Assignment 1-3

Xinyu Guo, Javed Hossain, Sriramjee Singh

September 19, 2017

```
#Running Previous model to get coefficients right

ULA_AIDS_beef <- lm(beef_w ~ log(beef_p) + log(pork_p) + log(poultry_p) + log(meat_exp/Stone) + Spring +

ULA_AIDS_pork <- lm(pork_w ~ log(beef_p) + log(pork_p) + log(poultry_p) + log(meat_exp/Stone) + Spring +

ULA_AIDS_poultry <- lm(poultry_w ~ log(beef_p) + log(pork_p) + log(poultry_p) + log(meat_exp/Stone) + Sp

#LAAIDS model
LA_AIDS_b <- beef_w ~ ab + s0b*Spring + s1b*Summer + s2b*Fall + trendb*t + gamma_beef*log(beef_p) + gamm
LA_AIDS_p <- pork_w ~ ap + s0p*Spring + s1p*Summer + s2p*Fall + trendp*t + gamma_beef_pork*log(beef_p) +

labels_LA <- list( "Beef", "Pork" )

start.values_LA <- c(trendb = as.numeric(coef(ULA_AIDS_beef)["t"]),
                     trendp = as.numeric(coef(ULA_AIDS_pork)["t"]),
                     ab =as.numeric(coef(ULA_AIDS_beef)["(Intercept)"]),
                     s0b = as.numeric(coef(ULA_AIDS_beef)["Spring"]),
                     s1b = as.numeric(coef(ULA_AIDS_beef)["Summer"]),
                     s2b = as.numeric(coef(ULA_AIDS_beef)["Fall"]),
                     gamma_beef = as.numeric(coef(ULA_AIDS_beef)["log(beef_p)"]),
                     gamma_beef_pork = as.numeric(coef(ULA_AIDS_beef)["log(pork_p)"]),
                     beta_beef = as.numeric(coef(ULA_AIDS_beef)["log(meat_exp/Stone)"]),
                     ap= as.numeric(coef(ULA_AIDS_pork)["(Intercept)"]),
                     s0p = as.numeric(coef(ULA_AIDS_pork)["Spring"]),
                     s1p = as.numeric(coef(ULA_AIDS_pork)["Summer"]),
                     s2p = as.numeric(coef(ULA_AIDS_pork)["Fall"]),
                     gamma_pork = as.numeric(coef(ULA_AIDS_pork)["log(pork_p)"]),
                     beta_pork = as.numeric(coef(ULA_AIDS_pork)["log(meat_exp/Stone)"]))


model_LA <- list( LA_AIDS_b, LA_AIDS_p )
model.LA <- nlsystemfit( "SUR", model_LA, start.values_LA, data = dta, eqnlabels=labels_LA, maxiter=100
```

```
#Rotterdam model

rot_beef <- s.dln.beef_q ~ ab + s0b*Spring + s1b*Summer + s2b*Fall + trendb*t + beta_beef*dln.M + gamma_

rot_pork <- s.dln.pork_q ~ ap + s0p*Spring + s1p*Summer + s2p*Fall + trendp*t + beta_pork*dln.M + gamma_

labels_rot <- list( "Beef", "Pork")

start.values_rot <- c(trendb = as.numeric(model.LA$b["trendb"]),
                      trendp = as.numeric(model.LA$b["trendp"]),
                      ab = as.numeric(model.LA$b["ab"]),
                      s0b = as.numeric(model.LA$b["s0b"]),
                      s1b = as.numeric(model.LA$b["s1b"]),
```

```r
                        s2b = as.numeric(model.LA$b["s2b"]),
                        ap= as.numeric(model.LA$b["ap"]),
                        s0p = as.numeric(model.LA$b["s0p"]),
                        s1p = as.numeric(model.LA$b["s1p"]),
                        s2p = as.numeric(model.LA$b["s2p"]),
                        gamma_beef = as.numeric(model.LA$b["gamma_beef"]),
                        gamma_beef_pork = as.numeric(model.LA$b["gamma_beef_pork"]),
                        gamma_pork = as.numeric(model.LA$b["gamma_pork"]),
                        beta_beef = as.numeric(model.LA$b["beta_beef"]),
                        beta_pork = as.numeric(model.LA$b["beta_pork"])) %>% round(3)

model.rot <- list(rot_beef, rot_pork)
model_rot <- nlsystemfit("SUR", model.rot, start.values_rot, data = dta, eqnlabels=labels_rot, maxiter=
```

Question 10

Verify that the Slutsky matrix from the model estimated in question 8 is semi-negative definite.

```r
#Recover parameters for the third equation
model_rot$b["gamma_pork_beef"] <- model_rot$b["gamma_beef_pork"]
model_rot$b["gamma_beef_poultry"] <- -(model_rot$b["gamma_beef"] + model_rot$b["gamma_beef_pork"])
model_rot$b["gamma_poultry_beef"] <- model_rot$b["gamma_beef_poultry"]
model_rot$b["gamma_pork_poultry"] <- -(model_rot$b["gamma_pork"] + model_rot$b["gamma_beef_pork"])
model_rot$b["gamma_poultry_pork"] <- model_rot$b["gamma_pork_poultry"]
model_rot$b["gamma_poultry"] <- -(model_rot$b["gamma_beef_poultry"] + model_rot$b["gamma_pork_poultry"])
model_rot$b["beta_poultry"] <- 1 - (model_rot$b["beta_beef"] + model_rot$b["beta_pork"])


#Income elasticity
eta_m <- function(X){
  mean(as.numeric(model_rot$b[paste("beta", X, sep="_")]/dta[,paste(X, "w", sep="_")]))
}


#Own price elasticity
eta_i_H <- function(X){
  mean(as.numeric((model_rot$b[paste("gamma", X, sep="_")])/dta[,paste(X, "w", sep="_")]))}

#Cross price elasticity
eta_ij_H <- function(X1,X2){
  mean(as.numeric((model_rot$b[paste("gamma", X1, X2, sep="_")])/dta[,paste(X1, "w", sep="_")]))}

p_load(stargazer)

#Table or elasticities - ULA-AIDS
Table_rot <- array(0,c(4,3))
rownames(Table_rot) <- c("Expenditure", "Beef", "Pork", "Poultry" )
colnames(Table_rot) <- c("Beef","Pork", "Poultry")
Table_rot["Expenditure","Beef"] <- eta_m("beef")
Table_rot["Beef","Beef"] <- eta_i_H("beef")
Table_rot["Pork","Beef"] <- eta_ij_H("pork", "beef")
Table_rot["Poultry","Beef"] <- eta_ij_H("poultry", "beef")
Table_rot["Expenditure","Pork"] <- eta_m("pork")
```

```r
Table_rot["Beef","Pork"] <- eta_ij_H("beef", "pork")
Table_rot["Pork","Pork"] <- eta_i_H("pork")
Table_rot["Poultry","Pork"] <- eta_ij_H("poultry", "pork")
Table_rot["Expenditure","Poultry"] <- eta_m("poultry")
Table_rot["Beef","Poultry"] <- eta_ij_H("beef", "poultry")
Table_rot["Pork","Poultry"] <- eta_ij_H("pork", "poultry")
Table_rot["Poultry","Poultry"] <- eta_i_H("poultry")

stargazer(signif(Table_rot,3), summary = FALSE, title = "Elasticities from linear AIDS model",header=FA
```

```
##
## \begin{table}[!htbp] \centering
##   \caption{Elasticities from linear AIDS model}
##   \label{}
## \begin{tabular}{@{\extracolsep{5pt}} cccc}
## \\[-1.8ex]\hline
## \hline \\[-1.8ex]
##  & Beef & Pork & Poultry \\
## \hline \\[-1.8ex]
## Expenditure & $1.220$ & $0.875$ & $0.591$ \\
## Beef & $$-$0.394$ & $0.307$ & $0.087$ \\
## Pork & $0.574$ & $$-$0.617$ & $0.043$ \\
## Poultry & $0.268$ & $0.071$ & $$-$0.339$ \\
## \hline \\[-1.8ex]
## \end{tabular}
## \end{table}
```

```r
beef_p_mean <- mean(dta$beef_p)
pork_p_mean <- mean(dta$pork_p)
poultry_p_mean <- mean(dta$poultry_p)
beef_q_mean <- mean(dta$beef_q)
pork_q_mean <- mean(dta$pork_q)
poultry_q_mean <- mean(dta$poultry_q)

#Calculate Slutsky matrix

a_bb <- beef_p_mean/beef_q_mean
a_bp <- beef_p_mean/pork_q_mean
a_bpo <- beef_p_mean/poultry_q_mean
a_pb <- pork_p_mean/beef_q_mean
a_pp <- pork_p_mean/pork_q_mean
a_ppo <- pork_p_mean/poultry_q_mean
a_pob <- poultry_p_mean/beef_q_mean
a_pop <- poultry_p_mean/pork_q_mean
a_popo <- poultry_p_mean/poultry_q_mean

h_bb <- -0.394*a_bb
h_bp <- 0.307*a_bp
h_bpo <- 0.087*a_bpo
h_pb <- 0.574*a_pb
h_pp <- -0.617*a_pp
h_ppo <- 0.043*a_ppo
h_pob <- 0.268*a_pob
h_pop <- 0.071*a_pop
```

```
h_popo <- -0.339*a_popo


# Calculate principal minor

slutsky_matrix_2 <- matrix(c(h_bb, h_pb, h_bp, h_pp), nrow = 2, ncol = 2)
slutsky_matrix_2
```

```
##           [,1]      [,2]
## [1,] -5.119942  5.883533
## [2,]  5.840753 -9.259192
```

```
det(slutsky_matrix_2)
```

```
## [1] 13.04227
```

```
slutsky_matrix <- matrix(c(h_bb, h_pb, h_pob, h_bp, h_pp, h_pop, h_bpo, h_ppo, h_popo), nrow = 3, ncol =
slutsky_matrix
```

```
##            [,1]        [,2]       [,3]
## [1,] -5.1199422  5.88353277  1.1768616
## [2,]  5.8407533 -9.25919223  0.4554731
## [3,]  0.2235844  0.08735664 -0.2944039
```

```
det(slutsky_matrix)
```

```
## [1] -2.171974e-15
```

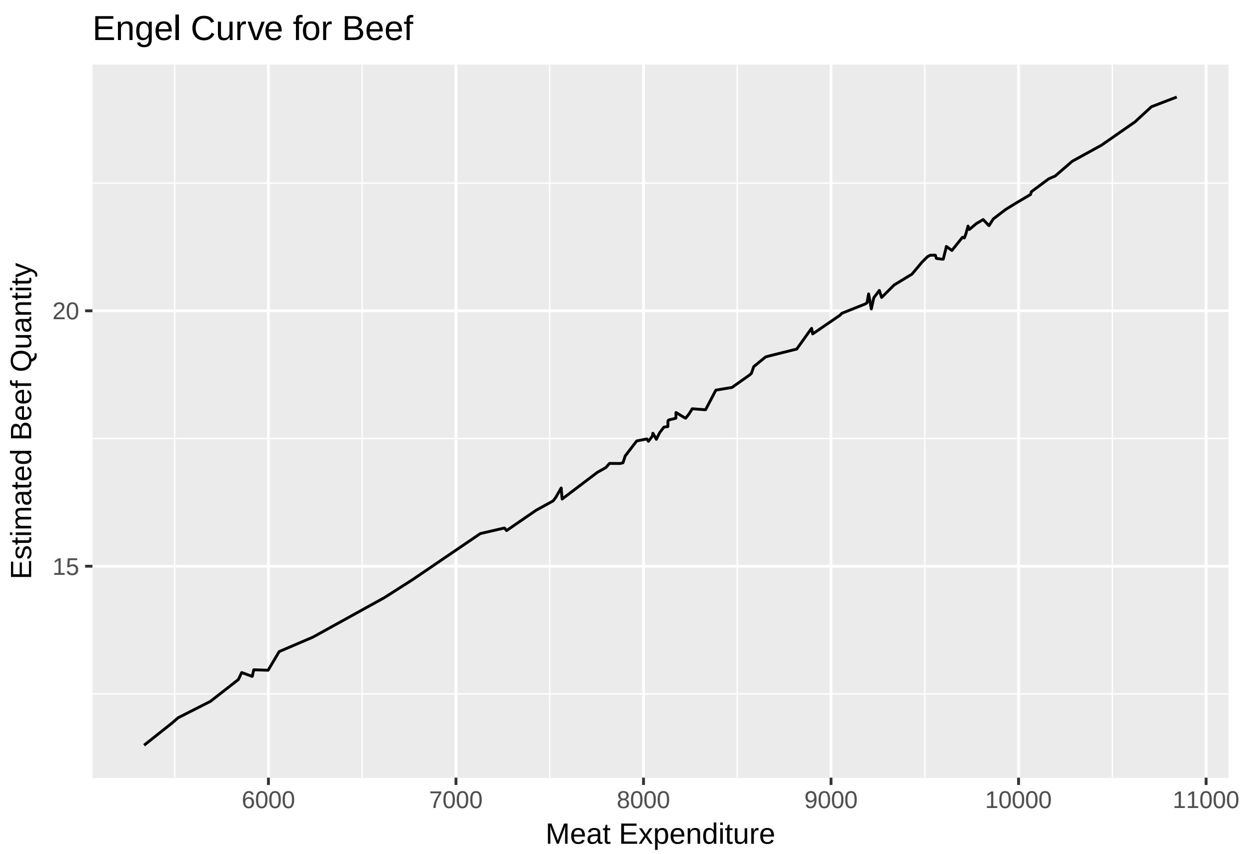
From above we can see,its kth order principal minors are nonpositive for k odd and nonnegative for k even. Thus slutsky matrix is nsd.

Question 11

Plot the Engel curves for beef for the model in questions 7. Discuss.

```
#relevent part of LA-AIDS model

LA_AIDS_b <- beef_w ~ ab + s0b*Spring + s1b*Summer + s2b*Fall + trendb*t + gamma_beef*log(beef_p) + gamm
LA_AIDS_p <- pork_w ~ ap + s0p*Spring + s1p*Summer + s2p*Fall + trendp*t + gamma_beef_pork*log(beef_p) +

labels_LA <- list( "Beef", "Pork" )

A <-lm(log(beef_w) ~ Spring + Summer + Fall + t + log(beef_p) + log(pork_p) + log(poultry_p) + log(meat_


start.values_LA <- c(trendb = as.numeric(coef(A)["t"]),
                     trendp = as.numeric(coef(A)["t"]),
                     ab =as.numeric(coef(A)["(Intercept)"]),
                     s0b = as.numeric(coef(A)["Spring"]),
                     s1b = as.numeric(coef(A)["Summer"]),
                     s2b = as.numeric(coef(A)["Fall"]),
                     gamma_beef = as.numeric(coef(A)["log(beef_p)"]),
                     gamma_beef_pork = as.numeric(coef(A)["log(pork_p)"]),
                     beta_beef = as.numeric(coef(A)["log(meat_exp/Stone)"]),
                     ap= as.numeric(coef(A)["(Intercept)"]),
                     s0p = as.numeric(coef(A)["Spring"]),
```

```r
                      s1p = as.numeric(coef(A)["Summer"]),
                      s2p = as.numeric(coef(A)["Fall"]),
                      gamma_pork = as.numeric(coef(A)["log(pork_p)"]),
                      beta_pork = as.numeric(coef(A)["log(meat_exp/Stone)"]))


model_LA <- list( LA_AIDS_b, LA_AIDS_p )
model.LA <- nlsystemfit( "SUR", model_LA, start.values_LA, data = dta, eqnlabels=labels_LA, maxiter=1000

#take mean of other variables
dta$mean_beef_p <- mean(dta$beef_p)
dta$mean_pork_p <- mean(dta$pork_p)
dta$mean_poultry_p <- mean(dta$poultry_p)
dta$mean_Spring <- mean(dta$Spring)
dta$mean_Summer <- mean(dta$Summer)
dta$mean_Fall <- mean(dta$Fall)
dta$mean_t <- mean(dta$t)

#estimation of beef share
dta$est_beef_w <- as.numeric(model.LA$b["ab"]) + as.numeric(model.LA$b["s0b"])*dta$mean_Spring+ as.numer

#estimation of beef quantity
dta$est_beef_q <- (dta$est_beef_w*dta$meat_exp)/(dta$mean_beef_p)

dta_graph <- dta %>%
  dplyr::select(meat_exp, est_beef_q)

plot <- dta_graph %>% ggplot(aes(x = meat_exp, y = est_beef_q)) +
geom_line(aes())

print(plot + labs(y="Estimated Beef Quantity", x="Meat Expenditure") + ggtitle("Engel Curve for Beef"))
```



Discussion:

Engel Curve is a relationship between quantity demanded and the income, when prices and other variables are kept constant. We take meat expenditure as the income assuming separability. For quantity, we are not taking the values given for beef but estimating them by keeping prices and trend constant using LA-AIDS model. The engel curve is a straight line with positive slope in our case, as theory suggests.

Question 12

Does the weak axiom of revealed preferences hold? Discuss.

```
nb <- nrow(dta)  #Number of rows

prices <- matrix(c(dta$beef_p, dta$pork_p, dta$poultry_p), nrow=nb)
quants <- matrix(c(dta$beef_q, dta$pork_q, dta$poultry_q), nrow=nb)

C <- prices %*% t(quants)  # expenditure of different bundle under different price
Invers_Expend_Matrix <- solve(diag(diag(C)))  #compute the inverse of a matrix
WARP_Matrix <- Invers_Expend_Matrix %*% C   #compute the final matrix
WARP_Matrix <- round(WARP_Matrix,4) #To round because of precision issues
```

```
Nb_violation <-0

for (i in  1:(nb-1)){
  for (j in (i+1):nb){
     if(WARP_Matrix[i,j]<1 && WARP_Matrix[j,i]<1){
      Nb_violation <- Nb_violation+1
```

```
    }
  }
}

Nb_possibleCombination= nb*(nb-1)/2

percent_Violation=Nb_violation/Nb_possibleCombination

Nb_violation
```

```
## [1] 25
```

```
Nb_possibleCombination
```

```
## [1] 4753
```

```
percent_Violation
```

```
## [1] 0.005259836
```

From above test, we find that the WARP does not hold for US meat consumption data. However, the percentage of violation is small.

Question 13

Discuss your findings about the weak axiom of revealed preferences.

Even though we find some violations when testing WARP, we can not jump to the conclusion that data is not consistent with economic theory, since the percentage of violation is relatively small (0.005259836). However, conventional understanding is that, WARP restriction is not very restrictive on emperical observations (the power of captureing violation is small) due to the fact that income shifts out more dramatically than the change in relative price. Yet, in our test, this issue is avoided since we use total expenditure on meat as income, suppose seperability assumption holds.