Prof. Dr. S. Kranz | Summer term 2018
M. Kies | 2018-04-17
Sheet 2

Exercises in Empirical Industrial Organization and Consumer Choice

(Presentation: Tuesday, 2018-05-08T14:15 [2:15 p.m.], He18 R120)

All R Code presented here can be downloaded via moodle for a better Copy & Paste Experience. It is recommended, that you to solve these exercises, so that encountered problems can be discussed in class. The solution is given by a separate document, downloadable via moodle.

Exercise 2 - A different model

After having sold her ice cream truck, Emma decides to go into selling paper-clips. Ice cream in our model has the property, that, given a certain amount of people, weather and other factors, there is a certain maximal demand for it. This is modelled via the parameter a_t, which gives us an upper bound of demand, even if the price is set to 0. On the other hand everybody loves paper-clips and you can't have enough of them (or so Emma assumes). Consequently she decides to use a different model than before:
Assume the sales in period t are given by the following demand function

$$q_t = A \cdot p_t^{\alpha} \cdot \exp(\varepsilon_t) \text{ with } \varepsilon_t \overset{\text{iid}}{\sim} N(0, \sigma_\varepsilon^2),$$

i.e. ε_t being independent, identically and normally distributed with mean 0 and standard deviation σ_ε. To keep things simple and since the paper-clip resources market is rather competitive, we assume

$$c_t \overset{\text{iid}}{\sim} N(\mu, \sigma_c^2)$$

to be the costs Emma faces per paper-clip. To keep it realistic, the parameters are assumed to be sensible, i.e.

$$A > 0,\ \alpha < -1,\ \sigma_\varepsilon > 0 \ \&\ \sigma_c > 0 \text{ but sufficiently small}, \mu \gg \sigma_c \text{ with } \mathrm{P}(c_t \leq 0) \approx 0.$$

(a) Which price should Emma chose, assuming she is aware of all parameters?

Solution:

We assume, that Emma wants to maximize her profit:

$$\pi_t = p_t \cdot q_t - c_t \cdot q_t = q_t \cdot (p_t - c_t) = A \cdot e^{\varepsilon_t} \cdot \left(p_t^{\alpha+1} - p_t^{\alpha} c_t\right)$$

To maximize this profit, we have to calculate the first order condition:

$$\begin{aligned}
\frac{\partial \pi_t}{\partial p_t} &= A \cdot e^{\varepsilon_t} \cdot \left[(\alpha+1)\, p_t^{\alpha} - \alpha p_t^{\alpha-1} c_t\right] \overset{!}{=} 0 \\
&\Leftrightarrow (\alpha+1) p_t^{\alpha} \overset{!}{=} \alpha p_t^{\alpha-1} c_t \\
&\Leftrightarrow \quad \frac{\alpha+1}{\alpha} \overset{!}{=} \frac{p_t^{\alpha-1} c_t}{p_t^{\alpha}} = \frac{c_t}{p_t} \\
&\quad \Rightarrow\ p_t^* = \frac{\alpha}{\alpha+1} \cdot c_t
\end{aligned}$$

To make sure we found a maximum, we have a look at the second order condition:

$$\left.\frac{\partial^2 \pi_t}{\partial^2 p_t}\right|_{p_t = p_t^*} = \underbrace{A \cdot e^{\varepsilon_t}}_{>0} \cdot \left[(\alpha+1)\, \alpha p_t^{*\alpha-1} - \alpha\,(\alpha-1)\, p_t^{*\alpha-2} c_t\right] \overset{?}{<} 0$$

It holds that

$$\begin{aligned}
&\left[(\alpha+1)\, \alpha p_t^{*\alpha-1} - \alpha\,(\alpha-1)\, p_t^{*\alpha-2} c_t\right] \\
&= (\alpha+1)\alpha \left(\frac{\alpha}{\alpha+1} c_t\right)^{\alpha-1} - \cancel{\alpha}(\alpha-1)\left(\frac{\alpha}{\alpha+1} c_t\right)^{\alpha-1} \cdot \frac{\alpha+1}{\cancel{\alpha}\cancel{c_t}} \cdot \cancel{c_t} \\
&= (\alpha+1)\left(\frac{\alpha}{\alpha+1} c_t\right)^{\alpha-1} [\alpha - (\alpha-1)] \\
&= \underbrace{(\alpha+1)}_{<\mathbf{0}} \underbrace{\left(\frac{\alpha}{\alpha+1} c_t\right)^{\alpha-1}}_{>0} < 0
\end{aligned}$$

Therefore we can be sure that p_t^* is actually a maximizating price.

(b) Why do we require $\alpha < -1$ and $\mu \gg \sigma_c$ with $\mathrm{P}(c_t \leq 0) \approx 0.$?

Solution:

Those requirements are necessary for the second order condition of p_t^*. They guarantee that it is a maximum. Additionally they allow us to divide by p_t^{α} and by $\alpha + 1$ when calculating p_t^*.

(c) Simulate the model in R by generating q_t and π_t (Emmas profit) and plot profits $[\pi_t]$ and output $[q_t]$, respectively, in regard to the prices $[p_t]$ by setting the prices

i. randomly within a reasonable range.

ii. in a profit maximizing fashion.

Hint: Nice parameters are $T = 1000$, $A = 1000$, $\mu = 5$, $\sigma_c = 0.05$, $\sigma_\varepsilon = 0.15$, $\alpha = -2$.

Solution:

```r
library(restorepoint) # For better bugfixing

##### Exercise 1
## c)

#Setting up basic parameters
T <- 1000
A <- 1000
mu <- 5
sigma_c <- 0.05
sigma_eps <- 0.15
c_t <- rnorm(T,mean=mu,sd=sigma_c)
alpha <- -2
eps_t <- rnorm(T,0,sigma_eps)

#Function which generates the prices
## option = "optimal" generates optimal prices [necessary parameters: alpha, c_t]
## option = "random" generates uniformly distributed random prices [necessary parameters:
    a, b, T]
generate.p_t <- function(alpha.=alpha, c_t. = c_t, a=c_t, b=c_t+1/sigma_c*2, T. = T,
    option="optimal"){
  restore.point("generate.p_t")
  if(option=="optimal"){
    p_t <- (alpha.*c_t.)/(alpha.+1)
  } else if (option=="random"){
    p_t <- runif(T.,a,b)
  } else {
    stop("Wrong argument")
  }
  return(p_t)
}

#Function which generates demand based on the non-linear demand function given in the
    Exercise
generate.q_t <- function(A.=A, p_t=generate.p_t(), alpha.=alpha, eps_t.=eps_t){
  q_t <- A.*p_t^(alpha.)*exp(eps_t.)
  return(q_t)
}

#Function which generates the profit
## Note that given no parameters the profit is identical to the option given as a default
    in generate.p_t (i.e. optimal)
generate.pi <- function(q_t=generate.q_t(),p_t=generate.p_t(), c_t. = c_t){
  pi <- q_t*p_t - c_t.*q_t
  return(pi)
}

### i.

```

```
p_t.random <- generate.p_t(option="random")
q_t.random <- generate.q_t(p_t=p_t.random)
pi.random <- generate.pi(p_t=p_t.random,q_t = q_t.random)

plot(p_t.random, pi.random)
plot(p_t.random, q_t.random)

### ii.

p_t.optimal <- generate.p_t(option="optimal")
q_t.optimal <- generate.q_t(p_t=p_t.optimal)
pi.optimal <- generate.pi(p_t=p_t.optimal,q_t = q_t.optimal)

plot(p_t.optimal, pi.optimal)
plot(p_t.optimal, q_t.optimal)
```

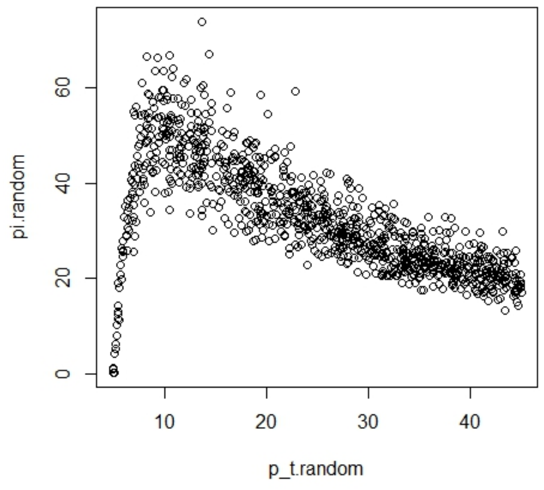

Figure 1: π_t given random pricing p_t.

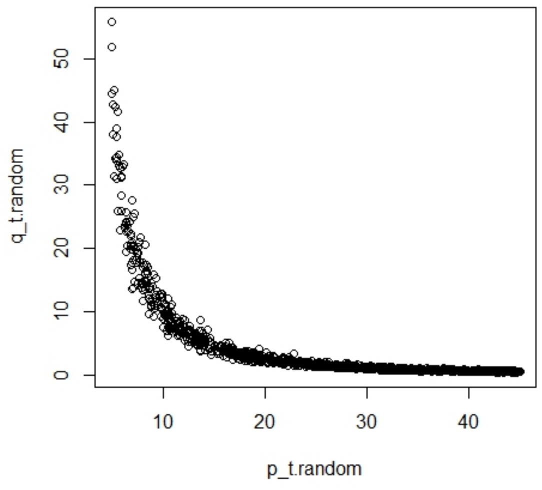

Figure 2: q_t given random pricing p_t

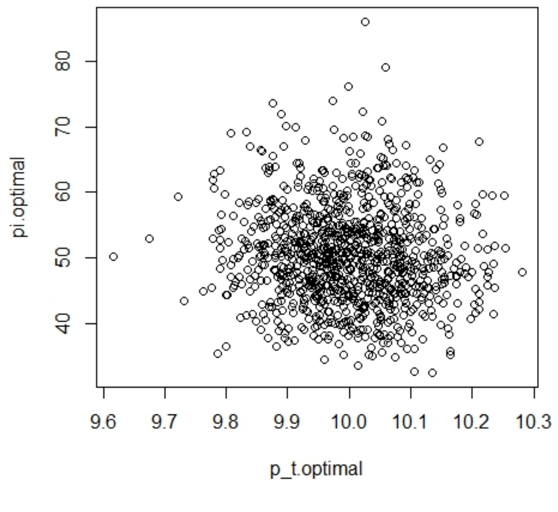

Figure 3: π_t given optimal pricing p_t.

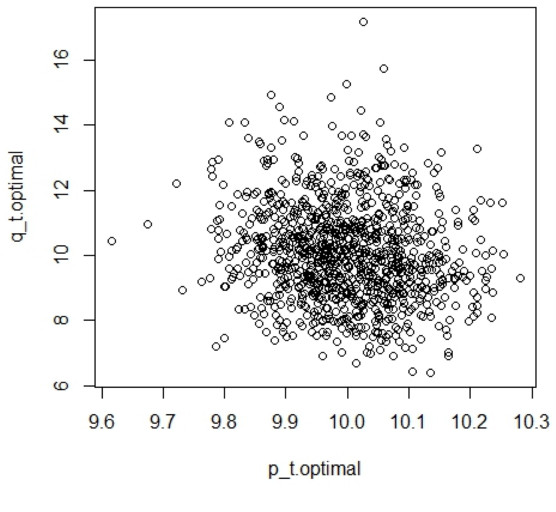

Figure 4: q_t given optimal pricing p_t

(d) How could you transform the data, so that linear regression can be used to estimate the parameters? Which of the two data sets from above is probably the better choice for an implementation? Write a program in *R* which estimates the parameters with linear regression.

Solution:

By design we can assume q_t to be positive, which means we may draw the logarithm of our model:

$$\begin{aligned}\ln(q_t) &= \ln(A \cdot p_t^\alpha \cdot \exp(\varepsilon_t)) \\ &= \ln(A) + \ln(p_t^\alpha) + \ln\big(\exp(\varepsilon_t)\big) \\ &= \ln(A) + \alpha\ln(p_t) + \varepsilon_t\end{aligned}$$

which is exactly of the form needed for the OLS. α may be estimated directly, but A will be exp(Intercept) of our OLS.
We use the random prices as basis for the OLS:

```
##d)

# Calculate regression based on the logarithmic values of q_t and p
   _t
lm <- lm(log(q_t.random)~log(p_t.random))

# Extract regressors
A.emp <- exp(coef(lm)[1])
alpha.emp <- coef(lm)[2]

# Build nice table for cleaner display
ex1table <- data.frame(cbind(c(A,alpha),c(A.emp,alpha.emp)),row.
   names=c("A","alpha"))
colnames(ex1table) <- c("True Value", "Empirical Value")
ex1table
```

Why do we use the random prices?

You may note, that (in contrast to the optimal prices of the linear model), our optimal p_t^* with

$$p_t^* = \frac{\alpha}{\alpha+1} \cdot c_t$$

does not depend on ε_t. But using the optimal prices results in false estimates:

```
> lm <- lm(log(q_t.optimal)~log(p_t.optimal))
> A.emp.opt <- exp(coef(lm)[1])
> alpha.emp.opt <- coef(lm)[2]
> ex1table[,"Empirical Value with opt prices"] <- c(A.emp.opt,alpha
   .emp.opt)
```

```
> ex1table
      True Value Empirical Value Empirical Value with opt prices
A           1000      957.866675                      385.625098
alpha         -2       -1.985623                       -1.585638
```

We have a problem with endogenity iff

$$\mathrm{cor}(\varepsilon_t, p_t^*) \neq 0$$

Empirically we see, that

```
> cor(p_t.random,eps_t)
[1] 0.05663789
> cor(p_t.optimal,eps_t)
[1] 0.02749994
```

or in other words, that endogenity is not the problem. This makes theoretically sense as well, as our c_t did not depend on ε_t in any way. But why is it still better to use random prices?
Lets have a look at the summery of p_t:

```
> summary(p_t.random)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  4.863  14.970  25.030  25.020  35.040  45.100
> summary(p_t.optimal)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  9.542   9.933  10.000  10.000  10.070  10.430
```

You see, that in this case the random prices have substantially higher variance. The OLS regression is better with more data points, especially, if the data points are very *different* from each other. In this case using the optimal prices would indeed provide the correct result, given infinitely many data points. Even the 1000 data points from the example are not enough, to ensure a high degree of confidence into in using the optimal prices.

For more information how the OLS can be let astray see the Bonus Exercise.

Bonus - Exercise - Be thorough in your data examination

This exercise will probably not be topic of the presentation and is not relevant for the exam. You are encouraged to solve it though for your own benefit.
Let's have a look at this rather complicated function which is executed on the data set *anscombe*, provided by *R*:

```
somestatistics <- function(data){
  #Definition
```

```r
  ncols <- ncol(data)
  nrows <- 2

  #Build up Matrix
  res1 <- array(rep(NA,nrows*ncols),dim=c(nrows,ncols))
  res1 <- as.data.frame(res1, row.names=c("mean", "variance"))
  colnames(res1) <- colnames(data)

  #Fill Matrix
  res1["mean",] <- format(apply(data,MARGIN=2,mean),digits=4)
  res1["variance",] <- format(apply(data,MARGIN=2,var),digits=4)

  res2 <- list()

  for (i in 1:(ncols/2)){
    lm <- lm(data[,ncols/2+i] ~ data[,i])
    res2[[i]] <- data.frame(
      Covariance=cov(data[,i],data[,ncols/2+i]),
      Intercept=coef(lm)[1],
      Regressor=coef(lm)[2])
    rownames(res2[[i]]) <- NULL
  }

  res <- list(vectorspecific=res1,combination=res2)

  return(res)
}
```

It expects a data frame whose columns have the logic $(x_1, ..., x_n, y_1, ..., y_n)$ with x_i and y_i defining a two-dimensional dataset (e.g. "sold quantities given prices" or "number of passengers given train miles"). Consequently the variable n is the number of our datasets; four in the case of *anscombe.*

(a) Try to understand the function:

i. What does it do?
(Hint: When you dissect a function it is helpful to work with restorepoint*s.)*

Solution:

somestatistics generates a list with two elements from the data set:

A. The first element is a data.frame, where for each column of the data set the mean and the variance of those columns have been calculated.

B. The second element is a list which holds for each of the combinations (x_i, y_i) the information about the covariance and the Intercept and the Regressor of the linear regression $y_i = \text{Intercept} + \text{Regressor} \cdot x_i$.

ii. What could be the intend behind certain constructs?

Solution:

There are several constructs in this code, which might be confusing:

A. The code

```
        format(apply(data,MARGIN=2,mean),digits=4)
```

can be split into two parts:

- *format* is a function, which allows for nicer formatting without problems regarding rounding.
- *apply* is one of the most powerful tools of R. It can be compared to a *for* loop. In this case we loop through all columns and execute a function (here: *mean*) on each function. *apply* has the advantage that it is severely faster than a for-loop due to optimizers behind the curtains.

B. The argument of the *for* loop

```
        i in 1:(ncols/2)
```

is used based on the structure of the data. It is used to access the pairs (x_i, y_i).

(b) Interpret the output of the function. What does it tell you about the data sets?

Solution:

Based on these statistics it seems, that they are fairly regular datasets. One could assume, that each of them looks the following way:

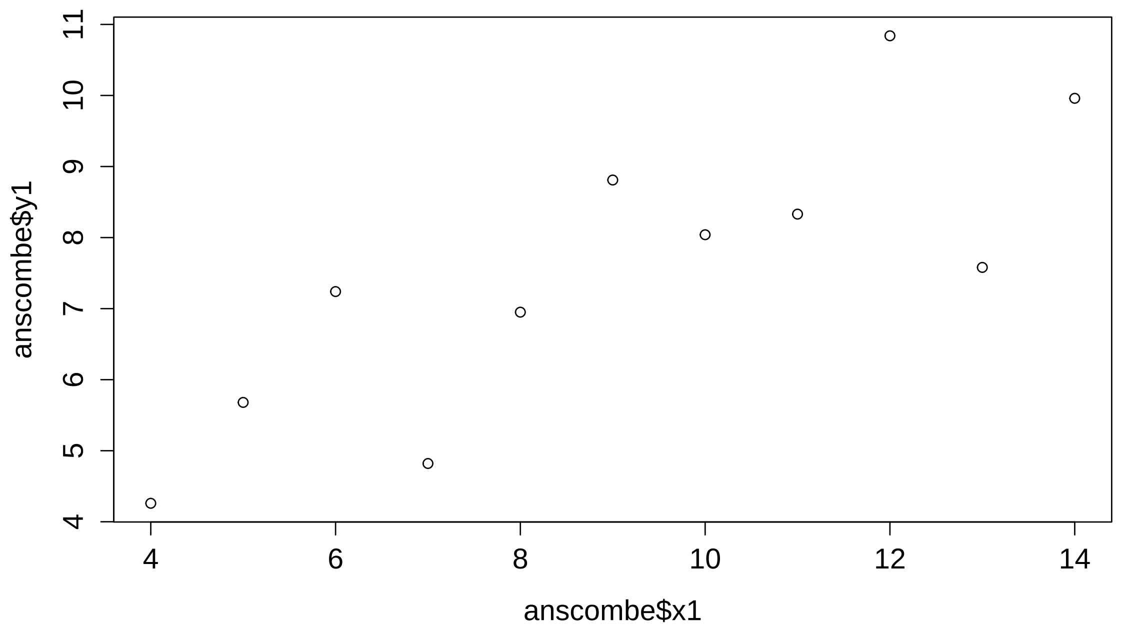

Figure 5: Plot of the first anscombe set.

(c) Would you say the four data sets are similar?

Solution:

The datasets are similar in so far as the calculated measures reflect, but not necessarily in another way. In the next task it can be seen, that they are quite different.

(d) Plot each of the data sets.

Solution:

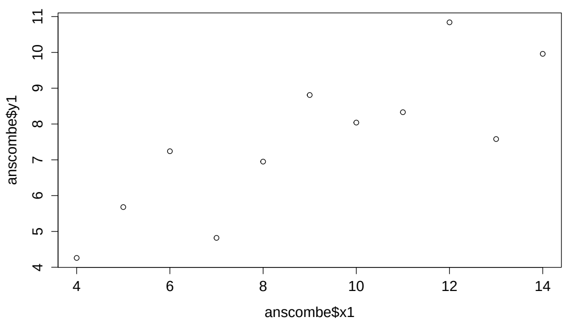

Figure 6: Plot of the first anscombe set.

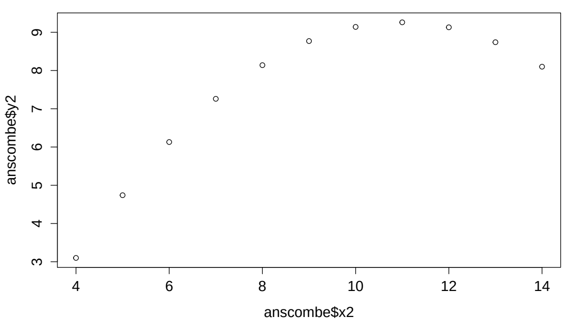

Figure 7: Plot of the second anscombe set.

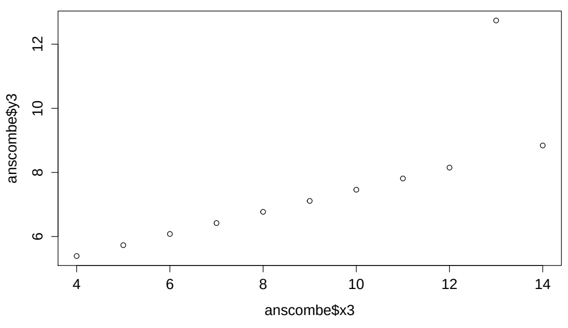

Figure 8: Plot of the third anscombe set.

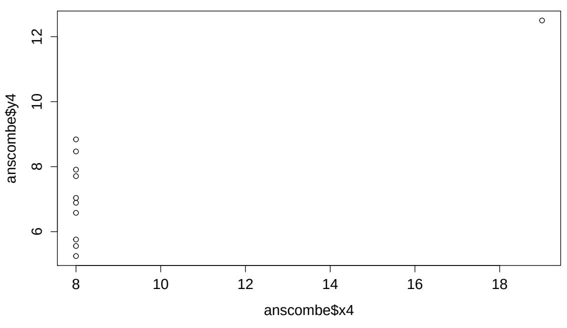

Figure 9: Plot of the fourth anscombe set.