The Merchant Guilds and the Political Economy of the Spanish Empire on the eve of Independence

Fernando Arteaga*

Online Appendix

*Economics Department, University of Pennsylvania, E-mail: arteaga@sas.upenn.edu.

A Online Appendix A: Model Details

A.1 Agent's Description and Attributes

Table 1 shows agents' attributes, as well as their initialization values. The population class has only an income attribute, which can take values from zero up to infinite. The initial values are set by an exogenous random exponential distribution ($\sim exp(\lambda) = exp(1)$) multiplied by ten— to replicate the idea that income and wealth are unequal within societies. The model is not a general equilibrium one, so the total income the population holds is independent of the goods available in the economy. It would be possible to complicate the model by implementing both the American mining sector as the income source and the European producers of the goods and creating a feedback mechanism between them. Yet, It would add little to the core mechanics I am interested in: the relations between the intermediary merchant guilds as wholesalers and distributors and the Crown's incentive to privilege some in exchange for taxing income. Moreover, historically speaking, the existence of a transatlantic market is something that still is disputed.[1]

The merchant class is characterized by six attributes: *i)* income, which can go from zero to infinite. It is also initialized by an exogenously-given random exponential distribution ($\sim exp(\lambda) = exp(1)$) multiplied by one hundred; *ii)* Goods, which are the amount of the good X that any merchant holds during the simulation. Just like population, income does not depend on the goods available. Its quantity is also not dependent on the population's income. At the beginning of the simulation, the initial quantity is set to be 50—which is allotted to the supplier, the Spanish merchant guild; *iii)* Profit, an unbounded attribute, is calculated as the difference between the selling and buying prices that each merchant faces during each period in the simulation; *iv)* There are two binary attributes—either being a supplier or a wholesaler – that allows distinguishing between the three different sets of merchant subclasses in the model. At the initialization of the model, one supplier and one wholesaler are identified; *v)* A third boolean attribute serves to identify if any merchant has become bankrupt during the simulation; bankruptcy is defined as the case in which the income of the merchant goes to zero.

The Crown class has three attributes: *i)* The Trade Risk attribute serves as an identifier of the three risk scenarios; It begins with the lowest risk profile. Hence its initialization value is zero; *ii)* Tax

[1]There are disparities in price between both areas for a set of goods, which imply lack of a unified market.

Agents	**Attributes**	**Values**	**Initialization**
Population	Income	$float[0, \infty)$	Random Exponential
Crown	Trade Risk	$Int[0, 2]$	0
	Tax Rate	$float(.05)$	.05
	Tax Revenue	$float[0, \infty)$	0
Merchants	Income	$float[0, \infty)$	Random Exponential
	Goods	$float[0, \infty)$	—
	Profit	$float[(-\infty, \infty)$	0
	Supplier	Boolean	—
	Wholesaler	Boolean	—
	Bankrupt	Boolean	False

Table 1: Agent's Attributes and Values

rate, which is the sales tax that the Crown applies to each population-retailer transaction. The value is set to be 5%; *iii)* Tax Revenue, which records the amount of taxes collected each period during the simulation. The value can go from zero up to infinite.

A.2 Agents' Behavior

The agents in the model follow simple heuristics. Merchants, population, and their interactions are modeled under the premise that they constitute zero intelligence traders. Following Becker (1962) and Gode and Sunder (1993), it is possible to recreate simple partial markets—where supply and demand exist—by acknowledging that income constraints are sufficient conditions for markets to behave as basic price theory predicts. The Crown acts as the mayor agent that optimizes its rent out of the political mechanisms already described.

A.2.1 First Market: Supplier-Wholesaler

By construction, only one supplier of good X exists within the model—it is assumed to be the Spanish Seville merchant guild. It initializes with 50 X goods that intends to distribute to the Americas. Each period, the supplier increases/decreases the amount of produced goods that it has to sell. The growth rate comes from a random normal distribution ($\sim N(\mu, \sigma) = N(.25, .5)$). The idea is to approximate the growth of a precapitalist economy, which tended to be characterized by a small positive growth trend but subject to big potential variations from time to time.

After good X is produced, the supplier ships it towards the Americas to be sold to the American wholesalers. At this moment, the supplier can lose part of its cargo, and the three different risk scenarios may apply (first, with pirates as a threat, then with the British and French Navy as potential harassers). It is important to note that before making the trip to America, the Spanish supplier calculates its potential demand according to the number of customers it has (the wholesalers). So, it sends separate shipments to the Americas. This is important because when only one wholesaler exists, only one shipment is made, and the probability of losing all the cargo is clustered on that shipment. Whereas when two or more wholesalers exist, two or more shipments are made, and the likelihood of losing cargo is distributed over the total shipments (the assumption is that the events are independent of each other).

After arriving to the Americas, the supplier negotiates with the American wholesalers. Initially, there is only one wholesaler, so the negotiation is one-to-one. However, during the course of the simulation, more wholesalers can arise. The Spanish supplier acts as a perfect discriminator, and hence it optimizes its income by selling at different prices/quantities to each wholesaler—according to the wholesaler's income.[2] A more realistic approach towards modeling the negotiations between Spanish suppliers and American wholesalers would resemble a kind of 'tug of war' between both merchant guilds.[3] However, given that the model is primarily interested in explaining the difficulties in transatlantic trade—rather than on the definitive winners of the windfall rents between the American and Spanish merchant guilds—the scenario is simplified and it is assumed that economic rents are totally captured by the American wholesalers. So, even though the supplier can discriminate, it does only so for accounting purposes within the model.

A.2.2 Second Market: Wholesalers-Retailers

After the wholesalers acquire the goods from the Spanish supplier, they redistribute them to the other American merchants: the regional and local retailers. It is essential to acknowledge that the main difference between wholesalers and retailers derives from the political privileges enjoyed by the first. For that reason, as we will see in the following sections, the Crown can grant wholesaler concessions to

[2]The income of the supplier i is given by the sum of all j wholesaler's income $I_i = \sum I_j$. Alternatively, the quantity of goods received by the j wholesalers for their payment is given by $Q_j = \frac{Q_i * I_j}{\sum I_j}$

[3]In historical terms, the bargaining positions—arbitraged by the Crown—did matter in deciding which side could profit the most from the rents generated in the Atlantic trade

other competing merchants.

The wholesalers also discriminate among retailers to make the most profit out of the operation. The same market pattern as the First Market is repeated: The wholesalers exchange all of their goods for retailers' money/income.[4]

A.2.3 Third Market: Retailers-Population

Given the larger amount of agents involved and its fewer rentier features in comparison with the intermediary markets, I proceed to model the consumer market as in *Figure 1.* Following standard practices among computational representations of partial markets—and given that there is only one good in the model— income relates one-to-one to the willingness to pay of the population. And given that the initial population's income is set by an exponential distribution, it is possible to derive a downward sloped demand curve. The supply curve is set by the total amount of goods available in the American economy; it equals the Spanish supplier production minus the amount lost in the transatlantic trade. I assume it to be inelastic because of the disconnection between the European production process and its demand in the Americas.[5] The other fundamental assumption is that the population satisfies their demand by consuming only one good. The Crown imposes a sale tax of 5%[6] on the price and derives its income from it.

The competitive price in the consumer market arises out of the interaction between the supply of goods and the population's income. It is important to stress that the population's income grows exogenously. The individual rate is given by a normal distribution ($\sim N(\mu, \sigma) = N(.25, .5)$).

It is important to stress two corollaries derived from the model's assumptions: *i)* The retailer and wholesaler's profitability do not necessarily positively correlate with total goods availability. It all depends on the particular convexity of the demand curve—which varies from simulation to simulation due to random initialization settings and random growth rates;[7] *ii)* The retailers face a fixed sell price

[4]The income of the wholesaler i is given by $I_i = \frac{\sum I_j * Q_i}{\sum Q_i}$. Alternatively, the quantity of goods received by the j retailer for their payment is given by $Q_j = \frac{\sum Q_i * I_j}{\sum I_j}$

[5]I could build a positively sloped curve by taking into account the different 'willingness to sell' prices of the retailers given their costs – the price they paid to the wholesalers. I didn't do it to simplify the market process.

[6]Admittedly, the tax rate is arbitrary. Yet the relevance is not on the specific number, but on the trends, it produces

[7]The aggregate demand curve in the retailer market arises out of the income constraints of the population. The curve is always negatively sloped, but its particular shape differs in-between simulations.

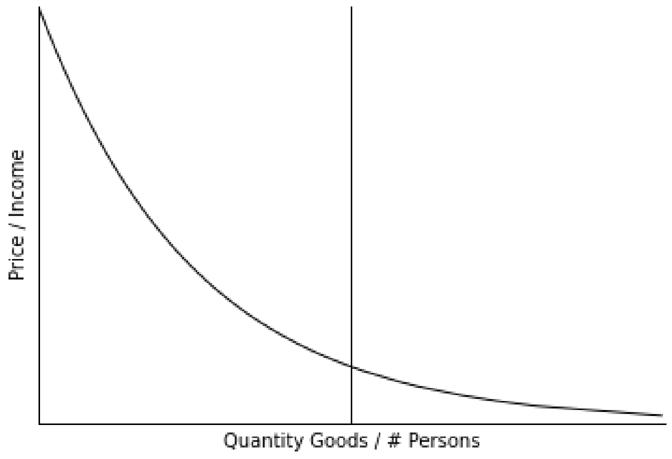


Figure 1: Retailer Market

while buying at differentiated prices. Hence, the model may generate industry concentration.

A second relevant thing to note is the model's assumption of limiting one good per person. It creates shortages that leave people with unmet demand. Historical literature has stressed the potential political problems arising out of this—out of political uncertainty.

A.2.4 Crown's Behavior

As explained in *Figure* **??**, the Crown acquires revenue by taxing the retailer's market. The Crown also has the prerogative to offer trade privileges to the retailers. To do so, the Crown follows a particular heuristic that requires three factors to be activated: *i)* There must be a potential individual retailer with enough industry power to bargain for its inclusion as a new wholesaler. I define that such an event happens whenever a unique retailer owns at least 66% of all the retailers' income;[8] *ii)* The Crown's revenue must appear to be following a decreasing trend. I quantify such a scenario by making the Crown remember their ten immediate predecessor income data points and by estimating the trend by a simple OLS regression; iii) The Crown must face increasing uncertainty in their income revenue. I estimate this by quantifying the difference between the coefficient of variances of the ten recent income data-points *vis a vis* the ten to second-last income data-points.

In short, the Crown will grant trade privileges whenever it has a perspective of decreasing income,

[8]It's possible to construct a Herfindahl Index within the model that self-updates the concentration. Ultimately I didn't incorporate it as it didn't change the results in a significant way.

increasing uncertainty, and when there is at least one potential retailer big enough to become a wholesaler. Thus, the first two requirements provide a venue for exogenous Crown change, while the last one allows the possible endogenous emergence of wholesalers. This resembles the historical case, as I argue in detail in the central historical section of the paper. The idea is that competing merchant guilds were granted licenses only when they had become large enough to have a voice and demand recognition from the Crown. Only after the trade opportunities of normal trade with the initial existing merchant guild were exhausted.

It is important to stress the Crown's privilege granting process is delimited in an *ad hoc* way by expecting it to happen only every five turns. The limitation is set up to avoid a natural pitfall in the modeling design: whenever the conditions exist in time t, it is probable that they will continue to do so in time $t+1$ and hence the need for a time limit. Historically speaking, this resembles the huge time discrepancies in the travel of information between the different parts of the empire. So significant decisions were always slow and not at all immediate affairs.

B Online Appendix B: Other Relevant Figures

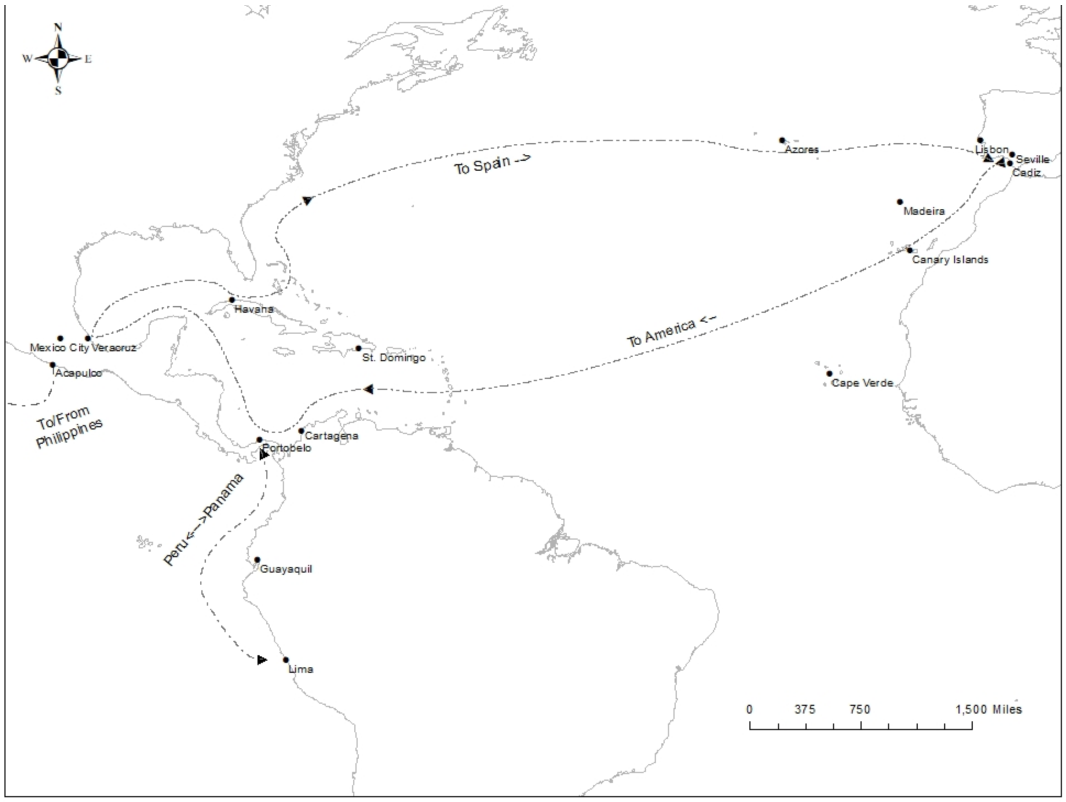
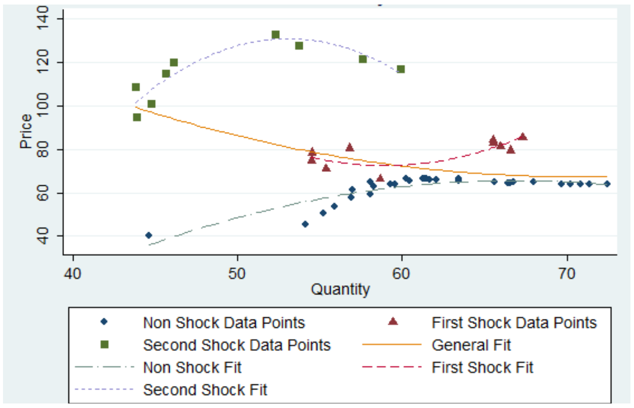
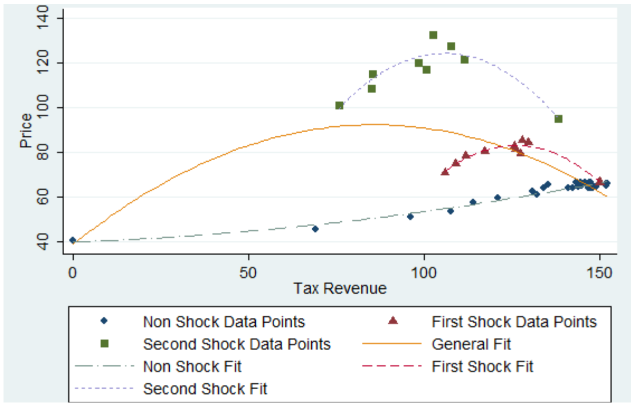


Figure 2: Spanish America and its Main Trading Routes

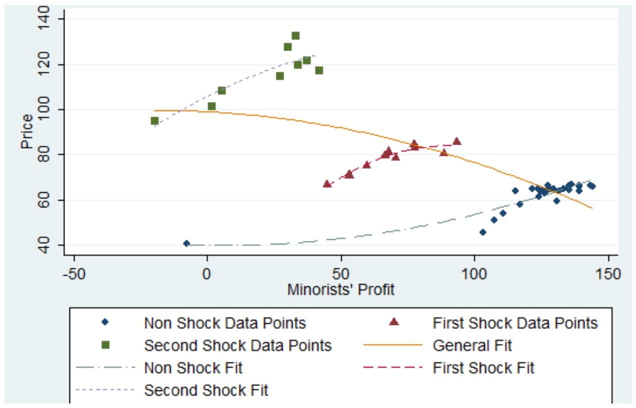
(a) Price and Quantity



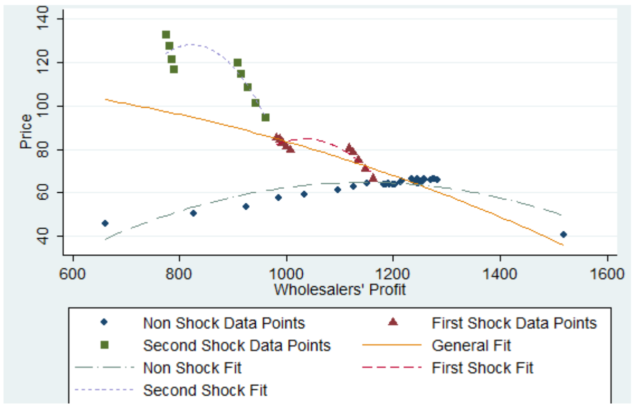
(b) Price and Tax Revenue



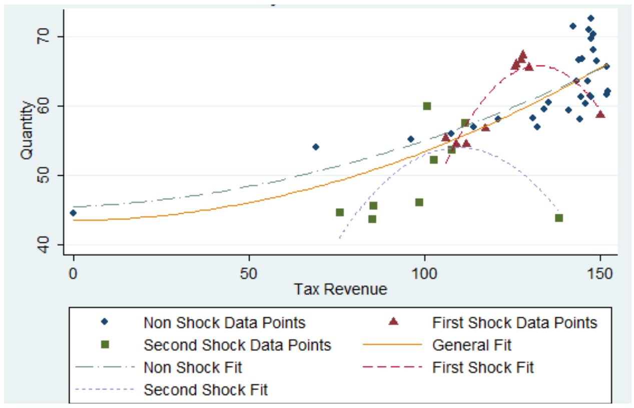
(c) Price and Retailers' Profit



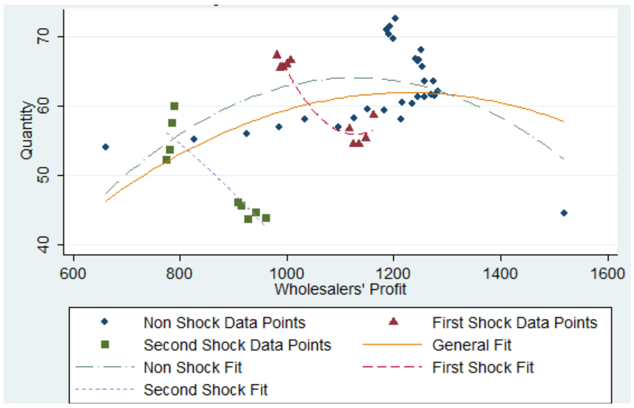
(d) Price and Wholesalers' Profit



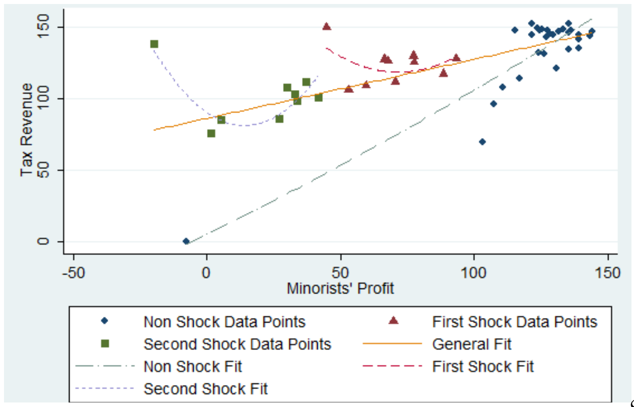
(e) Quantity and Crown's Tax Revenue



(f) Quantity and Wholesalers' Profit



(g) Crown's Tax Revenue and Retailers Profit



(h) Crown's Tax Revenue and W Profit

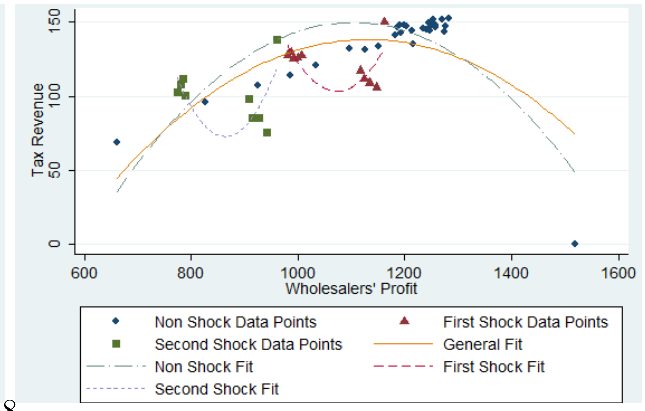


Figure 3: Scatter Plots from the Simulation Results

C Online Appendix C: Python Code

```
""" A simple model of Institutional Reform in the Spanish Empire, 18th Century
Author: Arteaga, Fernando
"""

import random as rd
import numpy as np
import matplotlib.pyplot as plt
import math as mt

time = 0
Price=0
Quantity=0
Quantity_List=[]
Price_list=[]
Crown_Revenue_List=[]
Avg_Wholesaler_Profit_List =[]
Avg_Minorist_Profit_List=[]
Number_Wholesalers_List =[]
Number_Minorists_List=[]
Her_Index=[]
Pop_Initial_Distribution =[]
Pop_Final_Distribution = []
Merch_Initial_Distribution =[]
Merch_Final_Distribution=[]

def GenBoundedRandomNormal(meanVal,stdDev,lowerBound,upperBound):
aRand = rd.gauss(meanVal,stdDev)
while (aRand < lowerBound or aRand > upperBound):
aRand = rd.gauss(meanVal,stdDev)
return aRand


class Population(object):

def __init__(self,ID):
self.ID = ID
self.income =mt.log(1 - rd.uniform(0, 1))/(-1)*10
self.past_income = self.income
self.consumption = False

def Step(self):
self.income = self.income + (self.past_income * (1 + \
(GenBoundedRandomNormal(.25,.5,-1.5,1.5) /100)))
self.consumption = 0


class Merchant(object):

def __init__(self,ID):
self.ID = ID
self.income = (mt.log(1 - rd.uniform(0, 1))/(-1))*100
self.past_income = 0
self.goods = 0
self.potential_goods = 0
self.total_goods=0
self.accounting_goods =[]
self.Wholesaler = False
self.Spanish_Merchant= False
self.profit = 0
self.bankrupt= False
self.Mkt_Powe = 0

def Step(self):
```

```
if self.income<= 0:
self.bankrupt=True


class Crown(object):

def __init__(self):
self.population_pool = [Population(x)for x in range(300)]
self.merchant_pool = [Merchant(y)for y in range(20)]
self.Trade_Risk = 0
self.tax_revenue = 0
self.tax_rates= 5

for i in self.merchant_pool:
if i.ID == 0:
i.Spanish_Merchant = True
i.income = 0
i.goods = 50
if i == max(self.merchant_pool, key=lambda p: p.income) and \
i.Spanish_Merchant == False:
i.Wholesaler = True
break

Tot_Income= []
for i in self.merchant_pool:
if i.Spanish_Merchant==False and i.Wholesaler==False:
Tot_Income.append(i.income)
for i in self.merchant_pool:
if i.Spanish_Merchant==False and i.Wholesaler==False:
i.Mkt_Powe= i.income/sum(Tot_Income)

def Step(self):

if time==2:
for i in self.population_pool:
Pop_Initial_Distribution.append(i.income)

for i in self.merchant_pool:
if i.Wholesaler ==False and i.Spanish_Merchant == False:
Merch_Initial_Distribution.append(i.income)

if time == 49:
for i in self.population_pool:
Pop_Final_Distribution.append(i.income)
for i in self.merchant_pool:
if i.Wholesaler ==False and i.Spanish_Merchant == False:
Merch_Final_Distribution.append(i.income)

self.tax_revenue = 0

if time >= 1:
for i in self.population_pool:
i.Step()

#1st, Spanish Merchant trades with Wholasalers
for i in self.merchant_pool:
if i.Spanish_Merchant==False and i.Wholesaler==False and \
i.bankrupt==False:
i.Step()

Tot_Income= []
for i in self.merchant_pool:
if i.Spanish_Merchant==False and i.Wholesaler==False:
Tot_Income.append(i.income)
for i in self.merchant_pool:
if i.Spanish_Merchant==False and i.Wholesaler==False:
i.Mkt_Powe= i.income/sum(Tot_Income)

for i in self.merchant_pool:
```

```
if i.Spanish_Merchant == True:
i.goods *= (1 + (GenBoundedRandomNormal(.25,.5,-1.5,1.5) /100))
Lista0=[]
for j in self.merchant_pool:
if j.Wholesaler==True:
if j.income !=0:
Lista0.append(j.income)
if j.income ==0:
print "mistake"
Lista0.append(1000)

for j in self.merchant_pool:
if j.Wholesaler==True:
j.past_income = 0
j.potential_goods = i.goods * \
((j.income) / float(sum(Lista0)))
j.income = 0
Lista1=[]
for j in self.merchant_pool:
if j.Wholesaler==True:
if self.Trade_Risk == 0:
z= np.random.chisquare(10, size=None)
if z >95:
z=95
j.goods = j.potential_goods * (1-(z/100))
j.total_goods = j.goods
Lista1.append(j.goods)
if self.Trade_Risk== 1:
z= np.random.chisquare(40, size=None)
if z >95:
z=95
j.goods = j.potential_goods * (1-(z/100))
j.total_goods = j.goods
Lista1.append(j.goods)
if self.Trade_Risk== 2:
z= np.random.chisquare(70, size=None)
if z >95:
z=95
j.goods = j.potential_goods * (1-(z/100))
j.total_goods = j.goods
Lista1.append(j.goods)

#2nd, Wholesalers trade with the rest of the Merchants
ProfitList1=[]
Lista2=[]
Lista3=[]
for i in self.merchant_pool:
if i.Wholesaler==False and i.Spanish_Merchant==False and \
i.bankrupt==False:
Lista2.append(i.income)

for i in self.merchant_pool:
if  i.Wholesaler == True:
for j in self.merchant_pool:
if j.Wholesaler == False and \
j.Spanish_Merchant== False and i.bankrupt==False:
j.goods += i.total_goods * \
((j.income) / float(sum(Lista2)))
i.goods -= i.total_goods * \
((j.income) / float(sum(Lista2)))
j.accounting_goods.append(i.total_goods * \
((j.income) / float(sum(Lista2))))
j.past_income = j.income
Lista3.append(j.goods)

for j in self.merchant_pool:
if j.Wholesaler==False and j.Spanish_Merchant==False and i.bankrupt==False:
pos=0
for i in self.merchant_pool:
```

```
if i.Wholesaler==True:
i.income+= (j.accounting_goods[pos]/  \
sum(j.accounting_goods) )*j.income
j.income-= (j.accounting_goods[pos]/  \
sum(j.accounting_goods) )*j.income
pos+=1

for i in self.merchant_pool:
if i.Wholesaler==True:
i.profit = i.income -i.past_income
ProfitList1.append(i.profit)

global Quantity
Quantity = sum(Lista3)
Quantity_List.append(sum(Lista3))
Avg_Wholesaler_Profit_List.append(np.mean(ProfitList1))

#3rd, Minorist Merchants sell to the Population
Price_Before_Tax = 0
Sorted_Pop_Income = sorted(self.population_pool, key=lambda \
p: p.income, reverse=True)
if len(Sorted_Pop_Income) < Quantity: # Price base limit is 1
global Price
Price = 1
Price_list.append(Price)
else:
P = Sorted_Pop_Income[int(Quantity)-1]
global Price
Price_Before_Tax = P.income
Price = Price_Before_Tax * 1.05
Price_list.append(Price)

ProfitList2=[]
for i in self.merchant_pool:
if i.Wholesaler == False and i.Spanish_Merchant== False and \
i.bankrupt==False:
for j in self.population_pool:
if j.income >= Price:
while j.consumption == False and i.goods>=1:
i.income += Price_Before_Tax
j.income -= Price_Before_Tax
self.tax_revenue += Price - Price_Before_Tax
i.goods-= 1
j.consumption=True
i.profit = i.income - i.past_income
if i.bankrupt == False:
ProfitList2.append(i.profit)
Avg_Minorist_Profit_List.append(np.mean(ProfitList2))

Crown_Revenue_List.append(self.tax_revenue)
z= []
LL=[]
XL=[]
for i in self.merchant_pool:
if i.Spanish_Merchant == False and i.Wholesaler == True and \
i.bankrupt == False:
XL.append(i)
i.accounting_goods=[]
if i.Spanish_Merchant == False and i.Wholesaler == False:
z.append(i)
if i.bankrupt == False:
LL.append(i)
Number_Minorists_List.append(len(LL))
Number_Wholesalers_List.append(len(XL))
Bounded_Revenue_List = Crown_Revenue_List[len(Crown_Revenue_List)-11: \
len(Crown_Revenue_List)-1]
Bounded_Revenue_list2= Crown_Revenue_List[len(Crown_Revenue_List)-11: \
len(Crown_Revenue_List)-2]
WW=0
```

```
if time >10:
XX = np.arange(0, len(Bounded_Revenue_List) )
YY = np.array(Bounded_Revenue_List)
zz = np.polyfit(XX,YY,1)
WW = float("{0}".format(*zz))
Coef_Var_A = np.std(Bounded_Revenue_List)/ \
abs(np.mean(Bounded_Revenue_List))
Coef_Var_B = np.std(Bounded_Revenue_list2)/ \
abs(np.mean(Bounded_Revenue_list2))

HIU=[]
for i in z:
HIU.append(i.Mkt_Powe**2)
Herf_Ind_Una= sum(HIU)
Her_Index.append(Herf_Ind_Una)

"""
Potential Use of Herfindahl Index - CURRENTLY NOT IN THE MODEL
if Herf_Ind_Una > .20:
if WW<0:
if Coef_Var_A > Coef_Var_B:
for i in z:
if i==max(z,key=lambda p: p.income):
i.Wholesaler = True
"""

if time % 5 ==0:
GG = (sum(i.income for i in z)/1.5)
if max(z,key=lambda p:p.income)> GG:
if WW<0:
if Coef_Var_A> Coef_Var_B:
for i in z:
if i==max(z,key=lambda p: p.income):
i.Wholesaler = True

if time>=30: #First Shock, 1760's
self.Trade_Risk = 1

if time>=40: #Second Shock, 1780s onwards,
self.Trade_Risk = 2

def SaveReport2File(self,fileName):

fileObj=open(fileName, 'w')
lineList = []
lineList.append("Quantity; Price Crown_Revenue;Avg_Wholesaler_Profit;Avg_Minorist_Profit \
;Number_Wholesalers;Number_Minorists;\n")
numSteps = len(Quantity_List)
for i in range(numSteps):
oneLine = "%s;%0.2f;%0.2f;%0.2f;%0.2f;%0.2f;%0.2f;\n" %(Quantity_List[i],Price_list[i], \
Crown_Revenue_List[i],Avg_Wholesaler_Profit_List[i],Avg_Minorist_Profit_List[i], \
Number_Wholesalers_List[i], Number_Minorists_List[i])
lineList.append(oneLine)
fileObj.writelines(lineList)
fileObj.close()


class Sim(object):

def __init__(self):
self.acrown= Crown()
def Run(self):
global time
for i in range(50):
self.acrown.Step()
time += 1
self.acrown.SaveReport2File("SpanishEmpire.txt")

if __name__ == '__main__':
```

```
aSim = Sim()
aSim.Run()
```

References

Becker, Gary, "Irrational Behavior and Economic Theory," *Journal of Political Economy*, 1962, *70* (1), 1–13.

Gode, Dhananjay and Shyam Sunder, "Allocative Efficiency of Markets with Zero-Intelligence Traders: Market as a Partial Substitute for Individual Rationality," *Journal of Political Economy*, 1993, *101* (1), 119–137.