TD1 Assignment Report

DRAME Alia MEIGNEN Erwan LAKAS Idriss

Exercise 1: Demand Curve

Background

We simulate a market of 1000 consumers, where each consumer's reservation price is randomly distributed between 0 and 1. The demand curve shows the relationship between market price and quantity demanded.

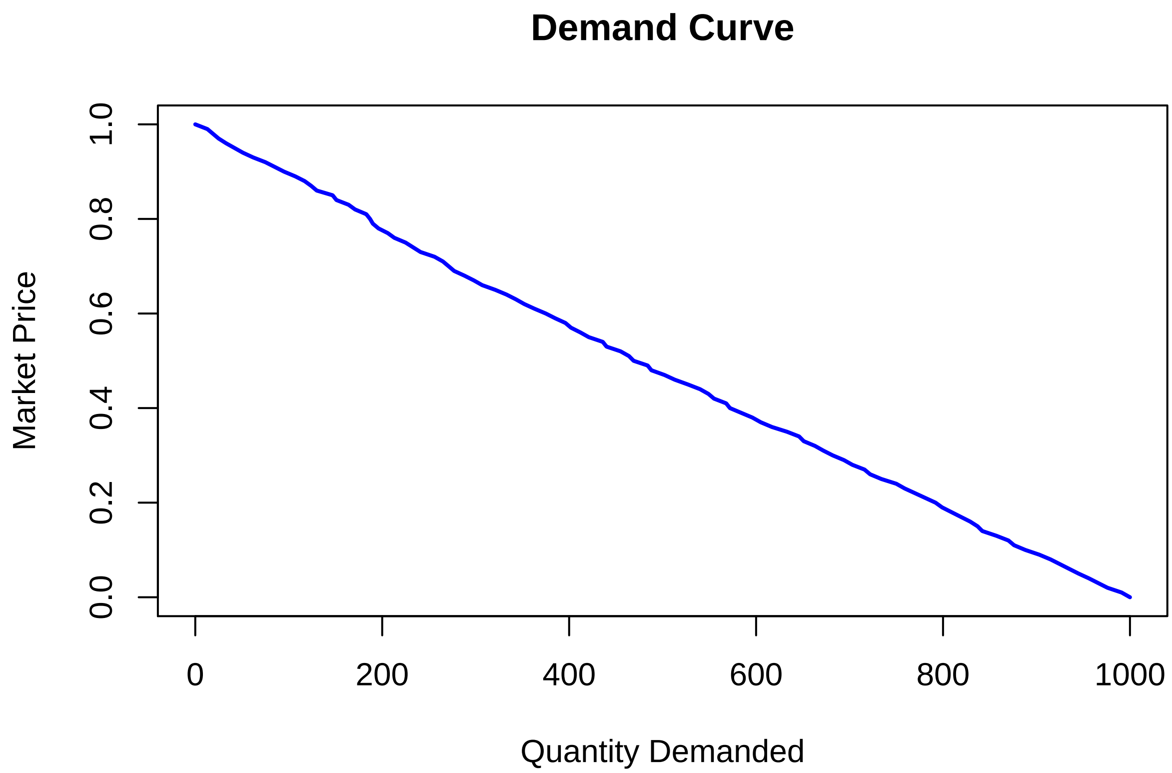
Simulation

Demand Function

```r
# Number of consumers
N <- 1000
# Generate random reservation prices
reservation_prices <- runif(N)

# Generate market prices
market_prices <- seq(0, 1, by = 0.01)

# Calculate quantity demanded for each market price
quantity_demanded <- sapply(market_prices, function(p) sum(reservation_prices >= p))
```

Plot

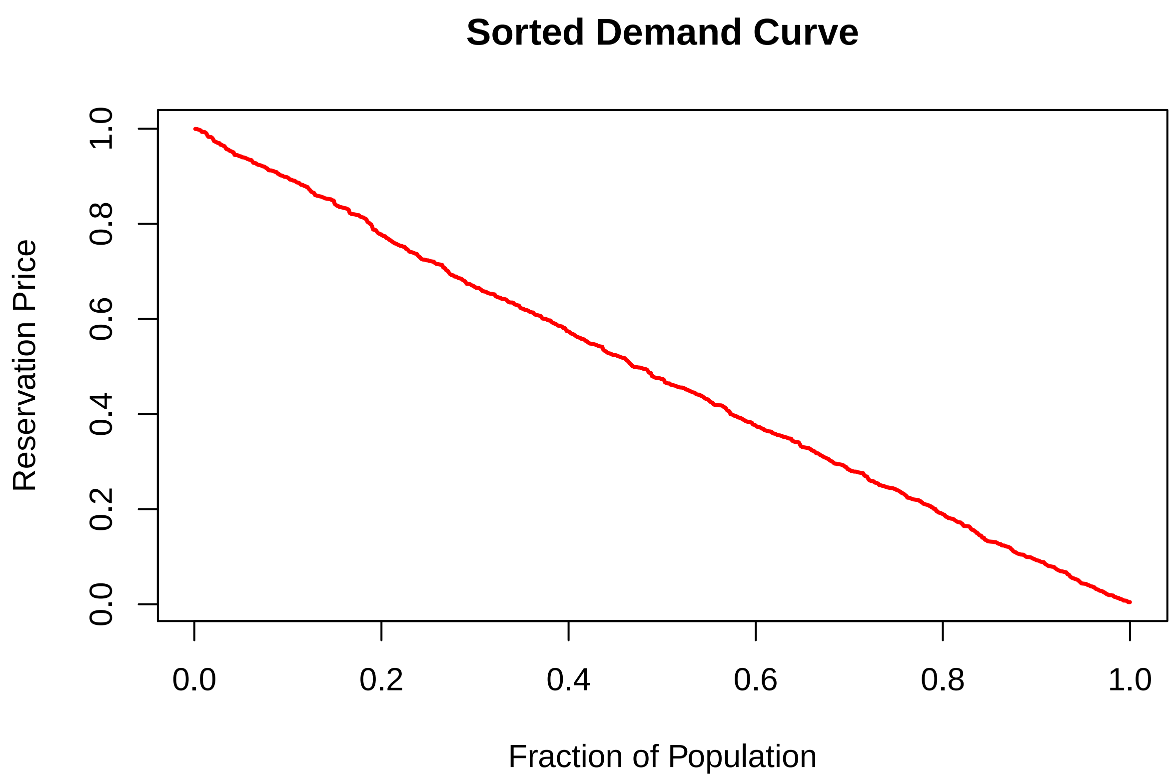
```r
# Plot demand curve
plot(quantity_demanded, market_prices, type = "l", col = "blue", lwd = 2,
     xlab = "Quantity Demanded", ylab = "Market Price",
     main = "Demand Curve")
```




```
# Comment: The demand curve shows how market price decreases as quantity demanded increases, reflecting
```

Alternative Representation

```r
# Sort reservation prices in descending order
sorted_reservation_prices <- sort(reservation_prices, decreasing = TRUE)

# Create fraction of population
fraction_population <- seq(1 / N, 1, by = 1 / N)

# Plot sorted reservation prices against fraction of population
plot(fraction_population, sorted_reservation_prices, type = "l", col = "red", lwd = 2,
     xlab = "Fraction of Population", ylab = "Reservation Price",
     main = "Sorted Demand Curve")
```




```
# Comment: This plot illustrates the distribution of reservation prices, showing how the fraction of th
```

Exercise 2: Monopoly Pricing

Background

We simulate a monopolist's pricing problem, where each day's buyers have reservation prices distributed uniformly on (0,1). The firm's goal is to maximize profits.

Simulation

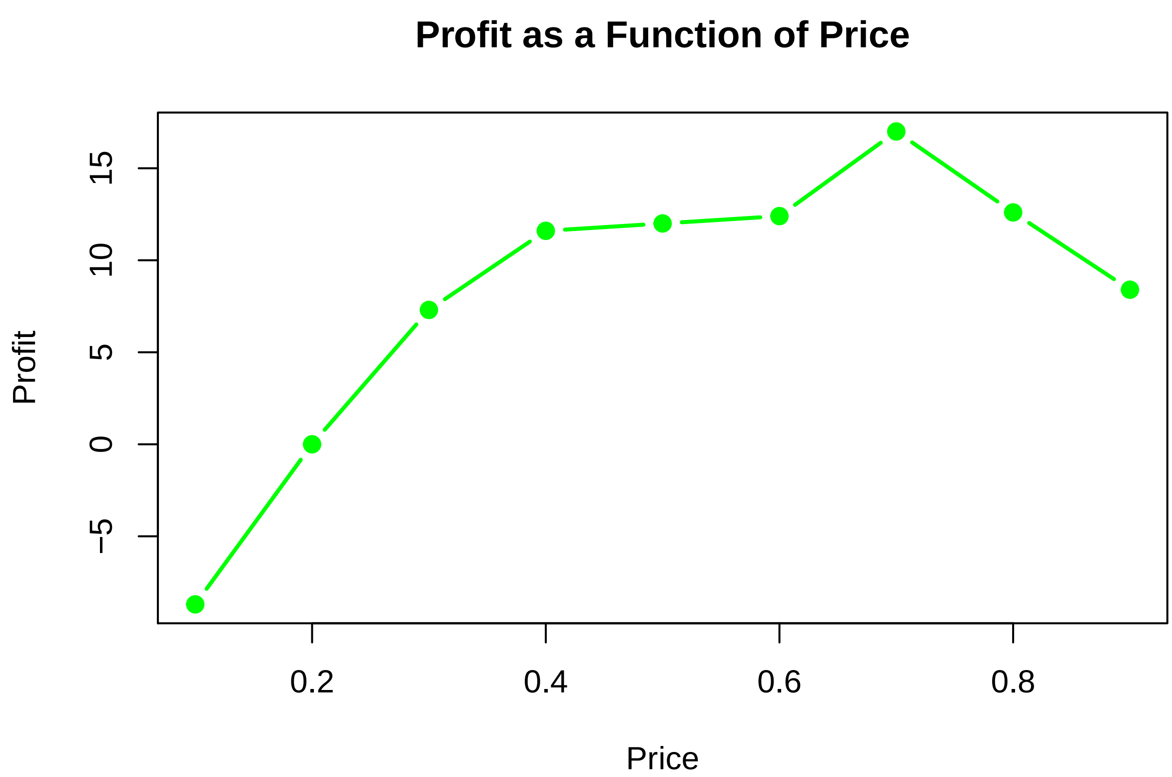
Profit Function

```
# Number of buyers per day
daily_buyers <- 100
# Marginal cost
c <- 0.2
# Simulate profits at different prices
prices <- seq(0.1, 0.9, by = 0.1)
profits <- sapply(prices, function(p) {
  quantity <- sum(runif(daily_buyers) >= p)
  revenue <- p * quantity
```

```
  cost <- c * quantity
  revenue - cost
})
```

Plot

```
# Plot profits against prices
plot(prices, profits, type = "b", col = "green", pch = 19, lwd = 2,
     xlab = "Price", ylab = "Profit", main = "Profit as a Function of Price")
```




```
# Comment: The profit curve peaks at an optimal price, where the balance between price and quantity sol
```

Exercise 3: Duopoly and Demand Estimation

Background

We estimate the Cournot duopoly equilibrium using a linear demand function.

Simulation

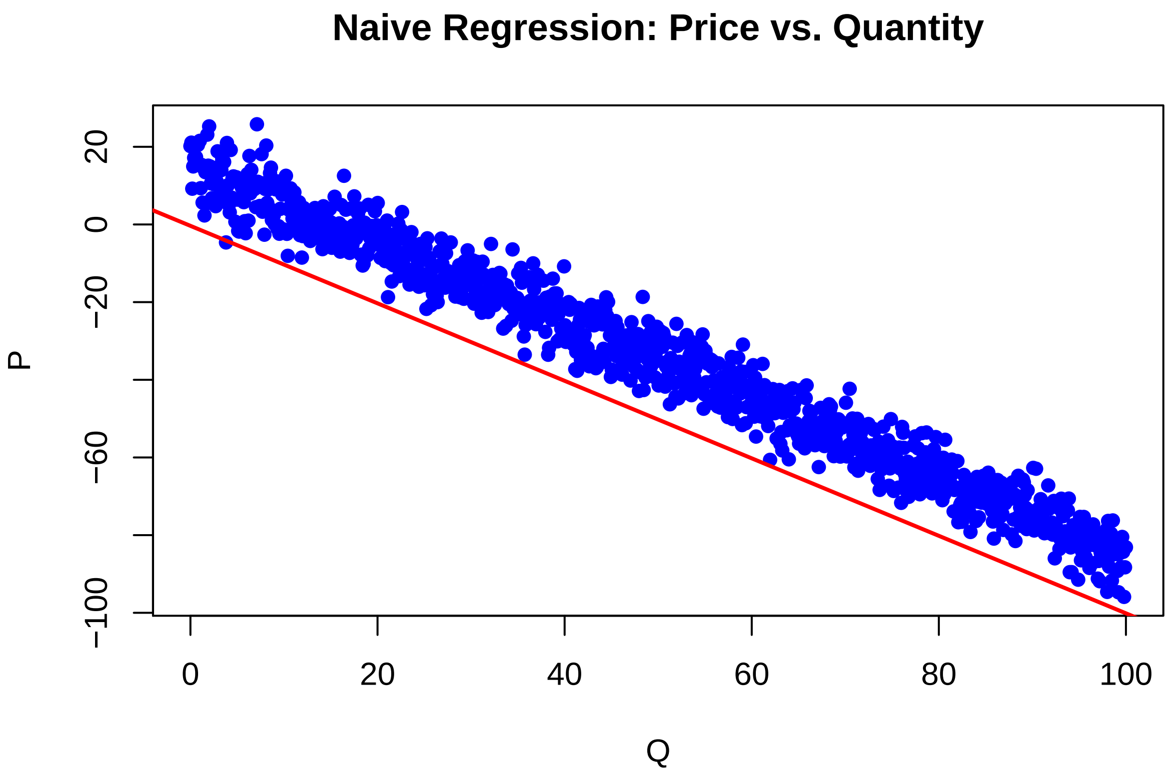
Simulated Observations and Regression

```r
# Simulation parameters
nsim <- 1000
beta_x <- 1.5
beta_q <- -1
beta_g <- 0.6

# Simulate variables
X <- rnorm(nsim, mean = 10, sd = 2)
w <- rnorm(nsim, mean = 2, sd = 1)
epsilon_D <- rnorm(nsim, mean = 0, sd = 4)
epsilon_S <- rnorm(nsim, mean = 0, sd = 1)

# Demand function: P = beta_x * X + beta_q * Q + epsilon_D
Q <- seq(0, 100, length.out = nsim)
P <- beta_x * X + beta_q * Q + epsilon_D

# Naive regression
model_naive <- lm(P ~ Q + X + w)  # Simple regression without AER
summary(model_naive)
```

```
##
## Call:
## lm(formula = P ~ Q + X + w)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.0216  -2.5842  -0.0549   2.6470  13.8669
##
## Coefficients:
##              Estimate Std. Error  t value Pr(>|t|)
## (Intercept) -0.355460   0.696691   -0.510    0.610
## Q           -0.997805   0.004308 -231.612   <2e-16 ***
## X            1.538885   0.059627   25.809   <2e-16 ***
## w           -0.023900   0.127968   -0.187    0.852
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.933 on 996 degrees of freedom
## Multiple R-squared:  0.9821, Adjusted R-squared:  0.982
## F-statistic: 1.818e+04 on 3 and 996 DF,  p-value: < 2.2e-16
```

```r
# Plot the results
plot(Q, P, pch = 16, col = "blue", main = "Naive Regression: Price vs. Quantity")
abline(model_naive, col = "red", lwd = 2)
```

```
## Warning in abline(model_naive, col = "red", lwd = 2): only using the first two
## of 4 regression coefficients
```


Naive Regression: Price vs. Quantity
P
20
0
−20
−60
−100
0
20
40
60
80
100
Q