Working from Self-driving Car - Results

Georg Hirte

03 03 2022

#1 Introduction

This markdown file prepares the results of the Monte-Carlo Simulation to the paper "Working form self-driving car". We use GAMS to perform the simulations. Here, we prepare the results for use in the paper.

Initialization

Define function for read in data

Read parameter data

function to read variables with two sets (not "a" or "b")

```
mydataVARcr.frame <- function(gdxfile) {
  colnames <- c("id_i","id_c","id_r");
  id_r <- rgdx.set(gdxfile,"ll",names = colnames,compress=TRUE,ts=TRUE)
  id_c <- rgdx.set(gdxfile,"ctr",names = colnames,compress=TRUE,ts=TRUE)
  id_i <- rgdx.set(gdxfile,"i",names = colnames,compress=TRUE,ts=TRUE)
  alpha.frame <- rgdx.param(gdxfile,"calc_alpha",
                            names = c(colnames[2],colnames[3],"alpha"),
                            compress=TRUE,ts=TRUE,squeeze=FALSE)
  epsilon.frame <- rgdx.param(gdxfile,"calc_epsilon",
                              names = c(colnames[2],colnames[3],"epsilon"),
                              compress=TRUE,ts=TRUE,squeeze=FALSE)
  vc.frame <- rgdx.param(gdxfile,"calc_hvc",
                         names = c(colnames[2],colnames[3],"vc"),
                         compress=TRUE,ts=TRUE,squeeze=FALSE)
  indVx.frame <- rgdx.param(gdxfile,"calc_indVx",
                            names = c(colnames[2],colnames[3],"indVx"),
                            compress=TRUE,ts=TRUE,squeeze=FALSE)
  Ubar.frame <- rgdx.param(gdxfile,"calc_Ubar",
                           names = c(colnames[2],colnames[3],"Ubar"),
                           compress=TRUE,ts=TRUE,squeeze=FALSE)
  wBstar.frame <- rgdx.param(gdxfile,"calc_wBstar",
                             names = c(colnames[2],colnames[3],"wBstar"),
                             compress=TRUE,ts=TRUE,squeeze=FALSE)
  xv.frame <- rgdx.param(gdxfile,"calc_xv",
                         names = c(colnames[2],colnames[3],"xv"),
                         compress=TRUE,ts=TRUE,squeeze=FALSE)
  pb.frame <- rgdx.param(gdxfile,"calc_pb",
                         names = c(colnames[2],colnames[3],"pb"),
                         compress=TRUE,ts=TRUE,squeeze=FALSE)
  chgmargcost.frame <- rgdx.param(gdxfile,"firm_chg_marg_costs",names = c(colnames[2],colnames[3],"chgma
```

```
  chgmargprofit.frame <- rgdx.param(gdxfile,"firm_chg_marg_profit",names = c(colnames[2],colnames[3],"cl
  chgmargprod.frame <- rgdx.param(gdxfile,"firm_chg_margprod",names = c(colnames[2],colnames[3],"chgmarg
  costA.frame <- rgdx.param(gdxfile,"firm_marg_costsA",
                            names = c(colnames[2],colnames[3],"costA"),
                            compress=TRUE,ts=TRUE,squeeze=FALSE)
  costB.frame <- rgdx.param(gdxfile,"firm_marg_costsB",
                            names = c(colnames[2],colnames[3],"costB"),
                            compress=TRUE,ts=TRUE,squeeze=FALSE)
  Deltac.frame <- rgdx.param(gdxfile,"firm_marg_Deltac",
                             names = c(colnames[2],colnames[3],"Deltac"),
                             compress=TRUE,ts=TRUE,squeeze=FALSE)
  Deltafc.frame <- rgdx.param(gdxfile,"firm_marg_Deltafc",
                              names = c(colnames[2],colnames[3],"Deltafc"),
                              compress=TRUE,ts=TRUE,squeeze=FALSE)
  Deltamc.frame <- rgdx.param(gdxfile,"firm_marg_Deltamc",
                              names = c(colnames[2],colnames[3],"Deltamc"),
                              compress=TRUE,ts=TRUE,squeeze=FALSE)

  FrameVARrc.frame <- merge(alpha.frame,epsilon.frame,by=c("id_r", "id_c"))
  FrameVARrc.frame <- merge(FrameVARrc.frame,vc.frame,by=c("id_r", "id_c"))
  FrameVARrc.frame <- merge(FrameVARrc.frame,indVx.frame,by=c("id_r", "id_c"))
  FrameVARrc.frame <- merge(FrameVARrc.frame,Ubar.frame,by=c("id_r", "id_c"))
  FrameVARrc.frame <- merge(FrameVARrc.frame,wBstar.frame,by=c("id_r", "id_c"))
  FrameVARrc.frame <- merge(FrameVARrc.frame,xv.frame,by=c("id_r", "id_c"))
  FrameVARrc.frame <- merge(FrameVARrc.frame,pb.frame,by=c("id_r", "id_c"))
  FrameVARrc.frame <- merge(FrameVARrc.frame,costA.frame,by=c("id_r", "id_c"))
  FrameVARrc.frame <- merge(FrameVARrc.frame,costB.frame,by=c("id_r", "id_c"))
  FrameVARrc.frame <- merge(FrameVARrc.frame,Deltac.frame,by=c("id_r", "id_c"))
  FrameVARrc.frame <- merge(FrameVARrc.frame,Deltafc.frame,by=c("id_r", "id_c"))
  FrameVARrc.frame <- merge(FrameVARrc.frame,Deltamc.frame,by=c("id_r", "id_c"))
  FrameVARrc.frame <- merge(FrameVARrc.frame,chgmargcost.frame,by=c("id_r", "id_c"))
  FrameVARrc.frame <- merge(FrameVARrc.frame,chgmargprofit.frame,by=c("id_r", "id_c"))
  FrameVARrc.frame <- merge(FrameVARrc.frame,chgmargprod.frame,by=c("id_r", "id_c"))
  return(FrameVARrc.frame)
}



### parameters (variables in GAMS simulation) with id_i =a" and "b"
mydataVARab.frame <- function(gdxfile) {
  colnames <- c("id_i","id_c","id_r");
  id_r <- rgdx.set(gdxfile,"ll",names = colnames,compress=TRUE,ts=TRUE)
  id_c <- rgdx.set(gdxfile,"ctr",names = colnames,compress=TRUE,ts=TRUE)
  id_i <- rgdx.set(gdxfile,"i",names = colnames,compress=TRUE,ts=TRUE)
  effort.frame <- rgdx.param(gdxfile,"calc_effort",
                       names = c(colnames[1],colnames[2],colnames[3],"effort"),
                       compress=TRUE,ts=TRUE,squeeze=FALSE)
  ell.frame <- rgdx.param(gdxfile,"calc_ell",
                         names = c(colnames[1],colnames[2],colnames[3],"ell"),
                         compress=TRUE,ts=TRUE,squeeze=FALSE)
  income.frame <- rgdx.param(gdxfile,"calc_income",
                  names = c(colnames[1],colnames[2],colnames[3],"income"),
```

```
                  compress=TRUE,ts=TRUE,squeeze=FALSE)
  incomeB.frame <- as.data.frame(income.frame[effort.frame$id_i =="b" | income.frame$id_i =="b",])
  indV.frame <- rgdx.param(gdxfile,"calc_indV",
                           names = c(colnames[1],colnames[2],colnames[3],"indV"),
                           compress=TRUE,ts=TRUE,squeeze=FALSE)
  v.frame <- rgdx.param(gdxfile,"calc_v",
                        names = c(colnames[1],colnames[2],colnames[3],"v"),
                        compress=TRUE,ts=TRUE,squeeze=FALSE)
  vo.frame <- rgdx.param(gdxfile,"calc_vo",
                         names = c(colnames[1],colnames[2],colnames[3],"vo"),
                         compress=TRUE,ts=TRUE,squeeze=FALSE)
  wB.frame <- rgdx.param(gdxfile,"calc_wB",
                         names = c(colnames[1],colnames[2],colnames[3],"wB"),
                         compress=TRUE,ts=TRUE,squeeze=FALSE)
  xB.frame <- rgdx.param(gdxfile,"calc_xB",
                         names = c(colnames[1],colnames[2],colnames[3],"xB"),
                         compress=TRUE,ts=TRUE,squeeze=FALSE)
  z.frame <- rgdx.param(gdxfile,"calc_z",
                        names = c(colnames[1],colnames[2],colnames[3],"z"),
                        compress=TRUE,ts=TRUE,squeeze=FALSE)
  tkm.frame <- rgdx.param(gdxfile,"calc_tkm",
                          names = c(colnames[1],colnames[2],colnames[3],"tkm"),
                          compress=TRUE,ts=TRUE,squeeze=FALSE)
  expwage.frame <- rgdx.param(gdxfile,"firm_avg_expwage",
                      names = c(colnames[1],colnames[2],colnames[3],"expwage"),
                            compress=TRUE,ts=TRUE,squeeze=FALSE)
#  avgsqm.frame <- rgdx.param(gdxfile,"firm_avg_sqm",
#  names = c(colnames[1],colnames[2],colnames[3],"avgsqm"),
#  compress=TRUE,ts=TRUE,squeeze=FALSE)
  firmlabord.frame <- rgdx.param(gdxfile,"firm_labor_demand",
                     names = c(colnames[1],colnames[2],colnames[3],"firmlabord"),
                                 compress=TRUE,ts=TRUE,squeeze=FALSE)
#  firmsqm.frame <- rgdx.param(gdxfile,"firm_sqm",
#  names = c(colnames[1],colnames[2],colnames[3],"firmsqm"),
# compress=TRUE,ts=TRUE,squeeze=FALSE)
  firmproduction.frame <- rgdx.param(gdxfile,"firm_production",
            names = c(colnames[1],colnames[2],colnames[3],"firmproduction"),
                                     compress=TRUE,ts=TRUE,squeeze=FALSE)
  margprod.frame <- rgdx.param(gdxfile,"firm_margprod",
                      names = c(colnames[1],colnames[2],colnames[3],"margprod"),
                              compress=TRUE,ts=TRUE,squeeze=FALSE)
  netprofits.frame <- rgdx.param(gdxfile,"firm_NetProfits",
        names = c(colnames[1],colnames[2],colnames[3],"netprofits"),
                                 compress=TRUE,ts=TRUE,squeeze=FALSE)
  worker_travelcost.frame <- rgdx.param(gdxfile,"zemployee_travelcost",
            names = c(colnames[1],colnames[2],colnames[3],"worker_travelcost"),
            compress=TRUE,ts=TRUE,squeeze=FALSE)
  worker_traveltime.frame <- rgdx.param(gdxfile,"zemployee_traveltime",
            names = c(colnames[1],colnames[2],colnames[3],"worker_traveltime"),
            compress=TRUE,ts=TRUE,squeeze=FALSE)
  VOT.frame <- rgdx.param(gdxfile,"zemployee_VOT",
                          names = c(colnames[1],colnames[2],colnames[3],"VOT"),
                          compress=TRUE,ts=TRUE,squeeze=FALSE)
```

```
  # generate dataframe with all variables with "a" and "b"
  FrameVARab.frame <- merge(effort.frame,ell.frame,by=c("id_i","id_r", "id_c"))
  FrameVARab.frame <- merge(FrameVARab.frame,income.frame,by=c("id_i","id_r", "id_c"))
  FrameVARab.frame <- merge(FrameVARab.frame,incomeB.frame,by=c("id_i","id_r", "id_c"))
  FrameVARab.frame <- merge(FrameVARab.frame,indV.frame,by=c("id_i","id_r", "id_c"))
  FrameVARab.frame <- merge(FrameVARab.frame,v.frame,by=c("id_i","id_r", "id_c"))
  FrameVARab.frame <- merge(FrameVARab.frame,vo.frame,by=c("id_i","id_r", "id_c"))
  FrameVARab.frame <- merge(FrameVARab.frame,wB.frame,by=c("id_i","id_r", "id_c"))
  FrameVARab.frame <- merge(FrameVARab.frame,xB.frame,
                            by=c("id_i","id_r", "id_c"))
  FrameVARab.frame <- merge(FrameVARab.frame,z.frame,
                            by=c("id_i","id_r", "id_c"))
  FrameVARab.frame <- merge(FrameVARab.frame,tkm.frame,
                            by=c("id_i","id_r", "id_c"))
  FrameVARab.frame <- merge(FrameVARab.frame,expwage.frame,
                            by=c("id_i","id_r", "id_c"))
#  FrameVARab.frame <- merge(FrameVARab.frame,avgsqm.frame,by=c("id_i","id_r", "id_c"))
  FrameVARab.frame <- merge(FrameVARab.frame,firmlabord.frame,
                            by=c("id_i","id_r", "id_c"))
#  FrameVARab.frame <- merge(FrameVARab.frame,firmsqm.frame,by=c("id_i","id_r", "id_c"))
  FrameVARab.frame <- merge(FrameVARab.frame,firmproduction.frame,
                            by=c("id_i","id_r", "id_c"))
  FrameVARab.frame <- merge(FrameVARab.frame,margprod.frame,
                            by=c("id_i","id_r", "id_c"))
  FrameVARab.frame <- merge(FrameVARab.frame,netprofits.frame,
                            by=c("id_i","id_r", "id_c"))
  FrameVARab.frame <- merge(FrameVARab.frame,worker_travelcost.frame,
                            by=c("id_i","id_r", "id_c"))
  FrameVARab.frame <- merge(FrameVARab.frame,worker_traveltime.frame,
                            by=c("id_i","id_r", "id_c"))
  FrameVARab.frame <- merge(FrameVARab.frame,VOT.frame,
                            by=c("id_i","id_r", "id_c"))
  return(FrameVARab.frame)
}

### df1.frame
df1.frame <- function(gdxfile) {
  MobileVARcr.frame <- mydataVARcr.frame(gdxfile)
  MobilePARAM.frame <- mydataPARAM.frame(gdxfile)
  MobileVARab.frame <- mydataVARab.frame(gdxfile)
  # generate file with only contract 'b'
  Mobileonlyb.frame <-
    as.data.frame(MobileVARab.frame[MobileVARab.frame$id_i =="a" | MobileVARab.frame$id_i =="b",])
  Mobileb.frame <- merge(MobilePARAM.frame, MobileVARcr.frame,
                         by=c("id_r", "id_c"))
  Mobileb.frame <- merge(Mobileb.frame, Mobileonlyb.frame,
                         by=c("id_r", "id_c"))
  # produce data.frame for table
  df1.frame <- Mobileb.frame
  return(df1.frame)
}

#gdxfile = "OutMobileWork_eff.gdx"
```

```
#nameparam = "eff"
#nameidc = "DE"
dfprintidc <- function(gdxfile,nameparam,nameidc) {
  df.frame <- df1.frame(gdxfile)
  df2.frame <- select(filter(df.frame, id_c == nameidc),
                      c(id_r,id_c,all_of(nameparam),
                        alpha,v,vc,vo,xB,wage,wB,pb,Deltac))
  df2.frame <- df2.frame %>% arrange(id_r) #sort data
  xtab1 <- xtable(df2.frame[1:10, 3:10])
  text0 <- nameparam
  text1 = "Parameter"
  text2 <- nameidc
  text1 <- paste(text1,text0,text2)
  textn0 = "./DataParam"
  textn1 <- nameidc
  textn2 = ".txt"
  textn <- paste(textn0,text0,textn1,textn2,sep="")
  textl1 <- nameidc
  textl01 = " "
  textl0 = "tab: "
  textlab <- paste(textl0,text0,textl01,textl1,sep="")
  pprint <- print(xtable(xtab1, caption = text1, label=textlab)
                  ,floating = TRUE, table.placement = "htb",
                  size = "footnotesize", latex.environments = "center",
                  math.style.negative = "TRUE", file = textn)
  return(pprint)
}



# print elasticites in Tex table Germany
#c("eff","beta","sqmemployee","rmonth","dB","dA","sp","xbar","gm","gx","gh","tb","wage")
#gdxfile = "OutMobileWork_eff.gdx"
#nameparam = "eff"
#nameidc = "DE"
dfprintElast <- function(gdxfile,nameparam,nameidc) {
  df.frame <- df1.frame(gdxfile)
  df2.frame <- select(filter(df.frame, id_c == nameidc),
                      c(id_r,id_c,all_of(nameparam), alpha,v,vc,vo,xB))
  df2.frame <- df2.frame %>% arrange(id_r) #sort data
  df2elast.frame <- data.frame(matrix(ncol = 8, nrow = 10))
  colnames(df2elast.frame) <- c("id_r","id_c",nameparam,"e_alpha","e_v","e_vc",
                                "e_vo", "e_xB")
  df2elast.frame <- data.frame(matrix(vector(),10,8,dimnames=list(c(),
              c("id_r","id_c",nameparam,"e_alpha","e_v","e_vc","e_vo", "e_xB"))),
                               stringsAsFactors=F)
  df2elast.frame$id_r <-  df2.frame$id_r
  df2elast.frame$id_c <-  df2.frame$id_c
  df2elast.frame[,3] <-  df2.frame[,3]
  for(i in 2:10){
    df2elast.frame$e_alpha[i] =
      ((df2.frame$alpha[i]-df2.frame$alpha[i-1])/df2.frame$alpha[i-1])/
      ((df2.frame[i,3]-df2.frame[i-1,3])/df2.frame[i-1,3])
```

```
    df2elast.frame$e_v[i] =
      ((df2.frame$v[i]-df2.frame$v[i-1])/df2.frame$v[i-1])/
      ((df2.frame[i,3]-df2.frame[i-1,3])/df2.frame[i-1,3])
    df2elast.frame$e_vc[i] =
      ((df2.frame$vc[i]-df2.frame$vc[i-1])/df2.frame$vc[i-1])/
      ((df2.frame[i,3]-df2.frame[i-1,3])/df2.frame[i-1,3])
    df2elast.frame$e_vo[i] =
      ((df2.frame$vo[i]-df2.frame$vo[i-1])/df2.frame$vo[i-1])/
      ((df2.frame[i,3]-df2.frame[i-1,3])/df2.frame[i-1,3])
    df2elast.frame$e_xB[i] =
      ((df2.frame$xB[i]-df2.frame$xB[i-1])/df2.frame$xB[i-1])/
      ((df2.frame[i,3]-df2.frame[i-1,3])/df2.frame[i-1,3])
  }
  str(df2elast.frame)
  # add median
  df2elast.frame <- rbind(df2elast.frame,c("median",nameidc,nameparam,
                          median(df2elast.frame$e_alpha,sort=TRUE,na.rm=TRUE),
                          median(df2elast.frame$e_v,sort=TRUE,na.rm=TRUE),
                          median(df2elast.frame$e_vc,sort=TRUE,na.rm=TRUE),
                          median(df2elast.frame$e_vo,sort=TRUE,na.rm=TRUE),
                          median(df2elast.frame$e_xB,sort=TRUE,na.rm=TRUE)))

# transform to numeric
  df2elast.frame$e_alpha <- as.numeric(df2elast.frame$e_alpha)
  df2elast.frame$e_v <- as.numeric(df2elast.frame$e_v)
  df2elast.frame$e_vc <- as.numeric(df2elast.frame$e_vc)
  df2elast.frame$e_vo <- as.numeric(df2elast.frame$e_vo)
  df2elast.frame$e_xB <- as.numeric(df2elast.frame$e_xB)

  number_hook <- function(x) {
  ifelse(abs(x) < 100 & abs(x) > 0.000,
         prettyNum(x, small.mark = ",", digits = 2), trunc(x))
  }
  xtable :: xtable(dplyr::mutate_if(df2elast.frame, is.numeric, number_hook))
  xtab1 <- xtable(dplyr::mutate_if(df2elast.frame, is.numeric, number_hook))
  #xtab1 <- xtable(df2elast.frame[2:10, 1:8],digits=2)
  text0 <- nameparam
  text1 = "Parameter "
  text2 <- nameidc
  text1 <- paste(text1,text0,text2)
  textn0 = "./ParamElast"
  textn1 <- nameidc
  textn2 = ".txt"
  textn <- paste(textn0,text0,textn1,textn2,sep="")
  textl1 = " "
  textl2 <- nameidc
  textl0 = "tab: "
  textlab <- paste(textl0,text0,textl1,sep="")
  pprintelast <- print(xtable(xtab1, caption = text1, label=textlab,digits=2)
                  ,floating = TRUE, table.placement = "htb",
                  size = "footnotesize", latex.environments = "center",
                  math.style.negative = "TRUE", file = textn)
  return(pprintelast)
```

```
}
```

UDF: calcprob

function calculate probabilities *Ü Monte Carlo Table* calculate all variables as between 0-1 (shares) * sort them, calculate the empirical cumulative density function * and print probabilities

UDF: dfmcframe

generate data frame from Monte Carlo results

UDF dfprintMonteCarlo

print latex table for MC

```
dfprintMonteCarlo <- function(gdxfile,nameparam, nameidc,asupport) {
    #dfmc.frame <- dfmc.frame %>% arrange(id_r) #sort data
  # prepare data for output table
  dfmc.frame <- dfmcframe(gdxfile,nameparam, nameidc,asupport)
  value <- c(0,0.1,0.2,0.25, 0.3,0.4,0.5,0.6,0.7,0.75,0.8,0.9,1,"median","mean")
  digit = 4
  namevar <- dfmc.frame$alpha
  alpha <- calcprob(namevar,digit,1)
  # call function to calculate probabilities and cumulative density ecdf
  namevar <- dfmc.frame$share_v
  sv <- calcprob(namevar,digit,1)
  namevar <- dfmc.frame$share_vc
  svc <- calcprob(namevar,digit,1)
  namevar <- dfmc.frame$share_vo
  svo <- calcprob(namevar,digit,1)
  namevar <- dfmc.frame$share_xsum
  sxsum <- calcprob(namevar,digit,1)
  namevar <- dfmc.frame$share_x
  sx <- calcprob(namevar,digit,1)
  namevar <- dfmc.frame$share_xo
  sxo <- calcprob(namevar,digit,1)
  #xtab1.frame <- data.frame(value,alpha,sv,svc,svo,sxsum,sx,sxo)
  xtab1.frame <- data.frame(value,alpha,sv,svc,svo,sx,sxo)
  summary(xtab1.frame)
  xtab1 <- xtable(xtab1.frame, rownames=F,)

  if (nameidc =="DE") {
    text1 <- "Monte-Carlo Simulation: DE, ["
  } else {
    text1 <- "Monte-Carlo Simulation: US, ["
  }
  text2 <- asupport
  text3 <- "]"
  text1 <- paste(text1,text2,text3)
  if (nameidc =="DE") {
    textn0 <- "./DataMC_DE.txt"
  } else {
    textn0 <- "./DataMC_US.txt"
  }
  #textn <- paste(textn0)
```

```
  pprintMC <- print(xtable(xtab1,  caption = text1), include.rownames=F
                    ,floating = TRUE, latex.environments = "center", file = textn0)
  return(pprintMC)
}
```

Evaluate results

PARAMETER VARIATION

Calculate results for single parameter variations

```
select1 <- c("eff","beta","sqmemployee","rmonth","dB","dA","sp","xbar","gm",
             "gx","gh","tb","wage")
for(nameparam in select1){
    text0 <- "OutMobileWork_"
    text1 <- nameparam
    text2 <- ".gdx"
    gdxfile <- paste(text0,text1,text2,sep="")
#    gdxfile = "OutMobileWork_eff.gdx"
#  nameparam <- select1
  for(nameidc in c("DE","US")) {
    dfpDE <- dfprintidc(gdxfile,nameparam,nameidc)
    dfpelastDE <- dfprintElast(gdxfile,nameparam,nameidc)
  }
}
```

```
## 'data.frame':    10 obs. of  8 variables:
##  $ id_r   : Factor w/ 10 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10
##  $ id_c   : Factor w/ 2 levels "DE","US": 1 1 1 1 1 1 1 1 1 1
##  $ eff    : num  1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2
##  $ e_alpha: num  NA 0.156 -0.138 -0.136 -0.127 ...
##  $ e_v    : num  NA -1.67e-15 -2.10 -2.15 -1.95 ...
##  $ e_vc   : num  NA 16.083 0.666 0 0 ...
##  $ e_vo   : num  NA -0.303 -2.33 -2.389 -2.217 ...
##  $ e_xB   : num  NA -4.06 -3.78 0 0 ...
## 'data.frame':    10 obs. of  8 variables:
##  $ id_r   : Factor w/ 10 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10
##  $ id_c   : Factor w/ 2 levels "DE","US": 2 2 2 2 2 2 2 2 2 2
##  $ eff    : num  1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2
##  $ e_alpha: num  NA 0.495 -0.358 -0.45 -0.371 ...
##  $ e_v    : num  NA 0 -1.76 -2.08 -1.88 ...
##  $ e_vc   : num  NA 38.8 1.57 0 0 ...
##  $ e_vo   : num  NA -0.438 -2.118 -2.389 -2.217 ...
##  $ e_xB   : num  NA -4.25 -10.09 0 0 ...
## 'data.frame':    10 obs. of  8 variables:
##  $ id_r   : Factor w/ 10 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10
##  $ id_c   : Factor w/ 2 levels "DE","US": 1 1 1 1 1 1 1 1 1 1
##  $ beta   : num  -51.7 -41.4 -31 -20.7 -10.3 ...
##  $ e_alpha: num  NA -Inf -43.581 -2.748 -0.956 ...
##  $ e_v    : num  NA 7.91e-12 5.79e-12 3.96e-12 2.39e-12 ...
##  $ e_vc   : num  NA 0 0 0 0 ...
##  $ e_vo   : num  NA 1.01e-11 7.42e-12 5.07e-12 3.07e-12 ...
##  $ e_xB   : num  NA 0 0 0 0 ...
## 'data.frame':    10 obs. of  8 variables:
```

```
##  $ id_r   : Factor w/ 10 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10
##  $ id_c   : Factor w/ 2 levels "DE","US": 2 2 2 2 2 2 2 2 2 2
##  $ beta   : num  -51.4 -41.2 -30.9 -20.6 -10.3 ...
##  $ e_alpha: num  NA NaN -Inf -103.41 -1.94 ...
##  $ e_v    : num  NA 3.17e-12 2.25e-12 1.49e-12 8.69e-13 ...
##  $ e_vc   : num  NA 5.10e-15 -4.08e-15 0.00 0.00 ...
##  $ e_vo   : num  NA 4.36e-12 3.10e-12 2.05e-12 1.20e-12 ...
##  $ e_xB   : num  NA -1.38e-13 1.10e-13 0.00 0.00 ...
## 'data.frame':    10 obs. of  8 variables:
##  $ id_r       : Factor w/ 10 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10
##  $ id_c       : Factor w/ 2 levels "DE","US": 1 1 1 1 1 1 1 1 1 1
##  $ sqmemployee: num  18.8 20.4 22 23.6 25.2 26.8 28.4 30 31.6 33.2
##  $ e_alpha    : num  NA 0.0672 0.0726 0.0778 0.083 ...
##  $ e_v        : num  NA -1.75e-15 1.90e-15 -2.05e-15 2.20e-15 ...
##  $ e_vc       : num  NA 0 0 0 0 0 0 0 0 0
##  $ e_vo       : num  NA 0 0 0 0 0 0 0 0 0
##  $ e_xB       : num  NA 0 0 0 0 0 0 0 0 0
## 'data.frame':    10 obs. of  8 variables:
##  $ id_r       : Factor w/ 10 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10
##  $ id_c       : Factor w/ 2 levels "DE","US": 2 2 2 2 2 2 2 2 2 2
##  $ sqmemployee: num  9.3 11.2 13 14.9 16.7 ...
##  $ e_alpha    : num  NA 0.211 0.243 0.272 0.3 ...
##  $ e_v        : num  NA 0 0 0 0 0 0 0 0 0
##  $ e_vc       : num  NA 0 0 0 0 0 0 0 0 0
##  $ e_vo       : num  NA 0 0 0 0 0 0 0 0 0
##  $ e_xB       : num  NA 0 0 0 0 0 0 0 0 0
## 'data.frame':    10 obs. of  8 variables:
##  $ id_r   : Factor w/ 10 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10
##  $ id_c   : Factor w/ 2 levels "DE","US": 1 1 1 1 1 1 1 1 1 1
##  $ rmonth : num  8.44 12.12 15.8 19.48 23.16 ...
##  $ e_alpha: num  NA 0.039 0.0551 0.0706 0.0857 ...
##  $ e_v    : num  NA -3.42e-16 4.91e-16 -6.40e-16 7.89e-16 ...
##  $ e_vc   : num  NA 0 0 0 0 0 0 0 0 0
##  $ e_vo   : num  NA 0 0 0 0 0 0 0 0 0
##  $ e_xB   : num  NA 0 0 0 0 0 0 0 0 0
## 'data.frame':    10 obs. of  8 variables:
##  $ id_r   : Factor w/ 10 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10
##  $ id_c   : Factor w/ 2 levels "DE","US": 2 2 2 2 2 2 2 2 2 2
##  $ rmonth : num  22.8 36.5 50.2 63.9 77.6 ...
##  $ e_alpha: num  NA 0.118 0.176 0.227 0.272 ...
##  $ e_v    : num  NA 0 0 0 0 0 0 0 0 0
##  $ e_vc   : num  NA 0 0 0 0 0 0 0 0 0
##  $ e_vo   : num  NA 0 0 0 0 0 0 0 0 0
##  $ e_xB   : num  NA 0 0 0 0 0 0 0 0 0
## 'data.frame':    10 obs. of  8 variables:
##  $ id_r   : Factor w/ 10 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10
##  $ id_c   : Factor w/ 2 levels "DE","US": 1 1 1 1 1 1 1 1 1 1
##  $ dB     : num  1 4 7 10 13 16 19 22 25 28
##  $ e_alpha: num  NA -0.00715 -0.02923 -0.0523 -0.07642 ...
##  $ e_v    : num  NA -4.97e-17 1.99e-16 -3.48e-16 4.97e-16 ...
##  $ e_vc   : num  NA 0 0 0 0 0 0 0 0 0
##  $ e_vo   : num  NA 0 0 0 0 0 0 0 0 0
##  $ e_xB   : num  NA 0 0 0 0 0 0 0 0 0
## 'data.frame':    10 obs. of  8 variables:
```

```
##  $ id_r   : Factor w/ 10 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10
##  $ id_c   : Factor w/ 2 levels "DE","US": 2 2 2 2 2 2 2 2 2 2
##  $ dB     : num  1 4 7 10 13 16 19 22 25 28
##  $ e_alpha: num  NA -0.00778 -0.03185 -0.0571 -0.08361 ...
##  $ e_v    : num  NA 0 0 0 0 0 0 0 0 0
##  $ e_vc   : num  NA 0 0 0 0 0 0 0 0 0
##  $ e_vo   : num  NA 0 0 0 0 0 0 0 0 0
##  $ e_xB   : num  NA 0 0 0 0 0 0 0 0 0
## 'data.frame':    10 obs. of  8 variables:
##  $ id_r   : Factor w/ 10 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10
##  $ id_c   : Factor w/ 2 levels "DE","US": 1 1 1 1 1 1 1 1 1 1
##  $ dA     : num  1 2 3 4 5 6 7 8 9 10
##  $ e_alpha: num  NA 0.00795 0.01577 0.02346 0.03104 ...
##  $ e_v    : num  NA -1.49e-16 2.98e-16 -4.47e-16 5.96e-16 ...
##  $ e_vc   : num  NA 0 0 0 0 0 0 0 0 0
##  $ e_vo   : num  NA 0 0 0 0 0 0 0 0 0
##  $ e_xB   : num  NA 0 0 0 0 0 0 0 0 0
## 'data.frame':    10 obs. of  8 variables:
##  $ id_r   : Factor w/ 10 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10
##  $ id_c   : Factor w/ 2 levels "DE","US": 2 2 2 2 2 2 2 2 2 2
##  $ dA     : num  1 2 3 4 5 6 7 8 9 10
##  $ e_alpha: num  NA 0.01 0.0199 0.0295 0.039 ...
##  $ e_v    : num  NA 0 0 0 0 0 0 0 0 0
##  $ e_vc   : num  NA 0 0 0 0 0 0 0 0 0
##  $ e_vo   : num  NA 0 0 0 0 0 0 0 0 0
##  $ e_xB   : num  NA 0 0 0 0 0 0 0 0 0
## 'data.frame':    10 obs. of  8 variables:
##  $ id_r   : Factor w/ 10 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10
##  $ id_c   : Factor w/ 2 levels "DE","US": 1 1 1 1 1 1 1 1 1 1
##  $ sp     : num  0 0.05 0.1 0.15 0.2 0.25 0.3 0.35 0.4 0.45
##  $ e_alpha: num  NA 0 0.00125 0.0025 0.00375 ...
##  $ e_v    : num  NA 0.00 1.49e-16 -2.98e-16 4.47e-16 ...
##  $ e_vc   : num  NA 0 0 0 0 0 0 0 0 0
##  $ e_vo   : num  NA 0 0 0 0 0 0 0 0 0
##  $ e_xB   : num  NA 0 0 0 0 0 0 0 0 0
## 'data.frame':    10 obs. of  8 variables:
##  $ id_r   : Factor w/ 10 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10
##  $ id_c   : Factor w/ 2 levels "DE","US": 2 2 2 2 2 2 2 2 2 2
##  $ sp     : num  0 0.05 0.1 0.15 0.2 0.25 0.3 0.35 0.4 0.45
##  $ e_alpha: num  NA 0 0.0019 0.0038 0.00568 ...
##  $ e_v    : num  NA 0.00 2.77e-16 0.00 0.00 ...
##  $ e_vc   : num  NA 0 0 0 0 0 0 0 0 0
##  $ e_vo   : num  NA 0.00 1.91e-16 3.83e-16 0.00 ...
##  $ e_xB   : num  NA 0 0 0 0 0 0 0 0 0
## 'data.frame':    10 obs. of  8 variables:
##  $ id_r   : Factor w/ 10 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10
##  $ id_c   : Factor w/ 2 levels "DE","US": 1 1 1 1 1 1 1 1 1 1
##  $ xbar   : num  2 12 22 32 42 52 62 72 82 92
##  $ e_alpha: num  NA 0.0267 0.1761 0.3328 0.483 ...
##  $ e_v    : num  NA 0.0186 0.0884 0.1612 0.2194 ...
##  $ e_vc   : num  NA 0.202 0.522 0.712 0.785 ...
##  $ e_vo   : num  NA 3.78e-12 -2.27e-11 -8.42e-16 -6.12e-16 ...
##  $ e_xB   : num  NA 0.346 0.759 0.32 0.247 ...
## 'data.frame':    10 obs. of  8 variables:
```

```
##  $ id_r   : Factor w/ 10 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10
##  $ id_c   : Factor w/ 2 levels "DE","US": 2 2 2 2 2 2 2 2 2 2
##  $ xbar   : num  2 17 32 47 62 77 92 107 122 137
##  $ e_alpha: num  NA 0.0258 0.2289 0.418 0.4618 ...
##  $ e_v    : num  NA 0.0161 0.1203 0.2035 0.1603 ...
##  $ e_vc   : num  NA 0.145 0.583 0.72 0.465 ...
##  $ e_vo   : num  NA 7.27e-13 -6.18e-12 0.00 -6.00e-16 ...
##  $ e_xB   : num  NA 0.334 0.356 0.582 0.647 ...
## 'data.frame':    10 obs. of  8 variables:
##  $ id_r   : Factor w/ 10 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10
##  $ id_c   : Factor w/ 2 levels "DE","US": 1 1 1 1 1 1 1 1 1 1
##  $ gm     : num  0.279 0.369 0.459 0.549 0.639 ...
##  $ e_alpha: num  NA 0.107 0.137 0.165 0.191 ...
##  $ e_v    : num  NA 4.62e-16 -6.11e-16 7.60e-16 -9.10e-16 ...
##  $ e_vc   : num  NA 0 0 0 0 0 0 0 0 0
##  $ e_vo   : num  NA 0 0 0 0 0 0 0 0 0
##  $ e_xB   : num  NA 0 0 0 0 0 0 0 0 0
## 'data.frame':    10 obs. of  8 variables:
##  $ id_r   : Factor w/ 10 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10
##  $ id_c   : Factor w/ 2 levels "DE","US": 2 2 2 2 2 2 2 2 2 2
##  $ gm     : num  0.193 0.296 0.399 0.502 0.605 ...
##  $ e_alpha: num  NA 0.0946 0.1381 0.1776 0.2137 ...
##  $ e_v    : num  NA 0 0 0 0 0 0 0 0 0
##  $ e_vc   : num  NA 0 0 0 0 0 0 0 0 0
##  $ e_vo   : num  NA 0 0 0 0 0 0 0 0 0
##  $ e_xB   : num  NA 0 0 0 0 0 0 0 0 0
## 'data.frame':    10 obs. of  8 variables:
##  $ id_r   : Factor w/ 10 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10
##  $ id_c   : Factor w/ 2 levels "DE","US": 1 1 1 1 1 1 1 1 1 1
##  $ gx     : num  0.038 0.0397 0.0414 0.0431 0.0448 0.0465 0.0482 0.0499 0.0516 0.0533
##  $ e_alpha: num  NA -0.0164 -0.0171 -0.0179 -0.0186 ...
##  $ e_v    : num  NA 0.000439 0.000458 0.000478 0.000498 ...
##  $ e_vc   : num  NA 0.00199 0.00208 0.00216 0.00225 ...
##  $ e_vo   : num  NA 4.28e-15 -4.47e-15 0.00 0.00 ...
##  $ e_xB   : num  NA -0.0173 -0.0181 -0.0188 -0.0196 ...
## 'data.frame':    10 obs. of  8 variables:
##  $ id_r   : Factor w/ 10 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10
##  $ id_c   : Factor w/ 2 levels "DE","US": 2 2 2 2 2 2 2 2 2 2
##  $ gx     : num  0.0157 0.0212 0.0267 0.0322 0.0377 0.0432 0.0487 0.0542 0.0597 0.0652
##  $ e_alpha: num  NA -0.0107 -0.0146 -0.0184 -0.0223 ...
##  $ e_v    : num  NA 0.000228 0.000308 0.000388 0.000467 ...
##  $ e_vc   : num  NA 0.000827 0.001117 0.001406 0.001695 ...
##  $ e_vo   : num  NA 0.00 0.00 0.00 -1.12e-15 ...
##  $ e_xB   : num  NA -0.0352 -0.0481 -0.0613 -0.0749 ...
## 'data.frame':    10 obs. of  8 variables:
##  $ id_r   : Factor w/ 10 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10
##  $ id_c   : Factor w/ 2 levels "DE","US": 1 1 1 1 1 1 1 1 1 1
##  $ gh     : num  1.04 1.64 2.24 2.84 3.44 4.04 4.64 5.24 5.84 6.44
##  $ e_alpha: num  NA -0.0235 -0.0376 -0.0521 -0.067 ...
##  $ e_v    : num  NA 0.000418 0.000659 0.0009 0.001141 ...
##  $ e_vc   : num  NA 0.0019 0.00299 0.00408 0.00517 ...
##  $ e_vo   : num  NA -3.32e-16 0.00 0.00 9.06e-16 ...
##  $ e_xB   : num  NA -0.0158 -0.0251 -0.0346 -0.0442 ...
## 'data.frame':    10 obs. of  8 variables:
```

```
##  $ id_r   : Factor w/ 10 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10
##  $ id_c   : Factor w/ 2 levels "DE","US": 2 2 2 2 2 2 2 2 2 2
##  $ gh     : num  7.02 7.1 7.19 7.28 7.37 ...
##  $ e_alpha: num  NA -0.221 -0.225 -0.228 -0.232 ...
##  $ e_v    : num  NA 0.003 0.00304 0.00307 0.00311 ...
##  $ e_vc   : num  NA 0.0109 0.011 0.0111 0.0113 ...
##  $ e_vo   : num  NA 0.00 1.55e-14 0.00 -1.59e-14 ...
##  $ e_xB   : num  NA -0.475 -0.484 -0.493 -0.502 ...
## 'data.frame':    10 obs. of  8 variables:
##  $ id_r   : Factor w/ 10 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10
##  $ id_c   : Factor w/ 2 levels "DE","US": 1 1 1 1 1 1 1 1 1 1
##  $ tb     : num  1.33 1.53 1.93 2.53 3.33 ...
##  $ e_alpha: num  NA 0.107 0.145 0.225 0.362 ...
##  $ e_v    : num  NA 0.157 0.177 0.213 0.262 ...
##  $ e_vc   : num  NA 0.709 0.74 0.783 0.826 ...
##  $ e_vo   : num  NA 1.25e-11 2.92e-12 3.69e-12 4.62e-10 ...
##  $ e_xB   : num  NA -0.581 -0.588 -0.619 -0.666 ...
## 'data.frame':    10 obs. of  8 variables:
##  $ id_r   : Factor w/ 10 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10
##  $ id_c   : Factor w/ 2 levels "DE","US": 2 2 2 2 2 2 2 2 2 2
##  $ tb     : num  1.24 1.44 1.84 2.44 3.24 ...
##  $ e_alpha: num  NA 0.144 0.191 0.289 0.452 ...
##  $ e_v    : num  NA 0.199 0.225 0.27 0.327 ...
##  $ e_vc   : num  NA 0.72 0.754 0.797 0.833 ...
##  $ e_vo   : num  NA 2.41e-12 7.03e-13 8.72e-13 1.02e-10 ...
##  $ e_xB   : num  NA -0.633 -0.796 -0.906 -0.649 ...
## 'data.frame':    10 obs. of  8 variables:
##  $ id_r   : Factor w/ 10 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10
##  $ id_c   : Factor w/ 2 levels "DE","US": 1 1 1 1 1 1 1 1 1 1
##  $ wage   : num  102 152 202 252 302 ...
##  $ e_alpha: num  NA 9.44e-16 5.62e-15 -2.49e-15 1.55e-15 ...
##  $ e_v    : num  NA 0.00 2.27e-15 -6.03e-16 7.52e-16 ...
##  $ e_vc   : num  NA 0.00 8.22e-15 0.00 0.00 ...
##  $ e_vo   : num  NA 0.00 5.83e-16 0.00 0.00 ...
##  $ e_xB   : num  NA 0.00 -7.18e-14 0.00 0.00 ...
## 'data.frame':    10 obs. of  8 variables:
##  $ id_r   : Factor w/ 10 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10
##  $ id_c   : Factor w/ 2 levels "DE","US": 2 2 2 2 2 2 2 2 2 2
##  $ wage   : num  80.6 130.6 180.6 230.6 280.6 ...
##  $ e_alpha: num  NA 0.00 -8.75e-16 2.42e-15 -4.63e-15 ...
##  $ e_v    : num  NA 0 0 0 0 0 0 0 0 0
##  $ e_vc   : num  NA 0 0 0 0 0 0 0 0 0
##  $ e_vo   : num  NA 3.08e-16 0.00 0.00 0.00 ...
##  $ e_xB   : num  NA 0 0 0 0 0 0 0 0 0
```

```
# print results for policy parameters
select1 <- c("tauw","rho","tauq","taup","taud","taus","tauf","tauc")
for(nameparam in select1){
    text0 <- "OutMobileWork_"
    text1 <- nameparam
    text2 <- ".gdx"
    gdxfile <- paste(text0,text1,text2,sep="")
#    gdxfile = "OutMobileWork_eff.gdx"
#  nameparam <- select1
```

```
  for(nameidc in c("DE","US")) {
    dfpDE <- dfprintidc(gdxfile,nameparam,nameidc)
    dfpelastDE <- dfprintElast(gdxfile,nameparam,nameidc)
  }
}
```

```
## 'data.frame':    10 obs. of  8 variables:
##  $ id_r   : Factor w/ 10 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10
##  $ id_c   : Factor w/ 2 levels "DE","US": 1 1 1 1 1 1 1 1 1 1
##  $ tauw   : num  0 0.075 0.15 0.225 0.3 0.375 0.45 0.525 0.6 0.675
##  $ e_alpha: num  NA 0 0.0161 0.0316 0.0467 ...
##  $ e_v    : num  NA 0.00 -2.98e-16 2.98e-16 4.47e-16 ...
##  $ e_vc   : num  NA 0 0 0 0 0 0 0 0 0
##  $ e_vo   : num  NA 0.00 -1.91e-16 3.83e-16 0.00 ...
##  $ e_xB   : num  NA 0 0 0 0 0 0 0 0 0
## 'data.frame':    10 obs. of  8 variables:
##  $ id_r   : Factor w/ 10 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10
##  $ id_c   : Factor w/ 2 levels "DE","US": 2 2 2 2 2 2 2 2 2 2
##  $ tauw   : num  0 0.075 0.15 0.225 0.3 0.375 0.45 0.525 0.6 0.675
##  $ e_alpha: num  NA 0 0.0158 0.0311 0.0459 ...
##  $ e_v    : num  NA 0 0 0 0 0 0 0 0 0
##  $ e_vc   : num  NA 0 0 0 0 0 0 0 0 0
##  $ e_vo   : num  NA 0.00 1.91e-16 0.00 0.00 ...
##  $ e_xB   : num  NA 0 0 0 0 0 0 0 0 0
## 'data.frame':    10 obs. of  8 variables:
##  $ id_r   : Factor w/ 10 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10
##  $ id_c   : Factor w/ 2 levels "DE","US": 1 1 1 1 1 1 1 1 1 1
##  $ rho    : num  0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9
##  $ e_alpha: num  NA 0 -0.0155 -0.0314 -0.0478 ...
##  $ e_v    : num  NA 0.00 -1.49e-16 2.98e-16 -4.47e-16 ...
##  $ e_vc   : num  NA 0 0 0 0 0 0 0 0 0
##  $ e_vo   : num  NA 0 0 0 0 0 0 0 0 0
##  $ e_xB   : num  NA 0 0 0 0 0 0 0 0 0
## 'data.frame':    10 obs. of  8 variables:
##  $ id_r   : Factor w/ 10 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10
##  $ id_c   : Factor w/ 2 levels "DE","US": 2 2 2 2 2 2 2 2 2 2
##  $ rho    : num  0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9
##  $ e_alpha: num  NA 0 -0.00772 -0.01555 -0.02351 ...
##  $ e_v    : num  NA 0 0 0 0 0 0 0 0 0
##  $ e_vc   : num  NA 0 0 0 0 0 0 0 0 0
##  $ e_vo   : num  NA 0 0 0 0 0 0 0 0 0
##  $ e_xB   : num  NA 0 0 0 0 0 0 0 0 0
## 'data.frame':    10 obs. of  8 variables:
##  $ id_r   : Factor w/ 10 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10
##  $ id_c   : Factor w/ 2 levels "DE","US": 1 1 1 1 1 1 1 1 1 1
##  $ tauq   : num  0.5 1 1.5 2 2.5 3 3.5 4 4.5 5
##  $ e_alpha: num  NA 0.0615 0.116 0.1644 0.2078 ...
##  $ e_v    : num  NA 1.49e-16 -2.98e-16 4.47e-16 -5.96e-16 ...
##  $ e_vc   : num  NA 0 0 0 0 0 0 0 0 0
##  $ e_vo   : num  NA 0 0 0 0 0 0 0 0 0
##  $ e_xB   : num  NA 0 0 0 0 0 0 0 0 0
## 'data.frame':    10 obs. of  8 variables:
##  $ id_r   : Factor w/ 10 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10
##  $ id_c   : Factor w/ 2 levels "DE","US": 2 2 2 2 2 2 2 2 2 2
```

```
##  $ tauq   : num  0.5 1 1.5 2 2.5 3 3.5 4 4.5 5
##  $ e_alpha: num  NA 0.0503 0.0958 0.1371 0.1748 ...
##  $ e_v    : num  NA 0 0 0 0 0 0 0 0 0
##  $ e_vc   : num  NA 0 0 0 0 0 0 0 0 0
##  $ e_vo   : num  NA 0 0 0 0 0 0 0 0 0
##  $ e_xB   : num  NA 0 0 0 0 0 0 0 0 0
## 'data.frame':    10 obs. of  8 variables:
##  $ id_r   : Factor w/ 10 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10
##  $ id_c   : Factor w/ 2 levels "DE","US": 1 1 1 1 1 1 1 1 1 1
##  $ taup   : num  0 0.5 1 1.5 2 2.5 3 3.5 4 4.5
##  $ e_alpha: num  NA 0 -0.00463 -0.0093 -0.01402 ...
##  $ e_v    : num  NA 0.00 4.47e-16 2.98e-16 -4.47e-16 ...
##  $ e_vc   : num  NA 0.0 2.7e-15 0.0 0.0 ...
##  $ e_vo   : num  NA 0 0 0 0 0 0 0 0 0
##  $ e_xB   : num  NA 0.00 -2.36e-14 0.00 0.00 ...
## 'data.frame':    10 obs. of  8 variables:
##  $ id_r   : Factor w/ 10 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10
##  $ id_c   : Factor w/ 2 levels "DE","US": 2 2 2 2 2 2 2 2 2 2
##  $ taup   : num  0 0.5 1 1.5 2 2.5 3 3.5 4 4.5
##  $ e_alpha: num  NA 0 -0.00586 -0.01179 -0.01778 ...
##  $ e_v    : num  NA 0 0 0 0 0 0 0 0 0
##  $ e_vc   : num  NA 0 0 0 0 0 0 0 0 0
##  $ e_vo   : num  NA 0 0 0 0 0 0 0 0 0
##  $ e_xB   : num  NA 0 0 0 0 0 0 0 0 0
## 'data.frame':    10 obs. of  8 variables:
##  $ id_r   : Factor w/ 10 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10
##  $ id_c   : Factor w/ 2 levels "DE","US": 1 1 1 1 1 1 1 1 1 1
##  $ taud   : num  -0.5 -0.4 -0.3 -0.2 -0.1 0 0.1 0.2 0.3 0.4
##  $ e_alpha: num  NA 0.0463 0.0374 0.0283 0.0191 ...
##  $ e_v    : num  NA -0.000805 -0.000644 -0.000483 -0.000322 ...
##  $ e_vc   : num  NA -0.00365 -0.00292 -0.00219 -0.00146 ...
##  $ e_vo   : num  NA 9.57e-16 0.00 -5.74e-16 7.66e-16 ...
##  $ e_xB   : num  NA 0.0308 0.0248 0.0187 0.0126 ...
## 'data.frame':    10 obs. of  8 variables:
##  $ id_r   : Factor w/ 10 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10
##  $ id_c   : Factor w/ 2 levels "DE","US": 2 2 2 2 2 2 2 2 2 2
##  $ taud   : num  -0.5 -0.4 -0.3 -0.2 -0.1 0 0.1 0.2 0.3 0.4
##  $ e_alpha: num  NA 0.1102 0.0902 0.0692 0.0472 ...
##  $ e_v    : num  NA -0.001645 -0.001315 -0.000986 -0.000657 ...
##  $ e_vc   : num  NA -0.00599 -0.00478 -0.00358 -0.00239 ...
##  $ e_vo   : num  NA 9.57e-16 -7.66e-16 5.74e-16 -3.83e-16 ...
##  $ e_xB   : num  NA 0.2142 0.179 0.1406 0.0983 ...
## 'data.frame':    10 obs. of  8 variables:
##  $ id_r   : Factor w/ 10 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10
##  $ id_c   : Factor w/ 2 levels "DE","US": 1 1 1 1 1 1 1 1 1 1
##  $ taus   : num  -0.2 -0.1 0 0.1 0.2 0.3 0.4 0.5 0.6 0.7
##  $ e_alpha: num  NA 0.00402 0.00201 0 -0.00202 ...
##  $ e_v    : num  NA -1.08e-04 -5.39e-05 0.00 5.39e-05 ...
##  $ e_vc   : num  NA -0.000488 -0.000244 0 0.000244 ...
##  $ e_vo   : num  NA 0 0 0 0 ...
##  $ e_xB   : num  NA 0.00424 0.00213 0 -0.00214 ...
## 'data.frame':    10 obs. of  8 variables:
##  $ id_r   : Factor w/ 10 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10
##  $ id_c   : Factor w/ 2 levels "DE","US": 2 2 2 2 2 2 2 2 2 2
```

```
##  $ taus   : num  -0.2 -0.1 0 0.1 0.2 0.3 0.4 0.5 0.6 0.7
##  $ e_alpha: num  NA 0.00656 0.00329 0 -0.00331 ...
##  $ e_v    : num  NA -1.37e-04 -6.86e-05 0.00 6.86e-05 ...
##  $ e_vc   : num  NA -0.000497 -0.000249 0 0.000248 ...
##  $ e_vo   : num  NA 0.00 1.91e-16 0.00 1.91e-16 ...
##  $ e_xB   : num  NA 0.0223 0.0113 0 -0.0115 ...
## 'data.frame':    10 obs. of  8 variables:
##  $ id_r   : Factor w/ 10 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10
##  $ id_c   : Factor w/ 2 levels "DE","US": 1 1 1 1 1 1 1 1 1 1
##  $ tauf   : num  0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9
##  $ e_alpha: num  NA 0 0.024 0.0468 0.0686 ...
##  $ e_v    : num  NA 0.00 1.49e-16 -2.98e-16 4.47e-16 ...
##  $ e_vc   : num  NA 0 0 0 0 0 0 0 0 0
##  $ e_vo   : num  NA 0 0 0 0 0 0 0 0 0
##  $ e_xB   : num  NA 0 0 0 0 0 0 0 0 0
## 'data.frame':    10 obs. of  8 variables:
##  $ id_r   : Factor w/ 10 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10
##  $ id_c   : Factor w/ 2 levels "DE","US": 2 2 2 2 2 2 2 2 2 2
##  $ tauf   : num  0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9
##  $ e_alpha: num  NA 0 0.0279 0.0543 0.0793 ...
##  $ e_v    : num  NA 0 0 0 0 0 0 0 0 0
##  $ e_vc   : num  NA 0 0 0 0 0 0 0 0 0
##  $ e_vo   : num  NA 0 0 0 0 0 0 0 0 0
##  $ e_xB   : num  NA 0 0 0 0 0 0 0 0 0
## 'data.frame':    10 obs. of  8 variables:
##  $ id_r   : Factor w/ 10 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10
##  $ id_c   : Factor w/ 2 levels "DE","US": 1 1 1 1 1 1 1 1 1 1
##  $ tauc   : num  0 1 2 3 4 5 6 7 8 9
##  $ e_alpha: num  NA 0 -0.00376 -0.00755 -0.01137 ...
##  $ e_v    : num  NA 0 0.000401 0.000802 0.001203 ...
##  $ e_vc   : num  NA 0 0.00181 0.00362 0.00542 ...
##  $ e_vo   : num  NA 0.00 -1.91e-16 0.00 5.74e-16 ...
##  $ e_xB   : num  NA 0 -0.0161 -0.0328 -0.05 ...
## 'data.frame':    10 obs. of  8 variables:
##  $ id_r   : Factor w/ 10 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10
##  $ id_c   : Factor w/ 2 levels "DE","US": 2 2 2 2 2 2 2 2 2 2
##  $ tauc   : num  0 1 2 3 4 5 6 7 8 9
##  $ e_alpha: num  NA 0 -0.0093 -0.019 -0.0289 ...
##  $ e_v    : num  NA 0.00 4.27e-04 3.44e-04 -4.15e-16 ...
##  $ e_vc   : num  NA 0 0.00155 0.00124 0 ...
##  $ e_vo   : num  NA 0.00 3.83e-16 -3.83e-16 -5.74e-16 ...
##  $ e_xB   : num  NA 0.00 -7.63e-02 -6.66e-02 -7.35e-15 ...
```

Summarize Results of MC Simulations

Calculate probabilities

Plot results of MC Simulation

Evaluate results: Regressions and Interactions

```
# For dev version - install package to make tables1

#install library mass, needed for robust regressions
```

```
#install.packages("MASS")
library(MASS)

#library(knitr)
#install.packages("texReg")
#library(texreg)

#install.packages("censReg")
library(censReg)
```

```
## Lade nötiges Paket: maxLik
```

```
## Lade nötiges Paket: miscTools
```

```
##
## Please cite the 'maxLik' package as:
## Henningsen, Arne and Toomet, Ott (2011). maxLik: A package for maximum likelihood estimation in R. C
##
## If you have questions, suggestions, or comments regarding the 'maxLik' package, please use a forum o
## https://r-forge.r-project.org/projects/maxlik/
```

```
##
## Please cite the 'censReg' package as:
## Henningsen, Arne (2017). censReg: Censored Regression (Tobit) Models. R package version 0.5. http://
##
## If you have questions, suggestions, or comments regarding the 'censReg' package, please use a forum
## https://r-forge.r-project.org/projects/sampleselection/
```

```
#install.packages("stargazer")
library(stargazer)
```

```
##
## Please cite as:
```

```
##  Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary Statistics Tables.
```

```
##  R package version 5.2.2. https://CRAN.R-project.org/package=stargazer
```

```
stargazer(dfmcDE2.frame)
```

% Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
% Date and time: Do, Mrz 03, 2022 - 18:38:32

```
stargazer(dfmcUS2.frame)
```

% Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
% Date and time: Do, Mrz 03, 2022 - 18:38:34

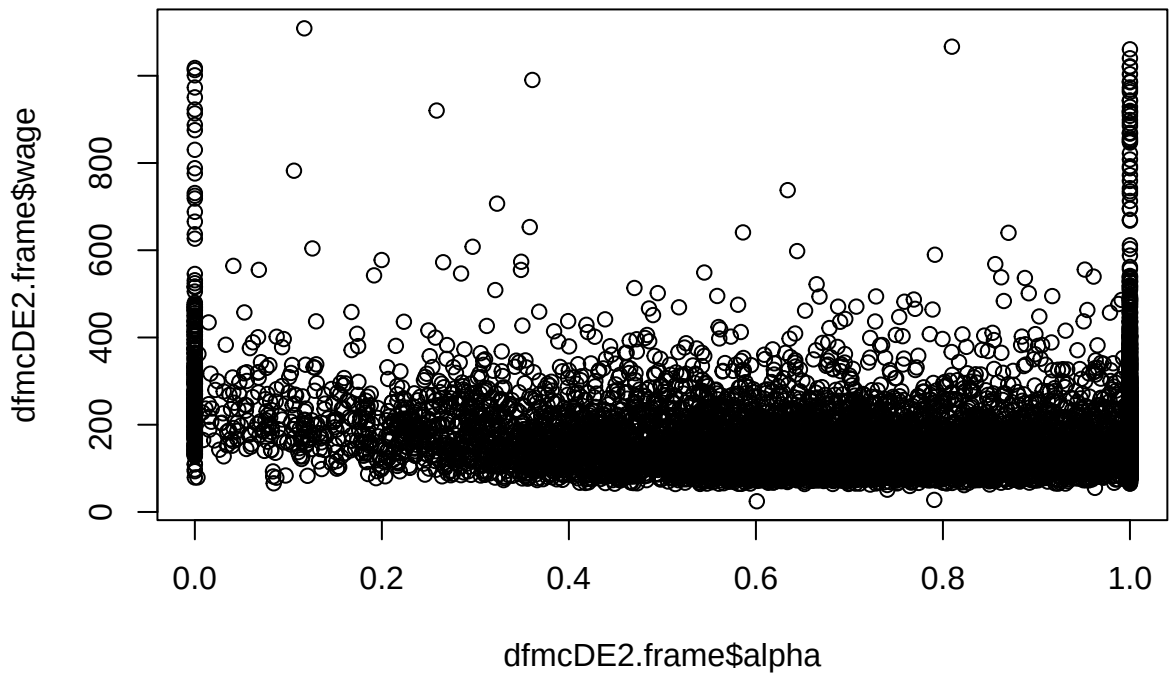
```
# correlation.matrix <- cor(dfmcDE2.frame[,c('wage','tkm','xbar','eff','beta',
#        'rmonth','sqmemployee','gm','gx','
#                                          gh','sp','dA','dB')])
# stargazer(correlation.matrix, title="Correlation Matrix DE")
# correlation.matrix <- cor(dfmcUS2.frame[,c('wage','tkm','xbar','eff','beta','rmonth','sqmemployee','g
#                                          'gh','sp','dA','dB')])
# stargazer(correlation.matrix, title="Correlation Matrix US")
plot(dfmcDE2.frame$alpha,dfmcDE2.frame$wage)
```

Table 1:

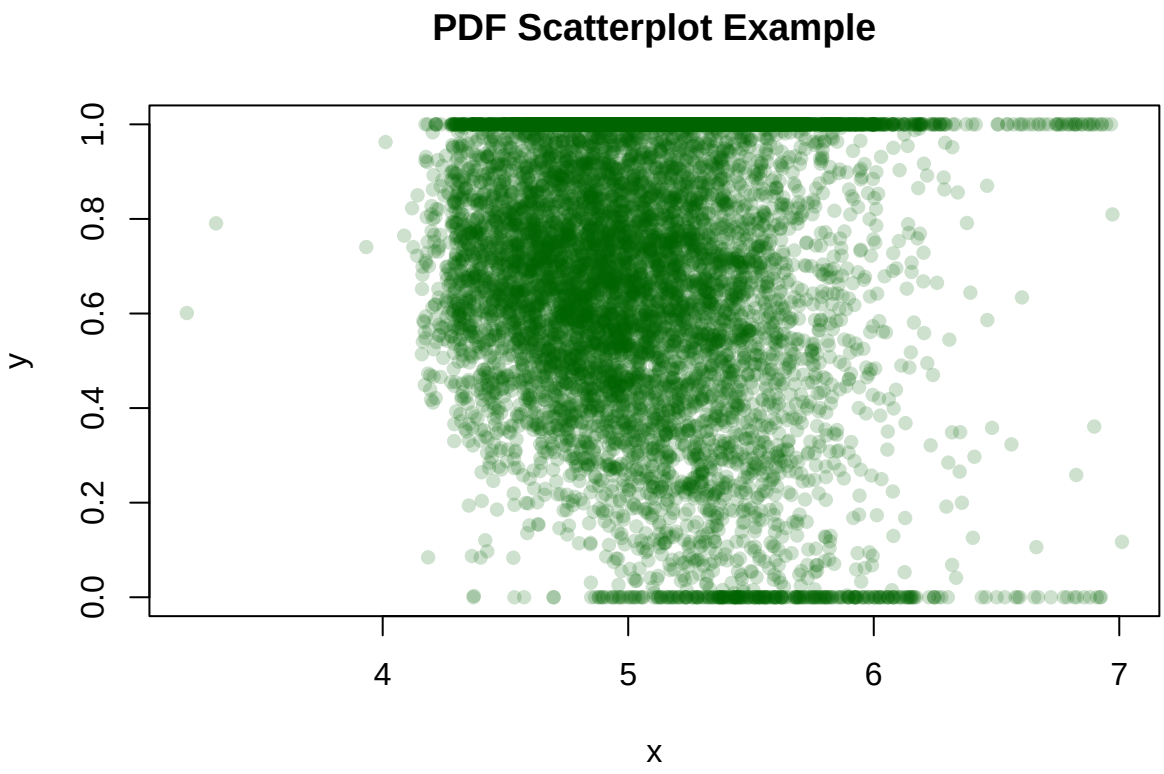
Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max
adistrib	10,000	45.000	0.000	45	45	45	45
alpha	10,000	0.725	0.277	0.000	0.552	1.000	1.000
share_v	10,000	0.475	0.198	0.188	0.320	0.586	1.000
share_vc	10,000	0.132	0.127	0.000	0.035	0.194	0.775
share_vo	10,000	0.343	0.157	0.182	0.239	0.388	1.000
share_xsum	10,000	80.919	3,192.889	0.004	0.736	3.569	273,975.400
share_x	10,000	0.135	0.149	0.000	0.037	0.203	1.000
share_xo	10,000	80.783	3,192.868	0.004	0.679	3.388	273,974.400
v	10,000	3.798	1.585	1.505	2.557	4.688	8.000
vc	10,000	1.055	1.017	0.000	0.279	1.549	6.203
vo	10,000	2.743	1.253	1.455	1.910	3.101	8.000
xbar	10,000	21.099	31.716	0.00000	1.537	27.438	332.863
xB	10,000	0.982	2.037	0.000	0.193	1.235	86.395
wage	10,000	176.552	98.403	24.590	115.189	208.876	1,108.667
wB	10,000	149.719	100.736	10.000	97.142	187.592	1,101.178
pb	10,000	0.850	3.281	0	0	0.4	113
Deltac	10,000	1.251	10.260	−56.105	−4.518	7.277	45.402
tkm	10,000	1.353	4.820	0.016	0.060	0.224	26.731
expwage	10,000	124.123	102.838	−26.796	59.032	162.073	1,107.473
hours	10,000	8.000	0.000	8	8	8	8
tauw	10,000	0.484	0.000	0.484	0.484	0.484	0.484
tauf	10,000	0.000	0.000	0	0	0	0
taus	10,000	0.000	0.000	0	0	0	0
taud	10,000	0.000	0.000	0	0	0	0
tauc	10,000	0.000	0.000	0	0	0	0
taup	10,000	0.000	0.000	0	0	0	0
tauq	10,000	0.500	0.000	0.500	0.500	0.500	0.500
rho	10,000	0.200	0.000	0.200	0.200	0.200	0.200
eff	10,000	1.998	0.409	1.009	1.701	2.293	2.983
beta	10,000	−0.178	20.495	−211.878	−11.411	10.834	187.541
dA	10,000	10.000	0.000	10	10	10	10
dB	10,000	9.971	2.048	5.011	8.469	11.449	14.936
gm	10,000	0.757	0.201	0.306	0.612	0.903	1.208
gx	10,000	0.047	0.005	0.038	0.042	0.051	0.055
gh	10,000	4.005	1.709	1.040	2.547	5.471	7.000
rmonth	10,000	21.391	8.412	8.274	13.991	27.676	45.285
sp	10,000	0.498	0.206	0.007	0.348	0.643	0.993
sqmemployee	10,000	26.830	4.659	18.807	22.768	30.874	34.898
tb	10,000	3.026	0.573	2.036	2.529	3.528	4.019

Table 2:

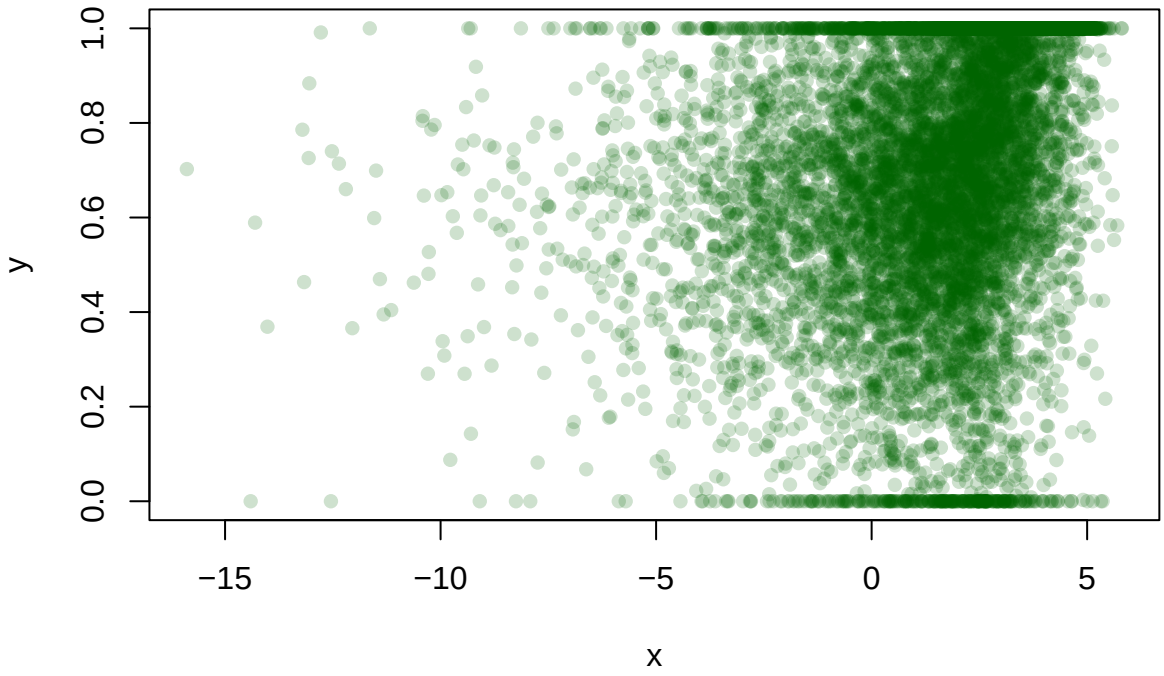
Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max
adistrib	10,000	60.000	0.000	60	60	60	60
alpha	10,000	0.686	0.304	0	0.5	1	1
share_v	10,000	0.391	0.166	0.184	0.279	0.444	1.000
share_vc	10,000	0.051	0.055	0.000	0.033	0.043	0.798
share_vo	10,000	0.340	0.159	0.182	0.238	0.378	1.000
share_xsum	10,000	173,443.400	12,125,841.000	0.047	2.099	5.626	1,192,170,945.000
share_x	10,000	0.105	0.184	0.000	0.023	0.107	1.000
share_xo	10,000	173,443.300	12,125,841.000	0.023	2.039	5.561	1,192,170,945.000
v	10,000	3.129	1.329	1.473	2.236	3.554	8.000
vc	10,000	0.407	0.440	0.000	0.263	0.343	6.381
vo	10,000	2.722	1.274	1.454	1.906	3.023	8.000
xbar	10,000	30.688	55.487	0.000	0.700	35.767	668.988
xB	10,000	1.445	6.122	0.000	0.001	1.944	272.591
wage	10,000	205.760	127.099	3.939	112.887	272.790	1,100.362
wB	10,000	194.857	128.554	10.000	96.647	262.631	1,137.274
pb	10,000	1.590	5.622	0.000	0.000	0.347	89.808
Deltac	10,000	−6.385	17.335	−126.664	−15.260	4.473	96.961
tkm	10,000	2.707	7.254	0.007	0.010	0.415	26.689
expwage	10,000	163.479	129.841	−51.918	66.882	234.119	1,081.960
hours	10,000	8.000	0.000	8	8	8	8
tauw	10,000	0.202	0.000	0.202	0.202	0.202	0.202
tauf	10,000	0.000	0.000	0	0	0	0
taus	10,000	0.000	0.000	0	0	0	0
taud	10,000	0.000	0.000	0	0	0	0
tauc	10,000	0.000	0.000	0	0	0	0
taup	10,000	0.000	0.000	0	0	0	0
tauq	10,000	0.500	0.000	0.500	0.500	0.500	0.500
rho	10,000	0.200	0.000	0.200	0.200	0.200	0.200
eff	10,000	2.010	0.405	1.027	1.722	2.297	2.986
beta	10,000	−0.263	24.549	−157.191	−11.846	11.203	159.641
dA	10,000	10.000	0.000	10	10	10	10
dB	10,000	10.032	2.026	5.116	8.596	11.479	14.921
gm	10,000	0.743	0.215	0.261	0.589	0.900	1.226
gx	10,000	0.047	0.014	0.024	0.036	0.059	0.071
gh	10,000	7.687	0.194	7.247	7.547	7.830	8.125
rmonth	10,000	91.125	30.299	22.842	68.782	113.374	159.669
sp	10,000	0.500	0.204	0.004	0.355	0.647	0.994
sqmemployee	10,000	18.557	5.376	9.298	13.869	23.223	27.868
tb	10,000	1.077	1.952	0.055	0.164	1.333	13.475




```
y <- dfmcDE2.frame$alpha
x <- log(dfmcDE2.frame$wage)
plot(x,y, main="PDF Scatterplot Example", col=rgb(0,100,0,50,maxColorValue=255), pch=16)
```




```
fit <- lm(y~x)
y <- dfmcDE2.frame$alpha
x <- log(dfmcDE2.frame$xbar)
plot(x,y, main="PDF Scatterplot Example", col=rgb(0,100,0,50,maxColorValue=255), pch=16)
```




```
#dev.off()

# add rent per sqm/employee to dataframe
dfmcDE2.frame <- mutate(dfmcDE2.frame, r = rmonth*sqmemployee)
# add rent per sqm/employee to dataframe
dfmcUS2.frame <- mutate(dfmcUS2.frame, r = rmonth*sqmemployee)

#
# regshare_vDE <- lm(share_v~ wage + tkm + log(eff) + xbar
#                    + gm
#                    + beta
#                    + rmonth + sqmemployee + gmethod = "MM", data = dfmcDE2.frame)
# summary(regshare_vDE)


regshare_vDE <- censReg(share_v~log(wage) + log(tkm) + log(eff) + log(xbar)
                  + log(gm)
                  + beta
                  + log(r) + log(gx) + log(gh)
                  + log(sp)  + dB,  left=0, right=1, data = dfmcDE2.frame)
summary(regshare_vDE)
```

Call: censReg(formula = share_v ~ log(wage) + log(tkm) + log(eff) + log(xbar) + log(gm) + beta + log(r) + log(gx) + log(gh) + log(sp) + dB, left = 0, right = 1, data = dfmcDE2.frame)

Observations: Total Left-censored Uncensored Right-censored 10000 0 9711 289

Coefficients: Estimate Std. error t value Pr(> t)
(Intercept) 9.486e-01 3.204e-02 29.608 <2e-16 ***log(wage) 1.316e-03 2.106e-03 0.625 0.532***
log(tkm) 1.127e-01 2.222e-03 50.718 <2e-16 ** log(eff) -6.835e-01 4.425e-03 -154.475 <2e-16 ***log(xbar)***
1.033e-01 1.392e-03 74.198 <2e-16 ** log(gm) -3.095e-03 3.273e-03 -0.946 0.344
beta 2.103e-05 4.564e-05 0.461 0.645
log(r) -9.852e-04 2.144e-03 -0.459 0.646
log(gx) -1.096e-02 8.614e-03 -1.272 0.203
log(gh) 1.601e-03 1.860e-03 0.861 0.389
log(sp) -1.122e-03 1.725e-03 -0.651 0.515
dB -2.623e-04 4.569e-04 -0.574 0.566
logSigma -2.372e+00 7.229e-03 -328.187 <2e-16 *** — Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Newton-Raphson maximisation, 9 iterations Return code 8: successive function values within relative tolerance limit (reltol) Log-likelihood: 8908.63 on 13 Df

```
regshare_vDEm <- margEff(regshare_vDE ,vcov = NULL,
   calcVCov = TRUE, returnJacobian = FALSE)
summary(regshare_vDEm)
```

```
       Marg. Eff.  Std. Error   t value Pr(>|t|)
```

log(wage) 1.3158e-03 2.1057e-03 0.6249 0.5321
log(tkm) 1.1272e-01 2.2224e-03 50.7180 <2e-16 ***log(eff) -6.8349e-01 4.4246e-03 -154.4751 <2e-16***
log(xbar) 1.0330e-01 1.3922e-03 74.1980 <2e-16 *** log(gm) -3.0955e-03 3.2732e-03 -0.9457 0.3443
beta 2.1034e-05 4.5638e-05 0.4609 0.6449
log(r) -9.8519e-04 2.1441e-03 -0.4595 0.6459
log(gx) -1.0958e-02 8.6142e-03 -1.2721 0.2034
log(gh) 1.6007e-03 1.8602e-03 0.8605 0.3895
log(sp) -1.1224e-03 1.7253e-03 -0.6506 0.5153
dB -2.6229e-04 4.5690e-04 -0.5741 0.5659
— Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

```
regshare_vcDE <- censReg(share_vc~log(wage) + log(tkm) + log(eff) + log(xbar)
                 + log(gm)
                 + beta
                 + log(r) + log(gx) + log(gh)
                 + log(sp)  + dB,  left=0, right=1, data = dfmcDE2.frame)
 summary(regshare_vcDE)
```

Call: censReg(formula = share_vc ~ log(wage) + log(tkm) + log(eff) + log(xbar) + log(gm) + beta + log(r) + log(gx) + log(gh) + log(sp) + dB, left = 0, right = 1, data = dfmcDE2.frame)

Observations: Total Left-censored Uncensored Right-censored 10000 151 9849 0

Coefficients: Estimate Std. error t value Pr(> t)
(Intercept) 2.317e-01 1.467e-02 15.792 < 2e-16 ***log(wage) -1.003e-04 9.634e-04 -0.104 0.917***
log(tkm) 1.907e-01 1.148e-03 166.064 < 2e-16 ** log(eff) 9.250e-03 2.000e-03 4.626 3.73e-06 ***log(xbar)***
1.652e-01 7.564e-04 218.410 < 2e-16 ** log(gm) -2.258e-03 1.498e-03 -1.507 0.132
beta 9.771e-05 2.089e-05 4.676 2.92e-06 ***log(r) -3.427e-04 9.819e-04 -0.349 0.727***
log(gx) 4.425e-03 3.941e-03 1.123 0.262
log(gh) 1.330e-03 8.500e-04 1.565 0.118
log(sp) -7.039e-05 7.888e-04 -0.089 0.929
dB -2.322e-05 2.090e-04 -0.111 0.912
logSigma -3.159e+00 7.125e-03 -443.416 < 2e-16 ** — Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Newton-Raphson maximisation, 9 iterations Return code 8: successive function values within relative tolerance limit (reltol) Log-likelihood: 17140.03 on 13 Df

```
regshare_vcDEm <- margEff(regshare_vcDE ,vcov = NULL,
   calcVCov = TRUE, returnJacobian = FALSE)
summary(regshare_vcDEm)
```

```
       Marg. Eff.  Std. Error  t value  Pr(>|t|)
```

log(wage) -1.0018e-04 9.6176e-04 -0.1042 0.9170
log(tkm) 1.9039e-01 1.1463e-03 166.0953 < 2.2e-16 ***log(eff) 9.2341e-03 1.9961e-03 4.6260 3.775e-06***
log(xbar) 1.6493e-01 7.5487e-04 218.4862 < 2.2e-16 ***log(gm) -2.2543e-03 1.4956e-03 -1.5073 0.1318***
beta 9.7542e-05 2.0858e-05 4.6764 2.958e-06 log(r) -3.4208e-04 9.8023e-04 -0.3490 0.7271
log(gx) 4.4175e-03 3.9343e-03 1.1228 0.2615
log(gh) 1.3279e-03 8.4849e-04 1.5650 0.1176
log(sp) -7.0270e-05 7.8746e-04 -0.0892 0.9289
dB -2.3179e-05 2.0866e-04 -0.1111 0.9116
— Signif. codes: 0 ‘***’ ***0.001*** ‘**’ *0.01* ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

```
regshare_voDE <- censReg(share_vo ~ log(wage) + log(tkm) + log(eff) + log(xbar)
                 + log(gm)
                 + beta
                 + log(r) + log(gx) + log(gh)
                 + log(sp) + dB,  right=1, data = dfmcDE2.frame)
summary(regshare_voDE)
```

Call: censReg(formula = share_vo ~ log(wage) + log(tkm) + log(eff) + log(xbar) + log(gm) + beta + log(r) + log(gx) + log(gh) + log(sp) + dB, right = 1, data = dfmcDE2.frame)

Observations: Total Left-censored Uncensored Right-censored 10000 0 9999 1

Coefficients: Estimate Std. error t value Pr(> t)
(Intercept) 7.922e-01 1.996e-02 39.687 < 2e-16 ***log(wage) 1.009e-03 1.312e-03 0.769 0.442***
log(tkm) -5.392e-03 1.386e-03 -3.891 9.97e-05 log(eff) -6.795e-01 2.723e-03 -249.507 < 2e-16 ***log(xbar) -4.448e-03 8.675e-04 -5.127 2.94e-07*** log(gm) -5.701e-04 2.039e-03 -0.280 0.780
beta -2.271e-05 2.845e-05 -0.798 0.425
log(r) -1.927e-03 1.336e-03 -1.442 0.149
log(gx) -2.869e-03 5.367e-03 -0.534 0.593
log(gh) 3.782e-04 1.159e-03 0.326 0.744
log(sp) -1.613e-03 1.075e-03 -1.500 0.134
dB -1.131e-04 2.846e-04 -0.398 0.691
logSigma -2.843e+00 7.072e-03 -402.012 < 2e-16 *** — Signif. codes: 0 ‘***’ ***0.001*** ‘**’ *0.01* ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Newton-Raphson maximisation, 10 iterations Return code 1: gradient close to zero (gradtol) Log-likelihood: 14236.32 on 13 Df

```
regshare_voDEm <- margEff(regshare_voDE ,vcov = NULL,
   calcVCov = TRUE, returnJacobian = FALSE)
summary(regshare_voDEm)
```

```
       Marg. Eff.  Std. Error   t value  Pr(>|t|)
```

log(wage) 1.0092e-03 1.3119e-03 0.7693 0.4417536
log(tkm) -5.3917e-03 1.3856e-03 -3.8913 0.0001004 ***log(eff) -6.7953e-01 2.7235e-03 -249.5069 < 2.2e-16*** log(xbar) -4.4478e-03 8.6750e-04 -5.1271 2.997e-07 *** log(gm) -5.7007e-04 2.0392e-03 -0.2796 0.7798260
beta -2.2711e-05 2.8446e-05 -0.7984 0.4246570
log(r) -1.9270e-03 1.3359e-03 -1.4424 0.1492122
log(gx) -2.8687e-03 5.3675e-03 -0.5345 0.5930305
log(gh) 3.7818e-04 1.1586e-03 0.3264 0.7441202
log(sp) -1.6126e-03 1.0750e-03 -1.5000 0.1336370

dB -1.1315e-04 2.8462e-04 -0.3976 0.6909593
— Signif. codes: 0 ‘’ ***0.001*** ’’ *0.01* ’’ 0.05 ‘.’ 0.1 ’ ’ 1

```
regalphaDE2 <- censReg(alpha~log(wage) + log(tkm) + log(eff) + log(xbar)
                 + log(gm)
                 + beta
                 + log(r)  + log(gx) + log(gh)
                 + log(sp)  + dB,  left=0, right=1, data = dfmcDE2.frame)
summary(regalphaDE2)
```

Call: censReg(formula = alpha ~ log(wage) + log(tkm) + log(eff) + log(xbar) + log(gm) + beta + log(r) + log(gx) + log(gh) + log(sp) + dB, left = 0, right = 1, data = dfmcDE2.frame)

Observations: Total Left-censored Uncensored Right-censored 10000 289 6679 3032

Coefficients: Estimate Std. error t value Pr(> t)
(Intercept) 0.1373822 0.0576738 2.382 0.0172 *
log(wage) 0.0568206 0.0041475 13.700 < 2e-16 ***log(tkm) 0.1301906 0.0038767 33.583 < 2e-16*** log(eff) 0.0725747 0.0078132 9.289 < 2e-16 ***log(xbar) 0.1267035 0.0024848 50.992 < 2e-16*** log(gm) 0.0380228 0.0058289 6.523 6.89e-11 ***beta 0.0172285 0.0001195 144.143 < 2e-16*** log(r) 0.0692233 0.0038409 18.023 < 2e-16 ***log(gx) -0.0269499 0.0153666 -1.754 0.0795 .***
log(gh) -0.0594736 0.0033616 -17.692 < 2e-16 log(sp) 0.0050810 0.0030794 1.650 0.0989 .
dB -0.0053354 0.0008128 -6.564 5.23e-11 ***logSigma -1.8983337 0.0088862 -213.627 < 2e-16*** — Signif. codes: 0 ‘’ ***0.001*** ’’ *0.01* ’’ 0.05 ‘.’ 0.1 ’ ’ 1

Newton-Raphson maximisation, 9 iterations Return code 1: gradient close to zero (gradtol) Log-likelihood: 1611.151 on 13 Df

```
regalphaDE2m <- margEff(regalphaDE2 ,vcov = NULL,
   calcVCov = TRUE, returnJacobian = FALSE)
summary(regalphaDE2m)
```

```
       Marg. Eff.  Std. Error  t value  Pr(>|t|)
```

log(wage) 0.05195185 0.00376773 13.7887 < 2.2e-16 ***log(tkm) 0.11903504 0.00352660 33.7535 < 2.2e-16*** log(eff) 0.06635604 0.00715367 9.2758 < 2.2e-16 ***log(xbar) 0.11584670 0.00224693 51.5577 < 2.2e-16*** log(gm) 0.03476477 0.00532779 6.5252 7.122e-11 ***beta 0.01575226 0.00010262 153.4985 < 2.2e-16*** log(r) 0.06329176 0.00350800 18.0421 < 2.2e-16 ***log(gx) -0.02464067 0.01404960 -1.7538 0.07949 .***
log(gh) -0.05437750 0.00307211 -17.7004 < 2.2e-16 log(sp) 0.00464565 0.00281547 1.6500 0.09897 .
dB -0.00487820 0.00074312 -6.5645 5.480e-11 *** — Signif. codes: 0 ‘’ ***0.001*** ’’ *0.01* ’’ 0.05 ‘.’ 0.1 ’ ’ 1

```
stargazer(regalphaDE2,regshare_vDE,regshare_vcDE,regshare_voDE,
          title="Regression Results (Germany)",algin=TRUE,
          dep.var.labels=c("alpha","v","vc","vo","xB"), omit.stat=c("LL","ser","f"),
         no.space=TRUE)
```

% Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
% Date and time: Do, Mrz 03, 2022 - 18:38:47

% Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
% Date and time: Do, Mrz 03, 2022 - 18:38:47

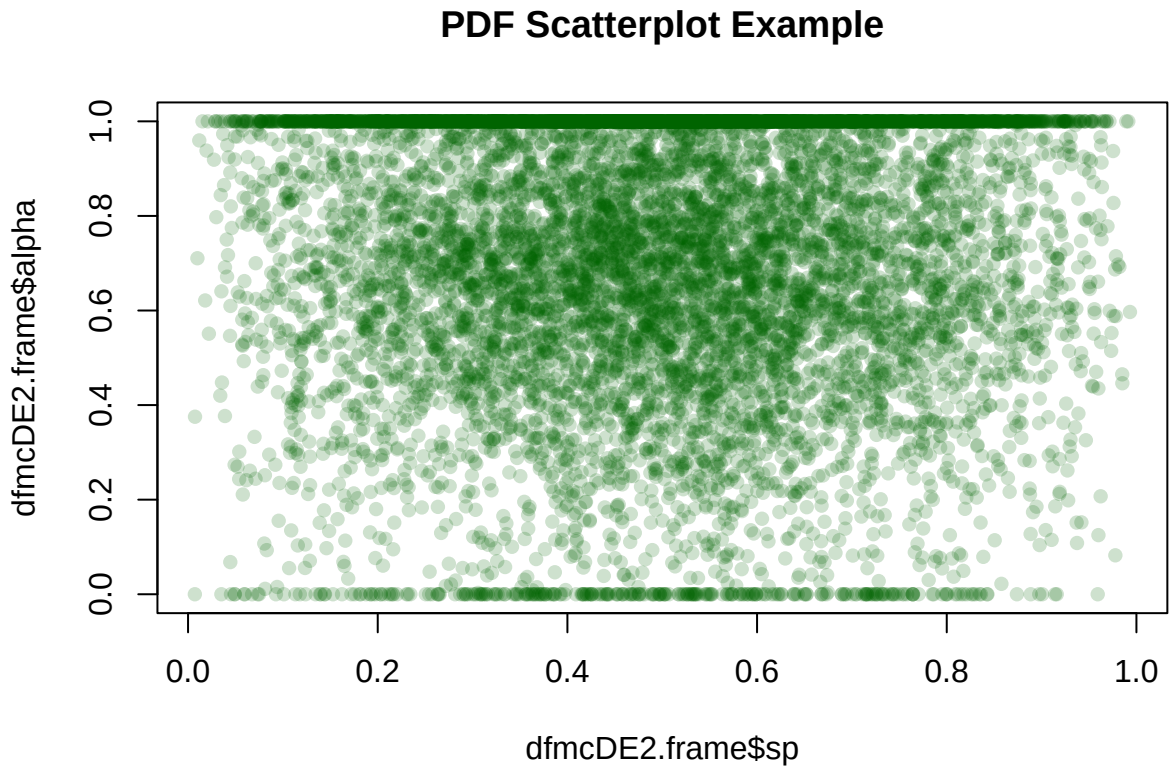
```
plot(dfmcDE2.frame$sp,dfmcDE2.frame$alpha, main="PDF Scatterplot Example",
     col=rgb(0,100,0,50,maxColorValue=255), pch=16)
```

Table 3: Regression Results (Germany)

	Dependent variable:			
	alpha	v	vc	vo
	(1)	(2)	(3)	(4)
log(wage)	0.057^{***}	0.001	−0.0001	0.001
	(0.004)	(0.002)	(0.001)	(0.001)
log(tkm)	0.130^{***}	0.113^{***}	0.191^{***}	-0.005^{***}
	(0.004)	(0.002)	(0.001)	(0.001)
log(eff)	0.073^{***}	-0.683^{***}	0.009^{***}	-0.680^{***}
	(0.008)	(0.004)	(0.002)	(0.003)
log(xbar)	0.127^{***}	0.103^{***}	0.165^{***}	-0.004^{***}
	(0.002)	(0.001)	(0.001)	(0.001)
log(gm)	0.038^{***}	−0.003	−0.002	−0.001
	(0.006)	(0.003)	(0.001)	(0.002)
beta	0.017^{***}	0.00002	0.0001^{***}	−0.00002
	(0.0001)	(0.00005)	(0.00002)	(0.00003)
log(r)	0.069^{***}	−0.001	−0.0003	−0.002
	(0.004)	(0.002)	(0.001)	(0.001)
log(gx)	-0.027^{*}	−0.011	0.004	−0.003
	(0.015)	(0.009)	(0.004)	(0.005)
log(gh)	-0.059^{***}	0.002	0.001	0.0004
	(0.003)	(0.002)	(0.001)	(0.001)
log(sp)	0.005^{*}	−0.001	−0.0001	−0.002
	(0.003)	(0.002)	(0.001)	(0.001)
dB	-0.005^{***}	−0.0003	−0.00002	−0.0001
	(0.001)	(0.0005)	(0.0002)	(0.0003)
logSigma	-1.898^{***}	-2.372^{***}	-3.159^{***}	-2.843^{***}
	(0.009)	(0.007)	(0.007)	(0.007)
Constant	0.137^{**}	0.949^{***}	0.232^{***}	0.792^{***}
	(0.058)	(0.032)	(0.015)	(0.020)
Observations	10,000	10,000	10,000	10,000
Akaike Inf. Crit.	−3,196.302	−17,791.260	−34,254.060	−28,446.640
Bayesian Inf. Crit.	−3,102.567	−17,697.530	−34,160.330	−28,352.900
Note:				*p<0.1; **p<0.05; ***p<0.01

Table 4: Regression Results (Germany)

TRUE




```
regshare_vUS <- censReg(share_v~log(wage) + log(tkm) + log(eff) + log(xbar)
                  + log(gm)
                  + beta
                  + log(r) + log(gx) + log(gh)
                  + log(sp) + dB,  left=0, right=1, data = dfmcUS2.frame)
summary(regshare_vUS)
```

Call: censReg(formula = share_v ~ log(wage) + log(tkm) + log(eff) + log(xbar) + log(gm) + beta + log(r) + log(gx) + log(gh) + log(sp) + dB, left = 0, right = 1, data = dfmcUS2.frame)

Observations: Total Left-censored Uncensored Right-censored 10000 0 9831 169

Coefficients: Estimate Std. error t value Pr(> t)
(Intercept) 9.456e-01 6.468e-02 14.620 <2e-16 ***
log(wage) -2.043e-04 1.135e-03 -0.180 0.857
log(tkm) 9.322e-03 9.350e-04 9.970 <2e-16 ** log(eff) -6.920e-01 3.697e-03 -187.175 <2e-16 ***
log(xbar) 1.405e-02 6.768e-04 20.754 <2e-16 ** log(gm) -2.607e-03 2.443e-03 -1.067 0.286
beta 1.014e-06 3.154e-05 0.032 0.974
log(r) -2.281e-03 1.590e-03 -1.435 0.151
log(gx) -3.783e-04 2.502e-03 -0.151 0.880
log(gh) -2.840e-02 3.067e-02 -0.926 0.354
log(sp) -1.987e-03 1.437e-03 -1.383 0.167
dB -7.701e-04 3.824e-04 -2.014 0.044 *
logSigma -2.559e+00 7.178e-03 -356.541 <2e-16 *** — Signif. codes: 0 ‘***’ ***0.001*** ‘**’ *0.01* ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Newton-Raphson maximisation, 9 iterations Return code 8: successive function values within relative tolerance limit (reltol) Log-likelihood: 10927.64 on 13 Df

```
regshare_vUSm <- margEff(regshare_vUS ,vcov = NULL,
   calcVCov = TRUE, returnJacobian = FALSE)
summary(regshare_vUSm)
```

```
       Marg. Eff.  Std. Error   t value Pr(>|t|)
```

log(wage) -2.0431e-04 1.1354e-03 -0.1799 0.85720
log(tkm) 9.3219e-03 9.3497e-04 9.9703 < 2e-16 ***log(eff) -6.9199e-01 3.6970e-03 -187.1749 < 2e-16***
log(xbar) 1.4046e-02 6.7680e-04 20.7535 < 2e-16 ** *log(gm) -2.6069e-03 2.4430e-03 -1.0671 0.28596*
beta 1.0136e-06 3.1539e-05 0.0321 0.97436
log(r) -2.2807e-03 1.5897e-03 -1.4347 0.15141
log(gx) -3.7827e-04 2.5024e-03 -0.1512 0.87985
log(gh) -2.8405e-02 3.0667e-02 -0.9262 0.35435
log(sp) -1.9871e-03 1.4371e-03 -1.3827 0.16678
dB -7.7014e-04 3.8240e-04 -2.0139 0.04404
— Signif. codes: 0 ‘***’ ***0.001*** ‘**’ *0.01* ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

```
regshare_vcUS <- censReg(share_vc ~ log(wage) + log(tkm) + log(eff) + log(xbar)
                 + log(gm)
                 + beta
                 + log(r) + log(gx) + log(gh)
                 + log(sp) + dB,  left=0, right=1, data = dfmcUS2.frame)
 summary(regshare_vcUS)
```

Call: censReg(formula = share_vc ~ log(wage) + log(tkm) + log(eff) + log(xbar) + log(gm) + beta + log(r) + log(gx) + log(gh) + log(sp) + dB, left = 0, right = 1, data = dfmcUS2.frame)

Observations: Total Left-censored Uncensored Right-censored 10000 317 9683 0

Coefficients: Estimate Std. error t value Pr(> t)
(Intercept) 1.210e-01 3.540e-02 3.418 0.00063 ***log(wage) 3.741e-05 6.207e-04 0.060 0.95194***
log(tkm) 4.364e-02 7.725e-04 56.495 < 2e-16 log(eff) 9.186e-03 2.017e-03 4.555 5.23e-06 ***log(xbar) 4.424e-02 6.339e-04 69.803 < 2e-16*** log(gm) 9.212e-04 1.335e-03 0.690 0.49014
beta 3.909e-05 1.726e-05 2.264 0.02355 *
log(r) -9.603e-04 8.671e-04 -1.107 0.26810
log(gx) -2.330e-04 1.368e-03 -0.170 0.86475
log(gh) -1.007e-02 1.679e-02 -0.600 0.54844
log(sp) -2.052e-06 7.860e-04 -0.003 0.99792
dB -2.915e-04 2.087e-04 -1.396 0.16258
logSigma -3.177e+00 7.185e-03 -442.194 < 2e-16 *** — Signif. codes: 0 ‘***’ ***0.001*** ‘**’ *0.01* ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Newton-Raphson maximisation, 10 iterations Return code 8: successive function values within relative tolerance limit (reltol) Log-likelihood: 16972.12 on 13 Df

```
regshare_vcUSm <- margEff(regshare_vcUS ,vcov = NULL,
   calcVCov = TRUE, returnJacobian = FALSE )
summary(regshare_vcUSm )
```

```
       Marg. Eff.  Std. Error t value  Pr(>|t|)
```

log(wage) 3.2090e-05 5.3242e-04 0.0603 0.95194
log(tkm) 3.7436e-02 6.5610e-04 57.0592 < 2.2e-16 ***log(eff) 7.8795e-03 1.7299e-03 4.5547 5.307e-06***
log(xbar) 3.7952e-02 5.3787e-04 70.5603 < 2.2e-16 ** *log(gm) 7.9017e-04 1.1450e-03 0.6901 0.49016*
beta 3.3532e-05 1.4807e-05 2.2646 0.02356
log(r) -8.2369e-04 7.4377e-04 -1.1074 0.26813
log(gx) -1.9986e-04 1.1734e-03 -0.1703 0.86475
log(gh) -8.6405e-03 1.4399e-02 -0.6001 0.54846
log(sp) -1.7598e-06 6.7424e-04 -0.0026 0.99792

dB -2.5003e-04 1.7905e-04 -1.3964 0.16262
— Signif. codes: 0 ‘’ ***0.001*** ’’ *0.01* ’’ 0.05 ‘.’ 0.1 ’ ’ 1

```
regshare_voUS <- censReg(share_vo~log(wage) + log(tkm) + log(eff) + log(xbar)
                  + log(gm)
                  + beta
                  + log(r) + log(gx) + log(gh)
                  + log(sp) + dB,  left=0, right=1, data = dfmcUS2.frame)
summary(regshare_voUS)
```

Call: censReg(formula = share_vo ~ log(wage) + log(tkm) + log(eff) + log(xbar) + log(gm) + beta + log(r) + log(gx) + log(gh) + log(sp) + dB, left = 0, right = 1, data = dfmcUS2.frame)

Observations: Total Left-censored Uncensored Right-censored 10000 0 9957 43

Coefficients: Estimate Std. error t value Pr(> t)
(Intercept) 8.316e-01 5.194e-02 16.013 <2e-16 ***log(wage) -3.852e-04 9.116e-04 -0.423 0.6726***
log(tkm) -1.045e-03 7.508e-04 -1.392 0.1640
log(eff) -6.982e-01 2.959e-03 -235.920 <2e-16 ** log(xbar) -6.513e-04 5.435e-04 -1.198 0.2308
log(gm) -3.329e-03 1.962e-03 -1.697 0.0897 .
beta 1.563e-05 2.533e-05 0.617 0.5373
log(r) -1.336e-03 1.276e-03 -1.046 0.2954
log(gx) -2.417e-04 2.009e-03 -0.120 0.9042
log(gh) -3.861e-03 2.463e-02 -0.157 0.8754
log(sp) -1.770e-03 1.154e-03 -1.533 0.1252
dB -4.156e-04 3.070e-04 -1.354 0.1758
logSigma -2.778e+00 7.099e-03 -391.342 <2e-16 *** — Signif. codes: 0 ‘’ ***0.001*** ’’ *0.01* ’’ 0.05 ‘.’ 0.1 ’ ’ 1

Newton-Raphson maximisation, 10 iterations Return code 1: gradient close to zero (gradtol) Log-likelihood: 13451.26 on 13 Df

```
regshare_voUSm <- margEff(regshare_voUS ,vcov = NULL,
   calcVCov = TRUE, returnJacobian = FALSE  )
summary(regshare_voUSm)
```

```
       Marg. Eff.  Std. Error   t value Pr(>|t|)
```

log(wage) -3.8521e-04 9.1161e-04 -0.4226 0.67263
log(tkm) -1.0451e-03 7.5084e-04 -1.3919 0.16400
log(eff) -6.9815e-01 2.9593e-03 -235.9201 < 2e-16 *** log(xbar) -6.5131e-04 5.4354e-04 -1.1983 0.23085
log(gm) -3.3291e-03 1.9615e-03 -1.6972 0.08969 .
beta 1.5625e-05 2.5327e-05 0.6169 0.53730
log(r) -1.3356e-03 1.2764e-03 -1.0464 0.29541
log(gx) -2.4175e-04 2.0093e-03 -0.1203 0.90424
log(gh) -3.8613e-03 2.4626e-02 -0.1568 0.87541
log(sp) -1.7697e-03 1.1540e-03 -1.5334 0.12520
dB -4.1561e-04 3.0702e-04 -1.3537 0.17587
— Signif. codes: 0 ‘’ ***0.001*** ’’ *0.01* ’’ 0.05 ‘.’ 0.1 ’ ’ 1

```
regalphaUS2  <- censReg(alpha~log(wage) + log(tkm) + log(eff) + log(xbar)
                  + log(gm)
                  + beta
                  + log(r) + log(gx) + log(gh)
                  + log(sp)  + dB,  left=0, right=1, data = dfmcUS2.frame)
summary(regalphaUS2)
```

Call: censReg(formula = alpha ~ log(wage) + log(tkm) + log(eff) + log(xbar) + log(gm) + beta + log(r) + log(gx) + log(gh) + log(sp) + dB, left = 0, right = 1, data = dfmcUS2.frame)

Observations: Total Left-censored Uncensored Right-censored 10000 574 6694 2732

Coefficients: Estimate Std. error t value Pr(> t)
(Intercept) -0.4024325 0.1419625 -2.835 0.004586 ** log(wage) 0.0403951 0.0025993 15.541 < 2e-16 ***log(tkm) -0.0174097 0.0019682 -8.845 < 2e-16*** log(eff) 0.0273138 0.0081445 3.354 0.000797 ***log(xbar) 0.0110152 0.0014129 7.796 6.37e-15*** log(gm) 0.0785401 0.0053467 14.690 < 2e-16 ***beta 0.0177293 0.0001157 153.225 < 2e-16*** log(r) 0.1725817 0.0035093 49.178 < 2e-16 ***log(gx) -0.0161689 0.0054806 -2.950 0.003176*** *log(gh) -0.1572064 0.0674119 -2.332 0.019699*
log(sp) 0.0133027 0.0031381 4.239 2.24e-05 ***dB -0.0070729 0.0008412 -8.409 < 2e-16*** logSigma -1.8883011 0.0088525 -213.308 < 2e-16 *** — Signif. codes: 0 ‘’ ***0.001*** ’’ *0.01* ’’ 0.05 ‘.’ 0.1 ’ ’ 1

Newton-Raphson maximisation, 9 iterations Return code 1: gradient close to zero (gradtol) Log-likelihood: 1645.791 on 13 Df

```
regalphaUS2m <- margEff(regalphaUS2 ,vcov = NULL,
   calcVCov = TRUE, returnJacobian = FALSE )
summary(regalphaUS2m )
```

```
       Marg. Eff.  Std. Error  t value  Pr(>|t|)
```

log(wage) 0.03821907 0.00244406 15.6375 < 2.2e-16 ***log(tkm) -0.01647185 0.00186215 -8.8456 < 2.2e-16*** log(eff) 0.02584244 0.00770850 3.3525 0.0008039 ***log(xbar) 0.01042179 0.00133637 7.7986 6.883e-15*** log(gm) 0.07430919 0.00505556 14.6985 < 2.2e-16 ***beta 0.01677426 0.00010196 164.5143 < 2.2e-16*** log(r) 0.16328485 0.00330666 49.3806 < 2.2e-16 ***log(gx) -0.01529786 0.00518542 -2.9502 0.0031834*** *log(gh) -0.14873786 0.06378118 -2.3320 0.0197203*
log(sp) 0.01258611 0.00296865 4.2397 2.259e-05 ***dB -0.00669193 0.00079560 -8.4111 < 2.2e-16*** — Signif. codes: 0 ‘’ ***0.001*** ’’ *0.01* ’’ 0.05 ‘.’ 0.1 ’ ’ 1

```
stargazer(regalphaUS2,regshare_vUS,regshare_vcUS,regshare_voUS,
          title="Regression Results (U.S.)",algin=TRUE,
          dep.var.labels=c("alpha","sv","svc","svo","sxb"),omit.stat=c("LL","ser","f"),
         no.space=TRUE)
```

% Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
% Date and time: Do, Mrz 03, 2022 - 18:39:01

% Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
% Date and time: Do, Mrz 03, 2022 - 18:39:01

Table 5: Regression Results (U.S.)

	Dependent variable:			
	alpha	sv	svc	svo
	(1)	(2)	(3)	(4)
log(wage)	0.040^{***}	-0.0002	0.00004	-0.0004
	(0.003)	(0.001)	(0.001)	(0.001)
log(tkm)	-0.017^{***}	0.009^{***}	0.044^{***}	-0.001
	(0.002)	(0.001)	(0.001)	(0.001)
log(eff)	0.027^{***}	-0.692^{***}	0.009^{***}	-0.698^{***}
	(0.008)	(0.004)	(0.002)	(0.003)
log(xbar)	0.011^{***}	0.014^{***}	0.044^{***}	-0.001
	(0.001)	(0.001)	(0.001)	(0.001)
log(gm)	0.079^{***}	-0.003	0.001	-0.003^{*}
	(0.005)	(0.002)	(0.001)	(0.002)
beta	0.018^{***}	0.00000	0.00004^{**}	0.00002
	(0.0001)	(0.00003)	(0.00002)	(0.00003)
log(r)	0.173^{***}	-0.002	-0.001	-0.001
	(0.004)	(0.002)	(0.001)	(0.001)
log(gx)	-0.016^{***}	-0.0004	-0.0002	-0.0002
	(0.005)	(0.003)	(0.001)	(0.002)
log(gh)	-0.157^{**}	-0.028	-0.010	-0.004
	(0.067)	(0.031)	(0.017)	(0.025)
log(sp)	0.013^{***}	-0.002	-0.00000	-0.002
	(0.003)	(0.001)	(0.001)	(0.001)
dB	-0.007^{***}	-0.001^{**}	-0.0003	-0.0004
	(0.001)	(0.0004)	(0.0002)	(0.0003)
logSigma	-1.888^{***}	-2.559^{***}	-3.177^{***}	-2.778^{***}
	(0.009)	(0.007)	(0.007)	(0.007)
Constant	-0.402^{***}	0.946^{***}	0.121^{***}	0.832^{***}
	(0.142)	(0.065)	(0.035)	(0.052)
Observations	10,000	10,000	10,000	10,000
Akaike Inf. Crit.	$-3{,}265.583$	$-21{,}829.280$	$-33{,}918.230$	$-26{,}876.530$
Bayesian Inf. Crit.	$-3{,}171.848$	$-21{,}735.540$	$-33{,}824.500$	$-26{,}782.790$
Note:				*p<0.1; **p<0.05; ***p<0.01

Table 6: Regression Results (U.S.)

TRUE