This document includes code to produce all of the results and run all of the models reported in "The Role of Mastery Learning in Intelligent Tutoring Systems: Principal Stratification on a Latent Variable."

The auxilliary files sourced here are available at our github repository, `https://github.com/adamSales/ctaiAdvance`.

First, load in and transform the (pre-imputed) data:

```
load('data/HSdata.RData')
load('data/advanceData.RData')
```

We'll use the `R` package `rstan` to run the models:

```
library(rstan)
rstan_options(auto_write = TRUE)
options(mc.cores = parallel::detectCores())
```

# 1 Data Description (Section ??)

This code produces the missigness information from Table 1, summarizing the student level data:

```
miss <- NULL
for(i in c('race','sex','spec','xirt')) miss <- rbind(miss,
 c(sum(is.na(covs[[i]])),mean(is.na(covs[[i]])),error[i,'error']))
miss <- as.data.frame(miss)
miss$`Error Type` <- c('PFC','PFC','PFC','SRMSE')
rownames(miss) <- c('Race/Ethnicity','Sex','Special Education','Pretest')
names(miss)[1:3] <- c('# Missing','% Missing','Imputation Error')
miss[,2] <- as.integer(round(miss[,2]*100))
miss[,1] <- as.integer(miss[,1])
miss['Pretest','Imputation Error'] <- sqrt(miss['Pretest','Imputation Error'])/sd(covs$xirt


print(xtable::xtable(miss))
```

| | # Missing | % Missing | Imputation Error | Error Type |
|---|---|---|---|---|
| Race/Ethnicity | 1071 | 8 | 0.23 | PFC |
| Sex | 526 | 4 | 0.35 | PFC |
| Special Education | 199 | 1 | 0.11 | PFC |
| Pretest | 2367 | 18 | 0.20 | SRMSE |

This code produces the covariate balance information:

```
covBal <- NULL
for(i in c('race','sex','spec')){
    covBal <- rbind(covBal,c(i,NA,NA,NA,NA))

    for(ll in levels(dat[[i]])){
        covBal <- rbind(covBal,c(NA,ll,round(c(mean(dat[[i]]==ll),mean(dat[[i]][dat$treatmen

    }
}
colnames(covBal) <- c('Covariate','Category','Overall Percent','Percent of Treated','Percent
print(xtable(covBal),floating=FALSE,include.rownames=FALSE)
```

| Covariate | Category | Overall Percent | Percent of Treated | Percent of Control |
|---|---|---|---|---|
| race | | | | |
| | WhiteAsian | 0.49 | 0.48 | 0.49 |
| | BlackMulti | 0.33 | 0.3 | 0.35 |
| | HispAIAN | 0.18 | 0.22 | 0.16 |
| sex | | | | |
| | F | 0.5 | 0.49 | 0.5 |
| | M | 0.5 | 0.51 | 0.5 |
| spec | | | | |
| | typical | 0.9 | 0.89 | 0.9 |
| | speced | 0.07 | 0.08 | 0.07 |
| | gifted | 0.03 | 0.04 | 0.03 |

The overall p-value for balance is:

```
library(RItools) ## using development version
balMod <- balanceTest(treatment~poly(xirt,2)+spec+race+sex+strata(pair)+cluster(schoolid2),d
print(balMod$overall['pair',])

## chisquare        df   p.value
##    8.4464    7.0000    0.2949
```

## 2 PS Model with $\bar{m}_T$

Here we estimate the model in Section **??** stratifying on $\bar{m}_T$.

First, we create the datasets:

```
source('R/prelimMbar.r')
```

The model is encoded in the file `psmodObs.stan`. It may be summarized as follows. The model for $\bar{m}_T$ is:

$$\bar{m}_{Ti} = \alpha_s^U + \boldsymbol{x}_i^T \boldsymbol{\beta}^{\boldsymbol{U}} + \epsilon_i^{Ui} + \epsilon_{t[i]}^{Ut} \tag{1}$$

where $\alpha_s^U$ is a separate intercept for each state, and $\boldsymbol{x}_i$ is a vector of covariates: dummy variables for racial/ethnic category, a dummy variable for sex, dummy variables for special education category, and linear and quadratic terms for pretest. The normally-distributed errors $\epsilon^{Ui}$ and $\epsilon_{t[i]}^{Ut}$ vary at the individual and teacher levels.

The model for $Y$ is

$$Y_i = \alpha_p^Y + \boldsymbol{x}_i^T \boldsymbol{\beta^Y} + a_1 \bar{m}_{Ti} + Z_i(b_0 + b_1 * \bar{m}_{Ti}) + \epsilon_i^{Yi} + \epsilon_{t[i]}^{Yt} + \epsilon_{s[i]}^{Ys} \qquad (2)$$

where $\alpha_p^Y$ is a separate intercept for each randomization block $p$, $Z_i$ is a dummy variable for treatment status, $\epsilon_{s[i]}^{Ys}$ is a normally distributed error at the school level, and the rest of the variables are analogous to those in (1). We run the model with the `stan` command from `rstan`:

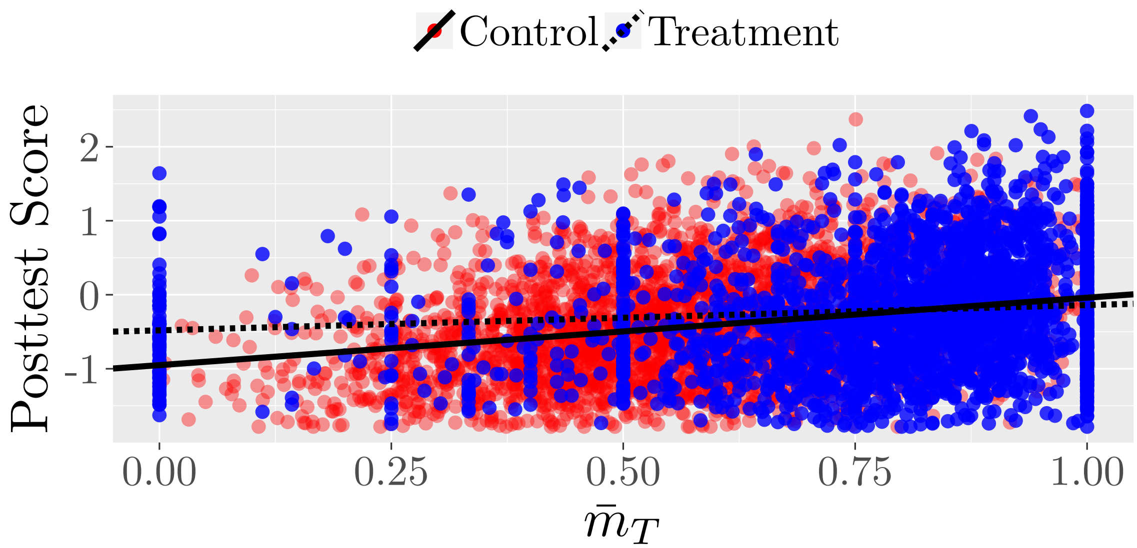
```
mbarMod <- stan('R/psmodObs.stan',data=sdatObs,seed=613)
```

Figure **??** can be replicated with the following code:

```
library(tikzDevice) ## allows latex code in figure
options( tikzLatexPackages = c(
getOption( "tikzLatexPackages" ),
"\\usepackage{amsmath,amsfonts}"
))

draw <- 1000

samps <- extract(mbarMod)
plotDatObs <- with(sdatObs,data.frame(Y=c(YtO,YtM,Yc),mbar=c(MbarTO,samps$MbarTM[draw,],samp
plotDatObs$treat <- ifelse(plotDatObs$Z==1,'Treatment','Control')
plotDatObs$slope <- ifelse(plotDatObs$treat=='Control',samps$a1[draw],samps$a1[draw]+samps$b
plotDatObs$int <- ifelse(plotDatObs$treat=='Control',samps$a0[draw],samps$a0[draw]+samps$b0[

plotDatObs <- within(plotDatObs, int <- int-( mean(int+slope*mbar)-mean(plotDatObs$Y)))
plotDatObs <- plotDatObs[order(plotDatObs$treat),]
plotDatObs$treat2 <- plotDatObs$treat

tikz(file = "figure/mbarModel.tex",
  standAlone = T,
  width  = 6, height  = 3)

ggplot(plotDatObs,aes(mbar,Y,fill=treat,group=treat,alpha=treat,color=treat))+geom_point(siz
    geom_abline(aes(intercept=int,slope=slope,linetype=treat2),color='black',size=2,alpha=1)
    scale_colour_manual(values=c('red','blue'))+
    labs(group=NULL,fill=NULL,alpha=NULL)+xlab('$\\bar{m}_T$')+
    ylab('Posttest Score')+theme(legend.position='top',text=element_text(size=20))+
    guides(color = guide_legend(title=NULL,override.aes=list(alpha=1),keywidth=3),linetype=g
dev.off()
```

pdf 2

```r
setwd('figure'); tools::texi2dvi('mbarModel.tex', pdf = T, clean = T); setwd('..')
```




To save memory, save and delete the $\bar{m}_T$ model:

```r
save(mbarMod,file='output/mbarMod.RData')
```

```r
rm(mbarMod); gc()
```

## 3 The Main PS Model

This section reproduces our paper's main model, described in Section **??**.

The data for the main model (similar to the $\bar{m}$ model but including student-section level mastery data) relies on a secondary file (available at github):

```r
source('R/prelimStan.r')
```

Since this is the main model, we will include full stan code in this online supplement:

```
data{
//Sample sizes
 int<lower=1> nsecWorked;
 int<lower=1> ncov;
 int<lower=1> nstud;
 int<lower=1> nteacher;
 int<lower=1> nsec;
 int<lower=1> nschool;
 int<lower=1> npair;

// indices
 int<lower=1,upper=nteacher> teacher[nstud];
 int<lower=1,upper=npair> pair[nstud];
 int<lower=1,upper=nschool> school[nstud];
 int<lower=1,upper=nstud> studentM[nsecWorked];
 int<lower=1,upper=nsec> section[nsecWorked];

// data data
 int<lower=0,upper=1> grad[nsecWorked];
 matrix[nstud,ncov] X;
 int<lower=0,upper=1> Z[nstud];
 real Y[nstud];

}
parameters{

 vector[nstud] studEff;

 vector[ncov] betaU;
 vector[ncov] betaY;

 real a1;
 real b0;
 real b1;

 real teacherEffY[nteacher];
 real teacherEffU[nteacher];
 real pairEffect[npair];
 real schoolEffU[nschool];
 real schoolEffY[nschool];
 real secEff[nsec];

 real<lower=0> sigTchY;
```

```
 real<lower=0> sigSclY;
 real<lower=0> sigY[2];
 real<lower=0> sigTchU;
 real<lower=0> sigSclU;
 real<lower=0> sigU;
}

model{
 real linPred[nsecWorked];
 vector[nstud] muY;
 vector[nstud] muU;
 real useEff[nstud];
 real trtEff[nstud];
 real sigYI[nstud];


// grad model
 for(i in 1:nsecWorked)
  linPred[i]= secEff[section[i]]+studEff[studentM[i]];

 for(i in 1:nstud){
  useEff[i]=a1*studEff[i];
  trtEff[i]=b0+b1*studEff[i];
  muU[i]=teacherEffU[teacher[i]]+schoolEffU[school[i]];
  muY[i]=teacherEffY[teacher[i]]+schoolEffY[school[i]]+pairEffect[pair[i]]+useEff[i]+Z[i]*tr
  sigYI[i]=sigY[Z[i]+1];
 }

 //priors
 betaY~normal(0,2);
 betaU~normal(0,2);
 pairEffect~normal(0,2);

 a1~normal(0,1);
 b0~normal(0,1);
 b1~normal(0,1);


 schoolEffY~normal(0,sigSclY);
 schoolEffU~normal(0,sigSclU);
 teacherEffU~normal(0,sigTchU);
 teacherEffY~normal(0,sigTchY);

 grad~bernoulli_logit(linPred);

 studEff~normal(muU+X*betaU,sigU);
```

```
 Y~normal(muY+X*betaY,sigYI);
}
```

This code runs the model:

```
main <- stan('R/psmod.stan',data =sdat,warmup=1500,chains=10,iter=5000)
save(main,file='output/mainModel.RData')
```

```
draws <- extract(main)

### for "multImp" and "trtEff"
set.seed(613)
U <- draws$studEff
Usamp <- U[sample(1:nrow(U),1000),]

### for sampleSizeEta & etaDiff
draw <- 1000
U <- U[,sort(unique(sdat$studentM))]
eta <- U[draw,]
etasd <- apply(U,2,sd)

### for "usageModel"
sdEta <- sqrt(mean(apply(draws$studEff,1,var)))
Eeta <- colMeans(draws$studEff)

draws$studEff <- Usamp

summMain <- summary(main)[[1]]

save(draws,draw,eta,etasd,sdEta,Eeta,summMain,Usamp,file='output/smallMain.RData')
```

# 4 Multiple Imputation Model Fit

To give some intuition on how the model fitting worked, and to what extent treatment effect moderation was discernable in this dataset anyway, we re-fit the model using (something akin to) multiple imputation. First, extract 1000 random draws of $\eta_T$ (denoted as `studEff` in the model code) from the fitted model. Then, for each draw, fit a standard HLM interacting treatment with the $\eta_T$ draw.

```
library(lme4)
set.seed(613)
## U <- extract(main,'studEff')[[1]]
## Usamp <- U[sample(1:nrow(U),1000),]
```

```
multImp <- apply(Usamp,1,
                 function(u) summary(
                                 lmer(Y~treatment*u+poly(xirt,2)+race+sex+spec+state+(1|scho
                                      data=dat))$coef['treatment:u',])
```

Does this give similar answers to the main model?

```
mean(multImp['Estimate',])
```

[1] -0.02832

```
## use "Rubin's Rule" to estimate SE
sqrt(var(multImp['Estimate',])+mean(multImp['Std. Error',]^2))
```

[1] 0.02489

```
## now for the main model:
## summary(main,par='b1',probs=c())[[1]]
summMain['b1',c(1:3,9:10)]
```

mean se$_measndn_effRhat-0.02940820.00051740.02498412331.57988911.0055037$

## 5 Figures Comparing PS with $\bar{m}_T$ to PS with $\eta$

Figure ??:

```
## smart jittering:
datObs$mbarJ <- datObs$mbar
datObs$nsecJ <- datObs$nsec
tab <- with(datObs,table(mbar,nsec))
mult <- which(tab>1,arr.ind=TRUE)
ms <- sort(unique(datObs$mbar))
ns <- sort(unique(datObs$nsec))
for(i in 1:nrow(mult)){
    w <- which(datObs$mbar==ms[mult[i,'mbar']] & datObs$nsec==ns[mult[i,'nsec']])
    s <- length(w)
    if(s>1){
        width=min(s*0.002,0.01)
        height=min(s*0.2,2)
        datObs$nsecJ[w] <- datObs$nsecJ[w]+runif(s,-width,width)
        datObs$mbarJ[w] <- datObs$mbarJ[w]+runif(s,-width,width)
    }
}
```

```
tikz(file='figure/mbarSampleSize.tex',
     standAlone=T,
     width=3,height=3)
ggplot(datObs,aes(mbarJ,nsecJ))+geom_point()+xlab('$\\bar{m}$')+ylab('$n_{sec}$')+theme(text
dev.off()
```

pdf 2

```
setwd('figure'); tools::texi2dvi('mbarSampleSize.tex', pdf = T, clean = T); setwd('..')
```

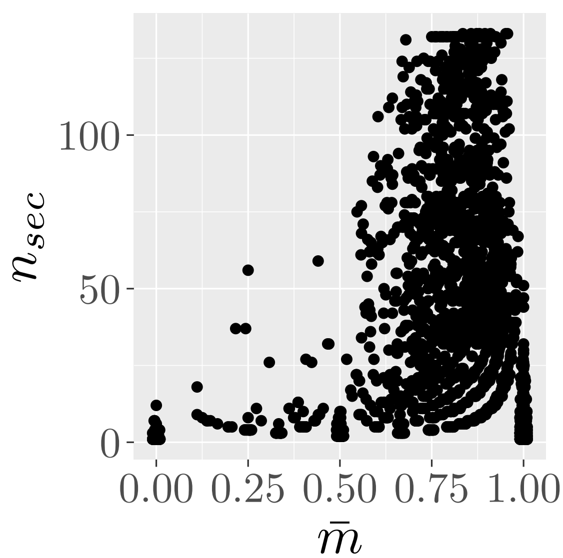



Figure **??**:

```
## draw <- 1000
sdatLat <- sdat
nsec <- as.vector(table(sdatLat$studentM))
## etaDraws <- extract(main,'studEff')[[1]][,sort(unique(sdatLat£studentM))]
## eta <- etaDraws[draw,]
## etasd <- apply(etaDraws,2,sd)

plotDat <- data.frame(nsec=nsec,eta=eta,etasd=etasd)

tikz(file='figure/etaSampleSize.tex',
     standAlone=T,
     width=3,height=3)
ggplot(plotDat,aes(eta,nsec,size=1/etasd))+geom_point()+ylab(NULL)+#ylab('£n_{sec}£')+
    labs(size='$1/\\text{SE}(\\eta_T)$')+scale_size(range=c(.5,2))+guides(size=FALSE)+xlab('
    theme(text=element_text(size=20))+
```

```
    theme(axis.title.y=element_blank(),
        axis.text.y=element_blank(),
        axis.ticks.y=element_blank())#+ggtitle('One Posterior Draw')#+xlab('£\\mathbb{E}\\et
dev.off()
```

pdf 2

```
setwd('figure'); tools::texi2dvi('etaSampleSize.tex', pdf = T, clean = T); setwd('..')
```

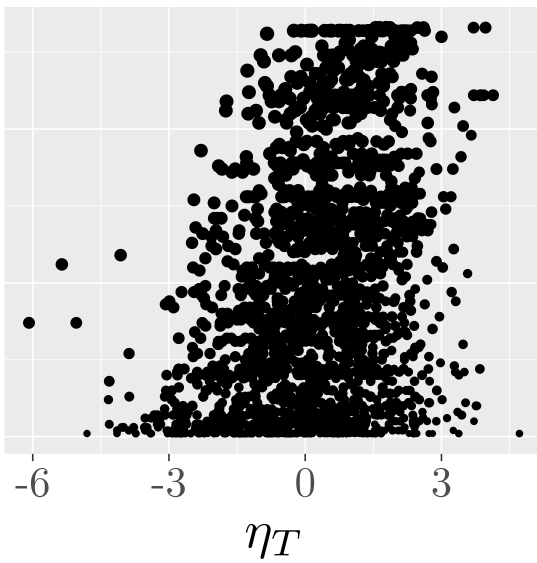



Figure ??:

```
secDiff <- -draws$secEff[draw,]
sss <- secDiff[sdatLat$sec]
mDiff <- aggregate(sss,list(stud=sdatLat$studentM),mean)
mbar <- aggregate(sdatLat$grad,list(sdatLat$studentM),mean)

mbarDiffDat <- data.frame(mbar=mbar$x,mDiff=mDiff$x)


tikz(file='figure/mbarDiff.tex',
     standAlone=T,
     width=3,height=3)
ggplot(mbarDiffDat,aes(mbar,mDiff))+geom_point()+xlab('$\\bar{m}$')+ylab('Avg. Sec. Difficul
    theme(text=element_text(size=20))
dev.off()
```

pdf 2

```
setwd('figure'); tools::texi2dvi('mbarDiff.tex', pdf = T, clean = T); setwd('..')
```

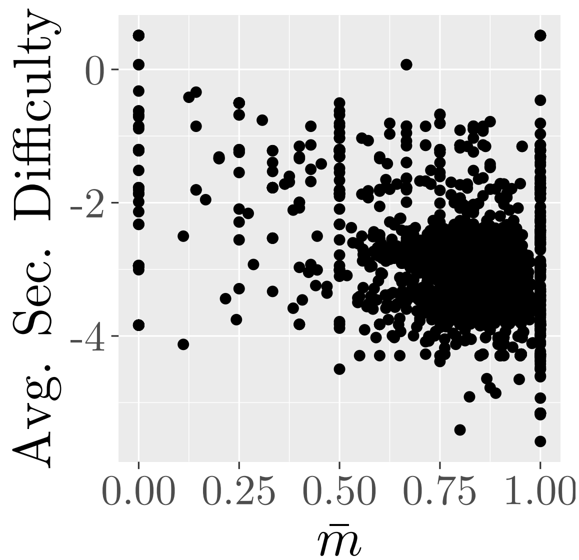
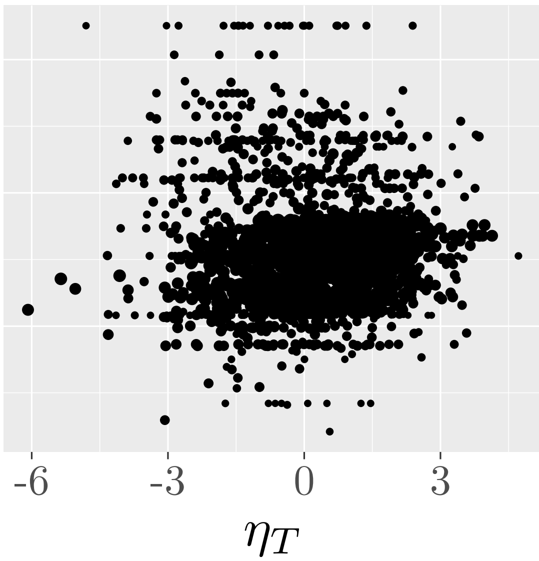



Figure **??**:

```
plotDat$mDiff <- mDiff$x
tikz(file='figure/etaDiff.tex',
     standAlone=T,
     width=3,height=3)
ggplot(plotDat,aes(eta,mDiff,size=1/etasd))+geom_point()+#ylab('£Avg. Section Difficulty£')+
    labs(size='$1/\\text{SE}(\\eta_T)$')+scale_size(range=c(.5,2))+guides(size=FALSE)+xlab('
    theme(text=element_text(size=20))+
    theme(axis.title.y=element_blank(),
        axis.text.y=element_blank(),
        axis.ticks.y=element_blank())#+ggtitle('One Posterior Draw')#+xlab('£\\mathbb{E}\\et
dev.off()
```

pdf 2

```
setwd('figure'); tools::texi2dvi('etaDiff.tex', pdf = T, clean = T); setwd('..')
```

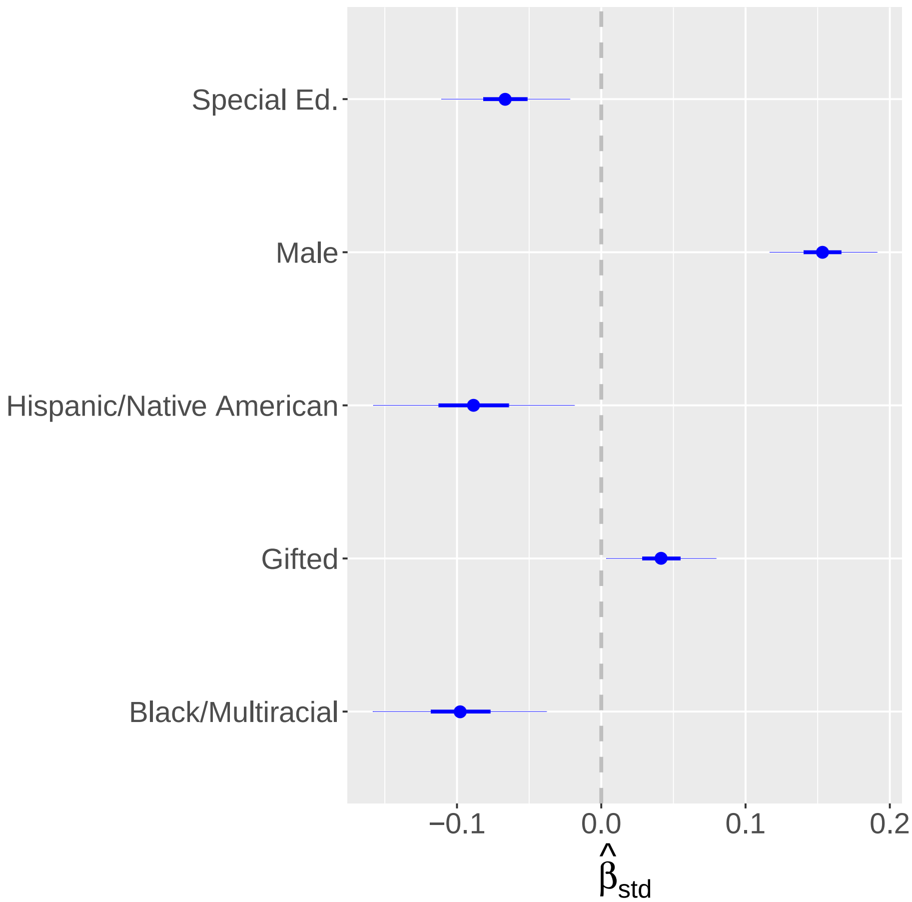



# 6 Main Model Results

Get all the MCMC draws from the main model:

## 6.1 Predicting Mastery

```
library(coefplot)

## coefs <- summary(main,'betaU')[[1]]
coefs <- summMain[grep('betaU',rownames(summMain)),]
rownames(coefs) <- colnames(sdat$X)

## sqrt of average (over the draws) of the variance of eta
## sdEta <- sqrt(mean(apply(draws$studEff,1,var)))

coefs <- coefs/apply(sdat$X,2,sd)/sdEta

cpdf <- data.frame(Value=coefs[3:7,1],Coefficient=c('Black/Multiracial','Hispanic/Native Ame

coefplot.data.frame(cpdf,title=NULL,xlab=expression(hat(beta)[std]),ylab=NULL)+theme(text=el
```




```
ggsave('figure/usageCoef.pdf',width=3,height=3)

pdf('figure/pretestEta.pdf')
omar <- par()$mar
par(mar=omar+c(0,1,-1,0))
## Eeta <- colMeans(draws£studEff)
plot(dat$xirt,Eeta/sdEta,col=ifelse(dat$treatment==1,'blue','red'),
     xlab='Pretest (std)',ylab=expression(paste('E[',eta[T],'|x]')),cex.lab=2)
X <- scale(model.matrix(~poly(xirt,2),data=dat)[,-1])
xpred <- (X[,1]*mean(draws$betaU[,1])+X[,2]*mean(draws$betaU[,2]))/sdEta
set.seed(613)
samp <- sample(1:4000,100)
for(ss in samp){
    xpredS <- (X[,1]*draws$betaU[ss,1]+X[,2]*draws$betaU[ss,2])/sdEta
    lines(sort(dat$xirt),xpredS[order(dat$xirt)],col=adjustcolor('pink',0.5))
}
```

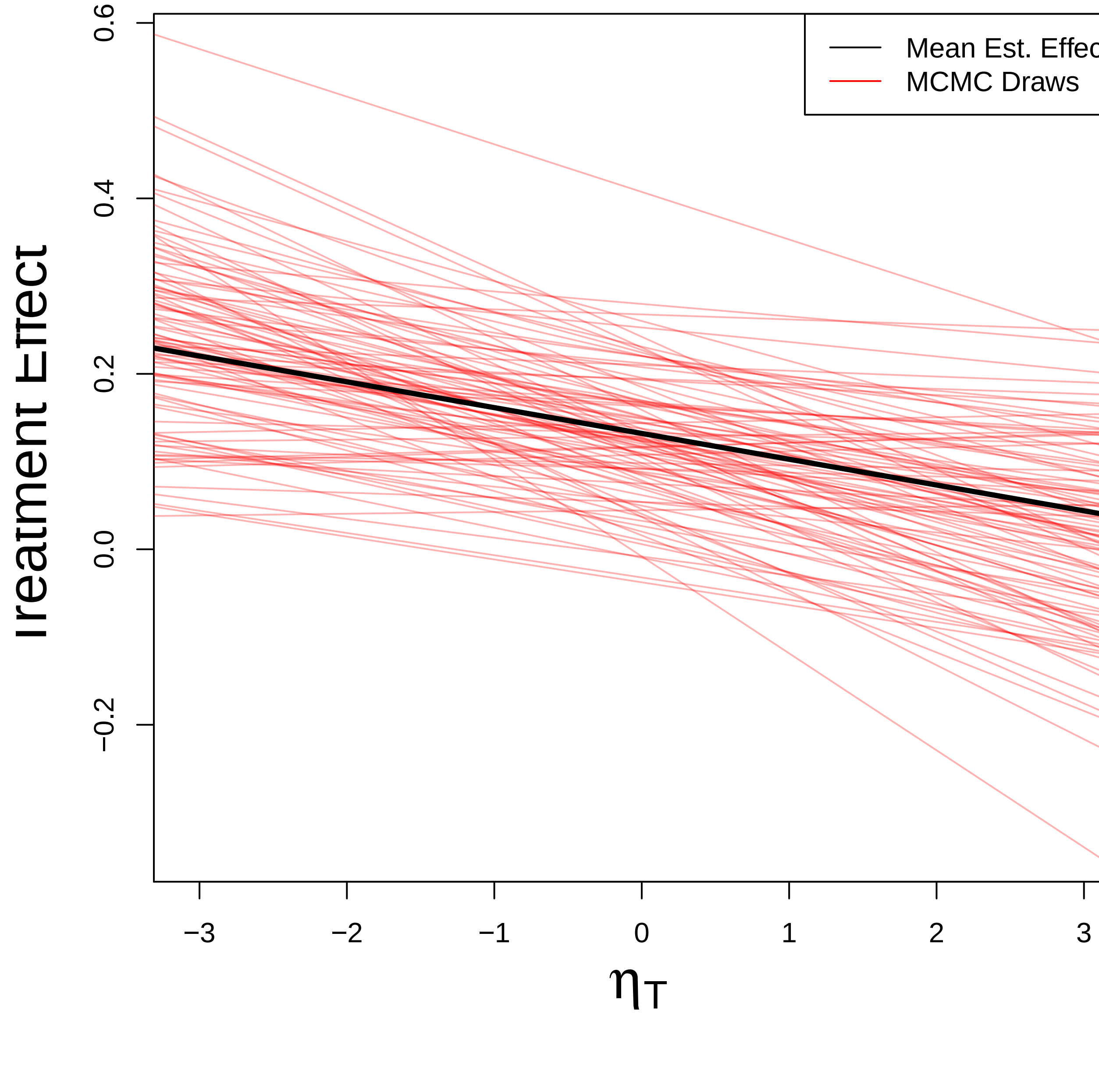
```
lines(sort(dat$xirt),xpred[order(dat$xirt)],lwd=2)
legend('bottomright',legend=c('Treated','Control (Imputed)','Model (Avg.)','Model (draws)'),
dev.off()
```

pdf 2

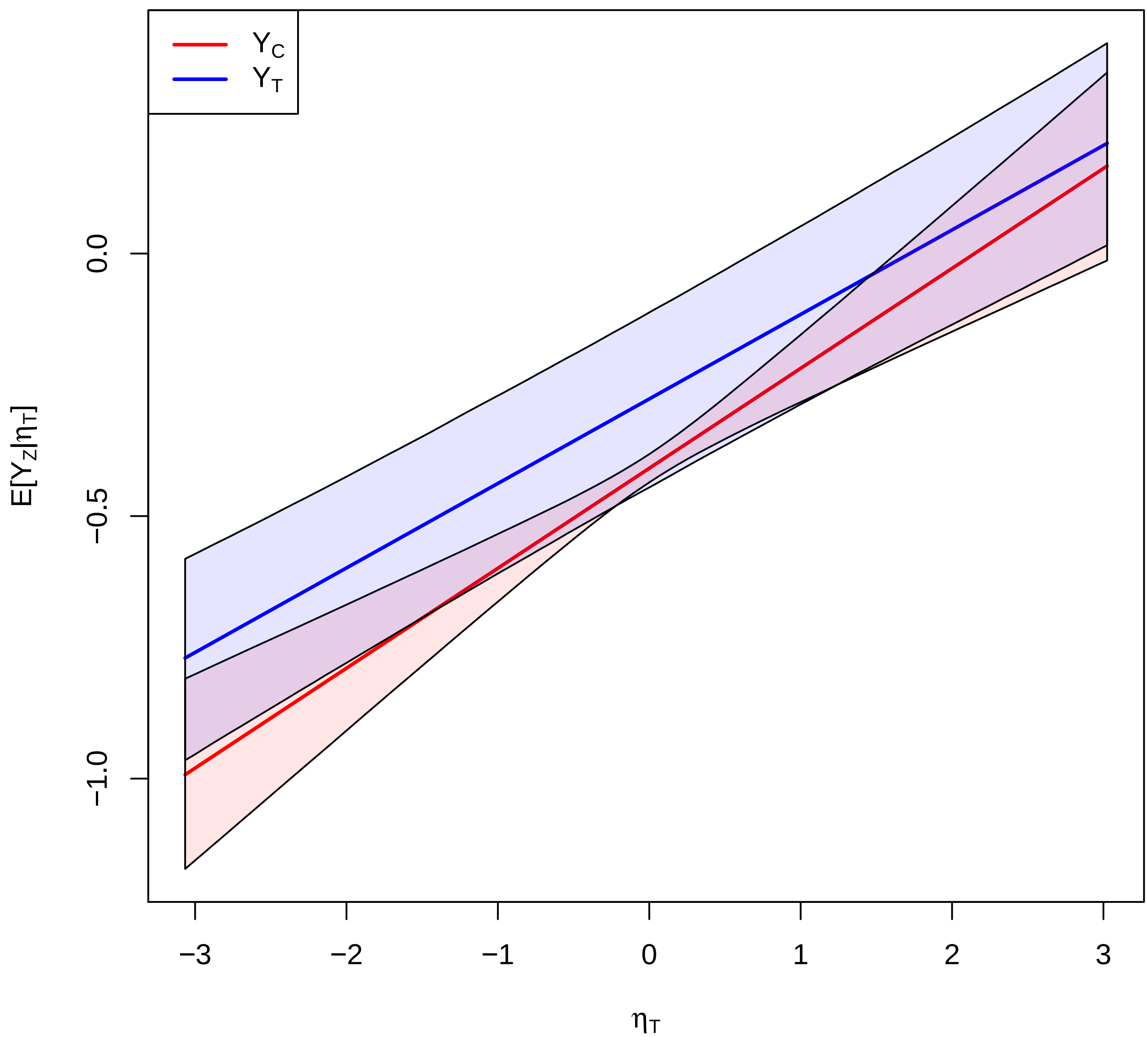
## 6.2 CTAI Treatment Effects

```
pdf('figure/treatmentEffects.pdf')
set.seed(613)
## samp <- sample(1:nrow(draws£studEff),1000)
## Usamp <- draws£studEff[samp,]
studEff95 <- quantile(Usamp,c(0.025,0.975))
Usamp[Usamp<studEff95[1] | Usamp>studEff95[2]] <- NA
trtEff <- sweep(sweep(Usamp,1,draws$b1[samp],'*'),1,draws$b0[samp],'+')

plot(1,1,type='n',ylim=range(trtEff,na.rm=TRUE),
      xlim=studEff95,xlab=expression(eta[T]),ylab='Treatment Effect',cex.lab=2)
sapply(samp,function(rr) invisible(abline(draws$b0[rr],draws$b1[rr],col=adjustcolor('red',.3
abline(mean(draws$b0),mean(draws$b1),lwd=3)
legend('topright',legend=c('Mean Est. Effect','MCMC Draws'),lty=1,col=c('black','red'))
dev.off()

pdf('figure/potentialOutcomes.pdf')
a0 <- rnorm(length(draws$a1),mean(sdat$Y[sdat$Z==0]),sd(sdat$Y[sdat$Z==0])/sqrt(sum(sdat$Z==
a1 <- draws$a1
b0 <- draws$b0
b1 <- draws$b1


xx <- seq(studEff95[1],studEff95[2],length=100)
Yc <- outer(a1,xx)
Yc <- sweep(Yc,1,a0,'+')

YcUp <- apply(Yc,2,function(x) quantile(x,0.975))
YcDown <- apply(Yc,2,function(x) quantile(x,0.025))


Yt <- outer(a1+b1,xx)
Yt <- sweep(Yt,1,a0+b0,'+')
YtUp <- apply(Yt,2,function(x) quantile(x,0.975))
YtDown <- apply(Yt,2,function(x) quantile(x,0.025))
```

```
curve(mean(a0)+mean(a1)*x,from=min(xx), to=max(xx),lwd=2,col='red',xlab=expression(eta[T]),y

curve(mean(a0)+mean(b0)+(mean(b1)+mean(a1))*x,add=TRUE,lwd=2,col='blue')
polygon(c(xx,rev(xx)),c(YcUp,rev(YcDown)),col=adjustcolor('red',0.1))
polygon(c(xx,rev(xx)),c(YtUp,rev(YtDown)),col=adjustcolor('blue',0.1))

 legend('topleft',legend=c(expression(Y[C]),expression(Y[T])),col=c('red','blue'),lwd=2)
 dev.off()
```



Mean Est. Effec
MCMC Draws
Treatment Effect
0.6
0.4
0.2
0.0
−0.2
−3
−2
−1
0
1
2
3
$\eta_T$



Y_C
Y_T
0.0
−0.5
−1.0
E[Y_Z|η_T]
−3
−2
−1
0
1
2
3
η_T