pset 4 675

```
#============================#
# ==== Metrics 675 ps 4 ====
#============================#


#========================================#
# ==== Load packages, clear workspace ====
#========================================#


  library(foreach)
  library(data.table)
  library(Matrix)
  library(ggplot2)
  library(sandwich)
  library(xtable)
  library(boot)
  library(CausalGAM)
  library(Hmisc)
library(mvtnorm)


  rm(list = ls(pos = ".GlobalEnv"), pos = ".GlobalEnv")
  options(scipen = 999)
  cat("\f")

#=====================================================#
# ==== Input data, add covariates and subset data ====
#=====================================================#

  lal_dt <- fread('C://Users/Nmath_000/Documents/MI_school/Second Year/675 Applied Econometrics/hw/hw4/l

  lal_dt[,log.re74 := log(re74+1)]
  lal_dt[,log.re75 := log(re75+1)]
  lal_dt[,age.sq := age^2]
  lal_dt[,educ.sq := educ^2]
  lal_dt[,age.cu := age^3]
  lal_dt[,black.u74 := black*u74]
  lal_dt[,educ.logre74 := educ*log.re74]

  # subset lal_dt for LaLonde control only
  lal_c1 <-  lal_dt[treat==1 | treat==0]

  # subset lal_dt for PSID control only
  lal_c2 <- lal_dt[treat==1 | treat==2]


  # Recode treatment indicate in PSID control dataset (recode 2's as 0's)
  lal_c2[,treat:=as.numeric(treat==1)]
```

```r
#================================#
# ==== Create covariate lists ====
#================================#


  z.a <- c("age", "educ", "black", "hisp", "married", "nodegr", "log.re74", "log.re75")
  z.b <- c(z.a,"age.sq","educ.sq", "u74", "u75" )
  z.c <- c(z.b, "age.cu", "black.u74", "educ.logre74")

  # make sure I didn't mis-type any (sohuld be empty sets)
  setdiff(z.a, colnames(lal_c1))
  setdiff(z.b, colnames(lal_c1))
  setdiff(z.c, colnames(lal_c1))


#=================================#
# ==== [1] Difference in means ====
#=================================#

  # run diff in diff
  dmeans.ll <- lm(re78~treat, data = lal_c1)
  dmeans.ps <- lm(re78~treat, data = lal_c2)

  # get robust se
  dmeans.ll <- data.table(tidy(coeftest(dmeans.ll, vcov = vcovHC(dmeans.ll, type = "HC1"))))
  dmeans.ps<- data.table(tidy(coeftest(dmeans.ps, vcov = vcovHC(dmeans.ps, type = "HC1"))))

  # just keep treat term
  dmeans.ll <- dmeans.ll[term == "treat", -c("term")]
  dmeans.ps <- dmeans.ps[term == "treat", -c("term")]

  dmeans.ll <- round(dmeans.ll, 2)
  dmeans.ps <- round(dmeans.ps, 2)

  # Compute 95% CIs
  dmeans.ll[, CI_lower := estimate - 1.96*std.error]
  dmeans.ll[, CI_upper := estimate + 1.96*std.error]
  dmeans.ll[, CI := paste0("(", CI_lower, ", ", CI_upper, ")")]

  dmeans.ps[, CI_lower := estimate - 1.96*std.error]
  dmeans.ps[, CI_upper := estimate + 1.96*std.error]
  dmeans.ps[, CI := paste0("(", CI_lower, ", ", CI_upper, ")")]


  # keep what I need and put them in same table
  dmeans.ll <- dmeans.ll[, c("estimate", "std.error", "CI" )]
  dmeans.ps <- dmeans.ps[, c("estimate", "std.error", "CI" )]

  setnames(dmeans.ll, colnames(dmeans.ll), paste0(colnames(dmeans.ll), '_exp'))
  setnames(dmeans.ps, colnames(dmeans.ps), paste0(colnames(dmeans.ps), '_PSID'))

  out_dmeans <- cbind(dmeans.ll, dmeans.ps)
```

```
#==================#
# ==== [2] OLS ====
#==================#


  # write wrapper function to get what I need
  # # define variables for line by line debug
  # in_dt=lal_c1
  # in_z = z.a
  # in_spec = "a"
  # in_c_label = "exp"

  # start function wrapper
  ols_wrap <- function(in_dt, in_z, in_spec, in_c_label ){

    # run ols
    reg_form <- reg_form <- as.formula(paste("re78~", paste(c("treat", in_z), collapse="+")))
    ols_out <- lm(reg_form, data = in_dt)

    # get robust se
    ols_out <- data.table(tidy(coeftest(ols_out, vcov = vcovHC(ols_out, type = "HC1"))))

    # grab the term we care about
    ols_out <- ols_out[term == "treat", -c("term")]

    # round
    ols_out <- round(ols_out, 2)

    # do CI
    ols_out[, CI_lower := estimate - 1.96*std.error]
    ols_out[, CI_upper := estimate + 1.96*std.error]
    ols_out[, CI := paste0("(", CI_lower, ", ", CI_upper, ")")]
    ols_out <- ols_out[, c("estimate", "std.error", "CI" )]

    # put label in
    setnames(ols_out, colnames(ols_out), paste0(colnames(ols_out),"_", in_c_label )  )
    ols_out[, specification := in_spec]
    return(ols_out[])
  }

  ols_wrap(in_z = z.a, in_spec = "a" ,in_dt=lal_c1, in_c_label = "exp" )

  # run on all expirmental data
  exp_ols <- mapply(ols_wrap,in_z = list(z.a, z.b, z.c), in_spec = list("a", "b", "c") ,in_dt= list(lal
  exp_ols <- rbindlist(exp_ols)

  # run on psid
  psid_ols <- mapply(ols_wrap, in_z = list(z.a, z.b, z.c), in_spec = list("a", "b", "c") ,in_dt= list(la
  psid_ols <- rbindlist(psid_ols)

  # put it in one table
  ols_out <- merge(psid_ols, exp_ols, by = "specification")
```

```
# cant get the standard errors this way. Gonna use a package to do it below
# #================================#
# # ==== Regression Imputation ====
# #================================#
#
#
#   # start function wrapper
#   in_dt = lal_dt
#   in_z = z.a
#   in_spec = "a"
#   reg_imp <- function(in_dt, in_z, in_spec ){
#
#     # run ols reg on each treatment group
#     reg_form <- reg_form <- as.formula(paste("re78~", paste(in_z, collapse="+")))
#     ols.treat     <- lm(reg_form, data = in_dt[treat == 1])
#     ols.control.ll  <- lm(reg_form, data = in_dt[treat == 0])
#     ols.control.ps  <- lm(reg_form, data = in_dt[treat == 2])
#
#     # make matrix of data so I can calculate ATE
#     # y values
#     Y.treat        = lal_dt[treat==1, "re78"]
#     Y.control.ll   = lal_dt[treat==0, "re78"]
#     Y.control.ps   = lal_dt[treat==2, "re78"]
#
#     # x values
#     X.treat <- data.table(const =1 , lal_dt[treat==1, in_z, with = FALSE])
#     X.control.ll <- data.table(const =0 , lal_dt[treat==1, in_z, with = FALSE])
#     X.control.ps <- data.table(const =2 , lal_dt[treat==1, in_z, with = FALSE])
#
#
#     # Impute `individual treatment effects`
#     tvec.ri.treat.ll      = as.matrix(X.treat)%*%(as.vector(ols.treat$coefficients)-as.vector(ols.con
#     tvec.ri.treat.ps      = as.matrix(X.treat)%*%(as.vector(ols.treat$coefficients)-as.vector(ols.con
#
#     tvec.ri.control.ll    = as.matrix(X.control.ll)%*%(as.vector(ols.treat$coefficients)-as.vector(ol
#     tvec.ri.control.ps    = as.matrix(X.control.ps)%*%(as.vector(ols.treat$coefficients)-as.vector(ol
#
#     # Compute ATEs
#     ate.ri.ll       = mean(c(tvec.ri.treat.ll,tvec.ri.control.ll))
#     ate.ri.ps       = mean(c(tvec.ri.treat.ps,tvec.ri.control.ps))
#
#     # Compute ATT
#     att.ri          = mean(tvec.ri.treat.ll)
#
#
#
#   }
#



#=============================================================#
# ==== IPW and Doubly Robust using the "CausalGAM" package ====
```

```
#============================================================#

  #=======================#
  # ==== run functions ====
  #=======================#


    # Covariates A, Lalonde control
      # make formula
      pscore_form <- as.formula(paste0("treat~", paste(z.a, collapse = " + ")))
      out_form    <- as.formula(paste0("re78~", paste(z.a, collapse = " + ")))
    ATE.ll.A <- estimate.ATE(pscore.formula = pscore_form,
                             pscore.family = binomial,
                             outcome.formula.t = out_form,
                             outcome.formula.c =out_form,
                             outcome.family = gaussian,
                             treatment.var = "treat",
                             data=as.data.frame(lal_c1),
                             divby0.action="t",
                             divby0.tol=0.001,
                             var.gam.plot=FALSE,
                             nboot=0)

    # Covariates B, Lalonde control
    pscore_form <- as.formula(paste0("treat~", paste(z.b, collapse = " + ")))
    out_form    <- as.formula(paste0("re78~", paste(z.b, collapse = " + ")))
    ATE.ll.B <- estimate.ATE(pscore.formula = pscore_form,
                             pscore.family = binomial,
                             outcome.formula.t = out_form,
                             outcome.formula.c =out_form,
                             outcome.family = gaussian,
                             treatment.var = "treat",
                             data=as.data.frame(lal_c1),
                             divby0.action="t",
                             divby0.tol=0.001,
                             var.gam.plot=FALSE,
                             nboot=0)

    # Covariates C, Lalonde control
    pscore_form <- as.formula(paste0("treat~", paste(z.c, collapse = " + ")))
    out_form    <- as.formula(paste0("re78~", paste(z.c, collapse = " + ")))
    ATE.ll.C <- estimate.ATE(pscore.formula = pscore_form,
                             pscore.family = binomial,
                             outcome.formula.t = out_form,
                             outcome.formula.c =out_form,
                             outcome.family = gaussian,
                             treatment.var = "treat",
                             data=as.data.frame(lal_c1),
                             divby0.action="t",
                             divby0.tol=0.001,
                             var.gam.plot=FALSE,
                             nboot=0)
    # # This doesnt run
```

```
  # # Covariates A, PSID control
  # # make formula
  # pscore_form <- as.formula(paste0("treat~", paste(z.a, collapse = " + ")))
  # out_form    <- as.formula(paste0("re78~", paste(z.a, collapse = " + ")))
  # ATE.ps.A <- estimate.ATE(pscore.formula = pscore_form,
  #                          pscore.family = binomial,
  #                          outcome.formula.t = out_form,
  #                          outcome.formula.c =out_form,
  #                          outcome.family = gaussian,
  #                          treatment.var = "treat",
  #                          data=as.data.frame(lal_c2),
  #                          divby0.action="t",
  #                          divby0.tol=0.01,
  #                          var.gam.plot=FALSE,
  #                          nboot=0)

  # Covariates B, PSID control
  pscore_form <- as.formula(paste0("treat~", paste(z.b, collapse = " + ")))
  out_form    <- as.formula(paste0("re78~", paste(z.b, collapse = " + ")))
  ATE.ps.B <- estimate.ATE(pscore.formula = pscore_form,
                           pscore.family = binomial,
                           outcome.formula.t = out_form,
                           outcome.formula.c =out_form,
                           outcome.family = gaussian,
                           treatment.var = "treat",
                           data=as.data.frame(lal_c2),
                           divby0.action="t",
                           divby0.tol=0.001,
                           var.gam.plot=FALSE,
                           nboot=0)

  # Covariates C, PSID control
  pscore_form <- as.formula(paste0("treat~", paste(z.c, collapse = " + ")))
  out_form    <- as.formula(paste0("re78~", paste(z.c, collapse = " + ")))
  ATE.ps.C <- estimate.ATE(pscore.formula = pscore_form,
                           pscore.family = binomial,
                           outcome.formula.t = out_form,
                           outcome.formula.c =out_form,
                           outcome.family = gaussian,
                           treatment.var = "treat",
                           data=as.data.frame(lal_c2),
                           divby0.action="t",
                           divby0.tol=0.001,
                           var.gam.plot=FALSE,
                           nboot=0)



#======================#
# ==== sort output ====
#======================#

  #NOTE this was a dumb way to do this but it is what it is
```

```
  #=========================#
  # ==== reg imputation ====
  #=========================#


    # Reg imputation results
    reg_imp_fun <- function(reg_output, in_spec, in_cont){
    tbl  =   data.table(estimate = reg_output$ATE.reg.hat,
                        std.error = reg_output$ATE.reg.asymp.SE,
                        CI_L = reg_output$ATE.reg.hat-1.96*reg_output$ATE.reg.asymp.SE,
                        CI_U = reg_output$ATE.reg.hat+1.96*reg_output$ATE.reg.asymp.SE)
    tbl <- round(tbl,2)
    tbl[, CI := paste0("(", CI_L, ", ", CI_U, ")")]
    tbl[, CI_L := NULL]
    tbl[, CI_U := NULL]
    setnames(tbl, colnames(tbl), paste0(colnames(tbl),"_", in_cont )  )
    tbl[, specification := in_spec]

    return(tbl[])
    }

    # do it with esp data
    out_RI <- list()
    out_RI[["a"]] <- reg_imp_fun(ATE.ll.A, "a", "exp")
    out_RI[["b"]] <- reg_imp_fun(ATE.ll.B, "b", "exp")
    out_RI[["c"]] <- reg_imp_fun(ATE.ll.C, "c", "exp")

    out_RI <-rbindlist(out_RI)

    # now do it with PSID
    out_RI2 <- list()
    out_RI2[["b"]] <- reg_imp_fun(ATE.ps.B, "b", "PSID")
    out_RI2[["c"]] <- reg_imp_fun(ATE.ps.C, "c", "PSID")

    out_RI2 <-rbindlist(out_RI2)

    # merge them
    out_RI <- merge(out_RI, out_RI2, by = "specification", all = TRUE)

#===============#
# ==== IPW ====
#===============#

    # ipw results
    ipw_fun <- function(reg_output, in_spec, in_cont){
      tbl  =   data.table(estimate = reg_output$ATE.IPW.hat,
                          std.error = reg_output$ATE.IPW.asymp.SE,
                          CI_L = reg_output$ATE.IPW.hat-1.96*reg_output$ATE.IPW.asymp.SE,
                          CI_U = reg_output$ATE.IPW.hat+1.96*reg_output$ATE.IPW.asymp.SE)
      tbl <- round(tbl,2)
      tbl[, CI := paste0("(", CI_L, ", ", CI_U, ")")]
      tbl[, CI_L := NULL]
      tbl[, CI_U := NULL]
```

```
  setnames(tbl, colnames(tbl), paste0(colnames(tbl),"_", in_cont )  )
  tbl[, specification := in_spec]

  return(tbl[])
}

# do it with esp data
out_IPW <- list()
out_IPW[["a"]] <- ipw_fun(ATE.ll.A, "a", "exp")
out_IPW[["b"]] <- ipw_fun(ATE.ll.B, "b", "exp")
out_IPW[["c"]] <- ipw_fun(ATE.ll.C, "c", "exp")

out_IPW <-rbindlist(out_IPW)

# now do it with PSID
out_IPW2 <- list()
out_IPW2[["b"]] <- ipw_fun(ATE.ps.B, "b", "PSID")
out_IPW2[["c"]] <- ipw_fun(ATE.ps.C, "c", "PSID")

out_IPW2 <-rbindlist(out_IPW2)

# merge them
out_IPW <- merge(out_IPW, out_IPW2, by = "specification", all = TRUE)


#================================#
# ==== Doubly robust results ====
#================================#


# DR results
DR_fun <- function(reg_output, in_spec, in_cont){
  tbl  =   data.table(estimate = reg_output$ATE.AIPW.hat,
                      std.error = reg_output$ATE.IPW.asymp.SE,
                      CI_L = reg_output$ATE.AIPW.hat-1.96*reg_output$ATE.AIPW.asymp.SE,
                      CI_U = reg_output$ATE.AIPW.hat+1.96*reg_output$ATE.AIPW.asymp.SE)
  tbl <- round(tbl,2)
  tbl[, CI := paste0("(", CI_L, ", ", CI_U, ")")]
  tbl[, CI_L := NULL]
  tbl[, CI_U := NULL]
  setnames(tbl, colnames(tbl), paste0(colnames(tbl),"_", in_cont )  )
  tbl[, specification := in_spec]

  return(tbl[])
}



# do it with esp data
out_dr <- list()
out_dr[["a"]] <- DR_fun(ATE.ll.A, "a", "exp")
out_dr[["b"]] <- DR_fun(ATE.ll.B, "b", "exp")
out_dr[["c"]] <- DR_fun(ATE.ll.C, "c", "exp")
```

```
      out_dr <-rbindlist(out_dr)

      # now do it with PSID
      out_dr2 <- list()
      out_dr2[["b"]] <- DR_fun(ATE.ps.B, "b", "PSID")
      out_dr2[["c"]] <- DR_fun(ATE.ps.C, "c", "PSID")

      out_dr2 <-rbindlist(out_dr2)

      # merge them
      out_dr <- merge(out_dr, out_dr2, by = "specification", all = TRUE)




#============================#
# ==== CONSTRUCT TABLE 1 ====
#============================#

# fill in statistic and stack data
out_dmeans[, statistic := "Mean Diff"]
ols_out[, statistic := "OLS"]
out_RI[, statistic := "Reg. Impute"]
out_IPW[, statistic := "IPW"]
out_dr[, statistic := "D. Robust"]

# stack them all
out_table <-rbind(out_dmeans, ols_out, out_RI, out_IPW, out_dr, fill = TRUE )

# set the column order
setcolorder(out_table, c("statistic", "specification", setdiff(colnames(out_table), c("statistic", "spe

# save it
write.csv(out_table, "C://Users/Nmath_000/Documents/Code/courses/econ 675/PS_4_tex/q2_results_R.csv", r


#============================================================#
# ==== now load in CSV and make the friggin latex table ====
#============================================================#

# load ATE
ATE_table <- fread("C://Users/Nmath_000/Documents/Code/courses/econ 675/PS_4_tex/Table1_ATE_resultq.csv




print(xtable(ATE_table, type = "latex"),
      file = "C://Users/Nmath_000/Documents/Code/courses/econ 675/PS_4_tex/q2table.tex",
      include.rownames = FALSE,
      floating = FALSE)

# ATT
att_table <- fread("C://Users/Nmath_000/Documents/Code/courses/econ 675/PS_4_tex/Table1_ATT_resultq.csv
```

```
print(xtable(ATE_table, type = "latex"),
      file = "C://Users/Nmath_000/Documents/Code/courses/econ 675/PS_4_tex/q2table_att.tex",
      include.rownames = FALSE,
      floating = FALSE)

# can't get this shit to work
# latex(round(ATE_table, 2),
#       file=paste0("C://Users/Nmath_000/Documents/Code/courses/econ 675/PS_4_tex/q2table.tex"),
#       append=FALSE,
#       table.env=FALSE
#       ,center="none",
#       title="",
#       n.cgroup=c(4, 4),
#       cgroup=c("Experimental Data", "PSID Control"),
#       colheads=c("$\\hat{\\tau}$", "s.e.", "C.I.", "", "$\\hat{\\tau}$", "s.e.", "C.I.", ""),
#       n.rgroup=c(1, rep(3, 6)),
#       rgroup=c("Mean Diff.", "OLS", "Reg. Impute", "IPW", "D. Robust", "N1 Match", "p Match"),
#       rowname=c("", rep(c("a", "b", "c"), 8)))



#=====================#
# ==== question 3 ====
#=====================#

# clear workspace

rm(list = ls(pos = ".GlobalEnv"), pos = ".GlobalEnv")
cat("\f")

# set attributes for plot to default ea theme
plot_attributes <- theme( plot.background = element_rect(fill = "lightgrey"),
                          panel.grid.major.x = element_line(color = "gray90"),
                          panel.grid.minor  = element_blank(),
                          panel.background = element_rect(fill = "white", colour = "black") ,
                          panel.grid.major.y = element_line(color = "gray90"),
                          text = element_text(size= 30),
                          plot.title = element_text(vjust=0, hjust = 0.5, colour = "#0B6357",face = "bo

##############################################################################
# Generate random data and simulate
##############################################################################

N     = 50
M     = 1000
SIGMA = matrix(c(1,0.85,0.85,1),2,2)

set.seed(1234)

# Generate covariates
W     = replicate(M,rmvnorm(N, mean = c(0,0), sigma = SIGMA, method="chol"))

# Generate errors
```

```r
E     = replicate(M,rnorm(50))

# Generate outcomes
Y     = sapply(1:M,function(i) rep(1,N)+W[,,i]%*%c(0.5,1)+E[,i])

# Get beta.hats
beta.hats = sapply(1:M,function(i) lm(Y[,i]~W[,,i])$coefficients[2])

# Get t-stats for gamma.hats
t.stats   = sapply(1:M,function(i) summary(lm(Y[,i]~W[,,i]))[["coefficients"]][, "t value"][3])

# Get beta.tildes
beta.tildes = sapply(1:M,function(i) lm(Y[,i]~W[,1,i])$coefficients[2])

# Construct betas if the model selection is used
beta.sel    = ifelse(t.stats>=1.96,beta.hats,beta.tildes)

#=========================================================#
# ==== [1] Summary Statistics for the different betas ====
#=========================================================#


# Summary statistics
beta.sum = data.table(rbind(summary(beta.hats),summary(beta.tildes),summary(beta.sel)))

# estimates
beta.sum[, estimator := c("i", "ii", "iii")]

# put in order
setcolorder(beta.sum, c("estimator", setdiff(colnames(beta.sum), "estimator")))

# Make kernenl desity plot
plot.dat = data.frame(beta = c(beta.hats,beta.tildes,beta.sel),Estimator=rep(c("hat", "tilde","sel"), ea

densplot = ggplot(plot.dat,aes(x=beta,fill=Estimator))+
  geom_density(alpha=0.5, kernel="e",bw="ucv")+
  ggtitle("Kernel Density Plots")+
  xlab("Point Estimator")+
  ylab("Density")+
  plot_attributes +
  scale_fill_discrete(
    name="Estimator",
    breaks=c("hat", "tilde", "sel"),
    labels=c("(i)", "(ii)", "(iii)"))+
  theme(legend.justification = c(0.05, 0.98), legend.position = c(0.05, 0.98))

# save summary stats and plot
print(xtable(beta.sum, type = "latex"),
      file = "C://Users/Nmath_000/Documents/Code/courses/econ 675/PS_4_tex/q3_sum_stats.tex",
      include.rownames = FALSE,
      floating = FALSE)

png(paste0("c:/Users/Nmath_000/Documents/Code/courses/econ 675/PS_4_tex/q4_den.png"), height = 800, widt
```

```r
print(densplot)
dev.off()


#############################################################################
# [2] Coverage rates
#############################################################################

# Compute coverage rate for beta.hat
beta.hats.se       = sapply(1:M,function(i) summary(lm(Y[,i]~W[,,i]))[["coefficients"]][, "Std. Error"]
beta.hats.CIs      = cbind(beta.hats-1.96*beta.hats.se,beta.hats+1.96*beta.hats.se)
beta.hats.covered  = ifelse(0.5>=beta.hats.CIs[,1]&0.5<=beta.hats.CIs[,2],1,0)
beta.hat.cr        = mean(beta.hats.covered)

# Compute coverage rate for beta.tilde
beta.tildes.se       = sapply(1:M,function(i) summary(lm(Y[,i]~W[,1,i]))[["coefficients"]][, "Std. Erro
beta.tildes.CIs      = cbind(beta.tildes-1.96*beta.tildes.se,beta.tildes+1.96*beta.tildes.se)
beta.tildes.covered  = ifelse(0.5>=beta.tildes.CIs[,1]&0.5<=beta.tildes.CIs[,2],1,0)
beta.tilde.cr        = mean(beta.tildes.covered)

# Compute coverage rate for beta.sel
beta.sel.CI.lower    = ifelse(beta.hats==beta.sel,beta.hats-1.96*beta.hats.se,beta.tildes-1.96*beta.til
beta.sel.CI.upper    = ifelse(beta.hats==beta.sel,beta.hats+1.96*beta.hats.se,beta.tildes+1.96*beta.til
beta.sel.CIs         = cbind(beta.sel.CI.lower,beta.sel.CI.upper)
beta.sel.covered     = ifelse(0.5>=beta.sel.CIs[,1]&0.5<=beta.sel.CIs[,2],1,0)
beta.sel.cr          = mean(beta.sel.covered)

# Put results together
cr.results           = rbind(beta.hat.cr,beta.tilde.cr,beta.sel.cr)
rownames(cr.results) = c("beta.hat.cr","beta.tilde.cr","beta.sel.cr")
colnames(cr.results) = c("Coverage Rate")

# save this shiz
print(xtable(cr.results, type = "latex"),
      file = "C://Users/Nmath_000/Documents/Code/courses/econ 675/PS_4_tex/q3_cov_rate.tex",
      include.rownames = FALSE,
      floating = FALSE)
```