# Lorenz Curve, Gini coefficient and an agent-based model of economic welfare

December 11, 2023

## Contents

# 1 the Lorenz curve

## 1.1 the concept of the Lorenz curve

The Lorenz curve was developed by Max O. Lorenz at the beginning of the 20th century. It is a graphical representation of the distribution of wealth.

The 'x' axis of the Lorenz curve represents the cumulative population and the 'y' axis represents the cumulative share of wealth. A Lorenz curve with an equal distribution of wealth (i.e. everyone has the same amount of wealth) would be a straight line with an upward slope of 1. However, since wealth is not equally distributed in practice, the Lorenz curve is usually a curve with an accelerating slope. The more bulbous the curve, the more unequally wealth is distributed in a population.

To construct the Lorenz curve, one needs to calculate the cumulative share of wealth given the cumulative population and connect these points. The population share is simply: $\frac{1}{N}$, where N is the size of the population. The cumulative wealth share can be calculated as follows.

$$l_j = \frac{\sum_{i=1}^{j} w_i}{\sum_{i=1}^{N} w_i} \quad (\text{and } l_0 = 0)$$

In this equation, $l_j$ is the cumulative wealth share up to person j and $w_i$ is the wealth of person i.

## 1.2 plotting of the Lorenz curve

For our project, we have created a function that produces a graph of the Lorenz curve based on the wealth data (curve in red) and a Lorenz curve for an equal distribution of wealth (line in blue). The inputs to the function are Wealth and Fill. Wealth must be a vector of reasonable length and should only contain non-negative numbers. If it contains negative numbers, they will be converted to positive numbers. You can also decide whether the space between the Lorenz curve of the entered vector and the Lorenz curve of an equal distribution should be filled by setting fill to TRUE or FALSE. There are no default values. Our code:

```
plot.lorenz <- function(wealth, fill) {

  #first we sort the unsorted input vector
```

```r
#and make sure it doesn't contain any negative values
sorted.wealth.lorenz <- sort(abs(wealth))

#we determine how many people the population contains
number.of.people <- length(sorted.wealth.lorenz)

#we calculate all the data we need to create the plot
#and put it into a dataframe df
relative.wealth.cummulated <-
  c(
    0,
    cumsum(sorted.wealth.lorenz/sum(sorted.wealth.lorenz))
  )

relative.people.cummulated <-
  seq(from =  0,
      to = 1,
      by = 1/number.of.people)

df <- data.frame(
  relative.people.cummulated,
  relative.wealth.cummulated
)

#we need to load ggplot2 (make sure it is installed)
library(ggplot2)

#we create the plot p
#and supply it with the necessary data
p <- ggplot(
  df,
  aes(relative.people.cummulated,
      relative.wealth.cummulated))+

  #constructing the red line
  geom_line(col = "red",
            linewidth = 1)+

  #constructing the blue line for an equal distribution
  geom_segment(x = 0, y = 0,
```

```
                xend = 1, yend = 1,
                col = "blue",
                linewidth = 1,
                linetype = "solid")+

  #adding a title to the diagram
  labs(
    title = "Lorenz curve",
    x = "cumulative population",
    y = "cumulative share wealth",
    )+

  #scaling x and y
  scale_x_continuous(breaks = seq(0, 1, by = 0.2))+
  scale_y_continuous(breaks = seq(0, 1, by = 0.2))+

  #changing the theme of the diagram
  theme_classic()+

  #adding the outer lines
  geom_segment(x = 0, y = 0, xend = 0, yend = 1)+
  geom_segment(x = 0, y = 1, xend = 1, yend = 1)+
  geom_segment(x = 1, y = 0, xend = 1, yend = 1)+
  geom_segment(x = 0, y = 0, xend = 1, yend = 0)+

  #adjusting the position of the title
  theme(plot.title = element_text(hjust = 0.5, vjust = 0.2))

#checking whether the space between the lines
#should be filled
  if (fill == TRUE) {

    #if so adapt p
    p <- p+

      #adding the filling
      geom_ribbon(
        aes(ymin = relative.people.cummulated,
            ymax = relative.wealth.cummulated),
        fill = "red",
```

```
            alpha = 0.5)
    }

  #returning the plot
    return(p)
}
```

As an example, we provide a vector with random elements ranging from 0 to 10 times the number of elements it contains. We choose a vector of length 10, so the wealth values are between 100. The more values the vector wealth contains, the "smoother" the Lorenz curve.

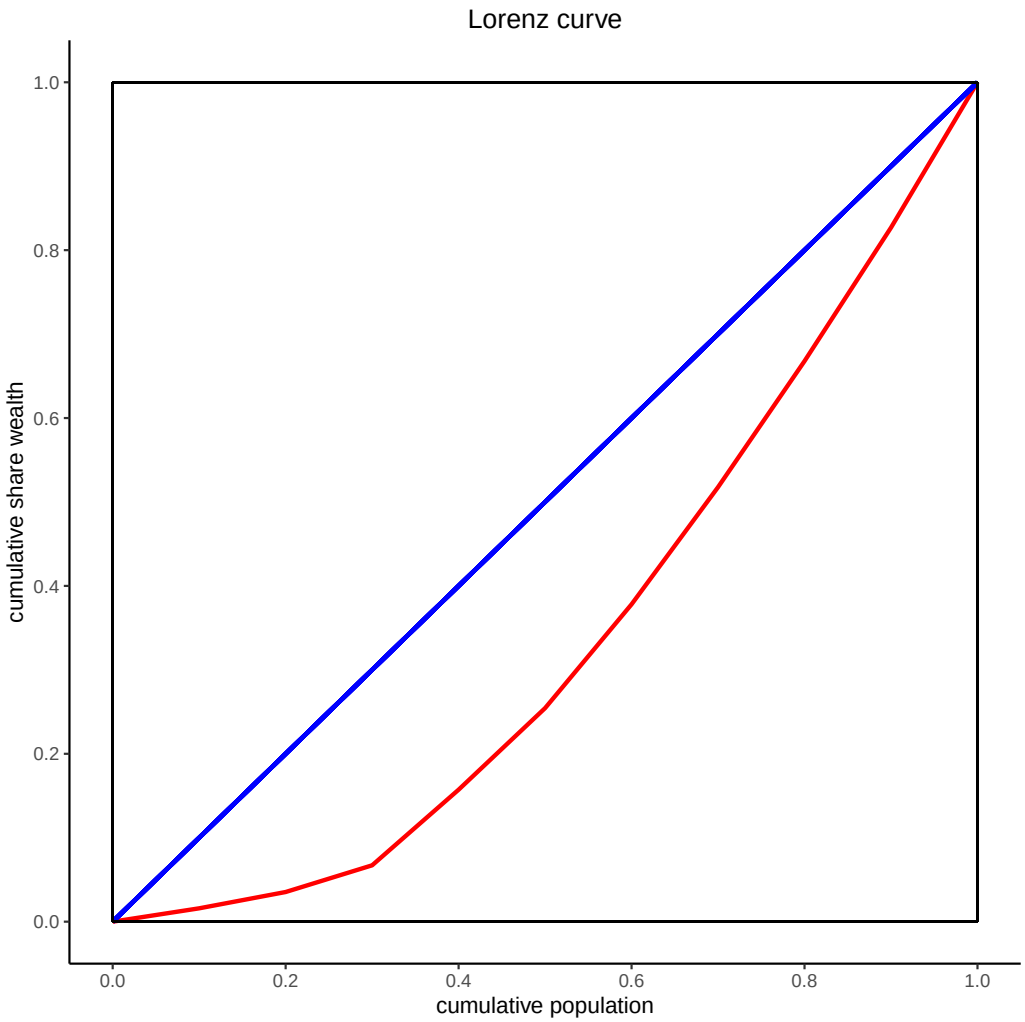
```
number.of.elements <- 10
random.wealth <-
  floor(10*number.of.elements*runif(number.of.elements))
```

In the first graph, the space between the two curves is not filled (fill = FALSE) - in the second, it is filled (fill = TRUE).

We also included an Lorenz curve of a maximal unequal wealth distribution (i.e. one person posses all the wealth).

```
plot.lorenz(random.wealth, fill = FALSE)
```

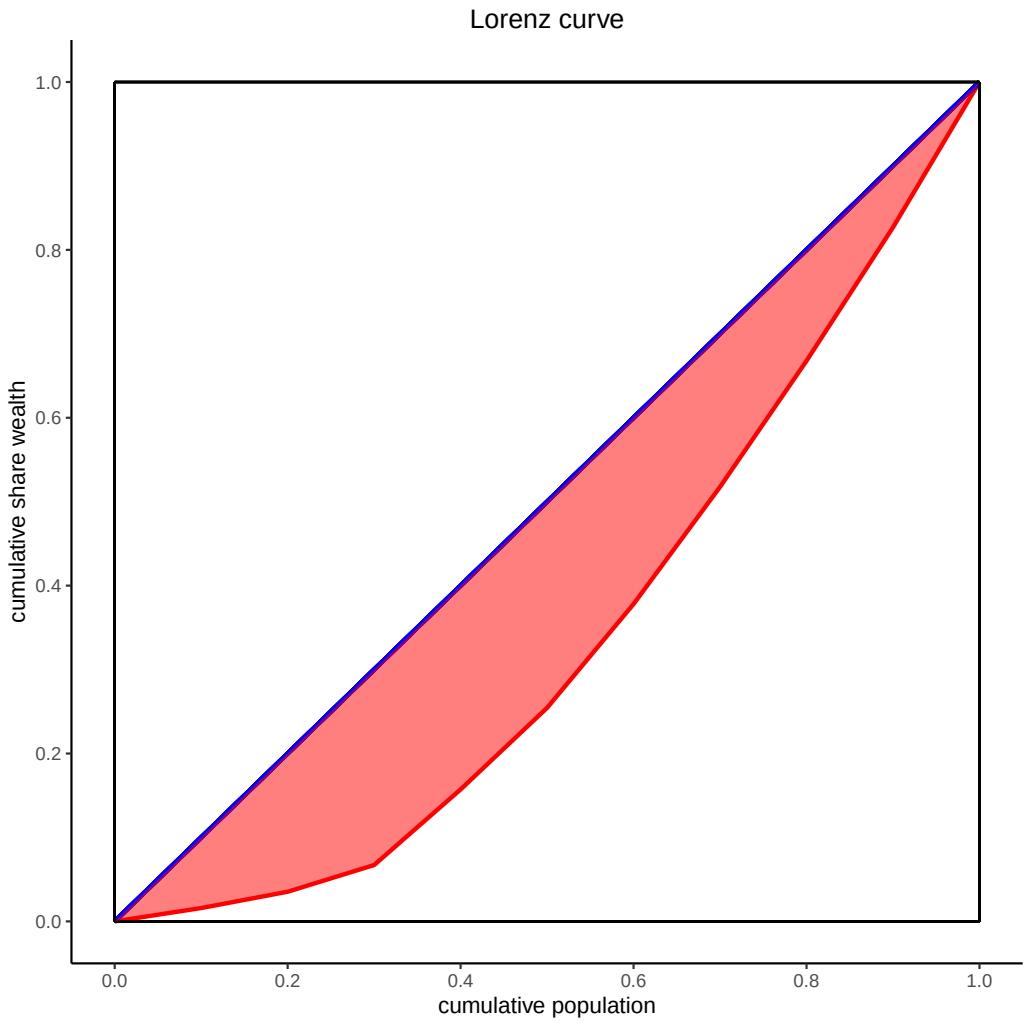
```
## Warning:  Paket 'ggplot2' wurde unter R Version 4.2.3
erstellt
```




```
## above lorenz curve for the given wealth distribution:
##  11 70 98 18 9  ...
```

Figure 1: Lorenz curve, unfilled

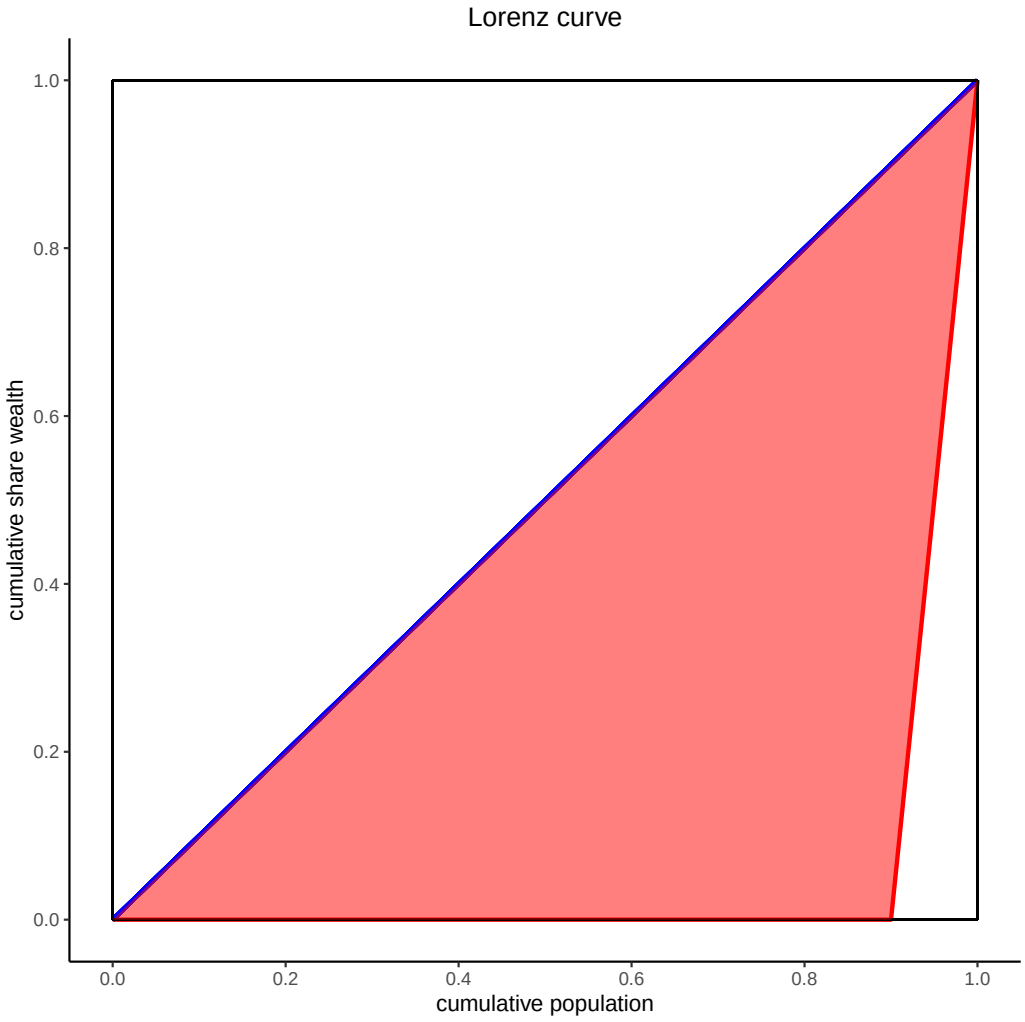
```
plot.lorenz(random.wealth, fill = TRUE)
```




```
## above lorenz curve for the given wealth distribution:
##  11 70 98 18 9  ...
```

Figure 2: Lorenz curve, filled

```
maximal.unequal <- c(rep(0, length(random.wealth)-1), 1)
plot.lorenz(maximal.unequal, fill = TRUE)
```




```
## above lorenz curve for the maximal unequal
##     distribution of wealth
```

Figure 3: Lorenz curve, maximal ineqaultiy, filled

# 2 the Gini coeffiecent

## 2.1 the concept of the Gini coefficient

The Gini coefficient is a measure for wealth inequality. Coorado Gini defined this quotient as the ratio between the area that is spanned by the Lorenz curve of an equal distribution of wealth with the Lorenz curve of the actual distribution (the red area in figure 2 Lorenz curve, filled) as well as the area of maximal inequality (the red area in figure 3 Lorenz curve, maximal ineqaultiy, filled).

The higher the coefficient, the more unequally wealth is distributed in a population.

We find the area between the Lorenz curve of equal distribution and actual distribution (L) by the following calculation. For the definition of the variables see chapter 1.1 the concept of the Lorenz curve.

$$L = \frac{1}{2} - \frac{1}{N}\left(\sum_{i=1}^{N} l_i - \frac{1}{2}\right)$$

The maximal area (M) is given by:

$$M = \frac{N-1}{2N}$$

Hence we find the Gini-coefficient:

$$G = \frac{L}{M}$$

## 2.2 calculating the Gini coefficient

We created a function that calculates the Gini coefficient for a given wealth distribution. The input to the function is wealth. Wealth must be a vector of reasonable length and should only contain non-negative numbers. If it contains negative numbers, they will be converted to positive numbers. There are no default values.

```
gini <- function (wealth) {
  if(missing(wealth)){
    stop("wealth is missing")
  }
```

```
  #first we sort the unsorted input vector
  sorted.wealth.gini <- sort(abs(wealth))

  #determining how many people there are
  number.of.people <- length(sorted.wealth.gini)

  #calculating l (~relative people)
  l_values <- cumsum(sorted.wealth.gini)/sum(sorted.wealth.gini)

  #calculating the area between the Lorenz curve and
  #the diagonal
  L <- 0.5 - (1/number.of.people)*(sum(l_values)- 0.5)

  #calculating the maximal area
  M <- (number.of.people - 1) / (2 * number.of.people)

  #calculating the gini coefficient
  gini_coefficient <- L/M
  return(gini_coefficient)
}
cat("The Gini coefficient for the given wealth
      distribution is: ", gini_coefficient, "\n")

## Error in cat("The Gini coefficient for the given wealth\n
distribution is:  ", :  Objekt 'gini_coefficient' nicht
gefunden
```

As an example, we used the wealth data from before (a randomised vector of length 10 with values between 1 and 100).

```
gini(random.wealth)

## [1] 0.3510012
```

# 3 EYSM

Definition - The extended yard sale model (EYSM) is an agent-based model for simulating interactions between autonomous agents. It is based on the principles of economic market and wealth redistribution.

In the EYSM we have starting wealths $w_1, w_2, \ldots$ and transactions between pairs of agents every iteration. The change in wealth from each transaction is subject to a redistribution among all agents, $\chi$, an expectation $\eta$ that the wealthier agent "wins" the transaction and takes an amount from the poorer agent, and $\gamma$, the amount of wealth taken from the other agent.

We can at most win or lose the amount of wealth of the poorer of the two agents, and the parameters $\chi, \gamma, \eta$ need to be set very precisely so the model works.

## 3.1 Mathematical Formulation

The core of the EYSM model involves several key variables and equations:

- $N$: Number of agents.
- $wealth$: An array representing the initial wealth of each agent.
- $gamma, zeta, chi$: Parameters governing the transactions.

The transactions are modeled as follows:

$$\Delta w_1 = \begin{cases} \chi\left(\frac{W}{N} - w_1\right) + \gamma \min(w_1, w_2), & \text{if agent 1 is wealthier and wins} \\ \chi\left(\frac{W}{N} - w_1\right) - \gamma \min(w_1, w_2), & \text{otherwise} \end{cases}$$

$$\Delta w_2 = \begin{cases} \chi\left(\frac{W}{N} - w_2\right) + \gamma \min(w_1, w_2), & \text{if agent 2 is wealthier and wins} \\ \chi\left(\frac{W}{N} - w_2\right) - \gamma \min(w_1, w_2), & \text{otherwise} \end{cases}$$

Where $\Delta w_1$ and $\Delta w_2$ are the changes in wealth for each agent in a transaction.

## 3.2 Usage in R

. The EYSM model is implemented in R, where the function `EYSM` takes into account parameters such as the number of internals (correlation), the prior richness distribution, and the sample parameters $gamma, zeta, chi$. The function plot correlations, update wealth , Gini coefficient etc. Calculate metrics, including the relative wealth of the richest person

```
# N represents the number of agents we want in our model
N = 100
# for the wealth we sample N values between 1000 and 100,000
wealth <- sample(1000:100000,N)
```

```r
# We define the EYSM function with inputs steps, or how many
#transactions we want,
# walth a vector of numerics or integers, gamma, zeta and chi.
#The zeta is a
# variable used in calculating the expectation of eta being 1
EYSM <- function(steps=100,wealth,gamma=0.1,zeta=0.001,chi=0){
  if(missing(steps)){
    warning("steps is using the default of 100")
  }
  if(missing(wealth)){
    stop("wealth is missing")
  }
  if(missing(gamma)){
    warning("gamma is using the default value of 0.1")
  }
  if(missing(zeta)){
    warning("zeta is using the default value of 0.001")
  }
  if(missing(chi)){
    warning("chi is using the default value of 0")
  }
  if(any(c(steps,wealth,gamma,zeta,chi)<0)){
    stop("you must only provide positive values")
  }

  # first we create a Nx2 matrix of the agent pairs
  agents <- matrix(wealth,ncol = 2)
  N <- length(wealth)

  # we also create empty vectors for the gini coefficients and
  #the relative wealth of the wealthiest agent
  gini <- vector(mode = "numeric",length = steps)
  oligarch <- vector(mode = "numeric",length = steps)

  for(i in 1:steps){
    #in every iteration we resample the combination of agents
    agents <- sample(agents,length(agents),replace = FALSE)
    agents <- matrix(agents,ncol = 2)

    # we create a random variable between -1 and 1, our eta
```

```
#and sample a random one every iteration
RV <- seq(-1,1,by = 0.0001)
eta <- sample(RV,1,replace = T)

# calculating the sum W, and the difference in weigths
#for every pair
W <- rowSums(agents)
difference <- agents[,2]-agents[,1]

# the expectiation of eta as laid out in the formula and
#the probability
E_eta <- (zeta/gamma)*(abs(difference)/(W/N))
p_eta <- (E_eta+1)/2

# adding probability and counter-probability to a matrix
prob <- cbind(p_eta,(1-p_eta))
# getting 1 with a probability of p_eta, otherwise 0,
#the probability of the wealthier agent to win
P_win <- rbinom(nrow(agents),1,prob = prob)

# the minima of each agent pair
mins <- apply(agents,1,FUN = min)

# nested if statements, covering all the outcomes we can
#have in terms of  who is richer and if the richer one wins
# given each outcome we calculate a version of the delta_w1
#given in the
#task and save it to the vector. We do the same with w_2
delta_w1 <- ifelse(agents[,1]>agents[,2] & P_win == 1,
     chi * (W/N - agents[, 1]) + gamma * mins,
     ifelse(agents[,1]>agents[,2] & P_win == 0,
            chi * (W/N - agents[, 1]) - gamma * mins,
            ifelse(agents[,2]>agents[,1] & P_win == 1,
                   chi * (W/N - agents[, 1]) - gamma * mins,
                   chi * (W/N - agents[, 1]) + gamma * mins
                   )))


delta_w2 <- ifelse(agents[,1]<agents[,2] & P_win == 1,
      chi * (W/N - agents[, 2]) + gamma * mins,
```

```
            ifelse(agents[,1]<agents[,2] & P_win == 0,
                   chi * (W/N - agents[, 2]) - gamma * mins,
                   ifelse(agents[,1]>agents[,2] & P_win == 1,
                          chi * (W/N - agents[, 2]) - gamma * mins,
                          chi * (W/N - agents[, 2]) + gamma * mins
                          )))

    # lastly we change the wealth of the agents accordingly
    agents[,1] <- agents[,1]+delta_w1
    agents[,2] <- agents[,2]+delta_w2

    # the relative wealth of the richest agent gets added to the
    #vector, same with the GINI index
    oligarch[i] <- max(agents)/sum(agents)
    gini[i] <- gini(agents)
  }
  # we return a dataframe of the relative wealth and the
  #GINI indices at each transaction
  return(list("wealth" = agents,"rel. wealth" = oligarch,
              "GINI" = gini))
}
```

### 3.3 End

As we can see the EYSM model provides insights into wealth distribution dynamics and factors affecting income inequality.

## 4 Monte Carlo Simulation

Definition - The Economic Yard Sales Model (EYSM) uses Monte Carlo simulation to examine the impact of different parameters on wealth distribution. We can observe how $\gamma$, $\zeta$, and $\chi$ affect the dynamics of wealth inequality (by running several simulations with different parameter values)

### 4.1 Mathematical Formulation

The simulation is repeated over a range of values for $\gamma$, $\zeta$, and $\chi$. Each of those affecting the interaction effects in the EYSM model. The objective is to

assess changes in the Gini coefficient and relative wealth as these coefficients change. The process is as follows:

.

- Start the matrix for Gini coefficients and relative wealth time serie.

- Iterate over a set of $\gamma$, $\zeta$, $\chi$ values

- Run the EYSM model for each combination and record the results

- Change the NA values and outliers of the data for more accurate interpretation

- Plot the time series of each run and include the median as a separate line

## 4.2 Usage in R

For the Monte-Carlo Simulation we try different values for $\gamma$, $\zeta$, $\chi$ and plot each time series of relative wealth as well as GINI coefficient of the course of the transactions. For executions we created matrices that will hold each runs' time series and return a list of the GINI time series and relative wealth time series.

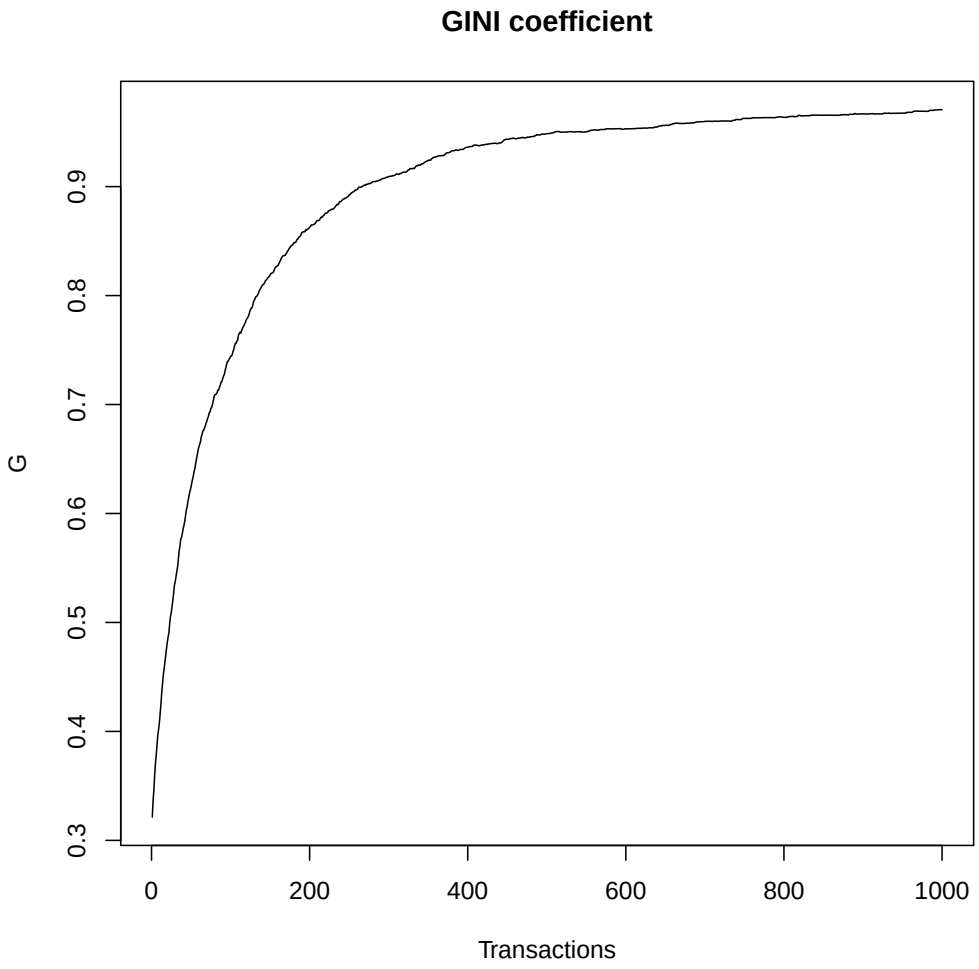
The vectors gamma, zeta and chi contain different values we want to try, due to the fact that the combination of those three matters, we will get some runs that do not work due to creating NA's in the probability calculation for $\eta$, these are also the warnings that might be displayed while running the procedure. The same problem leads to relative wealth above 1, which will be set to NA afterwards to keep reasonable plots and values only.

```
Monte_carlo<-function(steps, wealth, gamma, zeta, chi){
  gini_ts<-matrix(ncol=0, nrow=steps)
  wealth_ts<-matrix(ncol=0, nrow=steps)
    for (i in 1:length(gamma)) {
      for (j in 1:length(zeta)){
        for (l in 1:length(chi)){
          values<-EYSM(steps, wealth, gamma[i], zeta[j], chi[l])
          gini_ts<-cbind(gini_ts, values[[3]])
          wealth_ts<-cbind(wealth_ts, values[[2]])

        }
      }

    }
```

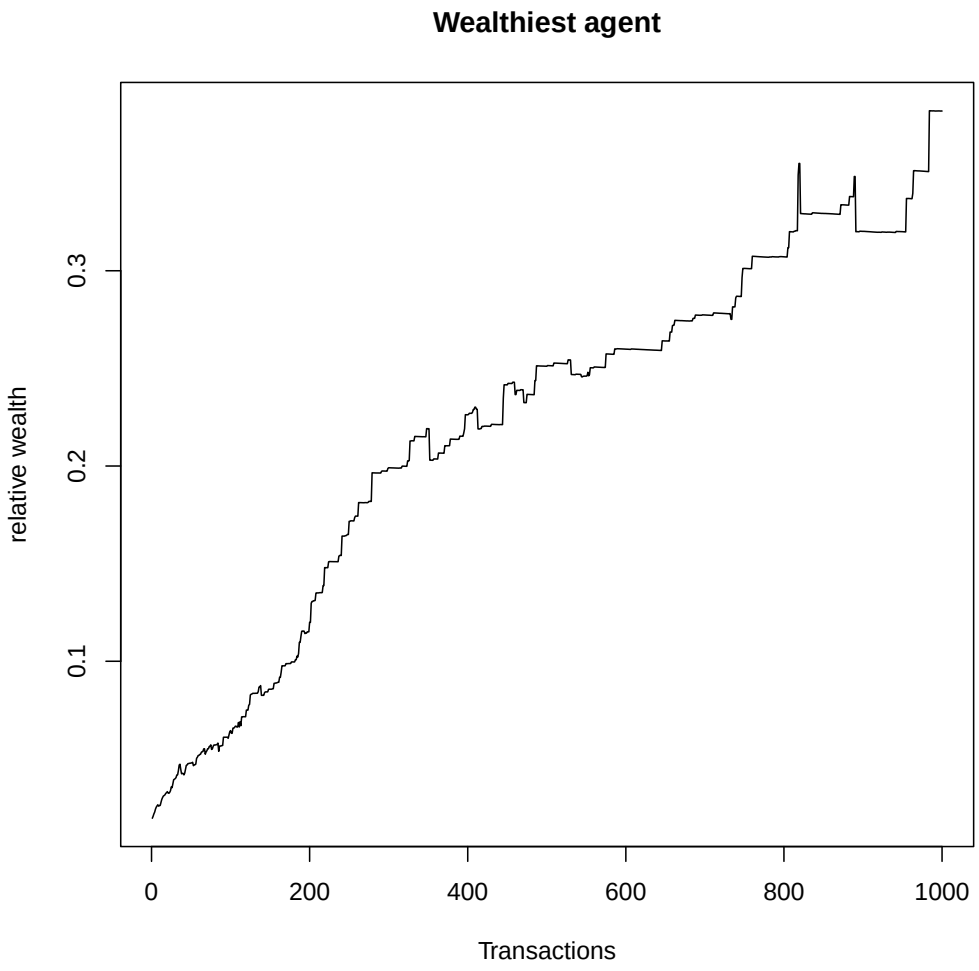
```
    return(list(gini_ts,wealth_ts))
  }
```

In the following plots we run Monte Carlo simulations similar to the ones in the assignment sheet, they may vary due to sample size, random wealth and pure luck of the agents. In the first two we use a $\gamma$ of 0.1 and a $\chi$ of 0.01 while in the second we use a $\gamma$ of 1 and a $\chi$ of 0.

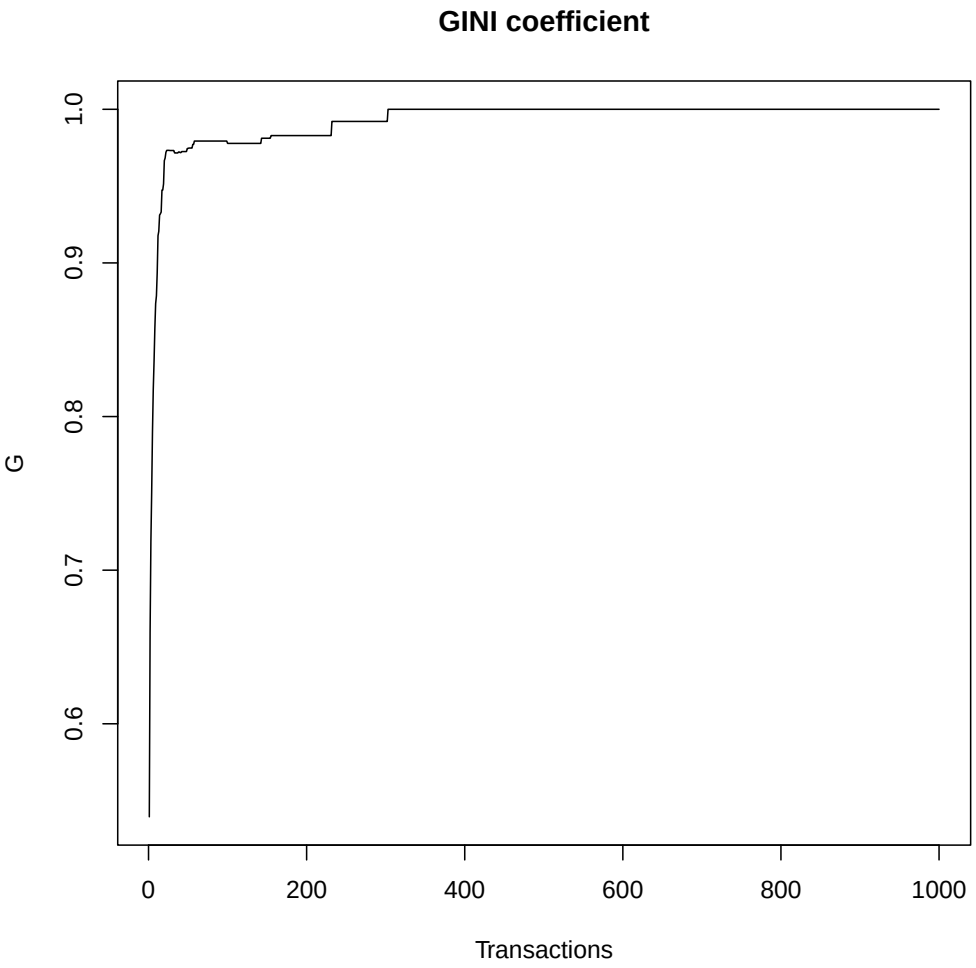
```
MC_simple <- Monte_carlo(1000,wealth,0.1,0.001,0.01)
val_g <- MC_simple[[1]]
plot(val_g,type = "l",main = "GINI coefficient",
     xlab = "Transactions",ylab = "G")
```



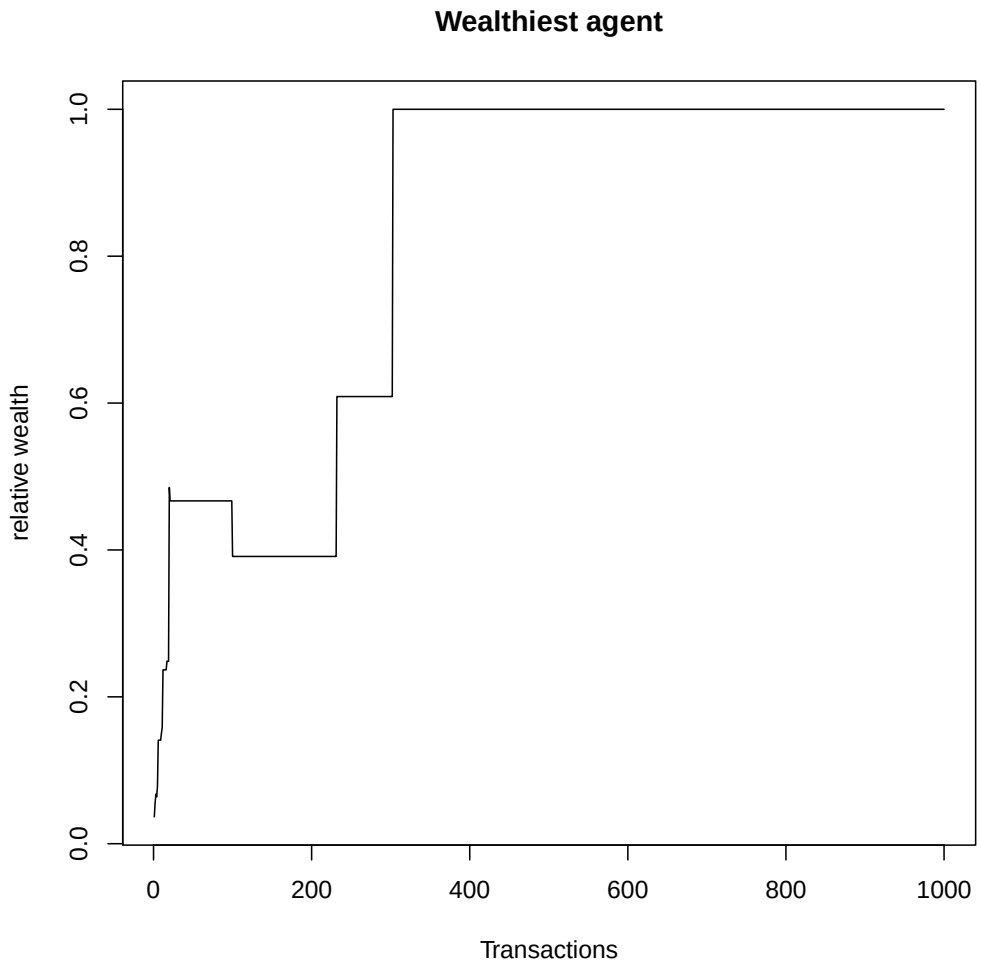
```
values <- MC_simple[[2]]
plot(values, type = "l",main = "Wealthiest agent",
     xlab = "Transactions",ylab = "relative wealth")
```




```
MC_simple2 <- Monte_carlo(1000,wealth,1,0.001,0)
val_g <- MC_simple2[[1]]
plot(val_g,type = "l",main = "GINI coefficient",
     xlab = "Transactions",ylab = "G")
```



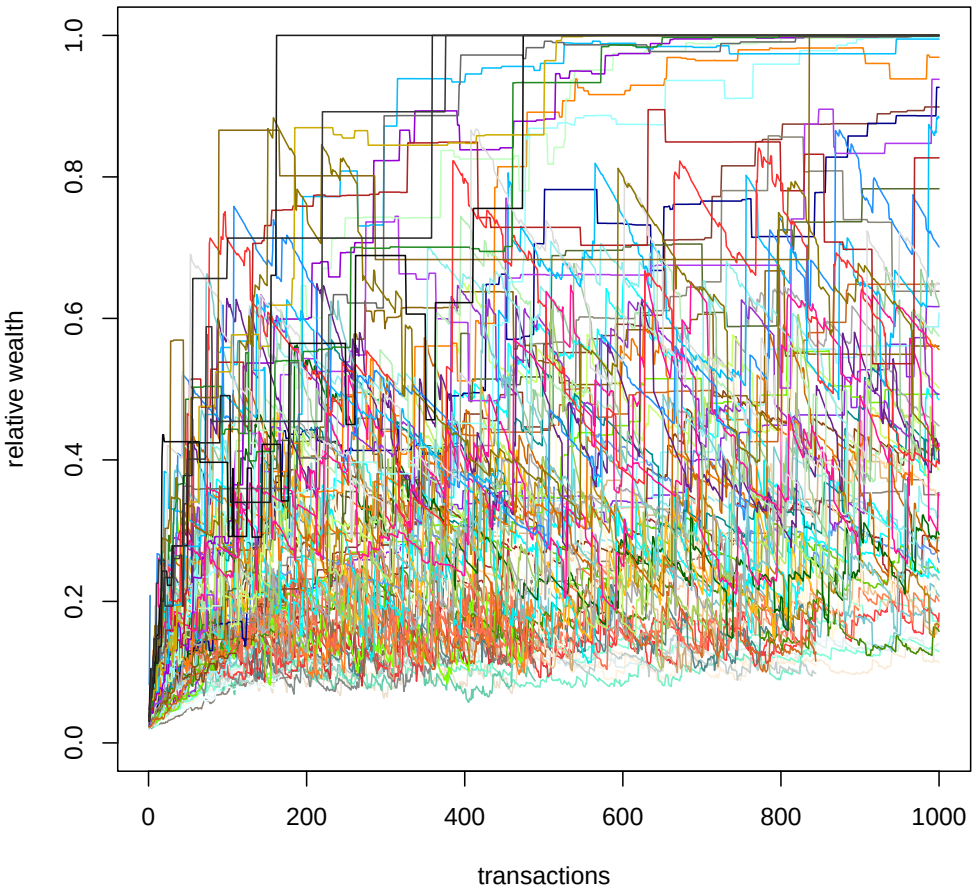

```
values <- MC_simple2[[2]]
plot(values, type = "l",main = "Wealthiest agent",
     xlab = "Transactions",ylab = "relative wealth")
```



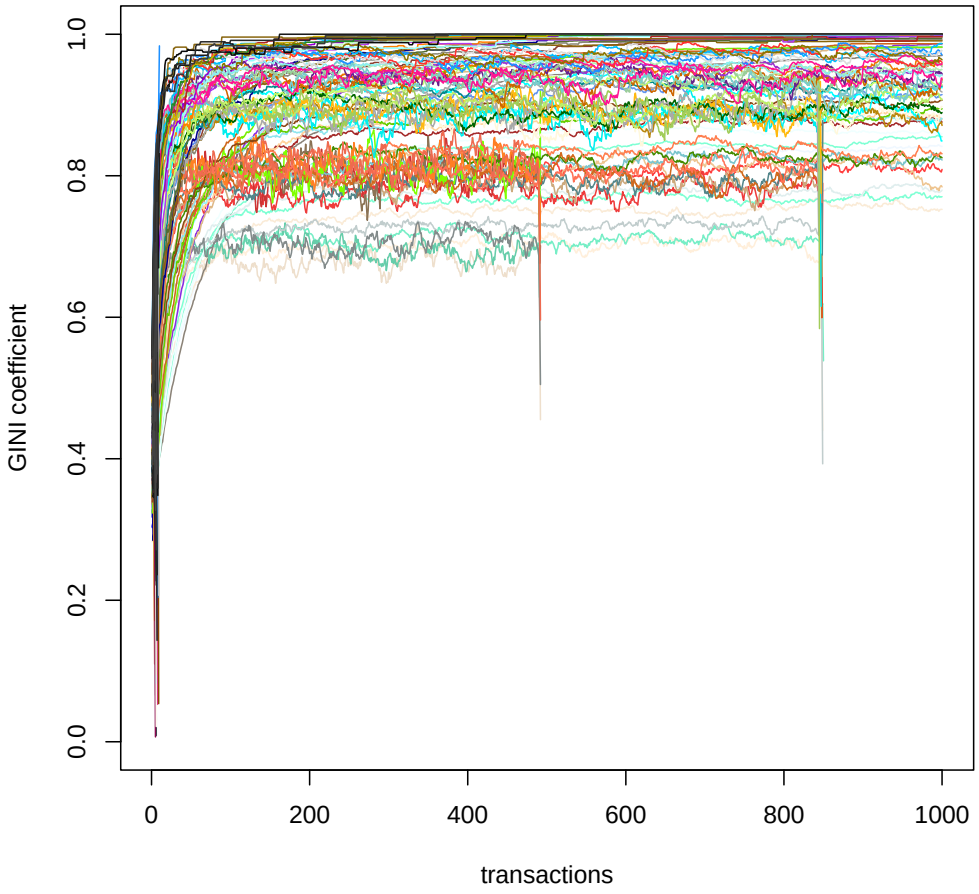

In the following we try many different combination of parameters to get a feel for how the GINI and relative wealth changes due to changes in them.

```
gamma<-c(0.1,0.2,0.4,0.6,0.8,1)
zeta<-seq(0.0008,0.0012, by = 0.0001)
chi<-c(0,0.2,0.4,0.6,0.8,1)

MC_run <- Monte_carlo(1000, wealth, gamma,zeta,chi)
# we extract the lists that contain either the gini
#coefficient of the rel. wealth
wealth_tsts <- MC_run[[2]]
wealth_tsts[which(wealth_tsts>1)] <- NA
```

```
gini_tsts <- MC_run[[1]]

# a vector that contains all the colors in R
color_vector<-colors()
plot(wealth_tsts[,1],type = "l", col=color_vector[1],
     ylim = c(0,1),ylab = "relative wealth",
     xlab="transactions")
for(i in 2:ncol(wealth_tsts)){
  lines(wealth_tsts[,i], col=color_vector[i])
}
```



```
plot(gini_tsts[,1],type = "l",col = color_vector[1],
     ylim = c(0,1),ylab = "GINI coefficient",
     xlab="transactions")
for(i in 2:ncol(gini_tsts)){
  lines(gini_tsts[,i], col=color_vector[i])
}
```




End Thanks to Monte Carlo simulation, we can examine the effects of various parameters on wealth distribution (in simulated economies). It helps us to understand role of economic factors in wealth inequality and complex interactions of them.

# 5 Analysis and Interpretation of Graphs

Key points of our graph analysis:

- The Gini coefficient tends to move towards 1(in most scenarios). This indicates a trend towards increased inequality over time.
- The relative wealth of the wealthiest agent shows significant dependence on the redistribution parameter ($\chi$) and the probability of the wealthier agent winning a transaction ($\zeta$ and $\alpha/\gamma$).
- Occurrence of NA values in the data is noteworthy. Primarily influenced by the interplay of parameters $\zeta$, $\chi$, $\gamma$, etc., as discussed in the text for the EYSM model.