```
# Industrial Organization II - Problem Set 1
# Demand Estimation (Vertical)
# Gabriel Gonzalez Sutil - gg2718

# Preliminaries -------------------------------------------------------------
rm(list = ls())
pacman :: p_load('data.table','dplyr','bbmle','AER','gmm','stargazer', 'optimx')
dir <- 'C:/Users/Gabriel Gonzalez Sut/Google Drive/PhD Sustainable - Columbia/'
course <- 'Sem 5 - Industrial Organization II/Problem Sets/'
ps <- 'PS1'
setwd(paste0(dir,course,ps))

# Import Data ---------------------------------------------------------------------------------------
auto <-  read.delim('ps1_blp_data_no_header.txt', header = FALSE)
colnames(auto) <- c('price','quantity','weight','hp','AC','firm')
auto$price <- as.numeric(auto$price)
auto$quantity <- as.numeric(auto$quantity)
auto$weight <- as.numeric(auto$weight)
auto$hp <- as.numeric(auto$hp)
auto$AC <- as.numeric(auto$AC)
auto$firm <- as.numeric(auto$firm)
M <- 100 * 10^6 # Market Size
lambda <- 4 * 10^(-6)

#---------------------------------------- Vertical Model ----------------------------------------------

vertical <- data.table(auto)

############## Preliminaries ####################
vertical[, share := (quantity/M)] #share
vertical <- vertical[order(price),] #order by price
# Cummulative Share
vertical[, cum.share := 0]
vertical$cum.share[1] <- 1 - sum(vertical$share)
for(i in 2:nrow(auto)){
  vertical$cum.share[i] <- vertical$cum.share[i-1] +  vertical$share[i-1]
}
# Price difference
vertical[, df.price := price - lag(price)]
vertical$df.price[1] <- vertical$price[1]
# Delta
vertical[, delta := 0]
vertical$delta[1] <- - (1/lambda) * log(vertical$cum.share[1]) * vertical$df.price[1]
for(i in 2:nrow(vertical)){
  vertical$delta[i] <- vertical$delta[i-1] - (1/lambda) * log(vertical$cum.share[i]) * vertical$df.price[i]
}
# Check delta
vertical[, df.delta := delta - lag(delta)]
vertical$df.delta[1] <- vertical$delta[1]
attach(vertical)
vertical$share.est <- 0
for(i in 1:nrow(vertical)){
  if(i < nrow(vertical)){
    vertical$share.est[i] <- pexp(df.delta[i]/df.price[i], rate = lambda) -
      pexp(df.delta[i+1]/ df.price[i+1], rate = lambda)
  }
  if(i == nrow(vertical)){
    vertical$share.est[i] <-  pexp(df.delta[i]/df.price[i], rate = lambda)
  }
}

png('vertical_delta.png')
plot(price, delta, xlab = 'Car Price', ylab = 'Delta')
dev.off()
plot(share, vertical$share.est, xlab = 'Share', ylab = 'Estimated Share', main = 'Test to Check Shares')

######################### Just Demand ###################################

# OLS
mod1 <- lm(delta ~ -1 + weight + hp + AC, data = vertical)
mod1b <- lm(delta ~ weight + hp + AC, data = vertical)

# GMM
X <- cbind(vertical[,c('weight','hp','AC')])

moment <- function(theta){
  xi <- delta - as.matrix(X[,1:3]) %*% as.matrix(theta[1:3])
  g <- (t(X) %*% xi)*(1/nrow(X))
  W <- as.matrix(t(X)) %*% as.matrix(X)
  obj <- t(g) %*% solve(W) %*% g
}
gmm.mod1 <- opm(par = c(0,0,0), fn = moment, method=c('BFGS'))

X <- cbind(rep(1, 131), vertical[,c('weight','hp','AC')])

moment <- function(theta){
  xi <- delta - as.matrix(X[,1:4]) %*% as.matrix(theta[1:4])
  g <- (t(X) %*% xi)*(1/nrow(X))
  W <- as.matrix(t(X)) %*% as.matrix(X)
  obj <- t(g) %*% solve(W) %*% g
}
gmm.mod1b <- opm(par = c(0,0,0,0), fn = moment, method=c('BFGS'))

# Results
mod1$coefficients
cbind(gmm.mod1$p1,gmm.mod1$p2,gmm.mod1$p3,gmm.mod1$p4)
mod1b$coefficients
cbind(gmm.mod1b$p1,gmm.mod1b$p2,gmm.mod1b$p3,gmm.mod1b$p4)
stargazer(mod1, mod1b, title="Parameters with Demand Data")

######################### Elasticity ###################################

# Define derivative matrix
D.vertical <- matrix(0,nrow(vertical), nrow(vertical))
for(i in 1:nrow(vertical)){
  for(j in 1:nrow(vertical)){
```

```
    if(j == i+1){
      D.vertical[i,j] <- lambda * df.delta[j]/(df.price[j])^2 * (1 - pexp(df.delta[j]/df.price[j], rate = lambda))
    }
    if(j == i-1){
      D.vertical[i,j] <- lambda * df.delta[i]/(df.price[i])^2 * (1 - pexp(df.delta[i]/df.price[i], rate = lambda))
    }
    if(j == i & i < nrow(vertical)){
      D.vertical[i,j] <- -lambda * (1 - pexp(df.delta[i+1]/df.price[i+1], rate = lambda)) * (df.delta[i+1]/(df.price[i+1])^2) -
        lambda * (1 - pexp(df.delta[i]/df.price[i], rate = lambda)) * (df.delta[i]/(df.price[i])^2)
    }
    if(j == i & i == nrow(vertical)){
      D.vertical[i,j] <- -lambda * ((df.delta[i]/(df.price[i])^2)*(1 - pexp(df.delta[i]/df.price[i], rate = lambda)))

    }
  }
}

# Check Matrix
D.vertical2 <- matrix(0,nrow(vertical), nrow(vertical))
for(i in 1:nrow(vertical)){
  for(j in 1:nrow(vertical)){
    if(j == i+1){
      D.vertical2[i,j] <- lambda * exp(-lambda * (delta[j] - delta[i])/(price[j] - price[i])) * (delta[j] - delta[i])/(price[j] - price[i])^2
    }
    if(j == i-1){
      D.vertical2[i,j] <- lambda * exp(-lambda * (delta[i] - delta[j])/(price[i] - price[j])) * (delta[i] - delta[j])/(price[i] - price[j])^2
    }
    if(j == i & i > 1 & i < nrow(vertical)){
      D.vertical2[i,j] <- -lambda *  (exp(-lambda * (delta[i+1] - delta[i])/(price[i+1] - price[i])) * (delta[i+1] - delta[i])/(price[i+1] -
price[i])^2
                                      +  exp(-lambda * (delta[i] - delta[i-1])/(price[i] - price[i-1])) * (delta[i] - delta[i-1])/(price[i] -
price[i-1])^2)
    }
    if(j == i & i == 1 & i < nrow(vertical)){
      D.vertical2[i,j] <- -lambda *  (exp(-lambda * (delta[i+1] - delta[i])/(price[i+1] - price[i])) * (delta[i+1] - delta[i])/(price[i+1] -
price[i])^2
                                      +  exp(-lambda * (delta[i]/price[i])) * delta[i]/(price[i])^2)
    }
    if(j == i & i == nrow(vertical)){
      D.vertical2[i,j] <- -lambda * (exp(-lambda * (delta[i] - delta[i-1])/(price[i] - price[i-1])) * (delta[i] - delta[i-1])/(price[i] - price[i-
1])^2)

    }
  }
}

# Define elasticity matrix
E.vertical <- matrix(0,nrow(vertical), nrow(vertical))
for(i in 1:nrow(vertical)){
  for(j in 1:nrow(vertical)){
    if(j == i){
      E.vertical[i,j] <- D.vertical[i,j] * price[i]/share[i]
    }else{
      E.vertical[i,j] <- D.vertical[i,j] * price[j]/share[i]
    }
  }
}

# Plots
png('vertical_price1.png')
plot(price, diag(E.vertical), ylim = c(-60000,0),
     xlab = "Car Price", ylab = "Price Elasticity");
lines(lowess(price,diag(E.vertical)), col = 'blue')
dev.off()

png('vertical_price2.png')
plot(df.price, diag(E.vertical), xlim = c(0,2000), ylim = c(-60000,0),
     xlab = "Difference in Car Price Pre-Car", ylab = "Price Elasticity");
lines(lowess(df.price,diag(E.vertical)), col = 'blue')
dev.off()

png('vertical_price3.png')
plot(df.price[2:131], diag(E.vertical[1:130,1:130]), xlim = c(0,2000), ylim = c(-60000,0),
     xlab = "Difference in Car Price Post-Car", ylab = "Price Elasticity");
lines(lowess(df.price,diag(E.vertical)), col = 'blue')
dev.off()

cross_pre <- rep(0,nrow(E.vertical)-1)
for(i in 2:length(cross_pre)+1){
  cross_pre[i-1] <- E.vertical[i,i-1]
}
cross_post <- rep(0,nrow(E.vertical)-1)
for(i in 1:length(cross_post)-1){
  cross_post[i] <- E.vertical[i,i+1]
}

png('vertical_cross1.png')
plot(df.price[2:131], cross_pre, xlim = c(0,2000), ylim = c(0,50000),
     xlab = "Difference in Car Price", ylab = "Pre Cross-Price Elasticity");
lines(lowess(df.price[2:131], cross_pre), col = 'blue')
dev.off()

png('vertical_cross2.png')
plot(df.price[2:131], cross_post, xlim = c(0,2000), ylim = c(0,50000),
     xlab = "Difference in Car Price", ylab = "Post Cross-Price Elasticity");
lines(lowess(df.price[2:131], cross_post), col = 'blue')
dev.off()

par(mfrow=c(1,1))

######################### Demand + Supply ###################################

# Define ownership matrix
H1 <- diag(nrow(vertical))
H2 <- diag(nrow(vertical))
for (i in 1:nrow(vertical)){
```

```
  for(j in 1:nrow(vertical)){
    if (vertical$firm[i] == vertical$firm[j]) H2[i,j] <- 1
  }
}
H3 <- matrix(1,nrow(vertical),nrow(vertical))

# Define marginal costs
vertical$mc1 <- price
vertical$mc2 <- price + solve(H1 * D.vertical) %*% share
vertical$mc3 <- price + solve(H2 * D.vertical) %*% share
vertical$mc4 <- price + solve(H3 * D.vertical) %*% share
attach(vertical)

# Instruments
x.pre <- data.table(pre.weight = lag(vertical$weight),
                    pre.hp = lag(vertical$hp),
                    pre.AC = lag(vertical$AC))
x.pre[1,] <- 0
x.post <- data.table(post = vertical[2:nrow(vertical),c('weight')],
                     post = vertical[2:nrow(vertical),c('hp')],
                     post = vertical[2:nrow(vertical),c('AC')])
x.post <- rbind(x.post,data.table(post.weight = 0,post.hp = 0,post.AC = 0))
Z <- cbind(x.pre, vertical[,c('weight','hp','AC')], x.post)
colnames(Z) <- c('Z1','Z2', 'Z3', 'Z4', 'Z5', 'Z6', 'Z7', 'Z8','Z9')
rm(x.post, x.pre)
vertical <- cbind(vertical,Z)

# Regression 2Sls (no constant)
mod2 <- ivreg(mc1 ~ -1 + quantity + weight + hp + AC | -1 + Z1 + Z2 + Z3 + weight + hp + AC +
                Z7 + Z8 + Z9, data = vertical)
mod3 <- ivreg(mc2  ~ - 1 + quantity + weight + hp + AC | -1 + Z1 + Z2 + Z3 + weight + hp + AC +
                Z7 + Z8 + Z9, data = vertical)
mod4 <- ivreg(mc3  ~ - 1 + quantity + weight + hp + AC | -1 + Z1 + Z2 + Z3 + weight + hp + AC +
                Z7 + Z8 + Z9, data = vertical)
mod5 <- ivreg(mc4  ~ - 1 + quantity + weight + hp + AC | -1 + Z1 + Z2 + Z3 + weight + hp + AC +
                Z7 + Z8 + Z9, data = vertical)
summary(mod2)
summary(mod3)
summary(mod4)
summary(mod5)

# GMM
library(optimx)
X <- cbind(quantity, weight, hp, AC)

moment1 <- function(theta){
  xi <- mc1 - as.matrix(X[,1:4]) %*% as.matrix(theta[1:4])
  g <- (t(Z) %*% xi)*(1/nrow(X))
  W <- as.matrix(t(Z)) %*% as.matrix(Z)
  obj <- t(g) %*% solve(W) %*% g
}
moment2 <- function(theta){
  xi <- mc2 - as.matrix(X[,1:4]) %*% as.matrix(theta[1:4])
  g <- (t(Z) %*% xi)*(1/nrow(X))
  W <- as.matrix(t(Z)) %*% as.matrix(Z)
  obj <- t(g) %*% solve(W) %*% g
}
moment3 <- function(theta){
  xi <- mc3 - as.matrix(X[,1:4]) %*% as.matrix(theta[1:4])
  g <- (t(Z) %*% xi)*(1/nrow(X))
  W <- as.matrix(t(Z)) %*% as.matrix(Z)
  obj <- t(g) %*% solve(W) %*% g
}
moment4 <- function(theta){
  xi <- mc4 - as.matrix(X[,1:4]) %*% as.matrix(theta[1:4])
  g <- (t(Z) %*% xi)*(1/nrow(X))
  W <- as.matrix(t(Z)) %*% as.matrix(Z)
  obj <- t(g) %*% solve(W) %*% g
}
gmm.mod2<- opm(par = c(0,0,0,0), fn = moment1, method=c('BFGS'))
gmm.mod3<- opm(par = c(0,0,0,0), fn = moment2, method=c('BFGS'))
gmm.mod4<- opm(par = c(0,0,0,0), fn = moment3, method=c('BFGS'))
gmm.mod5<- opm(par = c(0,0,0,0), fn = moment4, method=c('BFGS'))

# Results
mod2$coefficients
cbind(gmm.mod2$p1,gmm.mod2$p2,gmm.mod2$p3,gmm.mod2$p4)
mod3$coefficients
cbind(gmm.mod3$p1,gmm.mod3$p2,gmm.mod3$p3,gmm.mod3$p4)
mod4$coefficients
cbind(gmm.mod4$p1,gmm.mod4$p2,gmm.mod4$p3,gmm.mod4$p4)
mod5$coefficients
cbind(gmm.mod5$p1,gmm.mod5$p2,gmm.mod5$p3,gmm.mod5$p4)
stargazer(mod2,mod3,mod4,mod5, title="Parameters Supply")


# Regression 2Sls (no constant)
mod2b <- ivreg(mc1 ~ quantity + weight + hp + AC | Z1 + Z2 + Z3 + weight + hp + AC +
                Z7 + Z8 + Z9, data = vertical)
mod3b <- ivreg(mc2  ~ quantity + weight + hp + AC | Z1 + Z2 + Z3 + weight + hp + AC +
                Z7 + Z8 + Z9, data = vertical)
mod4b <- ivreg(mc3  ~ quantity + weight + hp + AC | Z1 + Z2 + Z3 + weight + hp + AC +
                Z7 + Z8 + Z9, data = vertical)
mod5b <- ivreg(mc4  ~ quantity + weight + hp + AC | Z1 + Z2 + Z3 + weight + hp + AC +
                Z7 + Z8 + Z9, data = vertical)

# GMM
X <- cbind(rep(1,131),quantity, weight, hp, AC)
Z <- cbind(rep(1,131),Z)

moment1 <- function(theta){
  xi <- mc1 - as.matrix(X[,1:5]) %*% as.matrix(theta[1:5])
  g <- (t(Z) %*% xi)*(1/nrow(X))
  W <- as.matrix(t(Z)) %*% as.matrix(Z)
  obj <- t(g) %*% solve(W) %*% g
}
```

```
moment2 <- function(theta){
  xi <- mc2 - as.matrix(X[,1:5]) %*% as.matrix(theta[1:5])
  g <- (t(Z) %*% xi)*(1/nrow(X))
  W <- as.matrix(t(Z)) %*% as.matrix(Z)
  obj <- t(g) %*% solve(W) %*% g
}
moment3 <- function(theta){
  xi <- mc3 - as.matrix(X[,1:5]) %*% as.matrix(theta[1:5])
  g <- (t(Z) %*% xi)*(1/nrow(X))
  W <- as.matrix(t(Z)) %*% as.matrix(Z)
  obj <- t(g) %*% solve(W) %*% g
}
moment4 <- function(theta){
  xi <- mc4 - as.matrix(X[,1:5]) %*% as.matrix(theta[1:5])
  g <- (t(Z) %*% xi)*(1/nrow(X))
  W <- as.matrix(t(Z)) %*% as.matrix(Z)
  obj <- t(g) %*% solve(W) %*% g
}
gmm.mod2b <- opm(par = c(0,0,0,0,0), fn = moment1, method=c('BFGS'))
gmm.mod3b <- opm(par = c(0,0,0,0,0), fn = moment2, method=c('BFGS'))
gmm.mod4b <- opm(par = c(0,0,0,0,0), fn = moment3, method=c('BFGS'))
gmm.mod5b <- opm(par = c(0,0,0,0,0), fn = moment4, method=c('BFGS'))


# Results
mod2b$coefficients
cbind(gmm.mod2b$p1,gmm.mod2b$p2,gmm.mod2b$p3,gmm.mod2b$p4,gmm.mod2b$p5)
mod3b$coefficients
cbind(gmm.mod3b$p1,gmm.mod3b$p2,gmm.mod3b$p3,gmm.mod3b$p4,gmm.mod3b$p5)
mod4b$coefficients
cbind(gmm.mod4b$p1,gmm.mod4b$p2,gmm.mod4b$p3,gmm.mod4b$p4,gmm.mod4b$p5)
mod5b$coefficients
cbind(gmm.mod5b$p1,gmm.mod5b$p2,gmm.mod5b$p3,gmm.mod5b$p4,gmm.mod5b$p5)
stargazer(mod2b,mod3b,mod4b,mod5b, title="Parameters Supply")
```