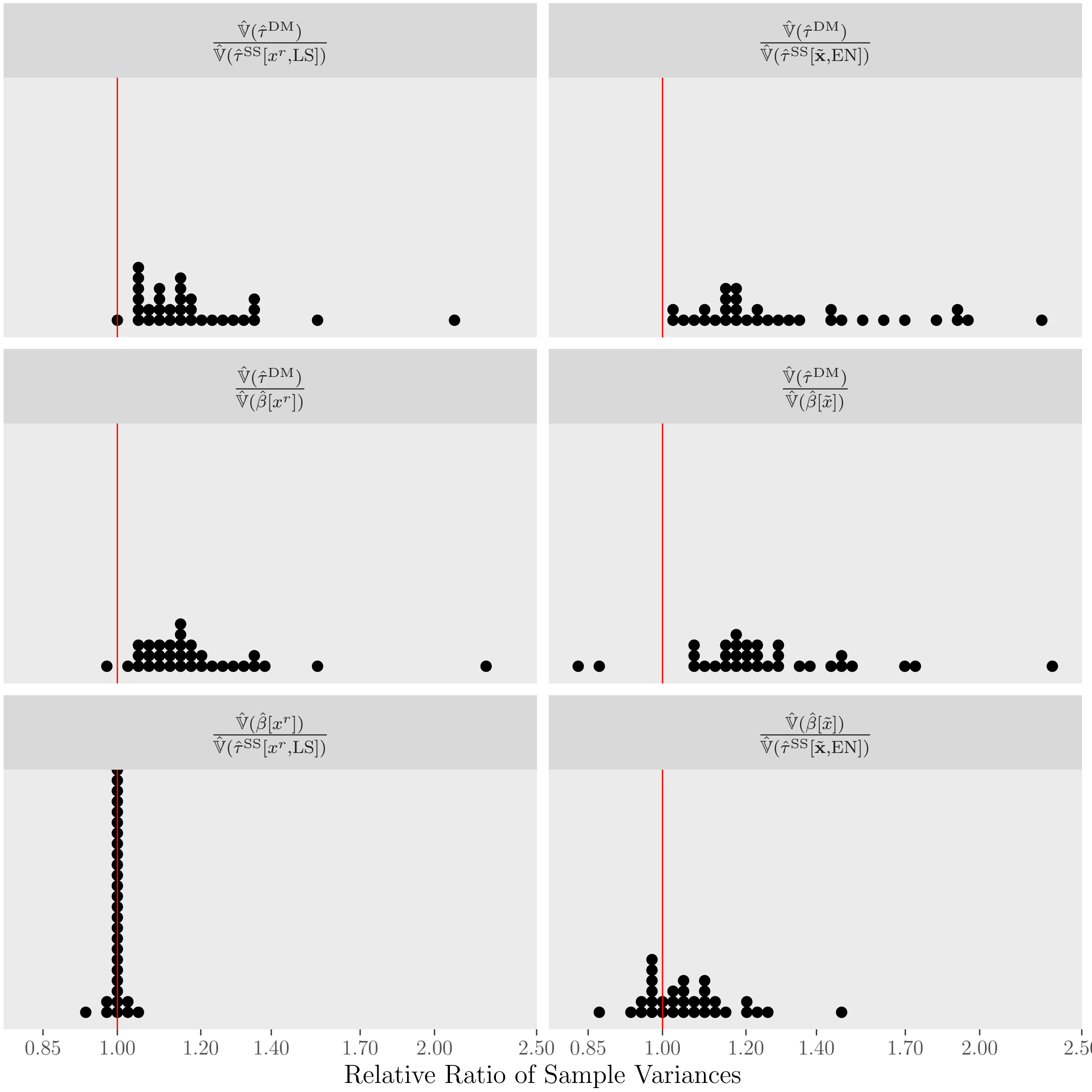
```
source('code/olsFigurePrep.r')
```

```
pOls3 <- ggplot(newcomp,aes(ssMult))+#,fill=exGroup))+
    geom_dotplot( method="histodot", binwidth = .01 )  +
    labs( x = "Relative Ratio of Sample Variances", y="" ) +
    geom_vline( xintercept = 1, col="red" ) +
    facet_wrap(~comp,nrow=3)+
    theme(legend.position = "none",
        panel.grid = element_blank(),
        axis.title.y = element_blank(),
        axis.text.y= element_blank(),
        axis.ticks.y = element_blank(),
        text=element_text(size=12),
        strip.text=element_text(size=12,lineheight=0.5))+
  scale_x_continuous(trans="log10",breaks=c(0.85,1,1.2,1.4,1.7,2,2.5))


tikz('figure/OlsReloop.tex',width=5,height=6,standAlone=FALSE,
  packages= c(getOption('tikzLatexPackages'),
  '\\usepackage{amsmath,amsfonts,amsthm,amssymb,thmtools}'))
print(pOls3)
dev.off()

## tikz output
##           2

print(pOls3)
```




```r
sessionInfo()

## R version 4.2.2 (2022-10-31)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Debian GNU/Linux 11 (bullseye)
##
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.9.0
```

```
## LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.9.0
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] forcats_0.5.2         estimatr_1.0.0        tikzDevice_0.12.3.1
##  [4] knitr_1.40            xtable_1.8-4          kableExtra_1.3.4
##  [7] loop.estimator_1.0.0  tidyr_1.2.0           purrr_0.3.4
## [10] tibble_3.1.8          ggplot2_3.4.1         dplyr_1.0.10
## [13] scales_1.2.1          languageserver_0.3.15 httpgd_1.3.1
##
## loaded via a namespace (and not attached):
##  [1] tinytex_0.41        tidyselect_1.1.2    xfun_0.32
##  [4] colorspace_2.0-3    vctrs_0.5.2         generics_0.1.3
##  [7] viridisLite_0.4.1   htmltools_0.5.3     utf8_1.2.2
## [10] rlang_1.0.6         later_1.3.0         pillar_1.8.1
## [13] glue_1.6.2          withr_2.5.0         DBI_1.1.3
## [16] lifecycle_1.0.3     stringr_1.4.1       munsell_0.5.0
## [19] gtable_0.3.0        rvest_1.0.3         evaluate_0.16
## [22] labeling_0.4.2      callr_3.7.3         fastmap_1.1.0
## [25] ps_1.7.1            parallel_4.2.2      fansi_1.0.3
## [28] highr_0.9           Rcpp_1.0.10         webshot_0.5.4
## [31] filehash_2.4-3      jsonlite_1.8.0      farver_2.1.1
## [34] systemfonts_1.0.4   digest_0.6.29       stringi_1.7.12
## [37] processx_3.7.0      grid_4.2.2          cli_3.6.0
## [40] tools_4.2.2         magrittr_2.0.3      randomForest_4.7-1.1
## [43] Formula_1.2-4       pkgconfig_2.0.3     ellipsis_0.3.2
## [46] xml2_1.3.3          svglite_2.1.0       assertthat_0.2.1
## [49] rmarkdown_2.16      httr_1.4.5          rstudioapi_0.14
## [52] R6_2.5.1            compiler_4.2.2
```