```
##############################
## Install missing packages ##
##############################

set.seed(47269801)

required.packages <- c("dplyr", "multiwayvcov", "multcomp", "reshape2", "knitr")
packages.missing <- required.packages[!required.packages %in%
installed.packages()[,"Package"]]

if(length(packages.missing) > 0) {install.packages(required.packages,
repo="https://cran.cnr.berkeley.edu/")}
lapply(required.packages, library, character.only = TRUE)

######################
## Define functions ##
######################

## RegTest conducts asymptotic tests from linear model ##

RegTest <- function(equation, clustvars, hypotheses, data) {

   model <- lm(equation, data = data, na.action = na.omit)

   if (missing(clustvars)) model$vcov <- vcov(model)
   else model$vcov <- cluster.vcov(model, cluster = clustvars)

   model$test <- summary(glht(model, linfct = hypotheses, vcov = model$vcov))$test

   numhyp <- length(hypotheses)

   EST <- matrix(nrow = numhyp, ncol = 4)

   for (i in 1:numhyp) {

      EST[i, 1] <- model$test$coefficients[i]
      EST[i, 2] <- model$test$tstat[i]
      EST[i, 3] <- model$test$sigma[i]
      EST[i, 4] <- model$test$pvalues[i]

   }

   colnames(EST) <- c("Estimate", "Tstat", "SE", "P")
```

```
    return(EST)

}

## PermTest returns MC approximations of the exact p-value ##

PermTest <- function(equation, treatvars, clustvars, hypotheses, iterations, data) {

    stopifnot(length(hypotheses) <= 1)

    obsEST <- RegTest(equation, clustvars, hypotheses, data)
    obsStat <- obsEST[1, 2]

    simEST <- matrix(ncol = 4)

    for (i in 1:iterations) {

        simTreat <- data[, treatvars, drop = FALSE]
        simTreat <- simTreat[sample(nrow(simTreat)),]

        simData <- cbind(simTreat, data[, !(names(data) %in% treatvars), drop = FALSE])
        colnames(simData)[1:length(treatvars)] <- treatvars

        simEST <- rbind(simEST, RegTest(equation, clustvars, hypotheses, data = simData))

    }

    simSTAT <- simEST[2:nrow(simEST), 2]
    countSTAT <- matrix(abs(simSTAT) >= abs(obsStat), ncol = 1)

    ExactP <- colSums(countSTAT) / iterations

    EST <- cbind(obsEST, ExactP)

    colnames(EST) <- c("Estimate", "Tstat", "SE", "P", "ExactP")

    return(EST)

}

## FDR returns minimum q-values ##
```

```
FDR <- function(pvals, step) {

    if (sum(is.na(pvals) == FALSE) <= 1) {return(pvals)}
    if (missing(step)) {step <- 0.001}

    allpvals <- cbind(as.matrix(pvals), matrix(1:nrow(as.matrix(pvals)), ncol = 1))

    pvals <- na.omit(allpvals)
    nump <- nrow(pvals)

    pvals <- pvals[order(pvals[, 1]), ]
    rank <- matrix(1:nump, ncol = 1)
    pvals <- cbind(pvals, rank, matrix(0, nrow = nump, ncol = 1))

    qval <- 1

    while (qval > 0) {

        qfirst <- qval / (1 + qval)
        fdrtemp <- (qfirst * rank) / nump

        subrank <- which(fdrtemp >= as.matrix(pvals[, 1]))

        if (length(subrank) < 1) {
            numreject <- 0
        } else numreject <- max(subrank)

        qsec <- qfirst * (nump / (nump - numreject))
        fdrtemp <- (qsec * rank) / nump

        subrank <- which(fdrtemp >= as.matrix(pvals[, 1]))

        if (length(subrank) < 1) {
            numreject <- 0
        } else numreject <- max(subrank)

        pvals[which(pvals[, 3] <= numreject), 4] <- qval

        qval <- qval - step

    }

    pvals <- pvals[order(pvals[, 2]), ]
```

```
    qvals <- matrix(nrow = nrow(allpvals), ncol = 1)
    qvals[match(pvals[, 2], allpvals[, 2]), 1] <- pvals[, 4]

    return(as.matrix(qvals))

}

## Interact returns a string of interacted variables ##

Interact <- function(d, x) {

    catstring <- ""

    for (var in x) {

        catstring <- paste(catstring, " + ", d, "*", var, sep = "")

    }

    return(substr(catstring, 3, nchar(catstring)))

}

#################
## Clean data ##
#################

varnames <- as.vector(read.delim(file = "K1__Field_Survey_v34+35_Appended.csv", sep = ",",
header = FALSE, stringsAsFactors = FALSE, na.strings = "", nrows = 1))
k1_df <- read.delim(file = "K1__Field_Survey_v34+35_Appended.csv", sep = ",", header =
FALSE, stringsAsFactors = FALSE, na.strings = "", skip = 2, nrows = 600, col.names =
varnames)

## Survey meta data ##

k1_df$start.time.mst <- as.POSIXct(as.character(k1_df$V3), format = "%m/%d/%y %H:%M")
k1_df$start.time.eat <- k1_df$start.time.mst + (3600 * 9)

k1_df$end.time.mst <- as.POSIXct(as.character(k1_df$V4), format = "%m/%d/%y %H:%M")
k1_df$end.time.eat <- k1_df$end.time.mst + (3600 * 9)

## Participant ID ##
```

```
k1_df$survey.id <- k1_df$V1
k1_df <- k1_df[complete.cases(k1_df$survey.id), ]

nonentry <- c("R_5ZkChbiXDIj6KsK", "R_7N8rNtLsxF5a9on", "R_97Tx2cAjy30fqMR",
"R_lYnRvOAJhuax6LS", "R_8dky4iSC7rfEuNc", "R_nczo7KPxLkkKkgo",
"R_5j1OiNu3wMp265N", "R_0GYX0scNN16ICfQ", "R_6PoFtAhwvSBNYzi",
"R_696kAyWai9bDkFI", "R_hPaLwAaYCnY0l69", "R_oGNBAVexMWYhMml",
"R_3rwTGdEULwOGH2y", "R_bhNv0SnArTa32Xe", "R_5txHVIbQY6twLIZ",
"R_kaSM6nunZ9Ynatj", "R_0ppcidBVCPaEkXK", "R_oWceQFJG5NnSokN",
"R_h5Cw4tvVUeDY8NI", "R_9ib4ASBi450NZPt", "R_mAlfPdxj5GQsJF5",
"R_kbW6NDTS1FWXyn3", "R_b1GC7jpoQrJKrFN", "R_5YpNpbPWOxnSNWc",
"R_mwLFSjScVyrgs9J", "R_1jbOtmMIrvgTgaE", "R_if5tz3h1N9MzTp2", "R_aFmo1jRrWIwsLjI",
"R_2itxVUcUstO3Syp", "R_9LP4exOnWpJ1TrE", "R_cQLu9PDCtFwbn7K",
"R_oFl39knSVL3SKpz", "R_11Y1KTzawxBJmvy")
k1_df <- k1_df[! k1_df$survey.id %in% nonentry, ]

## Treatment assignment ##

k1_df$treat[k1_df$condition == "poor"] <- 0
k1_df$treat[k1_df$condition == "individual"] <- 1
k1_df$treat[k1_df$condition == "community"] <- 2
k1_df$treat <- factor(k1_df$treat, labels = c("Pov", "Ind", "Com"))

k1_df$pov <- ifelse(k1_df$condition == "poor", 1, 0)
k1_df$ind <- ifelse(k1_df$condition == "individual", 1, 0)
k1_df$com <- ifelse(k1_df$condition == "community", 1, 0)

k1_df$msg1 <- recode(as.numeric(as.factor(k1_df$ORG_MESSAGE)), `2` = 0, `3` = 2)
k1_df$msg2 <- recode(as.numeric(as.factor(k1_df$ORG_MESSAGE_2)), `2` = 0)
k1_df$msg3 <- recode(as.numeric(as.factor(k1_df$ORG_MESSAGE_3)), `3` = 2)

## Self-efficacy ##

for (var in c(k1_df$sel.con, k1_df$sel.pers, k1_df$sel.com, k1_df$sel.prob, k1_df$sel.bett)) {

   var[var < 0] <- NA

}

k1_df$sel.score <- scale(k1_df$sel.con) + scale(k1_df$sel.pers) + scale(k1_df$sel.com) +
scale(k1_df$sel.prob) + scale(k1_df$sel.bett)
k1_df$sel.score.z <- scale(k1_df$sel.score)
```

```
## Stigma ##

for (var in c(k1_df$jud.fam, k1_df$jud.com, k1_df$jud.judg, k1_df$jud.emb, k1_df$jud.ups)) {

   var[var < 0] <- NA

}

k1_df$sti.score <- scale(6 - k1_df$jud.fam) + scale(6 - k1_df$jud.com) + scale(k1_df$jud.judg) +
scale(k1_df$jud.emb) + scale(k1_df$jud.ups)
k1_df$sti.score.z <- scale(k1_df$sti.score)

## Affect ##

for (var in c(k1_df$aff.pos, k1_df$aff.ash, k1_df$aff.pow, k1_df$aff.fina)) {

   var[var < 0] <- NA

}

k1_df$aff.score <- scale(k1_df$aff.pos) + scale(k1_df$aff.pow) + scale(7 - k1_df$aff.ash) +
scale(7 - k1_df$aff.fina)
k1_df$aff.score.z <- scale(k1_df$aff.score)

## Video selection ##

k1_df$vid.imp1 <- k1_df$vid.dec1 %in% c(3, 5)
k1_df$vid.imp2 <- k1_df$vid.dec2 %in% c(3, 5)
k1_df$vid.num <- k1_df$vid.imp1 + k1_df$vid.imp2

## Intertemporal choice ##

k1_df$sav.save <- k1_df$sav.dec > 1
k1_df$sav.save[k1_df$sav.dec < 0] <- NA

k1_df$sav.amt[k1_df$sav.dec == 1] = 0
k1_df$sav.amt[k1_df$sav.dec == 2] = 100
k1_df$sav.amt[k1_df$sav.dec == 3] = 200
k1_df$sav.amt[k1_df$sav.dec < 0] <- NA

## Query theory (savings) ##
```

```
que_df <- k1_df[names(k1_df) %in% c("survey.id", "que.rat1", "que.rat2", "que.rat3", "que.rat4",
"que.rat5")]

k1_df$que.nonm <- apply(que_df[, 1:5], 1, function(x) length(x[is.na(x) == FALSE]))

que_df <- melt(que_df, id = c("survey.id"))
que_df$variable <- as.numeric(que_df$variable)
que_df <- dcast(que_df[is.na(que_df$value) == FALSE & que_df$value > 0, ], survey.id ~ value,
median, value.var = "variable")
names(que_df) <- c("survey.id", "que.mri", "que.mrp")
k1_df <- merge(k1_df, que_df, all.x = TRUE)

k1_df$que.smrd <- (2 * (k1_df$que.mrp - k1_df$que.mri)) / k1_df$que.nonm
k1_df$que.smrd[is.na(k1_df$que.mrp)] <- 1
k1_df$que.smrd[is.na(k1_df$que.mri)] <- -1

# dealing with missing by filling in upper/lower bounds for now

## Message of support ##

k1_df$msg.dec[k1_df$msg.dec < 0] = NA
k1_df$msg.dec <- k1_df$msg.dec - 1

k1_df$msg.emp[k1_df$msg.emp < 0] = NA

k1_df$msg.lik[k1_df$msg.lik < 0] = NA
k1_df$msg.lik <- 7 - k1_df$msg.lik

k1_df$msg.avg <- (k1_df$msg.emp + k1_df$msg.lik) / 2

## Frame evaluation ##

k1_df$eva.poor[k1_df$msg1 == 0] <- k1_df$eva.msg1[k1_df$msg1 == 0]
k1_df$eva.poor[k1_df$msg2 == 0] <- k1_df$eva.msg2[k1_df$msg2 == 0]
k1_df$eva.poor[k1_df$msg3 == 0] <- k1_df$eva.msg3[k1_df$msg3 == 0]

k1_df$eva.ind[k1_df$msg1 == 1] <- k1_df$eva.msg1[k1_df$msg1 == 1]
k1_df$eva.ind[k1_df$msg2 == 1] <- k1_df$eva.msg2[k1_df$msg2 == 1]
k1_df$eva.ind[k1_df$msg3 == 1] <- k1_df$eva.msg3[k1_df$msg3 == 1]

k1_df$eva.com[k1_df$msg1 == 2] <- k1_df$eva.msg1[k1_df$msg1 == 2]
k1_df$eva.com[k1_df$msg2 == 2] <- k1_df$eva.msg2[k1_df$msg2 == 2]
k1_df$eva.com[k1_df$msg3 == 2] <- k1_df$eva.msg3[k1_df$msg3 == 2]
```

```
k1_df$eva.vid.poor <- k1_df$eva.rank.vid_8
k1_df$eva.vid.ind <- k1_df$eva.rank.vid_9
k1_df$eva.vid.com <- k1_df$eva.rank.vid_10

k1_df$eva.conf[k1_df$eva.conf < 0] <- NA

k1_df$eva.emp.poor <- k1_df$eva.rank.emp_5
k1_df$eva.emp.ind <- k1_df$eva.rank.emp_6
k1_df$eva.emp.com <- k1_df$eva.rank.emp_7

## Ladder scales ##

k1_df$ses.lad.now[k1_df$ses.lad.now < 0] <- NA
k1_df$ses.lad.now.z <- scale(k1_df$ses.lad.now)

k1_df$ses.lad.y2[k1_df$ses.lad.y2 < 0] <- NA
k1_df$ses.lad.y2.z <- scale(k1_df$ses.lad.y2)

k1_df$ses.lad.diff <- k1_df$ses.lad.y2 - k1_df$ses.lad.now
k1_df$ses.lad.avg <- (k1_df$ses.lad.y2 + k1_df$ses.lad.now) / 2

## Sociodemographics ##

k1_df$soc.age[k1_df$soc.age < 0] <- NA
k1_df$soc.pri <- as.numeric(k1_df$soc.edu > 3)
k1_df$soc.fem <- k1_df$soc.gen - 1
k1_df$soc.chr <- k1_df$soc.rel %in% c(1, 2)
k1_df$ses.unemp <- k1_df$ses.emp %in% c(1, 2)

k1_df$soc.inc[k1_df$soc.inc < 0] <- NA
k1_df$soc.inc.wins[k1_df$soc.inc <= quantile(k1_df$soc.inc, .99)] <-
k1_df$soc.inc[k1_df$soc.inc <= quantile(k1_df$soc.inc, .99)]
k1_df$soc.inc.wins.ln <- log(k1_df$soc.inc.wins + sqrt(k1_df$soc.inc.wins^2 + 1))

k1_df$soc.con[k1_df$soc.con < 0] <- NA
k1_df$soc.con.wins[k1_df$soc.con <= quantile(k1_df$soc.con, .99)] <-
k1_df$soc.con[k1_df$soc.con <= quantile(k1_df$soc.con, .99)]
k1_df$soc.con.wins.ln <- log(k1_df$soc.con.wins + sqrt(k1_df$soc.con.wins^2 + 1))

k1_df$soc.sav <- k1_df$soc.sav - 1

k1_df$soc.eme.z <- scale(k1_df$soc.eme)
```

```
## Survey validity ##

k1_df$end.hear <- k1_df$end.hear - 1
k1_df$end.hear[k1_df$end.hear < 0] <- NA

attach(k1_df)

## Center covariates ##

soc.fem.c <- scale(soc.fem, scale = FALSE)
soc.pri.c <- scale(soc.pri, scale = FALSE)
soc.age.c <- scale(soc.age, scale = FALSE)
ses.unemp.c <- scale(ses.unemp, scale = FALSE)
soc.inc.wins.ln.c <- scale(soc.inc.wins.ln, scale = FALSE)
soc.con.wins.ln.c <- scale(soc.con.wins.ln, scale = FALSE)
soc.sav.c <- scale(soc.sav, scale = FALSE)

#################
## Estimation ##
#################

## Randomization balance checks ##

hypotheses <- c("treatInd = 0", "treatCom = 1", "treatInd - treatCom = 0")
depvars <- c("soc.fem", "soc.pri", "soc.age", "ses.unemp", "soc.inc.wins.ln", "soc.con.wins.ln",
"soc.sav")

for (h in hypotheses) {

   RES <- matrix(nrow = 1, ncol = 5)

   for (depvar in depvars) {

      eqn <- paste(depvar, "~ treat", sep = " ")
      RES <- rbind(RES, PermTest(eqn, treatvars = c("treat", "pov", "ind", "com"), clustvars =
k1_df$survey.id, hypotheses = c(h), iterations = 10000, data = k1_df))

   }

   RES <- RES[2:nrow(RES), 1:ncol(RES)]
   RES <- cbind(RES, FDR(RES[, 4]))
```

```
    rownames(RES) <- depvars
    colnames(RES)[6] <- "Min. Q"

    print("----------------------------------------------------------------", quote = FALSE)
    print(paste("H_0:", h), quote = FALSE)
    print(RES, quote = FALSE)

}

## Plain OLS for primary outcomes ##

hypotheses <- c("treatInd = 0", "treatCom = 1", "treatInd - treatCom = 0")
depvars <- c("vid.num", "sav.amt", "msg.dec")

for (h in hypotheses) {

    RES <- matrix(nrow = 1, ncol = 5)

    for (depvar in depvars) {

        eqn <- paste(depvar, "~ treat", sep = " ")
        RES <- rbind(RES, PermTest(eqn, treatvars = c("treat", "pov", "ind", "com"), clustvars =
k1_df$survey.id, hypotheses = c(h), iterations = 10000, data = k1_df))

    }

    RES <- RES[2:nrow(RES), 1:ncol(RES)]
    RES <- cbind(RES, FDR(RES[, 4]))

    rownames(RES) <- depvars
    colnames(RES)[6] <- "Min. Q"

    print("----------------------------------------------------------------", quote = FALSE)
    print(paste("H_0:", h), quote = FALSE)
    print(RES, quote = FALSE)

}

## Plain OLS for secondary outcomes ##

hypotheses <- c("treatInd = 0", "treatCom = 1", "treatInd - treatCom = 0")
depvars <- c("sel.score.z", "sti.score.z", "aff.score.z", "msg.avg", "que.smrd", "ses.lad.now",
"ses.lad.y2", "ses.lad.diff")
```

```
for (h in hypotheses) {

   RES <- matrix(nrow = 1, ncol = 5)

   for (depvar in depvars) {

      eqn <- paste(depvar, "~ treat", sep = " ")
      RES <- rbind(RES, PermTest(eqn, treatvars = c("treat", "pov", "ind", "com"), clustvars =
k1_df$survey.id, hypotheses = c(h), iterations = 10000, data = k1_df))

   }

   RES <- RES[2:nrow(RES), 1:ncol(RES)]
   RES <- cbind(RES, FDR(RES[, 4]))

   rownames(RES) <- depvars
   colnames(RES)[6] <- "Min. Q"

   print("----------------------------------------------------------------", quote = FALSE)
   print(paste("H_0:", h), quote = FALSE)
   print(RES, quote = FALSE)

}

## Covariate adjustment for primary outcomes ##

hypotheses <- c("treatInd = 0", "treatCom = 1", "treatInd - treatCom = 0")
depvars <- c("vid.num", "sav.amt", "msg.dec")
covariates <- c("soc.fem.c", "soc.pri.c", "soc.age.c", "ses.unemp.c")

for (h in hypotheses) {

   RES <- matrix(nrow = 1, ncol = 5)

   for (depvar in depvars) {

      eqn <- paste(depvar, "~", Interact("treat", covariates), sep = " ")
      RES <- rbind(RES, PermTest(eqn, treatvars = c("treat", "pov", "ind", "com"), clustvars =
k1_df$survey.id, hypotheses = c(h), iterations = 10000, data = k1_df))

   }
```

```
    RES <- RES[2:nrow(RES), 1:ncol(RES)]
    RES <- cbind(RES, FDR(RES[, 4]))

    rownames(RES) <- depvars
    colnames(RES)[6] <- "Min. Q"

    print("-----------------------------------------------------------------", quote = FALSE)
    print(paste("H_0:", h), quote = FALSE)
    print(RES, quote = FALSE)

}

## Covariate adjustment for secondary outcomes ##

hypotheses <- c("treatInd = 0", "treatCom = 1", "treatInd - treatCom = 0")
depvars <- c("sel.score.z", "sti.score.z", "aff.score.z", "msg.avg", "que.smrd", "ses.lad.now",
"ses.lad.y2", "ses.lad.diff")
covariates <- c("soc.fem.c", "soc.pri.c", "soc.age.c", "ses.unemp.c")

for (h in hypotheses) {

    RES <- matrix(nrow = 1, ncol = 5)

    for (depvar in depvars) {

        eqn <- paste(depvar, "~", Interact("treat", covariates), sep = " ")
        RES <- rbind(RES, PermTest(eqn, treatvars = c("treat", "pov", "ind", "com"), clustvars =
k1_df$survey.id, hypotheses = c(h), iterations = 10000, data = k1_df))

    }

    RES <- RES[2:nrow(RES), 1:ncol(RES)]
    RES <- cbind(RES, FDR(RES[, 4]))

    rownames(RES) <- depvars
    colnames(RES)[6] <- "Min. Q"

    print("-----------------------------------------------------------------", quote = FALSE)
    print(paste("H_0:", h), quote = FALSE)
    print(RES, quote = FALSE)

}
```

```
## Heterogeneous effects for primary outcomes ##

depvars <- c("vid.num", "sav.amt", "msg.dec")
hetvars <- c("soc.fem", "soc.pri")

for (hetvar in hetvars) {

   hypotheses <- c(paste("treatInd:", hetvar, " = 0", sep = ""), paste("treatCom:", hetvar, " = 0",
sep = ""), paste("treatInd:", hetvar, " - ", "treatCom:", hetvar, " = 0", sep = ""))

   for (h in hypotheses) {

      RES <- matrix(nrow = 1, ncol = 5)

      for (depvar in depvars) {

         eqn <- paste(depvar, " ~ treat*", hetvar, sep = "")
         RES <- rbind(RES, PermTest(eqn, treatvars = c("treat", "pov", "ind", "com"), clustvars =
k1_df$survey.id, hypotheses = c(h), iterations = 10000, data = k1_df))

      }

      RES <- RES[2:nrow(RES), 1:ncol(RES)]
      RES <- cbind(RES, FDR(RES[, 4]))

      rownames(RES) <- depvars
      colnames(RES)[6] <- "Min. Q"

      print("----------------------------------------------------------------", quote = FALSE)
      print(paste("H_0:", h), quote = FALSE)
      print(RES, quote = FALSE)

   }

}

## Heterogeneous effects for secondary outcomes ##

depvars <- c("sel.score.z", "sti.score.z", "aff.score.z", "msg.avg", "que.smrd", "ses.lad.now",
"ses.lad.y2", "ses.lad.diff")
hetvars <- c("soc.fem", "soc.pri")

for (hetvar in hetvars) {
```

```
   hypotheses <- c(paste("treatInd:", hetvar, " = 0", sep = ""), paste("treatCom:", hetvar, " = 0",
sep = ""), paste("treatInd:", hetvar, " - ", "treatCom:", hetvar, " = 0", sep = ""))

   for (h in hypotheses) {

      RES <- matrix(nrow = 1, ncol = 5)

      for (depvar in depvars) {

         eqn <- paste(depvar, " ~ treat*", hetvar, sep = "")
         RES <- rbind(RES, PermTest(eqn, treatvars = c("treat", "pov", "ind", "com"), clustvars =
k1_df$survey.id, hypotheses = c(h), iterations = 10000, data = k1_df))

      }

      RES <- RES[2:nrow(RES), 1:ncol(RES)]
      RES <- cbind(RES, FDR(RES[, 4]))

      rownames(RES) <- depvars
      colnames(RES)[6] <- "Min. Q"

      print("----------------------------------------------------------------", quote = FALSE)
      print(paste("H_0:", h), quote = FALSE)
      print(RES, quote = FALSE)

   }

}
```