Table IX:

```
##############################
## Set Parameters        #####
##############################
#################################################################
hsig_list=c(0,0.125,0.25,0.5,1.0,2.0,4.0);
nj_list=c(20, 80);
al1_list=c(0, 1);
Simn=10000;
#################################################################
Bias=50
ind=0
model=1
#################################################################



#################################################################
# Import 'metafor' Package and set seed
#################################################################
library(tcltk)
library(metafor)
set.seed(7);

#################################################################
# Generating data for primary data
#################################################################
dgp <-function(al1, sigh,ind){
  #DGP: refer to step 1 in page 25;
  obs=sample(c(62,125,250,500,1000),1);
  x1=runif(obs, min = 100, max = 200);
  x2=x1+rnorm(obs, mean = 0, sd = 50);
  x3=x1+rnorm(obs, mean = 0, sd = 50);
  if (ind==0){
    z = 100 + al1*x1 + 0.5*x2 + rnorm(1,mean=0,sd=sigh)*x3+rnorm(obs,mean=0,sd=100);
  } else if (ind==1) {
    z = 100 + (al1+rnorm(1,mean=0,sd=sigh))*x1 + 0.5*x2 + rnorm(obs,mean=0,sd=100);}
  return (as.data.frame(cbind(z, x1 ,x2)))
}
```

```
#################################################################
# Generating data for MRA study
#################################################################
dtcollection <- function(al1, sigh, ssize, Bias,ind){
  output =  matrix(0, nrow=ssize, ncol=5);
  colnames(output) <- c("id","y","al_se","mj","Significant");
  num_publ=ssize*(Bias/100);
  for(i in 1:ssize) {
    output[i,1]=i;
    if (i<=num_publ){
      while (output[i,5]==0){
        data=dgp(al1,sigh,ind);
        model_selection=sample(c(1.1, 1.2),1);
        if (model_selection==1.1) {
          out <- lm(data[,1] ~ data[,2] + data[,3]);
          output[i,4]=0;
        } else if (model_selection==1.2){
          out <- lm(data[,1] ~ data[,2])
          output[i,4]=1;
        } else { cat("Model Selection Error", "\n"); }
        output[i,2]=coefficients(out)[2];
        output[i,3]=sqrt(diag(vcov(out)))[2];
        output[i,5]=((summary(out)$coefficients[2,4]<=0.05)*(0<=summary(out)$coefficients
            [2,1]));
      }
    } else if (i>num_publ){
      data=dgp(al1,sigh,ind);
      model_selection=sample(c(1.1, 1.2),1);
      if (model_selection==1.1) {
        out <- lm(data[,1] ~ data[,2] + data[,3])
        output[i,4]=0;
      } else if (model_selection==1.2){
        out <- lm(data[,1] ~ data[,2]);
        output[i,4]=1;
      } else { cat("Model Selection Error", "\n"); }
      output[i,2]=coefficients(out)[2];
      output[i,3]=sqrt(diag(vcov(out)))[2];
      output[i,5]=((summary(out)$coefficients[2,4]<=0.05)*(0<summary(out)$coefficients[2,1])
          );
    } else { cat("Publication Bias Error", "\n"); }
  }
  return(output)
}
#################################################################
#################################################################
```

```
88  ##################################################################
89  # Simulation Begins
90  ##################################################################
91  if(model==0){
92    mdl=y~mj;
93  }else if (model==1) {
94    mdl=y~al_se+mj;
95  }else if (model==2) {
96    mdl=y~al_se2+mj;
97  }
98
99
100 nrow=length(hsig_list)*length(nj_list)*length(al1_list)
101 output_third  =  matrix(0, nrow=nrow, ncol=11);
102 colnames(output_third) <- c("MRA_Size","h_sigma","TrueEffect","FE_MRA","RE_MRA","WLS_MRA","
      I2","RE-Bias","WLS-Bias", "RE-MSE", "WLS-MSE");
103 cnt=1;
104 start.time <- Sys.time()
105
106
107 for (k in al1_list){ al1=k;
108 for (j in nj_list) { nj=j;
109 for (l in hsig_list){
110   hsig=l;
111   output_second =  matrix(0, nrow=Simn, ncol=8);
112   colnames(output_second) <- c("FE-MRA","RE_MRA","WLS-MRA", "I2","RE_Coeff","WLS_Coeff","RE-
        Bias","WLS-Bias");
113   cat("Preparing row", cnt, "/", nrow, "\n");
114   pc=0;
115   for(i in 1:Simn) {
116     flag=TRUE
117     while (flag==TRUE){
118       flag=FALSE
119       MRAdata=as.data.frame(dtcollection(al1, hsig, nj, Bias, ind));
120       al_se2=MRAdata$al_se*MRAdata$al_se;
121       test=try(rma.uni(mdl,vi=al_se*al_se, intercept=TRUE, data=MRAdata, weighted=TRUE,
            method="REML", level=95, digits=5), silent =TRUE)
122       if(summary(test)[2]=="try-error") {flag=TRUE; }
123       if(flag==FALSE) { if(is.numeric(test$R2)==FALSE) {flag=TRUE;}}
124     }
125
126     reg_fe=rma.uni(mdl,vi=al_se*al_se, intercept=TRUE, data=MRAdata, weighted=TRUE, method="
          FE", level=95, digits=5)
127     output_second[i,1]=(reg_fe$ci.lb[1]<al1)*(al1<reg_fe$ci.ub[1])
128
```

```
    reg_re=rma.uni(mdl, vi=al_se*al_se, intercept=TRUE, data=MRAdata, weighted=TRUE, method="
        REML", level=95, digits=5)
    output_second[i,2]=(reg_re$ci.lb[1]<al1)*(al1<reg_re$ci.ub[1])
    output_second[i,5]=(coefficients(reg_re)[1]);
    output_second[i,7]=(coefficients(reg_re)[1])-al1;

    reg_wls=lm (mdl, data=MRAdata,weights=(1/(al_se*al_se)))
    output_second[i,3]=(confint(reg_wls)[1,1]<al1)*(al1<confint(reg_wls)[1,2])
    output_second[i,6]=(coefficients(reg_wls)[1]);
    output_second[i,8]=(coefficients(reg_wls)[1])-al1;

    output_second[i,4]=reg_re$I2;
    pc=pc+1;
    if((i-1)%%(Simn/10)==0){cat(10*(i-1)/(Simn/10), ".. ") }
  }
  output_third[cnt,]=c(nj, hsig, al1, mean(output_second[,1]), mean(output_second[,2]), mean
      (output_second[,3]), mean(output_second[,4])/100, abs(mean(output_second[,5])-al1),
      abs(mean(output_second[,6])-al1),mean(output_second[,7]*output_second[,7]),mean(output
      _second[,8]*output_second[,8]))

  cat("Done!", "\n");
  cnt=cnt+1;
}}}
output<-output_third
#####################################################################
#####################################################################
#####################################################################
end.time <- Sys.time()
time.taken <- end.time - start.time
time.taken

round(output,4)
```