```
##################################################################################
############################################

################################################ DATA HARMONIZATION PRACTICAL
################################################

##################################################################################
############################################


#####################

# PRELIMINARY STEPS #

#####################

##If you haven't already installed the necessary packages with libraries, please do so! For this
practical, you will need:

#install.packages("metafor")

#install.packages("plyr")

#install.packages("meta")

#install.packages("rmeta")

library(metafor)

library(plyr)

library(meta)

library(rmeta)


# Set working directory

setwd("") # Set the file path to your local directory


################################################################# PART 1
#################################################################


###################################################################

# UNDERSTANDING DATA AND DEFINING OUR GENETIC INSTRUMENTS FOR BMI #

###################################################################
```

1. We will be using results from the Locke et al. 2015 paper using data from the GIANT Consortium (downloaded from https://portals.broadinstitute.org/collaboration/giant/index.php/GIANT_consortium_data_files)

Note: If you want to have a look at the full GIANT data, then download, save to your working directory and load in the full results (this is a fairly large file) using the following code:

```
# giant_full <- read.table("./GIANT_raw.uniq", header = T)
```

In the Locke et al. paper, the authors describe a certain number of SNPs that are "approximately independently associated with BMI" across all ancestries. Let's have a look at these SNPs.

Note: This file was generated by taking the first rows from the Supplementary Table 8 of the Locke et al. paper

```
snps <- read.csv("./giant_snps_all.csv", header = T)
```

a. How many SNPs do authors describe as being independently associated with BMI in all ancestries?

```
dim(snps)

head(snps)
```

97 SNPs

b. What information does this file contain that are needed for Mendelian randomization analyses?

```
colnames(snps)
```

SNP, Effect allele, Other allele, Beta, SE

c. How many are associated within only Europeans?

```
table(snps$Sig_Analysis)
```

77+3+3+4 = 87 SNPs

d. Why might it be best to use the SNPs that have been identified as being associated with BMI in Europeans only?

Cardiogram contains CHD outcomes on Europeans. In MR we want to use effect sizes that are specific to the population that we are analyzing. If effect estimates differ across populations then this may provide misleading/biased results

2. Read in the second sheet of this file to get the estimates of the SNPs associated with BMI in Europeans that are significant in both sexes.

euro_snps <- read.csv("./giant_snps_euro.csv", header = T)

dim(euro_snps)

head(euro_snps)

a. Check that these are all associated with BMI at a conventional level of genome-wide significance

sort(euro_snps$P)

length(which(euro_snps$P<=5E-8))

b. Are all of these SNPs "good instruments"? What else might we want to check to see if they are strongly and independently associated with BMI?

Replication in an independent dataset

F statistic from the first stage regression

LD (independent of each other)

3. We're going to make sure the effect allele is the allele that increases BMI using the effect allele and beta column

Browse the data, are all SNPs coded so that the effect allele increases BMI?

a. Are all SNP effects in the same direction?

euro_snps[,c("SNP","Effect_Allele","Beta","SE","P")]

summary(euro_snps$Beta)

Yes

PART 2

##

#SNP LOOKUP IN CARDIOGRAM, A GWAS OF CORONARY HEART DISEASE#

##

1. These were downloaded from the CARDIOGRAM website http://www.cardiogramplusc4d.org/ and provided for you (however, this is quite a large file so we have truncated for you using the following code):

```
# CARDIOGRAM <- read.table("./CARDIoGRAM_GWAS_RESULTS.txt", header = T)

# BMI_SNPS_in_CARDIOGRAM <- CARDIOGRAM[CARDIOGRAM$SNP %in% euro_snps$SNP,]

# top_1000 <- CARDIOGRAM[c(1:1000),]

# bottom_1000 <- CARDIOGRAM[c((nrow(CARDIOGRAM)-1000):(nrow(CARDIOGRAM))),]

# CARDIOGRAM_TRUNC <- rbind(BMI_SNPS_in_CARDIOGRAM, top_1000, bottom_1000)

# write.table(CARDIOGRAM_TRUNC, "./CARDIOGRAM_CLEANED.txt", quote = F, col.names = T, row.names = F, sep = "\t")
```

a. How many SNPs are in this truncated CARDIOGRAM dataset?

```
CARDIOGRAM <- read.table("./CARDIOGRAM_CLEANED.txt", sep="\t", header=T, colClasses = "character")

dim(CARDIOGRAM)

head(CARDIOGRAM)
```

2078

b. Does this file contain everything that is required to perform a two-sample Mendelian randomization analysis?

```
colnames(CARDIOGRAM)
```

Yes- SNP, reference allele, other allele, ref allele frequency, log odds, log odds se.

2. How many of the BMI SNPs are included in the CARDIOGRAM dataset?

```
BMI_SNPs <- euro_snps$SNP

BMI_SNPs <- as.vector(BMI_SNPs)

matches <- unique(grep(paste(BMI_SNPs, collapse="|"), CARDIOGRAM$SNP, value=TRUE))

length(matches)
```

77

```
# View the data from CARDIOGRAM for our BMI SNPs

CARDIOGRAM_BMI <- CARDIOGRAM[grepl(paste(BMI_SNPs, collapse="|"), CARDIOGRAM$SNP),]

CARDIOGRAM_BMI
```

3. Merge the GIANT and CARDIOGRAM SNP summary associations

First, make sure the column headings are easy to understand (i.e., add "BMI" and "CHD" onto the respective datasets)

colnames(euro_snps) <- paste("BMI", colnames(euro_snps), sep = "_")

head(euro_snps)

colnames(CARDIOGRAM) <- paste("CHD", colnames(CARDIOGRAM), sep = "_")

head(CARDIOGRAM)

merged <- merge(euro_snps,CARDIOGRAM, by.x="BMI_SNP", by.y="CHD_SNP")

dim(merged)

head(merged)

PART 3

###

#HARMONIZING THE EFFECT ALLELES IN THE GIANT AND CARDIOGRAM DATASETS#

###

1. Make sure that the effect alleles in the CARDIOGRAM and GIANT datasets are the same. We want the cardiogram effect allele to be the allele that increases BMI

But be careful of palindromic SNPs or SNPs on different strands

First we need to see whether the effect alleles are the same

Browse the data

merged[,c("BMI_SNP", "BMI_Effect_Allele","CHD_reference_allele","BMI_Other_Allele","CHD_other_allele","BMI_EAF","CHD_ref_allele_frequency")]

# a.	How can we tell if the CARDIOGRAM and GIANT SNPs are coded using the same reference strand?

The BMI_Effect_Allele and the CHD_reference_allele should be the same, and likewise the BMI_Other allele and CHD_other_allele should be the same

b. Are CARDIOGRAM and the GIANT SNPs coded using the same reference strand?

No

c. Are there any palindromic SNPs?

```
palindromic_at<-subset(merged,BMI_Effect_Allele %in% "A" & BMI_Other_Allele %in% "T")
palindromic_ta<-subset(merged,BMI_Effect_Allele %in% "T" & BMI_Other_Allele %in% "A")
palindromic_gc<-subset(merged,BMI_Effect_Allele %in% "G" & BMI_Other_Allele %in% "C")
palindromic_cg<-subset(merged,BMI_Effect_Allele %in% "C" & BMI_Other_Allele %in% "G")
dim(palindromic_at)
dim(palindromic_ta)
dim(palindromic_gc)
dim(palindromic_cg)
```

Yes- three SNPs (rs1558902, rs17001654, rs4256980)

d. How can we tell whether the effect alleles are the same in both datasets for palindromic SNPs (i.e., the allele that increases BMI is the same as the reference allele in CARDIOGRAM)?

Examine the allele frequencies to confirm that they match. For example, the SNP rs17001654 has effect allele G and an allele frequency of 0.153 in the BMI GWAS. The same SNP in the CHD GWAS is indexed by an effect allele C with an allele frequency of 0.837 and therefore the G allele would be about frequency 0.15 also. This suggests that the genotyping has been performed on the same strand in both GWAS. This procedure becomes less reliable as the minor allele frequency approaches 0.5.

2. Make sure the CARDIOGRAM log odds ratio reflects the allele that increases BMI in the GIANT data

```
# First, find the positions of SNPs with different effect alleles
head(merged)
effect_diff <- which(merged$BMI_Effect_Allele != merged$CHD_reference_allele) # The position of SNPs where effect alleles are different
merged[effect_diff,c("BMI_SNP", "BMI_Effect_Allele","CHD_reference_allele","BMI_Other_Allele","CHD_other_allele","BMI_EAF","CHD_ref_allele_frequency")]
```

a. How many SNPs have effect alleles that are coded in the opposite direction?

38 which is about half the alleles (which is what we would expect purely by chance)

b. Where the effect alleles are different, flip the direction of the log odds ratio by multiplying it by minus one

If you want, you can also generate new columns that reflect the effect allele change but this isn't used in the causal estimate

```
merged$CHD_flip_log_odds <- as.numeric(merged$CHD_log_odds) # Make log odds ratio numeric

merged$CHD_log_odds_se <- as.numeric(merged$CHD_log_odds_se) # Make standard error numeric

merged$CHD_flip_log_odds[effect_diff] <- merged$CHD_flip_log_odds[effect_diff]*(-1)

head(merged)

dim(merged)
```

c. Check that all of the effect estimates have been flipped appropriately

```
merged[effect_diff,c("BMI_SNP", "BMI_Effect_Allele","CHD_reference_allele", "CHD_log_odds", "CHD_flip_log_odds")]

merged[-effect_diff,c("BMI_SNP", "BMI_Effect_Allele","CHD_reference_allele", "CHD_log_odds", "CHD_flip_log_odds")]
```

d. Check that the effect allele frequencies are correlated

Check correlation of effect allele frequency between BMI and CARDIOGRAM datasets before harmonising alleles

```
merged$BMI_EAF <- as.numeric(merged$BMI_EAF)

merged$CHD_ref_allele_frequency <- as.numeric(merged$CHD_ref_allele_frequency)

cor(merged$BMI_EAF,merged$CHD_ref_allele_frequency)
```

Check correlation of effect allele frequency between BMI and CARDIOGRAM datasets after harmonising alleles

```
merged$CHD_ref_allele_frequency[effect_diff] <- 1-merged$CHD_ref_allele_frequency[effect_diff]

cor(merged$BMI_EAF,merged$CHD_ref_allele_frequency)
```

e. What happened and why?

The choice of effect allele was random before harmonizing the alleles and so the correlation in allele frequency should be about zero. After harmonizing the alleles, the correlation is a lot better.

```
####################################################################### PART 4
#######################################################################

######################################
#ESTIMATE THE EFFECT OF BMI ON CHD#
######################################

### 1. Estimate the Wald ratios for each SNP and their delta approximated standard errors

gp <- merged$BMI_Beta # The effect of the SNP on BMI

segp <- merged$BMI_SE # The standard error of the SNP effect on BMI

gd <- merged$CHD_flip_log_odds # The log odds ratio for CHD (that were harmonized to reflect an increase in BMI)

segd <- merged$CHD_log_odds_se # Standard error of the log odds ratio

wald_ratio <- gd/gp # The log odds ratios of CHD per unit change in BMI

Cov <- 0 # Only required when the SNP-BMI and SNP-CHD associations are estimated in the same participants (therefore for two-sample MR with non-overlapping samples, this is set to 0)

wald_ratio_se <- sqrt((segd^2/gp^2) + (gd^2/gp^4)*segp^2 - 2*(gd/gp^3)*Cov) # Delta approximated standard error of the wald ratio; see Thomas, D. C., Lawlor, D. a, & Thompson, J. R. (2007). Re: Estimation of bias in nongenetic observational studies using Mendelian triangulation by Bautista et al. Annals of Epidemiology, 17(7), 5113. doi:10.1016/j.annepidem.2006.12.005

z <- wald_ratio/wald_ratio_se # Z statistic for the wald ratio

p <- 2*pnorm(abs(z) ,lower.tail=F) # P value for the z statistics under the null hypothesis that there is not effect

wald_ratio_var = wald_ratio_se^2 # Variance

weight <- 1/wald_ratio_var # Inverse variance weight

snps <- merged$BMI_SNP # SNPs that we will use in the estimates


### 2. Combine the Wald ratios by fixed effects meta-analysis

meta_results <- metagen(wald_ratio,wald_ratio_se,comb.fixed=T,sm="OR") #combine the SNPs by fixed effects meta-analysis

meta_results
```

```
### 3. Create a nice table of the results, which you could export to other programs e.g. excel, STATA etc

mr_results <- data.frame(matrix(c(as.character(merged$BMI_SNP),round(wald_ratio,2),round(wald_ratio_se,2),round(p,3)),nrow=length(merged$BMI_SNP),ncol=4))

names(mr_results) <- c("SNP","log_odds","se","p")

mr_results_order <- mr_results[order(mr_results$log_odds),]

overall_genetic_effect <- data.frame(matrix(c("Overall genetic effect",meta_results$TE.fixed,meta_results$seTE.fixed,meta_results$pval.fixed),nrow=1,ncol=4))

names(overall_genetic_effect) <- c("SNP","log_odds","se","p")

overall_genetic_effect

twosampleResults <- rbind.fill(mr_results_order,overall_genetic_effect)

twosampleResults


# Calculate heterogeneity p value

p_het <- pchisq(meta_results$Q,meta_results$df.Q,lower.tail=F)

twosampleResults$p_chi<-NA


# Add the p value for heterogeneity to the results table

twosampleResults$p_chi[twosampleResults$SNP=="Overall genetic effect"] <- p_het

write.table(twosampleResults,"./twosample_results_BMI_CHD.txt",sep="\t",col.names=T,row.names=F,quote=F)

twosampleResults


### 4. Create a forest plot of the results and compare the genetic and observational associations

# The observational effect is 1.23 (95% CI: 1.17, 1.29) per 4.56 kg/m2 (i.e., per SD) increase in BMI

# Formula for SE from 95% confidence interval: (log(uci)-log(lci))/(1.96*2)

effect <- c(wald_ratio[order(wald_ratio)],meta_results$TE.fixed,log(1.23))

se <- c(wald_ratio_se[order(wald_ratio)],meta_results$seTE.fixed,(log(1.29)-log(1.17))/(1.96*2))

snps <- c(as.character(merged$BMI_SNP)[order(wald_ratio)],"Overall genetic effect","Observational effect")

dev.off()
```

```
metaplot(effect,se,labels=snps,conf.level=0.95,logeffect=T,nn=0.1,boxsize=1,xlab="Odds ratio & 95% confidence interval",ylab="SNP",cex=0.7)
```

```
### 5. Interpret the results

# a. Is the MR-derived effect similar to the observational association?

# b. Is there evidence of heterogeneity in the genetic effects? How do you interpret this?

# c. Can you think of reasons for caution?
```

Point estimate for the MR derived effect overlaps with the confidence intervals for the observational association. However, substantial heterogeneity in causal effect estimates across the different SNPs. This suggests potential pleiotropy and so caution is warranted.

```
################################################################# PART 5 #################################################################


######################

#SENSITIVITY ANALYSES#

######################

## All that is required is summary level results for each SNP (remember gp, segp, gd, segd from PART 4)

#gp <- merged$BMI_Beta # The effect of the SNP on BMI

#segp <- merged$BMI_SE # The standard error of the SNP effect on BMI

#gd <- merged$CHD_flip_log_odds # The log odds ratio for CHD (that were harmonized to reflect an increase in BMI)

#segd <- merged$CHD_log_odds_se # Standard error of the log odds ratio


### 1. These functions define the IVW, MR-Egger, weighted median and weighted mode estimators, respectively, and a function that wraps up the results

#set seed for replication purposes

set.seed(50)

two.sample.iv.ivw <- function(x, y, sigmax, sigmay) {

 beta.ivw.fit  = summary(lm(y~x-1, weights=sigmay^-2))

 beta.ivw.fit.only  = lm(y~x-1, weights=sigmay^-2)

 beta.ivw      = beta.ivw.fit$coef[1,1]
```

```
  beta.se.ivw   = beta.ivw.fit$coef[1,2]/min(beta.ivw.fit$sigma,1)
  beta.df.ivw   = length(y) - 1
  beta.p.ivw    = 2*(1-pt(abs(beta.ivw/beta.se.ivw),beta.df.ivw))
  beta.lower.ivw   = beta.ivw + (-1*qt(df=beta.df.ivw, 0.975)*beta.se.ivw)
  beta.upper.ivw   = beta.ivw + (1*qt(df=beta.df.ivw, 0.975)*beta.se.ivw)

return(list(beta.ivw=beta.ivw,beta.se.ivw=beta.se.ivw,beta.lower.ivw=beta.lower.ivw,beta.upper.iv
w=beta.upper.ivw,beta.t.ivw=beta.ivw/beta.se.ivw,beta.p.ivw=beta.p.ivw,
            beta.ivw.fit.only=beta.ivw.fit.only,beta.df.ivw=beta.df.ivw,beta.ivw.fit=beta.ivw.fit))
}
weighted.median <- function(x, w) {
  N    = length(x)
  ord  = order(x);
  x    = x[ord];
  w    = w[ord];
  Sn   = cumsum(w)
  S_N  = Sn[N]
  Pn   = (100/S_N)*(Sn-w/2)
  if(sort(abs(Pn-50))[1] == 0){M = which(Pn==50); return(x[M])}
  Q  = length(Pn[sign(Pn-50)==-1])
  V1 = Q;  V2 = Q+1
  M  = x[V1] + (50 - Pn[V1])*(x[V2]-x[V1])/(Pn[V2]-Pn[V1])
  return(list(beta.median=M,CumSum.median=Sn,ordX.median=x))
}
weighted.median.boot <- function(x, y, sigmax, sigmay, Nsim, alpha, W) {
  med = NULL
  for (i in 1:Nsim){
    y_boot  = rnorm(length(y), mean=y, sd=sigmay)
    x_boot  = rnorm(length(x), mean=x, sd=sigmax)
    iv_boot = y_boot/x_boot
    run = weighted.median(iv_boot,W)
    med[i]  = run$beta.median
```

```
 }
 lower = Nsim*alpha/2
 upper = Nsim*(1-alpha/2)
 Sort  = sort(med)
 lowerCI = Sort[lower]
 upperCI = Sort[upper]
 se    = sd(med)
 t     = mean(med)/se
 p     = 2*(1-pt(abs(t),length(y)-1))


return(list(beta.se.median=se,beta.lower.median=lowerCI,beta.upper.median=upperCI,beta.t.media
n=t,beta.p.median=p))
}
two.sample.iv.egger <- function(x, y, sigmax, sigmay) {
 egger.fit       = summary(lm(y~x, weights=sigmay^-2))
 df.egger        = length(y) - 2
 beta.egger      = egger.fit$coef[2,1]
 beta.se.egger   = egger.fit$coef[2,2] / min(egger.fit$sigma, 1)
 beta.p.egger    = 2*(1-pt(abs(beta.egger/beta.se.egger),df.egger))
 beta.lower.egger   = beta.egger + (-1*qt(df=df.egger, 0.975)*beta.se.egger)
 beta.upper.egger   = beta.egger + (1*qt(df=df.egger, 0.975)*beta.se.egger)
 alpha.egger     = egger.fit$coef[1,1]
 alpha.se.egger  = egger.fit$coef[1,2] / min(egger.fit$sigma, 1)
 alpha.p.egger   = 2*(1-pt(abs(alpha.egger/alpha.se.egger),df.egger))
 alpha.lower.egger   = alpha.egger + (-1*qt(df=df.egger, 0.975)*alpha.se.egger)
 alpha.upper.egger   = alpha.egger + (1*qt(df=df.egger, 0.975)*alpha.se.egger)

return(list(beta.egger=beta.egger,beta.se.egger=beta.se.egger,beta.lower.egger=beta.lower.egger,b
eta.upper.egger=beta.upper.egger,beta.t.egger=beta.egger/beta.se.egger,beta.p.egger=beta.p.egge
r,

alpha.egger=alpha.egger,alpha.se.egger=alpha.se.egger,alpha.lower.egger=alpha.lower.egger,alpha.
```

```
upper.egger=alpha.upper.egger,alpha.t.egger=alpha.egger/alpha.se.egger,alpha.p.egger=alpha.p.egger))

}

ModeEstimator <- function(x, y, sigmax, sigmay, phi=c(1,0.5,0.25), n_boot=1e4, alpha=0.05) {

  beta <- function(BetaIV.in, seBetaIV.in) {

    s <- 0.9*(min(sd(BetaIV.in), mad(BetaIV.in)))/length(BetaIV.in)^(1/5)

    weights <- seBetaIV.in^-2/sum(seBetaIV.in^-2)

    beta <- NULL

    for(cur_phi in phi) {

      h <- s*cur_phi

      densityIV <- density(BetaIV.in, weights=weights, bw=h)

      beta[length(beta)+1] <- densityIV$x[densityIV$y==max(densityIV$y)]

    }

    return(beta)

  }

  boot <- function(BetaIV.in, seBetaIV.in, beta_Mode.in) {

    beta.boot <- matrix(nrow=n_boot, ncol=length(beta_Mode.in))

    for(i in 1:n_boot) {

      BetaIV.boot      <- rnorm(length(BetaIV.in), mean=BetaIV.in, sd=seBetaIV.in[,1])

      BetaIV.boot_NOME <- rnorm(length(BetaIV.in), mean=BetaIV.in, sd=seBetaIV.in[,2])

      beta.boot[i,1:length(phi)]                    <- beta(BetaIV.in=BetaIV.boot, seBetaIV.in=rep(1, length(BetaIV)))

      beta.boot[i,(length(phi)+1):(2*length(phi))]   <- beta(BetaIV.in=BetaIV.boot, seBetaIV.in=seBetaIV.in[,1])

      beta.boot[i,(2*length(phi)+1):(3*length(phi))] <- beta(BetaIV.in=BetaIV.boot_NOME, seBetaIV.in=rep(1, length(BetaIV)))

      beta.boot[i,(3*length(phi)+1):(4*length(phi))] <- beta(BetaIV.in=BetaIV.boot_NOME, seBetaIV.in=seBetaIV.in[,2])

    }

    return(beta.boot)

  }

  BetaIV   <- y/x

  seBetaIV <- cbind(sqrt((sigmay^2)/(x^2) + ((y^2)*(sigmax^2))/(x^4)), sigmay/abs(x))
```

```
  beta_SimpleMode       <- beta(BetaIV.in=BetaIV, seBetaIV.in=rep(1, length(BetaIV)))
  beta_WeightedMode     <- beta(BetaIV.in=BetaIV, seBetaIV.in=seBetaIV[,1])
  beta_WeightedMode_NOME <- beta(BetaIV.in=BetaIV, seBetaIV.in=seBetaIV[,2])
  beta_Mode <- rep(c(beta_SimpleMode, beta_WeightedMode,
                beta_SimpleMode, beta_WeightedMode_NOME))
  beta_Mode.boot <- boot(BetaIV.in=BetaIV, seBetaIV.in=seBetaIV, beta_Mode.in=beta_Mode)
  se_Mode <- apply(beta_Mode.boot, 2, mad)
  CIlow_Mode <- beta_Mode-qnorm(1-alpha/2)*se_Mode
  CIupp_Mode <- beta_Mode+qnorm(1-alpha/2)*se_Mode
  P_Mode <- pt(abs(beta_Mode/se_Mode), df=length(x)-1, lower.tail=F)*2
  Method <- rep(c('Simple', 'Weighted', 'Simple (NOME)', 'Weighted (NOME)'), each=length(phi))
  Results <- data.frame(Method, phi, beta_Mode, se_Mode, CIlow_Mode, CIupp_Mode, P_Mode)
  colnames(Results) <- c('Method', 'phi', 'Estimate', 'SE', 'CI_low', 'CI_upp', 'P')
  return(Results)
}
MR_output <- function(ivw,egger,median, mode) {
  output = data.frame(matrix(NA, nrow=5, ncol=7))
  names(output) = c("test", "parameter", "estimate", "se", "lower_CI", "upper_CI","p_value")
  output[1:5,1] = c("IVW","MR-Egger","MR-Egger","Weighted_median","Weighted_mode")
  output[1:5,2] = c("beta","beta","alpha","beta","beta")
  output[1,3:7] =
c(IVW$beta.ivw,IVW$beta.se.ivw,IVW$beta.lower.ivw,IVW$beta.upper.ivw,IVW$beta.p.ivw)
  output[2,3:7] =
c(Egger$beta.egger,Egger$beta.se.egger,Egger$beta.lower.egger,Egger$beta.upper.egger,Egger$bet
a.p.egger)
  output[3,3:7] =
c(Egger$alpha.egger,Egger$alpha.se.egger,Egger$alpha.lower.egger,Egger$alpha.upper.egger,Egger
$alpha.p.egger)
  output[4,3:7] =
c(Median$beta.median,MedianBoot$beta.se.median,MedianBoot$beta.lower.median,MedianBoot$
beta.upper.median,MedianBoot$beta.p.median)
  output[5,3:7] = c(Mode$Estimate[Mode$Method=="Weighted (NOME)" &
Mode$phi==1.00],Mode$SE[Mode$Method=="Weighted (NOME)" &
Mode$phi==1.00],Mode$CI_low[Mode$Method=="Weighted (NOME)" &
```

```
Mode$phi==1.00],Mode$CI_upp[Mode$Method=="Weighted (NOME)" & Mode$phi==1.00],Mode$P[Mode$Method=="Weighted (NOME)" & Mode$phi==1.00])

  return(output)

}


### 2. Use the functions to estimate the results

IVW <- two.sample.iv.ivw(gp,gd,segp,segd)

Egger <- two.sample.iv.egger(gp,gd,segp,segd)

Median <- weighted.median(wald_ratio,weight)

MedianBoot <- weighted.median.boot(gp,gd,segp,segd,1000,0.05,weight)

Mode <- ModeEstimator(gp,gd,segp,segd)

sensitivity <- MR_output(IVW,Egger,Median,Mode)

sensitivity

write.table(sensitivity,"./twosample_sensitivity_BMI_CHD.txt",sep="\t",col.names=T,row.names=F,quote=F)
```

###3 Interpret the results

a. How consistent do the estimates of the causal effect look across the different approaches?

The coefficients are reasonably consistent across the different sensitivity analyses

b. Does a p value of p=0.64 for the MR Egger intercept indicate the pleiotropy is not present in the data?

This result suggests that directional pleiotropy is not present in the data (balanced pleiotropy may still be present- i.e. when the pleiotropic effects of SNPs tend to balance out)

c. Should we be concerned that the p value for the slope of MR Egger is > 0.05?

No MR Egger has low power in general- more important is whether the coefficient is similar to that obtained using the other approaches