Advanced Marketing Modeling

Prof. Dr. Daniel Klapper

Humboldt University Berlin
Institute of Marketing

SS 2018

Outline

Outline

A Simple Marketing System

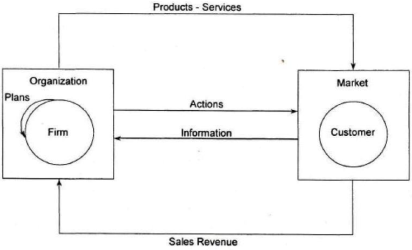
$$A_t = f(P_{t-1}, Q_{t-1})$$

where

$A_t =$ Firm's advertising expenditures at time t

$P_{t-1} =$ Price of the product at time $t-1$

$Q_{t-1} =$ Firm's sales in units at time $t-1$



Source: Parsons and Schultz (1976, p. 4)

Verbal and Mathematical Model

New-product growth often starts slowly, until some people (early triers) become aware of the product. These early triers interact with nontriers to lead to acceleration of sales growth. Finally, as market potential is approached, growth slows down.

Mathematical model

$$\frac{\partial x}{\partial t} = (a + bx)(N - x)$$

where

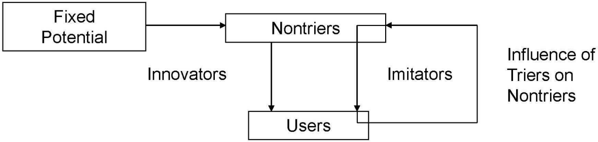
$x =$ Number of purchases by t

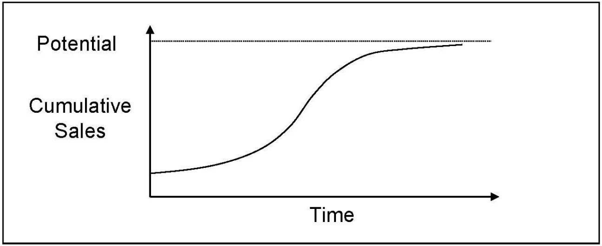
$N =$ Market potential

$a, b =$ Constants

Source: Lilien, Kotler and Moorthy 1992, p. 7

Graphical and Conceptual Model





Source: Lilien, Kotler and Moorthy 1992, p. 7

Marketing Data

Classification of marketing data:

- Primary data
  - collected by a researcher for the sole purpose of addressing a specific problem at hand
- Secondary data
  - data that were gathered for purposes other than the problem at hand
  - easily accessible
  - relatively inexpensive
  - quickly obtained

Source: Leeflang, Wieringa, Bijmolt and Pauwels 2015, p.71-83

Marketing Data

Secondary data:

- Internal
  - generated within the organization
  - e.g. from a CRM system
- External
  - generated by sources outside the organization
  - e.g. a country's national bureau of statistics
- Big data
- Subjective data

Source: Leeflang, Wieringa, Bijmolt and Pauwels 2015, p.71-83

Marketing Data

More on internal data (1):

Most companies gather (a subset of) the following types of internal variables

- Information on key performance indicators
  - Sales (revenue/units)
  - Recency (time since most recent visit/purchase)
  - Frequency (number of visits/purchases)
  - Loyalty/satisfaction measures (e.g. NPS scores)
  - Conversion rates
  - Cross-selling rates
  - Customer Lifetime Value (CLV)

Source: Leeflang, Wieringa, Bijmolt and Pauwels 2015, p.71-83

Marketing Data

More on internal data (2):

- Information about possible explanatory variables
  - Number and/or quality of inbound and outbound contracts
  - Prices paid
  - Discounts given
  - Promotions received
  - Use of loyalty cards
  - Mailings send, etc.
- Information about customer characteristics
  - Demographics
  - Interests (lifestyle data)
  - Activities (memberships etc.)

Source: Leeflang, Wieringa, Bijmolt and Pauwels 2015, p.71-83

Marketing Data

More on external data:

- Store (Retail) level
- Manufacturer level
- Scanner data
  - Volumes (at the SKU-level)
  - Revenues
  - Actual prices
  - ACV = the all commodity volume of the store or revenue
  - Baseline sales
- HandScan Panel data

Source: Leeflang, Wieringa, Bijmolt and Pauwels 2015, p.71-83

Marketing Data

More on Big Data:

- Increasing data volume
- Increasing data velocity
- Increasing data variety

INFORMS Marketing Science on Big Data:
"High volume implies the need for models that are scalable, high velocity open opportunity for real-time, or virtually real-time, making decision making that may or may not be automated; and high variety may require integration across disciplines with the corresponding sensitivity to various methods and philosophies of research."

Source: Leeflang, Wieringa, Bijmolt and Pauwels 2015, p.71-83

Outline

Stages in the Model-Building Process

1. Opportunity identification
2. Model purpose
3. Model scope
4. Data availability
5. Model-building criteria: simple, complete on important issues, adaptive, robust
6. Model specification
7. Parameterization
8. Validation
9. Cost-benefit consideration
10. Use
11. Updating

Source: Leeflang, Wittink, Wedel and Naert 2000, p. 52

Specification of the Model

1. Which variables enter the model?
2. What are the effects of the independent variables on the dependent variable?
3. Do dynamic effects exist?
4. Are there any interaction effects among independent variables?
5. What are the characteristics of the error term?

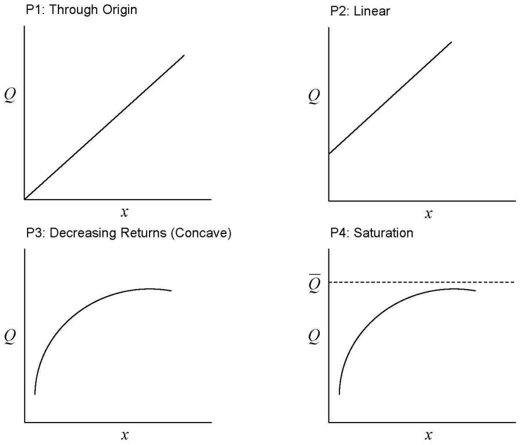
Information for model specification:

- Marketing manager
- Marketing theory and economic theory
- Statistical procedures

Dimensions of Model Development

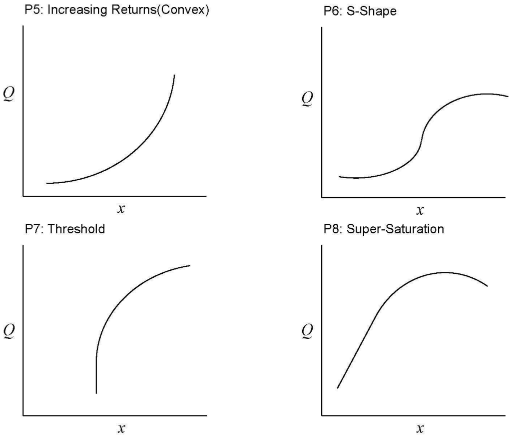
Dimension	Examples
1. Mathematical Form	
Linear in parameters and variables	$Q = a_0 + a_1 X$
Nonlinear in variables, linear in parameters	$Q = a_0 + a_1 X + a_2 X^2$
Nonlinear in parameters, linearizable	$Q = a_0 X_1^{a_1} X_2^{a_2}$
Inherently nonlinear	$Q = a_0(1 - e^{-a_1 x})$
2. Dynamic Effects	
Discrete time	$Q_1 = a_0 + a_1 X_t + \lambda Q_{t-1}$
Continuous time	$\frac{\partial Q}{\partial t} = \frac{rX(V-Q)}{V} - \lambda Q$
3. Uncertainty	
Deterministic	$Q = a_0 + a_1 X$
Deterministic with stochastic error	$Q = a_0 + a_1 X + \varepsilon$
Inherently stochastic	$p = f(\text{past purchase behavior})$
4. Level of Aggregation	
Individual	$p = f(\text{past behavior, marketing variables})$
Segment of market	$Q_i = a_0 + a_i X$
5. Level of Demand	
Product class	$V = f(\text{demographic trends, total marketing spending})$
Brand sales	$Q = SV$
Market share	$S = \frac{\text{us}}{\text{us+them}}$

Shape of Response Functions



Source: Lilien, Kotler and Moorthy 1992, p. 654

Shape of Response Functions



Source: Lilien, Kotler and Moorthy 1992, p. 654

Response Models with Diminishing Returns to Scale

Semilogarithmic model

$$Q = a_0 + a_1 \ln X$$

with positive sales from $e^{-\frac{a_0}{a_1}}$

Marketing elasticity:

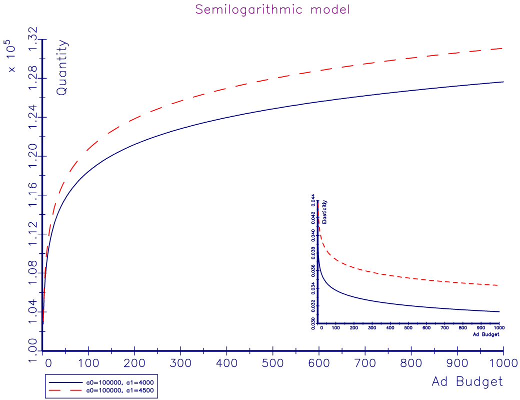
$$\varepsilon_X = \frac{\partial Q}{\partial X} \cdot \frac{X}{Q} = \frac{a_1}{Q}$$

$\Rightarrow$ Elasticity decreases with increasing sales

Source: Hanssens, Parsons and Schultz 2001, p. 100-101

Response Models with Diminishing Returns to Scale

Semilogarithmic model



Response Models with Diminishing Returns to Scale

Constant elasticity model

$$Q = a_0 X^{a_1} \text{ respectively } Q = e^{\beta_0} \cdot X^{\beta_1}$$

where $0 < a_1 < 1$ and $0 < \beta_1 < 1$.

Linear in parameters by taking the logarithms:

$$\ln Q = \ln a_0 + a_1 \ln X \qquad \text{and} \qquad \ln Q = \ln \beta_0 + \beta_1 \ln X$$

Computing the elasticity:

$$\frac{\partial Q}{\partial X} = a_0 a_1 X^{a_1 - 1} = \frac{a_1 Q}{X}$$

$$\varepsilon_X = \frac{\partial Q}{\partial X} \cdot \frac{X}{Q} = \frac{a_1 Q}{X} \cdot \frac{X}{Q} = a_1$$

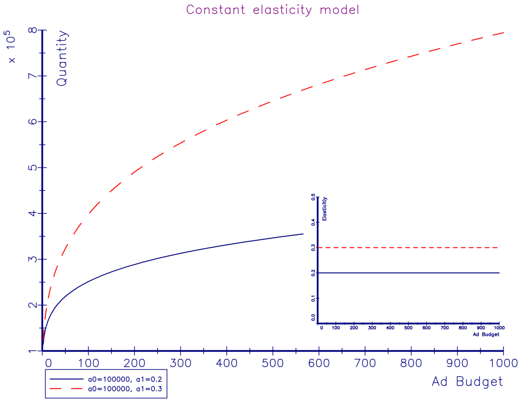
Multiplicative model:

$$Q = e^{\beta_0} X_1^{\beta_1} X_2^{\beta_2} \ldots X_k^{\beta_k}$$

Source: Hanssens, Parsons and Schultz 2001, p. 101-102

Response Models with Diminishing Returns to Scale

Constant elasticity model



Response Models with Increasing Returns to Scale

Exponential model

$$Q = e^{\beta_0} e^{\beta_1 X}$$

Price would typically be represented as $\frac{1}{p}$

Alternatively

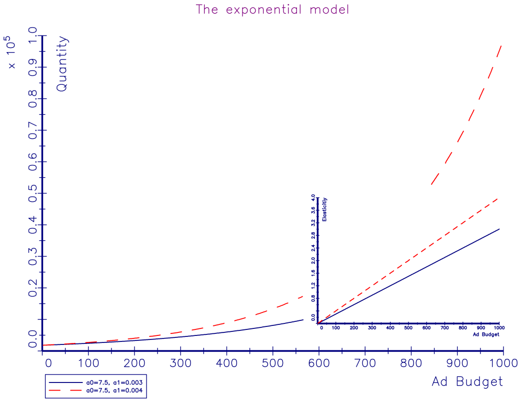
$$\begin{aligned} Q &= e^{\beta_0} e^{-\beta_1 p} = Q_0 e^{-\beta_1 p} \\ Q_0 &= \text{Saturation sales} \end{aligned}$$

$\Rightarrow$ Price and promotion model by Blattberg and Wisniewski

Source: Hanssens, Parsons and Schultz 2001, p. 105-106

Response Models with Increasing Returns to Scale

Exponential model



Response Models with an S-Shaped Function

Log-reciprocal function

$$Q = e^{\beta_0 - \frac{\beta_1}{X}} \qquad \text{or} \qquad \ln Q = \beta_0 - \frac{\beta_1}{X} \qquad \beta_0, \beta_1 > 0$$

The function is progressively increasing with $X < \frac{\beta_1}{2}$

The function has a maximum at e^{β_0}

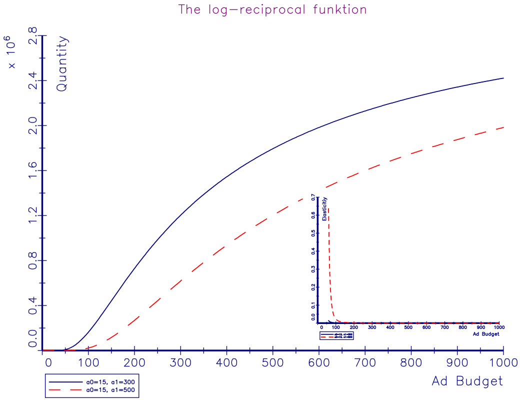
Marketing elasticity:

$$\varepsilon_X = \frac{\partial Q}{\partial X} \cdot \frac{X}{Q} = \frac{\beta_1}{X}$$

Source: Hanssens, Parsons and Schultz 2001, p. 106-109

Response Models with an S-Shaped Function

Log-reciprocal function



Response Models with an S-Shaped Function

Logistic model

$$Q = \frac{Q_{max}}{1 + \exp[-(\beta_0 + \sum_{j=1}^{J} \beta_j X_j)]}$$

$$\ln\left(\frac{Q}{Q_{max} - Q}\right) = \beta_0 + \sum_{j=1}^{J} \beta_j X_j$$

Inflection point: $Q = \frac{Q_{max}}{2}$

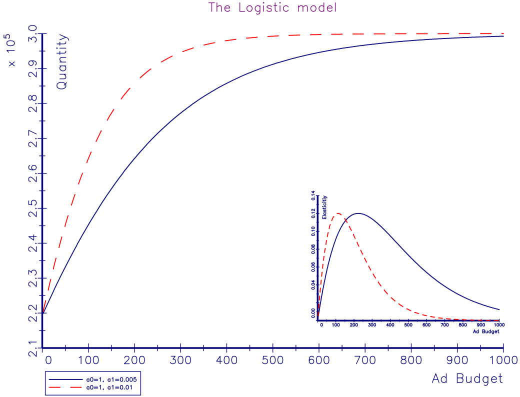
Marketing elasticity:

$$\varepsilon_X = \frac{\partial Q}{\partial X} \cdot \frac{X}{Q} = \beta_j X_j \frac{Q_{max} - Q}{Q_{max}}$$

Source: Hanssens, Parsons and Schultz 2001, p. 106-109

Response Models with an S-Shaped Function

Logistic model



Response Models with Saturation

Log-reciprocal function

$$Q = \exp(\beta_0 - \frac{\beta_1}{X}) \qquad \beta_0, \beta_1 > 0$$

$$X \to \infty, \qquad Q \to \exp(\beta_0) = Q_{max}$$

Modified exponential model

$$Q = e^{\beta_0}(1 - e^{-\beta_1 X}) = Q_{max}(1 - e^{-\beta_1 X})$$

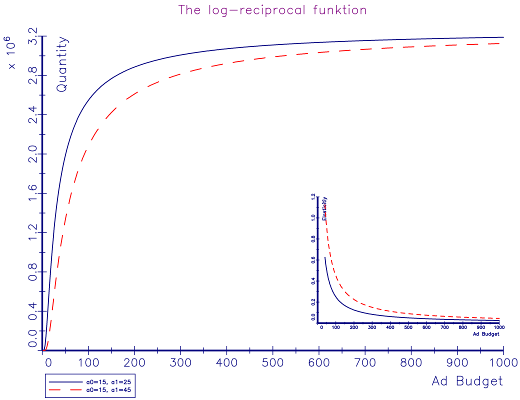
Marketing elasticity:

$$\varepsilon_X = \frac{\partial Q}{\partial X} \cdot \frac{X}{Q} = \frac{Q_{max} - Q}{Q} a_1 X$$

$\Rightarrow$ Identical to the logistic function if $Q_{min} = 0$

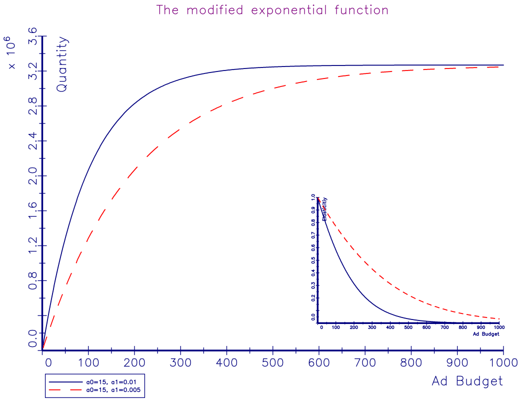
Response Models with Saturation

Log-reciprocal function



Response Models with Saturation

Modified exponential function



Response Models with Saturation

ADBUDG function

$$Q = \beta_0 + (\beta_1 - \beta_0)\frac{X^{\beta_2}}{\beta_3^{\beta_2} + X^{\beta_2}}$$

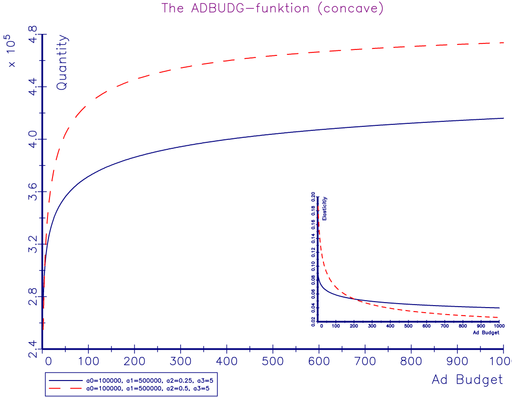
where

$$\begin{aligned}
\beta_0 &= \text{Intercept} \\
\beta_1 &= \text{Saturation level} \\
\beta_2, \beta_3 &= \text{Shape parameters}
\end{aligned}$$

Source: Hanssens, Parsons and Schultz 2001, p. 111

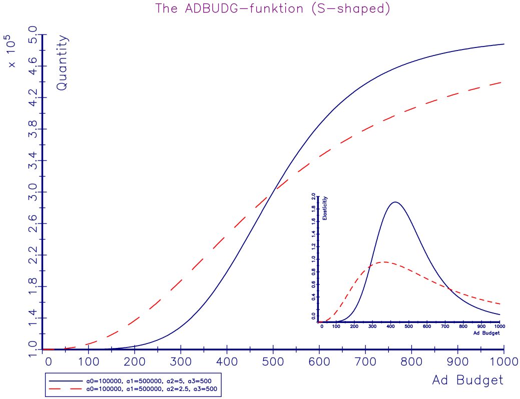
Response Models with Saturation

ADBUDG function (concave)



Response Models with Saturation

ADBUDG function (S-shaped)



Response Models with Supersaturation

Quadratic model

$$Q = \beta_0 + \beta_1 X + \beta_2 X^2$$

where $\beta_1 > 0, \beta_2 < 0, |\beta_2| < \beta_1$

Marketing elasticity:

$$\varepsilon_X = \frac{\partial Q}{\partial X} \cdot \frac{X}{Q} = \frac{\beta_1 X + 2\beta_2 X^2}{Q}$$

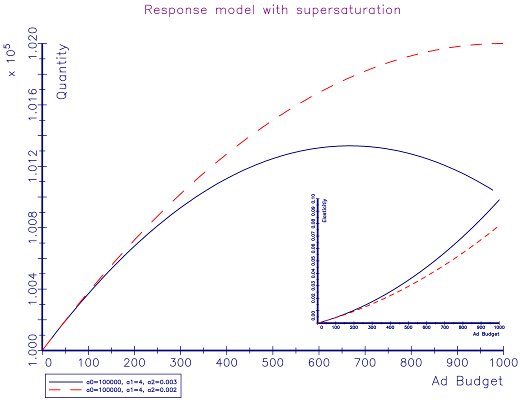
$$\varepsilon_X > 0 \qquad \text{for} X > -\frac{\beta_1}{2\beta_2}$$

Supersaturation at $X > -\frac{\beta_1}{2\beta_2}$

Source: Hanssens, Parsons and Schultz 2001, p. 112

Response Models with Supersaturation

Response models with supersaturation



Market Share Models

$$s_i = \frac{A_i}{\sum_{j=1}^{m} A_j}$$

MCI model:

$$A_i = \exp(\alpha_i) \prod_{k=1}^{K} X_{ki}^{\beta_k} \varepsilon_i$$

MNL model:

$$A_i = \exp(\alpha_i + \sum_{k=1}^{K} \beta_k X_{ki} + \varepsilon_i)$$

Market Share Models

where

$s_i =$ market share of brand i

$A_i =$ attraction of brand i

$m =$ number of brands

$X_{ki} =$ value of the k-th explanatory variable X_k for brand i
(e.g. prices, product attributes, expenditures for advertising, distribution, sales force)

$K =$ number of explanatory variables

$\beta_k =$ parameter to be estimated

$\alpha_i =$ parameter for the constant influence of brand i

$\varepsilon_i =$ error term

Market Share Models

Log-centering transformation applied to the MCI-Model

$$\log s_i = \alpha_i + \sum_{k=1}^{K} \beta_k \log X_{ki} + log\, \varepsilon_i - \log\Big\{ \sum_{k=1}^{m} \Big(\alpha_j \prod_{k=1}^{K} X_{kj}^{\beta_k} \varepsilon_j \Big) \Big\}$$

$$\log \tilde{s} = \tilde{\alpha} + \sum_{k=1}^{K} \beta_k \log \tilde{X}_k + \log \tilde{\varepsilon} - \log\Big\{ \sum_{k=1}^{m} \Big(\alpha_j \prod_{k=1}^{K} X_{kj}^{\beta_k} \varepsilon_j \Big) \Big\}$$

$$\log\Big(\frac{s_i}{\tilde{s}} \Big) = {\alpha_i}^* + \sum_{k=1}^{K} \beta_k \log\Big(\frac{X_{ki}}{\tilde{X}_k} \Big) + {\varepsilon_i}^*$$

$$\alpha_i^* = (\alpha_i - \bar{\alpha})$$

$$\varepsilon_i^* = \log\Big(\frac{\varepsilon_i}{\tilde{\varepsilon}} \Big)$$

Market Share Models

Log-centering transformation applied to the MNL model

$$\log\left(\frac{s_i}{\tilde{s}}\right) = (\alpha_i - \bar{\alpha}) + \sum_{k=1}^{K} \beta_k (X_{ki} - \bar{X}_k) + (\varepsilon_i - \bar{\varepsilon})$$

Simpler regression models that yield the same β-estimates

MCI:

$$\log s_{it} = \alpha_1 + \sum_{j=2}^{m} \alpha_j{}' d_j + \sum_{u=2}^{T} \gamma_u D_u + \sum_{k=1}^{K} \beta_k \log X_{kit} + \varepsilon_{it}$$

MNL:

$$\log s_{it} = \alpha_1 + \sum_{j=2}^{m} \alpha_j{}' d_j + \sum_{u=2}^{T} \gamma_u D_u + \sum_{k=1}^{K} \beta_k X_{kit} + \varepsilon_{it}$$

Market Share Elasticities

$$e_{s_i} = \frac{\partial s_i}{\partial X_{ki}} \frac{X_{ki}}{s_i}$$

MCI:

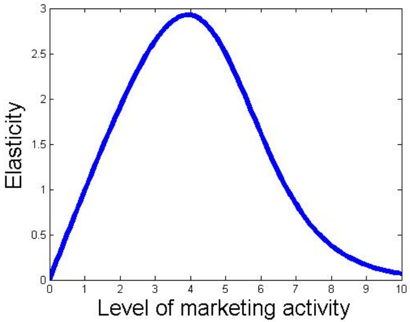
$$e_{s_i} = \beta_k (1 - s_i)$$

MNL:

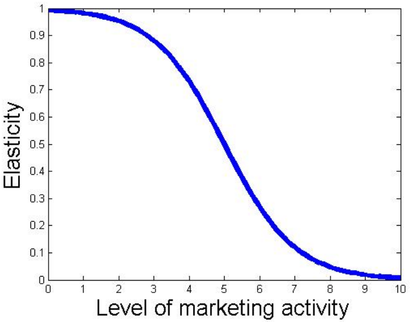
$$e_{s_i} = \beta_k (1 - s_i) X_{ki}$$

Market Share Elasticities

MCI-model



MNL-model



Differential Effectiveness

$$A_i = \exp(\alpha_i + \varepsilon_i) \prod_{k=1}^{K} f_k(X_{ki}^{\beta_{ki}})$$

$$s_i = \frac{A_i}{\sum_{j=1}^{m} A_j}$$

MCI:

$$f_k(X_{ki}) = X_{ki}$$

MNL:

$$f_k(X_{ki}) = \exp(X_{ki})$$

Market Share Elasticities

Differential direct elasticities

MCI:

$$e_{s_i} = \beta_{ki}(1 - s_i)$$

MNL:

$$e_{s_i} = \beta_{ki}(1 - s_i)X_{ki}$$

Differential cross elasticities

MCI:

$$e_{s_i j} = -\beta_{kj} s_j$$

MNL:

$$e_{s_i j} = -\beta_{kj} X_{kj} s_j$$

Pooling

⇒ combination of time series and cross-section data

	Assumptions about	
Model	Intercept	Vector of Slope Coefficients
I	Common for all i, t	Common for all i, t
II	Varying over i (or t)	Common for all i, t
III	Varying over i, t	Common for all i, t
IV	Varying over i (or t)	Varying over i (or t)
V	Varying over i, t	Varying over i, t

Source: Hanssens, Parsons and Schultz 2001, p. 120

Outline

Overview

Download R and R-Studio

https://www.r-project.org/

https://www.rstudio.com/products/rstudio/download/

Basics of the R Language

Website for the book by Chris Chapmann and Elea McDonnell Feit:
http://r-marketing.r-forge.r-project.org/index.html
Website for all data files:
http://r-marketing.r-forge.r-project.org/data.html

Objects

- We'll cover some of the basic object types in R
- Objects in R include variables, data sets, and functions

Basic Objects

The assignment operator <- assigns a value to a named object.

```
x <- c(2, 4, 6, 8)
x
```

```
[1] 2 4 6 8
```

Object names are case sensitive. Instead of `x`, `X` produces an error:

```
X
```

```
Error: Objekt 'X' not found
```

Vectors

We've just seen how to create a vector: the c() function concatenates individual items into a vector.

```
xNum  <- c(1, 3.14159, 5, 7)
xNum
```

```
[1] 1.00000 3.14159 5.00000 7.00000
```

```
xLog  <- c(TRUE, FALSE, TRUE, TRUE)
xLog
```

```
[1]  TRUE FALSE  TRUE  TRUE
```

```
xChar <- c("foo", "bar", "boo", "far")
xChar
```

```
[1] "foo" "bar" "boo" "far"
```

Vectors: Type Coercion

A vector can only hold a single type of value (number, text, etc). Values are coerced to the most general type.

```
xMix  <- c(1, TRUE, 3, "Hello, world!")
xMix
```

```
[1] "1"   "TRUE"   "3"   "Hello, world!"
```

More about Vectors

c() can be used to add vectors just as it adds single items:

```
x2 <- c(x, x)
x2
```

```
[1] 2 4 6 8 2 4 6 8
```

Type coercion will be applied as needed:

```
c(x2, 100)
c(x2, "Hello")
```

```
[1]   2   4   6   8   2   4   6   8 100
[1] "2"   "4"   "6"   "8"   "2"   "4"   "6"   "8"   "Hello"
```

Forcing Coercion

```
xMix
```

```
[1] "1"    "TRUE"    "3"    "Hello, world!"
```

```
xMix[1] # we'll see more on indices later
```

```
[1] "1"
```

```
as.numeric(xMix[1]) # forces it to "numeric"
```

```
[1] 1
```

```
as.numeric(xMix[1]) + 1.5
```

```
[1] 2.5
```

Help!

There are many ways to get help for R:

Command/Source	Note
R: `?someword`	to get help on *someword* that R knows
R: `??someword`	to search all R help files for the word in text
R: `?` or `??''some string''`	search for a string, character, etc. that doesn't work as a word
R: `vignette()`	list all the vignettes available
R: `vignette(''zoo'')`	open the vignette named (for package) "zoo"
`Web`: CRAN task view	Suggested packages by area such as Econometrics, Clustering, etc. https://cran.r-project.org/web/views/
`Web`: R help list	Monitored by volunteers with many R experts and authors
`Web`: Google	Understands "R" in many contexts
`Web`: Stack Overflow	Often great contributions, http://stackoverflow.com/questions/tagged/r

Summary

The `summary()` function summarizes an object in a way that is (usually) appropriate for its data type:

```
summary(xNum)
```

```
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
1.000   2.606   4.071   4.035   5.500   7.000
```

```
summary(xChar)
```

```
Length     Class      Mode
     4 character character
```

Math on Vectors

```
x2
```

```
[1] 2 4 6 8 2 4 6 8
```

```
x2 + 1
```

```
[1] 3 5 7 9 3 5 7 9
```

```
x2 * pi
```

```
[1]  6.283185 12.566371 18.849556 25.132741  6.283185
[6] 12.566371 18.849556 25.132741
```

More Complex Math

R will generalize operations across multiple vectors (or matrices) as best it can:

```
x
```

```
[1] 2 4 6 8
```

```
x2 # longer than x
```

```
[1] 2 4 6 8 2 4 6 8
```

```
(x+cos(0.5)) * x2 # x is recycled to match x2
```

```
[1]  5.755165 19.510330 41.265495 71.020660  5.755165
[6] 19.510330 41.265495 71.020660
```

Length and Structure

So vectors can be recycled. How do you find the length?

```
length(x)
```

```
[1] 4
```

```
length(x2)
```

```
[1] 8
```

A more general solution is to investigate the structure:

```
str(x2)
```

```
num [1:8] 2 4 6 8 2 4 6 8
```

```
str(xChar)
```

```
chr [1:4] "foo" "bar" "boo" "far"
```

Sequences

Basic 1-by-1 integer sequences are constructed with the ":" operator:

```
xSeq <- 1:10
xSeq
```

```
[1] 1  2  3  4  5  6  7  8  9  10
```

Be careful with operator precedence ... clarify liberally with parentheses:

```
1:5*2
```

```
[1] 2  4  6  8 10
```

```
1:(5*2)
```

```
[1] 1  2  3  4  5  6  7  8  9  10
```

Indexing a Vector 1

Basic indexing uses a set of integers to select positions:

```
xNum
```

```
[1] 1.00000 3.14159 5.00000 7.00000
```

```
xNum[2:4]
```

```
[1] 3.14159 5.00000 7.00000
```

```
xNum[c(1, 3)]
```

```
[1] 1 5
```

Variables and math operators can be used as well:

```
myStart <- 2
xNum[myStart:sqrt(myStart + 7)]
```

```
[1] 3.14159 5.00000
```

Negative Indexing

A negative index omits elements (returns everything else):

```
1:(5*2)
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

```
xSeq[-5:-7]
```

```
[1]  1  2  3  4  8  9 10
```

Indexing with Boolean

Boolean values of `TRUE` and `FALSE` can be used to select items:

```
xNum
```

```
[1] 1.00000 3.14159 5.00000 7.00000
```

```
xNum[c(FALSE, TRUE, TRUE, TRUE)]
```

```
[1] 3.14159 5.00000 7.00000
```

This is most often used with comparative operators to select items:

```
xNum > 3
```

```
[1] FALSE  TRUE  TRUE  TRUE
```

```
xNum[xNum > 3]
```

```
[1] 3.14159 5.00000 7.00000
```

Missing and Interesting Values

```
my.test.scores <- c(91, 93, NA, NA)
mean(my.test.scores)
```

```
[1] NA
```

```
max(my.test.scores)
```

```
[1] NA
```

```
mean(my.test.scores, na.rm = TRUE)
```

```
[1] 92
```

```
max(my.test.scores, na.rm = TRUE)
```

```
[1] 93
```

Other Ways to Omit

```
na.omit(my.test.scores)
```

```
[1] 91 93
attr(,"na.action")
[1] 3 4
attr(,"class")
[1] "omit"
```

```
mean(na.omit(my.test.scores))
```

```
[1] 92
```

```
is.na(my.test.scores)
```

```
[1] FALSE FALSE  TRUE  TRUE
```

```
my.test.scores[!is.na(my.test.scores)]
```

```
[1] 91 93
```

Data Frames

Data frames are the most common way to handle data sets in R.

Data Frames 1

```
x.df <- data.frame(xNum, xLog, xChar)
x.df
```

```
     xNum  xLog xChar
1 1.00000  TRUE   foo
2 3.14159 FALSE   bar
3 5.00000  TRUE   boo
4 7.00000  TRUE   far
```

Data frames have *names* and are indexed *row, column*.

```
x.df[2,1]
```

```
[1] 3.14159
```

```
x.df[1,3]
```

```
[1] foo
Levels: bar boo far foo
```

Data Frames 2

By default, text data is converted to factors. You'll often want to turn that off:

```
x.df[1,3]
```

```
[1] foo
Levels: bar boo far foo
```

```
x.df <- data.frame(xNum, xLog, xChar,
                   stringsAsFactors = FALSE)
x.df[1,3]
```

```
[1] "foo"
```

Indexing Data Frames

```
x.df[2, ] # all of row 2
     xNum  xLog xChar
2 3.14159 FALSE   bar

x.df[ ,3] # all of column 3
[1] "foo" "bar" "boo" "far"

x.df[2:3, ]
     xNum  xLog xChar
2 3.14159 FALSE   bar
3 5.00000  TRUE   boo

x.df[ ,1:2]
     xNum  xLog
1 1.00000  TRUE
2 3.14159 FALSE
3 5.00000  TRUE
4 7.00000  TRUE
```

Negative Indexing Data Frames

```
x.df[-3, ] # omit the third observation
```

```
     xNum  xLog xChar
1 1.00000  TRUE   foo
2 3.14159 FALSE   bar
4 7.00000  TRUE   far
```

```
x.df[, -2] # omit the second column
```

```
     xNum xChar
1 1.00000   foo
2 3.14159   bar
3 5.00000   boo
4 7.00000   far
```

Let's Create more Interesting Data

Warning: we're about to delete everything first

```
rm(list = ls()) # caution, deletes all objects
```

Store Data

```
store.num <- factor(c(3, 14, 21, 32, 54)) # store id
store.rev <- c(543, 654, 345, 678, 234)   # store revenue,$K
store.visits <- c(45, 78, 32, 56, 34)     # visits, 1000s
store.manager <- c("Annie", "Bert", "Carla", "Dave", "Ella")

store.df <- data.frame(store.num, store.rev,
                        store.visits, store.manager,
                        stringsAsFactors = FALSE)
store.df
```

```
  store.num store.rev store.visits store.manager
1         3       543           45         Annie
2        14       654           78          Bert
3        21       345           32         Carla
4        32       678           56          Dave
5        54       234           34          Ella
```

Some Data Checks

```
summary(store.df) # always recommended!
```

```
store.num   store.rev        store.visits store.manager
3 :1       Min.   :234.0   Min.   :32    Length:5
14:1       1st Qu.:345.0   1st Qu.:34    Class :character
21:1       Median :543.0   Median :45    Mode  :character
32:1       Mean   :490.8   Mean   :49
54:1       3rd Qu.:654.0   3rd Qu.:56
           Max.   :678.0   Max.   :78
```

```
store.df$store.manager
```

```
[1] "Annie" "Bert"  "Carla" "Dave"  "Ella"
```

```
mean(store.df$store.rev)
```

```
[1] 490.8
```

Read and Write CSVs

```
write.csv(store.df, row.names = FALSE)
"store.num","store.rev","store.visits","store.manager"
"3",543,45,"Annie"
"14",654,78,"Bert"
"21",345,32,"Carla"
"32",678,56,"Dave"
"54",234,34,"Ella"
```

```
write.csv(store.df, file = "store-df.csv",
          row.names=FALSE)
read.csv("store-df.csv")  # "file=" is optional
  store.num store.rev store.visits store.manager
1         3       543           45         Annie
2        14       654           78          Bert
3        21       345           32         Carla
4        32       678           56          Dave
5        54       234           34          Ella
```

Further Topics

- Basic Functions
- Sequences, again
- Interesting numbers
- Load and save raw data

Writing Basic Functions

```
se <- function(x) {sd(x) / sqrt(length(x))}
se(store.df$store.visits)
```

```
[1] 8.42615
```

```
mean(store.df$store.visits) + 1.96 * se(store.df
   $store.visits)
```

```
## [1] 65.51525
```

A function has:

- an assigned name (created with '<-')
- zero or more arguments that it operates on (in ())
- a body (usually in { }) with lines of code
- a return value (the last computed value, by default)

Document Your Functions Inline!

```
se <- function(x) {
  # computes standard error of the mean
  tmp.sd <- sd(x)      # standard deviation
  tmp.N  <- length(x) # sample size
  tmp.se <- tmp.sd / sqrt(tmp.N) # std error of the mean
  return(tmp.se)       # return() is optional but clear
}
se(store.df$store.visits)
```

```
[1] 8.42615
```

This is much better! You can examine it to see what it does:

```
se
```

```
function(x) {
   # computes standard error of the mean
   tmp.sd <- sd(x)      # standard deviation
   tmp.N  <- length(x) # sample size
   tmp.se <- tmp.sd / sqrt(tmp.N)  # std error of the mean
   return(tmp.se)       # return() is optional but clear
}
```

Other Ways to Make Sequences

The seq() function constructs sequences in various ways:

```
seq(from=-5, to=28, by=4)
[1] -5 -1  3  7 11 15 19 23 27

seq(from=-5, to=28, length=6)
[1] -5.0  1.6  8.2 14.8 21.4 28.0
```

The rep() (repeat) function is also useful. It is especially good for constructing indices into data sets with repeating structure:

```
rep(c(1,2,3), each=3)
[1] 1 1 1 2 2 2 3 3 3

rep(seq(from=-3, to=13, by=4), c(1, 2, 3, 2, 1))
[1] -3  1  1  5  5  5  9  9 13
```

Infinite and Impossible Numbers

```
1/0
[1] Inf
```

```
log(c(-1,0,1))
Warning in log(c(-1, 0, 1)): NaNs wurden erzeugt
[1]  NaN -Inf    0
```

```
sqrt(-2)
Warning in sqrt(-2): NaNs wurden erzeugt

[1] NaN
```

```
sqrt(2i)
[1] 1+1i
```

You can use these values yourself (occasionally it makes sense):

```
10 < Inf
[1] TRUE
```

Loading and Saving Raw Data Formats

```
save(store.df, file="store-df-backup.RData")
rm(store.df)
mean(store.df$store.rev)      # error
```

```
Error in mean(store.df$store.rev): Object 'store.df'
not found
```

```
load("store-df-backup.RData")
mean(store.df$store.rev)      # works now
```

```
[1] 490.8
```

Loading Data has Silent Overwrite

```
store.df <- 5
store.df
```

```
[1] 5
```

```
load("store-df-backup.RData")
store.df
```

```
  store.num store.rev store.visits store.manager
1         3       543           45         Annie
2        14       654           78          Bert
3        21       345           32         Carla
4        32       678           56          Dave
5        54       234           34          Ella
```

Install the following R Packages

- AER
- BLPestimatoR
- car
- cluster
- coefplot
- corrplot
- data.table
- gpairs
- gmnl
- lattice
- lme4
- MASS
- mclust
- mlogit
- multcomp
- psych
- RColorBrewer
- rworldmap

Load the Data

The book walks through **simulation** of nearly all the data sets ... check that out, as there is much more about R in those sections.
For today, we'll just load the data from the website:

```
store.df <- read.csv("http://goo.gl/QPDdMl")
summary(store.df)
   storeNum          Year          Week          p1sales       p2sales
 Min.   :101.0   Min.   :1.0   Min.   : 1.00   Min.   : 73   Min.   : 51.0
 1st Qu.:105.8   1st Qu.:1.0   1st Qu.:13.75   1st Qu.:113   1st Qu.: 84.0
 Median :110.5   Median :1.5   Median :26.50   Median :129   Median : 96.0
 Mean   :110.5   Mean   :1.5   Mean   :26.50   Mean   :133   Mean   :100.2
 3rd Qu.:115.2   3rd Qu.:2.0   3rd Qu.:39.25   3rd Qu.:150   3rd Qu.:113.0
 Max.   :120.0   Max.   :2.0   Max.   :52.00   Max.   :263   Max.   :225.0

    p1price          p2price          p1prom         p2prom         country
  Min.   :2.190   Min.   :2.29   Min.   :0.0   Min.   :0.0000   AU:104
  1st Qu.:2.290   1st Qu.:2.49   1st Qu.:0.0   1st Qu.:0.0000   BR:208
  Median :2.490   Median :2.59   Median :0.0   Median :0.0000   CN:208
  Mean   :2.544   Mean   :2.70   Mean   :0.1   Mean   :0.1385   DE:520
  3rd Qu.:2.790   3rd Qu.:2.99   3rd Qu.:0.0   3rd Qu.:0.0000   GB:312
  Max.   :2.990   Max.   :3.19   Max.   :1.0   Max.   :1.0000   JP:416
                                                                 US:312
```

Descriptives 1

table() for categorical variable

```
table(store.df$p1price)
```

```
2.19 2.29 2.49 2.79 2.99
 395  444  423  443  375
```

The counts can be converted to proportions with prop.table()

```
table(store.df$p1price)
```

```
     2.19      2.29      2.49      2.79      2.99
0.1899038 0.2134615 0.2033654 0.2129808 0.1802885
```

Table as an Object

Tables are objects that can be assigned and indexed:

```
p1.table <-table(store.df$p1price)
p1.table
```

```
2.19 2.29 2.49 2.79 2.99
 395  444  423  443  375
```

```
p1.table[3]
```

```
2.49
 423
```

```
str(p1.table)
```

```
 'table' int [1:5(1d)] 395 444 423 443 375
 - attr(*, "dimnames")=List of 1
  ..$ : chr [1:5] "2.19" "2.29" "2.49" "2.79" ...
```

Two-Way Tables

```
table(store.df$p1price, store.df$p1prom)
```

```
        0   1
2.19 354  41
2.29 398  46
2.49 381  42
2.79 396  47
2.99 343  32
```

Note that tables index [row, column] like most things in R!

Core Descriptive Functions

Distribution functions that operate on a numeric vector

Describe	Function	Value
Extremes	min(x)	Minimum value
	max(x)	Maximum value
Central tendency	mean(x)	Arithmetic mean
	median(x)	Median
	var(x)	Variance around the mean
Dispersion	sd(x)	Standard deviation (sqrt(var(x)))
	IQR(x)	Interquartile range, 75th — 25th percentile
	mad(x)	Median absolute deviation (a robust variance estimator)
Points	quantile(x, probs=c(...))	Percentiles

Core Descriptive Functions

```
min(store.df$p1sales)
```

```
[1] 73
```

```
max(store.df$p2sales)
```

```
[1] 225
```

```
mean(store.df$p1prom)
```

```
[1] 0.1
```

```
median(store.df$p2sales)
```

```
[1] 96
```

Core Descriptive Functions

```
var(store.df$p1sales)
```

```
[1] 805.0044
```

```
sd(store.df$p1sales)
```

```
[1] 28.3726
```

```
IQR(store.df$p1sales)
```

```
[1] 37
```

```
mad(store.df$p1sales)
```

```
[1] 26.6868
```

Percentile (Quantile) Function

```
quantile(store.df$p1sales)  # default = 0:4*0.25
  0%  25%  50%  75% 100%
  73  113  129  150  263
```

```
quantile(store.df$p1sales, probs=c(0.25, 0.75))  #
    Interquartile
25% 75%
113 150
```

```
quantile(store.df$p1sales, probs=c(0.025,0.975))  #
    central 95%
 2.5% 97.5%
   88   199
```

```
quantile(store.df$p1sales, probs=1:10/10)  # shortcut
  10%   20%   30%   40%   50%   60%   70%   80%   90%  100%
100.0 109.0 117.0 122.6 129.0 136.0 145.0 156.0 171.0 263.0
```

Summary of Data Frame

```
summary(store.df)
```

```
   storeNum          Year          Week          p1sales       p2sales
 Min.   :101.0   Min.   :1.0   Min.   : 1.00   Min.   : 73   Min.   : 51.0
 1st Qu.:105.8   1st Qu.:1.0   1st Qu.:13.75   1st Qu.:113   1st Qu.: 84.0
 Median :110.5   Median :1.5   Median :26.50   Median :129   Median : 96.0
 Mean   :110.5   Mean   :1.5   Mean   :26.50   Mean   :133   Mean   :100.2
 3rd Qu.:115.2   3rd Qu.:2.0   3rd Qu.:39.25   3rd Qu.:150   3rd Qu.:113.0
 Max.   :120.0   Max.   :2.0   Max.   :52.00   Max.   :263   Max.   :225.0

    p1price          p2price          p1prom         p2prom         country
 Min.   :2.190   Min.   :2.29   Min.   :0.0   Min.   :0.0000   AU:104
 1st Qu.:2.290   1st Qu.:2.49   1st Qu.:0.0   1st Qu.:0.0000   BR:208
 Median :2.490   Median :2.59   Median :0.0   Median :0.0000   CN:208
 Mean   :2.544   Mean   :2.70   Mean   :0.1   Mean   :0.1385   DE:520
 3rd Qu.:2.790   3rd Qu.:2.99   3rd Qu.:0.0   3rd Qu.:0.0000   GB:312
 Max.   :2.990   Max.   :3.19   Max.   :1.0   Max.   :1.0000   JP:416
                                                               US:312
```

Summary of Data Frame Elements

```
summary(store.df$p1sales)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
     73     113     129     133     150     263
```

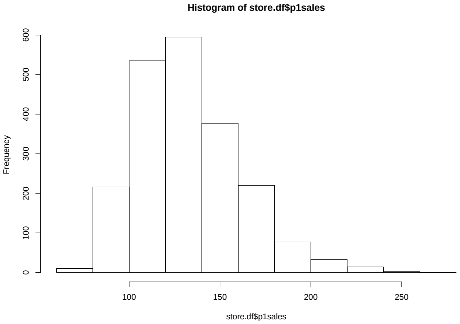
```
summary(store.df$p1sales,digits=2)  # round
   output
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
     73     110     130     130     150     260
```

Visualization: Steps to Prettify (1)

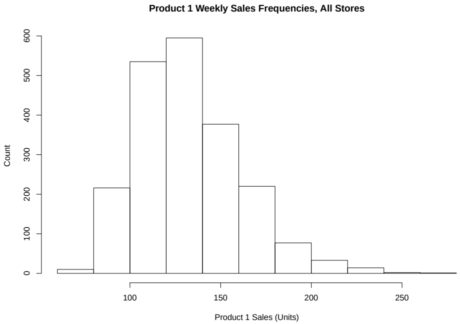
hist() for basic plot

```
hist(store.df$p1sales)
```



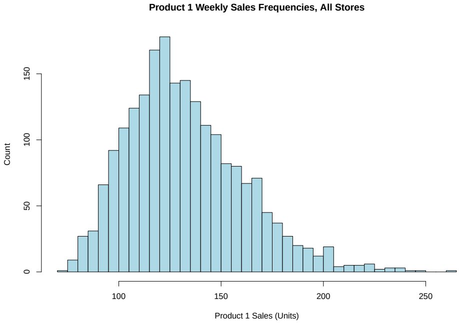
Improve it with Labels

```
hist(store.df$p1sales,
     main="Product 1 Weekly Sales Frequencies, All
         Stores",
     xlab="Product 1 Sales (Units)",
     ylab="Count")
```



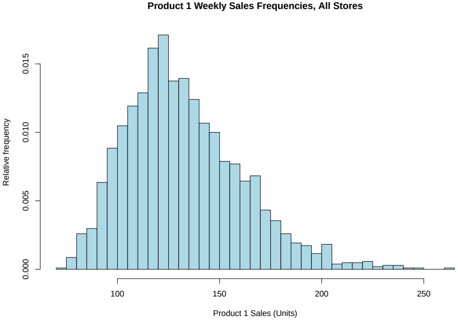
Make it more Granular and Colorful

```
hist(store.df$p1sales,
     main="Product 1 Weekly Sales Frequencies, All Stores",
     xlab="Product 1 Sales (Units)",
     ylab="Count",
     breaks=30,              # more columns
     col="lightblue")        # color the bars
```



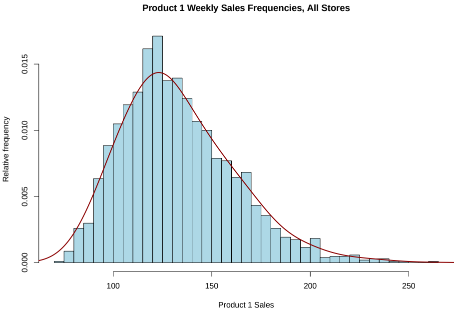
Change Counts to Proportions

```
hist(store.df$p1sales,
    main="Product 1 Weekly Sales Frequencies, All Stores",
    xlab="Product 1 Sales (Units)",
    ylab="Relative frequency", # changed
    breaks=30,
    col="lightblue",
    freq=FALSE)  # freq=FALSE for density
```



Add Density Curve

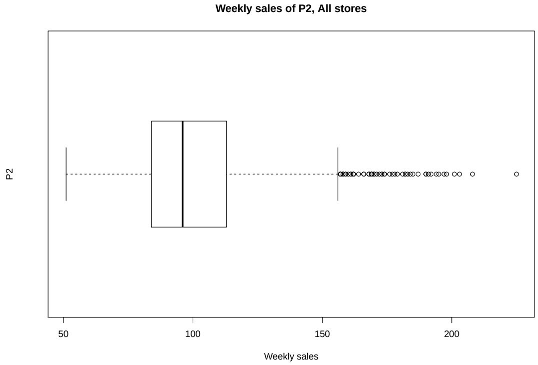
```
hist(store.df$p1sales,
     main="Product 1 Weekly Sales Frequencies, All Stores",
     xlab="Product 1 Sales", ylab="Relative frequency",
     breaks=30, col="lightblue", freq=FALSE)

lines(density(store.df$p1sales, bw=10), # bw = smoothing
      type="l", col="darkred", lwd=2)   # lwd = line width
```



Boxplot

Basic boxplot is good for data exploration:

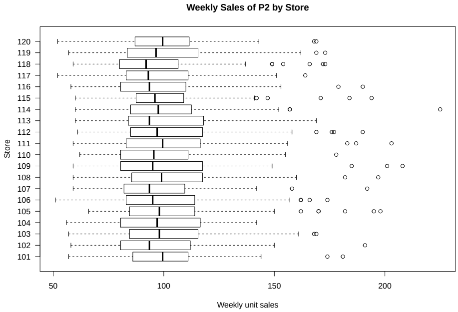
```
boxplot(store.df$p2sales, xlab="Weekly sales", ylab="P2",
        main="Weekly sales of P2, All stores",
        horizontal=TRUE)
```



Boxplot Broken out by Factor

Plot **DV ~ IV** to condition on a factor:

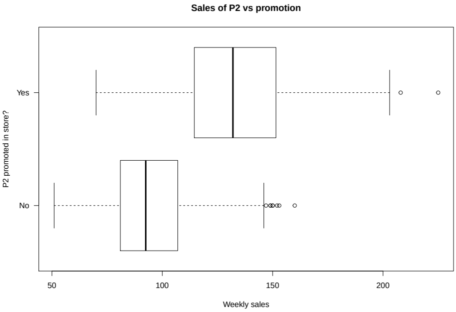
```
boxplot(store.df$p2sales ~store.df$storeNum,horizontal=TRUE,
        ylab="Store", xlab="Weekly unit sales", las=1,
        main="Weekly Sales of P2 by Store")
```



Boxplot with Data and Axes

Use **data=** to specify df, and **axis** to label axes better:

```
boxplot(p2sales ~ p2prom, data=store.df, horizontal=TRUE,
        yaxt="n", ylab="P2 promoted in store?",
        xlab="Weekly sales",main="Sales of P2 vs promotion")
axis(side=2), at=c(1,2), labels=c("No","Yes"), las=1)
```



by()

by() is one way to split data by a factor and apply a function to each group:

```
by(store.df$p1sales, store.df$storeNum, mean)
```

```
store.df$storeNum: 101
[1] 130.5385
------------------------------------------------------------
store.df$storeNum: 102
[1] 134.7404
------------------------------------------------------------
store.df$storeNum: 103
[1] 136.0385
------------------------------------------------------------
store.df$storeNum: 104
[1] 131.4423
------------------------------------------------------------
store.df$storeNum: 105
[1] 129.5288
------------------------------------------------------------
store.df$storeNum: 106
[1] 133.7981
------------------------------------------------------------
store.df$storeNum: 107
[1] 133.8077
------------------------------------------------------------
store.df$storeNum: 108
[1] 133.6923
------------------------------------------------------------
store.df$storeNum: 109
[1] 131.5481
```

by() Statement and more than one Function (1)

First define a function that computes several statistics. then include this function into the the by() statement:

```
desc <- function(x) {list(mean = mean(x), sd = sd(x))}
by(store.df$p1sales, store.df$storeNum, desc)
```

Alterative way:

```
library(psych)    # install package if needed
describeBy(store.df$p1sales, store.df$storeNum)
```

by() Statement and more than one Function (2)

Use data tables:

```
library(data.table) # install package if needed
# convert data to table
store.dt <- as.data.table(store.df)

# create descriptive measures manually
store.dt[, list(mean = mean(p1sales),
           sd = sd(p1sales)), keyby = storeNum]

# ... re-use the custom 'desc' function
store.dt[, desc(p1sales), keyby = storeNum]

# ... or the simple 'describe' function from the psych
    package
store.dt[, describe(p1sales), keyby = storeNum]
```

aggregate()

aggregate() collects results similar to by() into an object:

```
storeMean <-aggregate(store.df$p1sales,
                          by=list(store=store.df$
                              storeNum), mean)
storeMean
```

```
   store        x
1    101 130.5385
2    102 134.7404
3    103 136.0385
4    104 131.4423
5    105 129.5288
6    106 133.7981
7    107 133.8077
8    108 133.6923
9    109 131.5481
10   110 132.0962
11   111 130.4519
```

Additional Topics

- Indexing with tables
- Using `describe()` esp. for survey data
- Easy choropleth world map

Indexing with Tables

```
table(store.df$p1price, store.df$p1prom)
```

```
        0   1
  2.19 354  41
  2.29 398  46
  2.49 381  42
  2.79 396  47
  2.99 343  32
```

Note that tables index [row, column] like most things in R!
Two-way tables are also assignable & indexable:

```
p1.table2 <- table(store.df$p1price, store.df$p1prom)
p1.table2[, 2] / (p1.table2[, 1] + p1.table2[, 2])
```

```
      2.19       2.29       2.49       2.79       2.99 
0.10379747 0.10360360 0.09929078 0.10609481 0.08533333
```

Describe (psych Package)

`describe()` is especially useful for survey data.

```
library(psych)  # must install first
```

```
describe(store.df)
```

```
         vars    n   mean    sd median trimmed   mad    min    max range  skew kurtosis   se
storeNum    1 2080 110.50  5.77 110.50  110.50  7.41 101.00 120.00  19.0  0.00    -1.21 0.13
Year        2 2080   1.50  0.50   1.50    1.50  0.74   1.00   2.00   1.0  0.00    -2.00 0.01
Week        3 2080  26.50 15.01  26.50   26.50 19.27   1.00  52.00  51.0  0.00    -1.20 0.33
p1sales     4 2080 133.05 28.37 129.00  131.08 26.69  73.00 263.00 190.0  0.74     0.66 0.62
p2sales     5 2080 100.16 24.42  96.00   98.05 22.24  51.00 225.00 174.0  0.99     1.51 0.54
p1price     6 2080   2.54  0.29   2.49    2.53  0.44   2.19   2.99   0.8  0.28    -1.44 0.01
p2price     7 2080   2.70  0.33   2.59    2.69  0.44   2.29   3.19   0.9  0.32    -1.40 0.01
p1prom      8 2080   0.10  0.30   0.00    0.00  0.00   0.00   1.00   1.0  2.66     5.10 0.01
p2prom      9 2080   0.14  0.35   0.00    0.05  0.00   0.00   1.00   1.0  2.09     2.38 0.01
country*   10 2080   4.55  1.72   4.50    4.62  2.22   1.00   7.00   6.0 -0.29    -0.81 0.04
```

Aggregate Sales by Country

```
p1sales.sum <- aggregate(store.df$p1sales,
                         by=list(country=store.
                            df$country), sum)
p1sales.sum
```

```
  country     x
1      AU 14544
2      BR 27836
3      CN 27381
4      DE 68876
5      GB 40986
6      JP 55381
7      US 41737
```

Plot Sales by Country with rworldmap()

First we load the packages, and set up a map.
In our aggregated data, we use the **country** column to tell the map where to put the **p1sales.sum** aggregated mean.

```
library(rworldmap)   # must be installed
```

```
Loading required package: sp

For a short introduction type :   vignette('rworldmap')
```

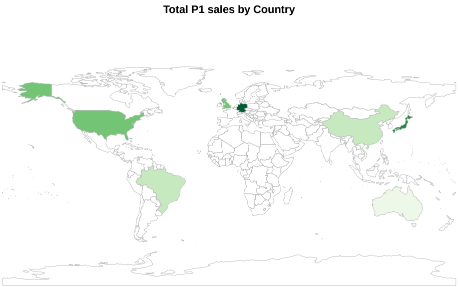
```
library(RColorBrewer) # must be installed

p1sales.map <- joinCountryData2Map(p1sales.sum,
                                   joinCode ="ISO2",
                                   nameJoinColumn ="country")
```

```
7 codes from your data successfully matched countries in the map
0 codes from your data failed to match with a country code in the map
235 codes from your data weren't represented in your data
```

Draw the Map

Once the data is “mapped” to the locations, we can draw the visualization:

```
mapCountryData(p1sales.map, nameColumnToPlot="x",
               mapTitle="Total P1 sales by Country",
               colourPalette=brewer.pal(7, "Greens"),
               catMethod="fixedWidth", addLegend=FALSE)
```



Load CRM Data

This is example data with data on customers' visits, transactions, and spending for online and retail purchases:

```
cust.df <- read.csv("http://goo.gl/PmPkaG")
str(cust.df)
```

```
'data.frame':    1000 obs. of  12 variables:
 $ cust.id          : int  1 2 3 4 5 6 7 8 9 10 ...
 $ age              : num  22.9 28 35.9 30.5 38.7 ...
 $ credit.score     : num  631 749 733 830 734 ...
 $ email            : Factor w/ 2 levels "no","yes": 2 2 2 2 1 2 2 2 1
 $ distance.to.store: num  2.58 48.18 1.29 5.25 25.04 ...
 $ online.visits    : int  20 121 39 1 35 1 1 48 0 14 ...
 $ online.trans     : int  3 39 14 0 11 1 1 13 0 6 ...
 $ online.spend     : num  58.4 756.9 250.3 0 204.7 ...
 $ store.trans      : int  4 0 0 2 0 0 2 4 0 3 ...
 $ store.spend      : num  140.3 0 0 95.9 0 ...
 $ sat.service      : int  3 3 NA 4 1 NA 3 2 4 3 ...
 $ sat.selection    : int  3 3 NA 2 1 NA 3 3 2 2 ...
```

Converting Data to Factors

Text data is automatically converted to factors when reading CSVs.
However, sometimes data that appears to be numeric is really not.
The factor() function will convert data to nominal factors:

```
str(cust.df$cust.id)
```

```
 int [1:1000] 1 2 3 4 5 6 7 8 9 10 ...
```

```
cust.df$cust.id <- factor(cust.df$cust.id)
str(cust.df$cust.id)
```

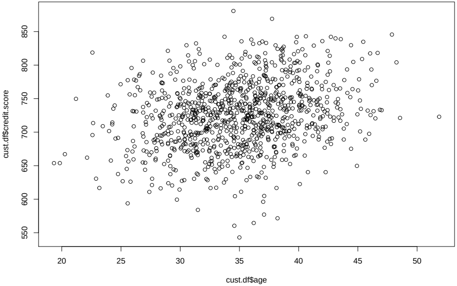
```
 Factor w/ 1000 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10
```

Option: ordered=TRUE (or ordered() function) creates ordinal factors.

Basic Scatterplot

Let's look at scatterplots. How does age relate to credit score?

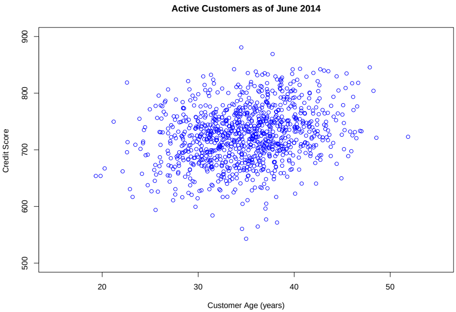
```
plot(x=cust.df$age, y=cust.df$credit.score)
```



A better Plot

Add color, labels, and adjust the axis limits:

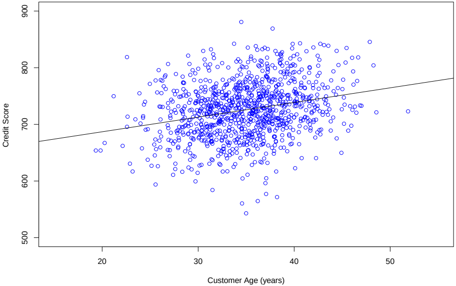
```
plot(cust.df$age, cust.df$credit.score,
     col="blue",
     xlim=c(15,55), ylim=c(500, 900),
     main="Active Customers as of June 2014",
     xlab="Customer Age (years)", ylab="Credit Score")
```



Add a Regression Line

abline() adds a regression line from a linear model (we'll discuss regression in depth later)

```
plot(cust.df$age, cust.df$credit.score,
     col="blue", xlim=c(15, 55), ylim=c(500, 900),
     xlab="Customer Age (years)", ylab="Credit Score")
abline(lm(cust.df$credit.score ~ cust.df$age))
```



Zero-Inflated and Skewed Data

Question: How do online sales relate to in-store sales? Does online decrease or increase online sales?
Before starting on this, note that customer data often has two issues:

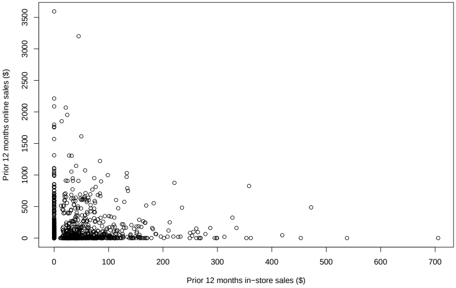
- **Skew**: many small transactions, few large ones
- **Zero inflation**: many zero values (because many customers spend $0 in any period)

There are various ways to handle these issues, but an easy approach is to use logarithmic plots. Instead of plotting on raw scales, plot on logarithmic scale.
Let's look at this in steps.

Scatterplot with Skewed Data

How does in-store spending relate to online spending? It's hard to tell because both are skewed! The apparent negative slope is very misleading.

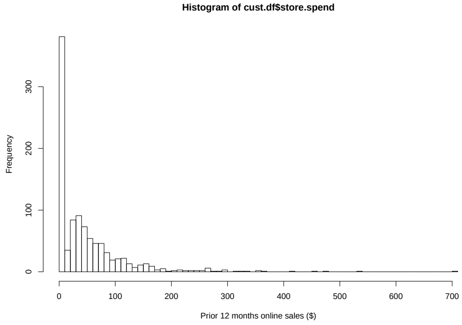
```
plot(cust.df$store.spend, cust.df$online.spend,
     xlab="Prior 12 months in-store sales ($)",
     ylab="Prior 12 months online sales ($)")
```



Looking at the Skew

A histogram reveals the skew, e.g., for in-store spending:

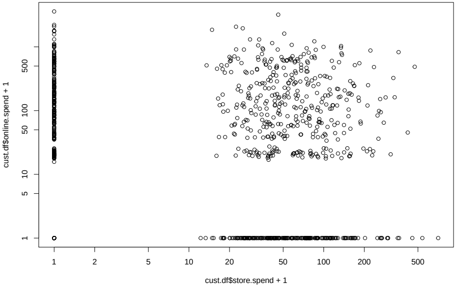
```
hist(cust.df$store.spend,
    breaks=(0:ceiling(max(cust.df$store.spend)/
       10))*10,
    xlab="Prior 12 months online sales ($)")
```



Using Logarithmic Axes

Use the log= argument to set one or both axes to logarithmic. **Caution**: $\log(x \leq 0)$ is not defined, so make sure values are positive. Add a constant such as +1 if needed. Now we see little or no relationship:

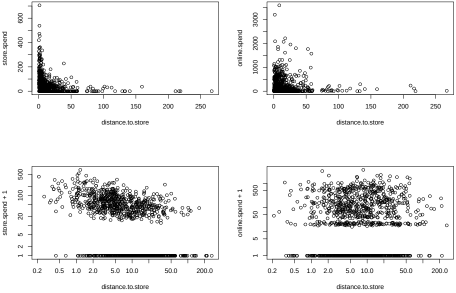
```
plot(cust.df$store.spend +1,
     cust.df$online.spend +1, log="xy")
```



Multi-Panel Plots

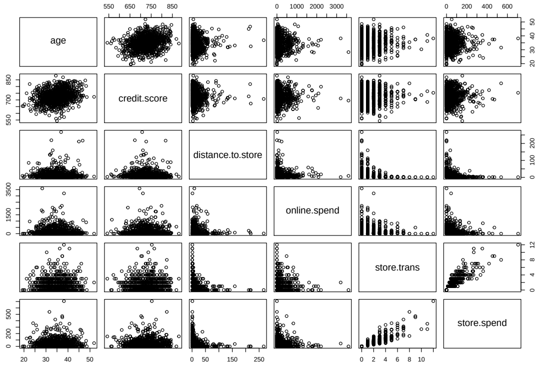
How does distance relate to in-store and online spending? Use par(mfrow=c(ROWS, COLUMNS)) to put multiple plots together:

```
par(mfrow=(2,2))
with(cust.df, plot(distance.to.store, store.spend))
with(cust.df, plot(distance.to.store, online.spend))
with(cust.df, plot(distance.to.store, store.spend+1,log="xy"))
with(cust.df, plot(distance.to.store, online.spend+1,log="xy"))
```



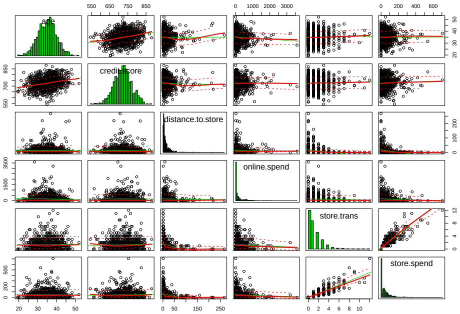
Scatterplot Matrix: Quick 2-Way Visualization

```
pairs(formula = ~age + credit.score + distance.to.store
 + online.spend + store.trans + store.spend, data=cust.df)
```



Fancy Alternative: scatterplotMatrix in “car”

```
library(car)          # install if needed
scatterplotMatrix(formula = ~ age + credit.score +
                  distance.to.store + online.spend +
                  store.trans + store.spend,}
                  data=cust.df, diagonal="histogram")
```



Correlation

A basic inferential test of Pearson's r can be done with cor.test():

```
cor.test(cust.df$age, cust.df$credit.score)
```

```
Pearson's product-moment correlation

data:  cust.df$age and cust.df$credit.score
t = 8.3138, df = 998, p-value = 3.008e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.1955974 0.3115816
sample estimates:
      cor
0.2545045
```

Age is associated with credit score here, $r = 0.25, p < .05$.

Correlation Matrix

The cor() function computes *r* between all pairs of variables:

```
cor(cust.df[, c(2, 3}, 5:12)])  # only numeric cols}
```

```
                          age credit.score distance.to.store online.visits online.trans
age                1.000000000  0.254504457        0.00198741   -0.06138107  -0.06301994
credit.score       0.254504457  1.000000000       -0.02326418   -0.01081827  -0.00501840
distance.to.store  0.001987410 -0.023264183        1.00000000   -0.01460036  -0.01955166
online.visits     -0.061381070 -0.010818272       -0.01460036    1.00000000   0.98732805
online.trans      -0.063019935 -0.005018400       -0.01955166    0.98732805   1.00000000
online.spend      -0.060685729 -0.006079881       -0.02040533    0.98240684   0.99334666
store.trans        0.024229708  0.040424158       -0.27673229   -0.03666932  -0.04024588
store.spend        0.003841953  0.042298123       -0.24149487   -0.05068554  -0.05224465
sat.service                 NA           NA                NA            NA           NA
sat.selection               NA           NA                NA            NA           NA
                  online.spend store.trans  store.spend sat.service sat.selection
age               -0.060685729  0.02422971  0.003841953          NA            NA
credit.score      -0.006079881  0.04042416  0.042298123          NA            NA
distance.to.store -0.020405326 -0.27673229 -0.241494870          NA            NA
online.visits      0.982406842 -0.03666932 -0.050685537          NA            NA
online.trans       0.993346657 -0.04024588 -0.052244650          NA            NA
online.spend       1.000000000 -0.04089133 -0.051690053          NA            NA
store.trans       -0.040891332  1.00000000  0.892756851          NA            NA
store.spend       -0.051690053  0.89275685  1.000000000          NA            NA
sat.service                 NA          NA           NA           1            NA
sat.selection               NA          NA           NA          NA             1
```

Redoing that with Complete Cases

Add use="complete.obs" argument:

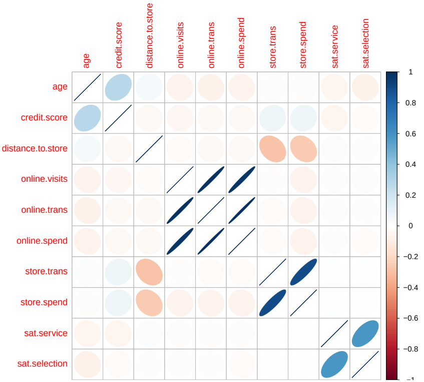
```
cor(cust.df[, c(2, 4, 5:12)], use="complete.obs")
```

```
                          age credit.score distance.to.store online.visits online.trans
age                1.00000000   0.27384005        0.04606521   -0.06334468  -0.07282280
credit.score       0.27384005   1.00000000       -0.03444605   -0.04337523  -0.03041161
distance.to.store  0.04606521  -0.03444605        1.00000000   -0.02680514  -0.03046099
online.visits     -0.06334468  -0.04337523       -0.02680514    1.00000000   0.98349553
online.trans      -0.07282280  -0.03041161       -0.03046099    0.98349553   1.00000000
online.spend      -0.06857108  -0.03344978       -0.03224989    0.97645451   0.99306906
store.trans        0.01917930   0.07147923       -0.28777128   -0.01833510  -0.02671777
store.spend       -0.01101162   0.07319630       -0.25249002   -0.06022874  -0.06321920
sat.service       -0.05846361  -0.05095454        0.02561875   -0.01614200  -0.01762744
sat.selection     -0.07411506  -0.02350937        0.01293211   -0.01837661  -0.01846859

                  online.spend  store.trans  store.spend  sat.service sat.selection
age                -0.06857108  0.019179304 -0.011011624 -0.058463613  -0.074115061
credit.score       -0.03344978  0.071479231  0.073196297 -0.050954538  -0.023509365
distance.to.store  -0.03224989 -0.287771277 -0.252490015  0.025618746   0.012932114
online.visits       0.97645451 -0.018335097 -0.060228738 -0.016142000  -0.018376612
online.trans        0.99306906 -0.026717771 -0.063219201 -0.017627444  -0.018468588
online.spend        1.00000000 -0.025572587 -0.061704685 -0.011873515  -0.021148403
store.trans        -0.02557259  1.000000000  0.892855470  0.001821736   0.001309098
store.spend        -0.06170469  0.892855470  1.000000000  0.007466294   0.008642354
sat.service        -0.01187352  0.001821736  0.007466294  1.000000000   0.587855775
sat.selection      -0.02114840  0.001309098  0.008642354  0.587855775   1.000000000
```

Visualize Correlation Matrix

Use the "corrplot" package. See book (and help) for more options.

```
library(corrplot)    # install if needed
corrplot(corr=cor(cust.df[ , c(2,3,, 5:12)],
              use="complete.obs"), method ="ellipse")
```



Data Transformation

R makes it easy to explore common transformations (log, 1/x, sqrt, etc.)

```
cor(cust.df$distance.to.store, cust.df$store.spend)
```

```
[1] -0.2414949
```

```
cor(1/cust.df$distance.to.store, cust.df$store.spend)}
```

```
[1] 0.4329997
```

```
cor(1/sqrt(cust.df$distance.to.store), cust.df$store.spend)
```

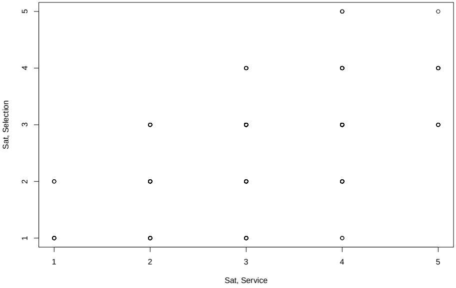
```
[1] 0.4843334
```

We see that closeness to store (1/*distance*) is strongly relation to spend. Having a good transformation gives a better estimate of strength.
See book for more on common transformations and how to find an optimal transform.

Polychoric Correlation

With ordinal variables such as satisfaction ratings, consider polychoric correlation instead of Pearson's r. We need it because the ratings are not fine grained, and plot on top of one another:

```
plot(cust.df$sat.service, cust.df$sat.selection,
     xlab="Sat, Service", ylab="Sat, Selection")
```



Polychoric Correlation Test

First identify respondents with data:

```
resp <- !is.na(cust.df$sat.service)
```

Now compare Pearson correlation to polychoric:

```
cor(cust.df$sat.service[resp], cust.df$sat.selection[resp])
```

```
[1] 0.5878558
```

```
library(psych)   # install if needed
```

```
polychoric(cbind(cust.df$sat.service[resp],
                 cust.df$sat.selection[resp]))
```

```
1 cells were adjusted for 0 values using the correction for continuity. Examine your data carefully.

Call: polychoric(x = cbind(cust.df$sat.service[resp], cust.df$sat.selection[resp]))
Polychoric correlations
   C1   C2
R1 1.00
R2 0.62 1.00

 with tau of
        1     2    3   4
[1,] -1.83 -0.72 0.54 1.7
[2,] -0.99  0.12 1.26 2.4
```

Load Segmentation/Subscription Data

As usual, check the book for details on the data simulation. For now:

```
seg.df <- read.csv("http://goo.gl/qw303p")
summary(seg.df)
```

```
      age            gender        income            kids        ownHome  
 Min.   :19.26   Female:157   Min.   : -5183   Min.   :0.00   ownNo :159  
 1st Qu.:33.01   Male  :143   1st Qu.: 39656   1st Qu.:0.00   ownYes:141  
 Median :39.49                Median : 52014   Median :1.00               
 Mean   :41.20                Mean   : 50937   Mean   :1.27               
 3rd Qu.:47.90                3rd Qu.: 61403   3rd Qu.:2.00               
 Max.   :80.49                Max.   :114278   Max.   :7.00               

  subscribe         Segment   
 subNo :260   Moving up : 70  
 subYes: 40   Suburb mix:100  
              Travelers : 80  
              Urban hip : 50  
```

Descriptives: Selecting by Group

```
mean(seg.df$income[seg.df$Segment == "Moving up"])
```

```
[1] 53090.97
```

```
mean(seg.df$income[seg.df$Segment == "Moving up" &}
                   seg.df$subscribe=="subNo"])
```

```
[1] 53633.73
```

This quickly gets tedious!

Descriptives: Apply a Function by Group

by(VARIABLE of interest, GROUPING variable, FUNCTION)

```
by(seg.df$income, seg.df$Segment, mean)
```

```
seg.df$Segment: Moving up
[1] 53090.97
------------------------------------------------------------
seg.df$Segment: Suburb mix
[1] 55033.82
------------------------------------------------------------
seg.df$Segment: Travelers
[1] 62213.94
------------------------------------------------------------
seg.df$Segment: Urban hip
[1] 21681.93
```

Descriptives: Apply a Function by Group

Use list() to have more than one grouping variable:

```
by(seg.df$income, list(seg.df$Segment, seg.df$subscribe), mean)
```

```
: Moving up
: subNo
[1] 53633.73
--------------------------------------------------------
: Suburb mix
: subNo
[1] 54942.69
--------------------------------------------------------
: Travelers
: subNo
[1] 62746.11
--------------------------------------------------------
: Urban hip
: subNo
[1] 22082.11
--------------------------------------------------------
: Moving up
: subYes
[1] 50919.89
--------------------------------------------------------
: Suburb mix
: subYes
[1] 56461.41
--------------------------------------------------------
: Travelers
```

Aggregate: Use a Formula!

Break out *income by segment*, in data "*seg.df*", computing the *mean*:

```
aggregate(income ~ Segment, data=seg.df, mean)
```

```
     Segment   income
1  Moving up 53090.97
2 Suburb mix 55033.82
3  Travelers 62213.94
4  Urban hip 21681.93
```

This extends easily to multiple dimensions:

```
aggregate(income ~ Segment + ownHome, data=seg.df, mean)
```

```
     Segment ownHome   income
1  Moving up   ownNo 54497.68
2 Suburb mix   ownNo 54932.83
3  Travelers   ownNo 63188.42
4  Urban hip   ownNo 21337.59
5  Moving up  ownYes 50216.37
6 Suburb mix  ownYes 55143.21
7  Travelers  ownYes 61889.12
8  Urban hip  ownYes 23059.27
```

Aggregate Returns a Data Frame

```
agg.data <- aggregate(income ~ Segment + ownHome,
                      data=seg.df, mean)
str(agg.data)
```

```
'data.frame':    8 obs. of  3 variables:
 $ Segment: Factor w/ 4 levels "Moving up","Suburb mix",..: 1 2 3 4 1 2 3 4
 $ ownHome: Factor w/ 2 levels "ownNo","ownYes": 1 1 1 1 2 2 2 2
 $ income : num  54498 54933 63188 21338 50216 ...
```

```
agg.data[2, ]}
```

```
     Segment ownHome   income
2 Suburb mix   ownNo 54932.83
```

```
agg.data[2, 3]
```

```
[1] 54932.83
```

Tables

Reminder – a table counts occurrences of a single value, such as one level of a factor.

```
table(seg.df$Segment, seg.df$ownHome)
```

```
            ownNo ownYes
  Moving up    47     23
  Suburb mix   52     48
  Travelers    20     60
  Urban hip    40     10
```

Telling R to use *seg.df* for everything is easy with **with()**:

```
with(seg.df, table(Segment, ownHome))
```

```
            ownHome
Segment      ownNo ownYes
  Moving up    47     23
  Suburb mix   52     48
  Travelers    20     60
  Urban hip    40     10
```

Note that table() uses R standard *(X, Y)* == *Row, Column* order.

prop.table()

Reminder – get proportions for a table by wrapping **table()** with **prop.table()**:

```
with(seg.df, prop.table(table(Segment, ownHome)))
```

```
            ownHome
Segment          ownNo     ownYes
  Moving up  0.15666667 0.07666667
  Suburb mix 0.17333333 0.16000000
  Travelers  0.06666667 0.20000000
  Urban hip  0.13333333 0.03333333
```

The default computes full table proportions. Obtain marginal proportions by specifying rows (*margin=1*) or columns (*margin=2*):

```
with(seg.df, prop.table(table(Segment, ownHome), margin=1))
```

```
            ownHome
Segment         ownNo    ownYes
  Moving up  0.6714286 0.3285714
  Suburb mix 0.5200000 0.4800000
  Travelers  0.2500000 0.7500000
  Urban hip  0.8000000 0.2000000
```

Doing Math in a Table

Reminder – aggregate() can be used to apply a function to data, computing the result within with each group.
For instance, to add up the total number of kids in each segment, use *sum*:

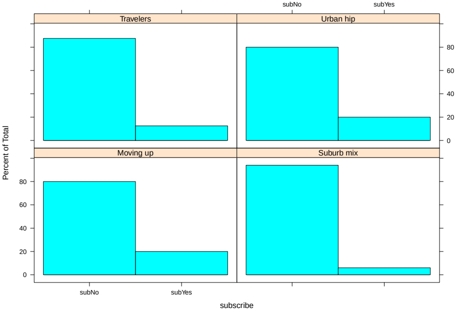
```
aggregate(kids ~ Segment, data=seg.df, sum)
```

```
     Segment kids
1  Moving up  134
2 Suburb mix  192
3  Travelers    0
4  Urban hip   55
```

Visualization: Counts by Group

histogram() in the **lattice** package plots proportional frequency by group. This is an alternative to basic `hist()` that we saw in an earlier chapter.
To get subscribers (**~subscribe**) by segment (**| Segment**):

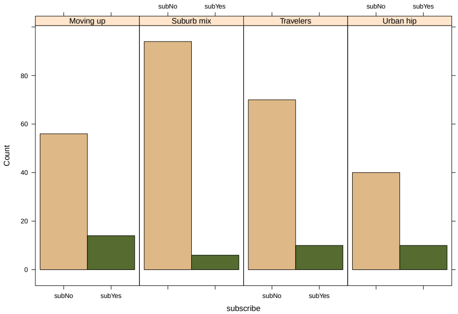
```
library(lattice)
histogram(~subscribe | Segment, data=seg.df)
```



Histograms Continued

You can plot counts instead of proportions with **type="count"**. There are options for the layout (cols, rows in this case) and colors:

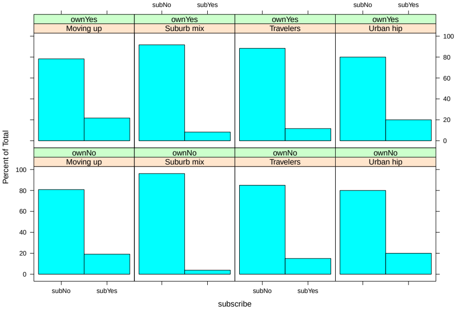
```
histogram(~subscribe | Segment, data=seg.df,
             type="count", layout=c(4,1),
             col=c("burlywood", "darkolivegreen"))
```



Histograms by 2 Factors

Break out by multiple factors using **| var1 + var2 + ...**:

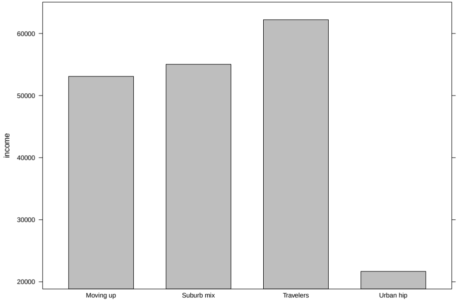
```
histogram(~subscribe | Segment + ownHome, data=seg.df)
```



Continuous Data: "Spreadsheet" Style

The general process is to **aggregate()** the data that you want, then plot that. For example: mean income by segment, using a **barchart**:

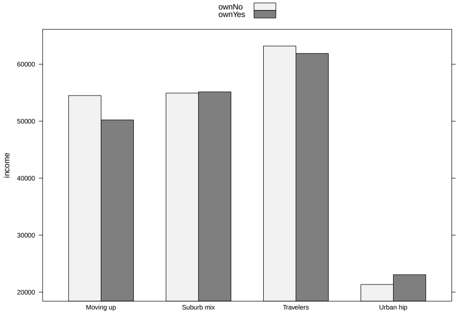
```
seg.mean <- aggregate(income ~ Segment, data=seg.df,
                        mean)
library(lattice)
barchart(income ~ Segment, data=seg.mean, col="grey")
```



Continuous Data by Two Factors

Use aggregate with + to break out multiple factors:

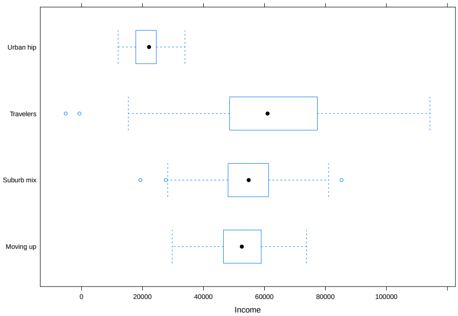
```
seg.agg <- aggregate(income ~ Segment + ownHome,
                     data=seg.df, mean)
barchart(income ~ Segment, data=seg.agg,
         groups=ownHome, auto.key=TRUE,
         par.settings = simpleTheme(col=c("gray95",
                                          "gray50")) )
```



Continuous Data: “Statistics” Style

Boxplots show much more information about the data distribution (see book for details). `bwplot()` from `lattice` is an upgrade over `boxplot()` that we saw in earlier chapters:

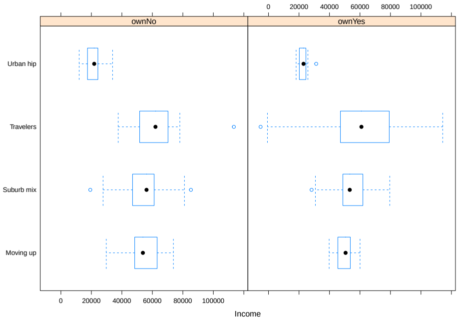
```
library(lattice)
bwplot(Segment ~ income, data=seg.df, horizontal=TRUE,
       xlab = "Income")
```



Boxplots with Two-Way Grouping

You can add a "conditioning" variable using |:

```
bwplot(Segment ~ income | ownHome, data=seg.df,
       horizontal=TRUE, xlab="Income")
```



Extra Slides

- Language: `for()`
- Language: `if()` and `ifelse()`

Language: for()

`for()` loops over a sequence of values, assigning them in turn to an index variable:

```
for (i in 1:10) { print(i) }
```

```
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
[1] 6
[1] 7
[1] 8
[1] 9
[1] 10
```

Advanced R programmers often avoid `for()` ... but if it makes sense to you then go ahead and use it!

Integers are not Required, just a Sequence

```
i.seq <- seq(from=2.1, to=6.2, by=0.65)
for (i in i.seq ) { print(i) }
```

```
[1] 2.1
[1] 2.75
[1] 3.4
[1] 4.05
[1] 4.7
[1] 5.35
[1] 6
```

```
for (i in c(5, 4, 3, 5, 3, 0, -100, 19)) { cat(i, " ")
    }
```

```
5  4  3  5  3  0  -100  19
```

```
for (i in c("Hello ", "world, ","welcome to R!")) {
    cat(i) }
```

if()

`if()` is used for basic program flow control.

`if (A) { B else C }` means:

"If A is true, compute B *[any commands inside {}]*, otherwise compute C."

```
x <- 2
if (x > 0) {
  print ("Positive!")
} else {
  print ("Zero or negative!")
}
```

```
[1] "Positive!"
```

Rules of brackets are confusing, so simplify: always use **{** and **}** !

`else C` is optional. If !A and no C block, nothing will occur.

ifelse()

ifelse() is a vectorized version of if(). Use it to *create a vector* using logic, *not* to control program flow.

```
x <- -2:2

if (x > 0) {        # bad code -- only tests once!
  "pos"
} else {
  "neg/zero"
}
```

The correct way to do this is:

```
ifelse(x > 0, "pos", "neg/zero")
```

```
[1] "neg/zero" "neg/zero" "neg/zero" "pos"      "pos"
```

Instead of simply getting values as the result, you could perform actions (e.g., by calling functions to do something).

Load the Data

```
seg.df <- read.csv("http://goo.gl/qw303p")
summary(seg.df)
```

```
     age               gender         income             kids      
 Min.   :19.26    Female:157   Min.   : -5183   Min.   :0.00  
 1st Qu.:33.01    Male  :143   1st Qu.: 39656   1st Qu.:0.00  
 Median :39.49                 Median : 52014   Median :1.00  
 Mean   :41.20                 Mean   : 50937   Mean   :1.27  
 3rd Qu.:47.90                 3rd Qu.: 61403   3rd Qu.:2.00  
 Max.   :80.49                 Max.   :114278   Max.   :7.00  

   ownHome         subscribe          Segment   
 ownNo :159      subNo :260    Moving up : 70  
 ownYes:141      subYes: 40    Suburb mix:100  
             Travelers : 80
             Urban hip : 50
```

Chi-Square Test

Tests equality of marginal counts in groups. *Important*: compile a **table** first (don't use raw data). Then use `chisq.test()`.
Let's see this for simple, fake data first:

```
tmp.tab <- table(rep(c(1:4), times=c(25,25,25,20)))
tmp.tab
```

```
 1  2  3  4
25 25 25 20
```

```
chisq.test(tmp.tab)
```

```
Chi-squared test for given probabilities

data:  tmp.tab
X-squared = 0.78947, df = 3, p-value = 0.852
```

chisq.test "significant" and "not significant"

```
tmp.tab <- table(rep(c(1:4), times=c(25,25,25,20)))
chisq.test(tmp.tab)
```

```
Chi-squared test for given probabilities

data:  tmp.tab
X-squared = 0.78947, df = 3, p-value = 0.852
```

```
tmp.tab <- table(rep(c(1:4), times=c(25,25,25,10)))
tmp.tab
```

```
 1  2  3  4
25 25 25 10
```

```
chisq.test(tmp.tab)
```

```
Chi-squared test for given probabilities

data:  tmp.tab
X-squared = 7.9412, df = 3, p-value = 0.04724
```

chisq.test with Segment Data

Are the segments the same size? (Not a very interesting question, perhaps.)

```
table(seg.df$Segment)
```

```
 Moving up Suburb mix  Travelers  Urban hip
        70        100         80         50
```

```
chisq.test(table(seg.df$Segment))
```

```
Chi-squared test for given probabilities

data:  table(seg.df$Segment)
X-squared = 17.333, df = 3, p-value = 0.0006035
```

chisq.test with Segment Data

Do they have the same rate of subscription by ownership?

```
table(seg.df$subscribe, seg.df$ownHome)
```

```
        ownNo ownYes
  subNo    137    123
  subYes    22     18
```

```
chisq.test(table(seg.df$subscribe, seg.df$
   ownHome))
```

```
Pearson's Chi-squared test with Yates' continuity correction

data:  table(seg.df$subscribe, seg.df$ownHome)
X-squared = 0.010422, df = 1, p-value = 0.9187
```

chisq.test with Segment Data

Without correction (matches traditional formula):

```
chisq.test(table(seg.df$subscribe, seg.df$
   ownHome), correct=FALSE)
```

```
Pearson's Chi-squared test

data:  table(seg.df$subscribe, seg.df$ownHome)
X-squared = 0.074113, df = 1, p-value = 0.7854
```

Proportions: Binomial Test

`binom.test(x=successes, n=trials, p=proportion)` tests whether the count of *successes* in a certain number of *trials* matches an expected *proportion*:

```
binom.test(12, 20, p=0.5)
```

```
Exact binomial test

data:  12 and 20
number of successes = 12, number of trials = 20, p-value = 0.5034
alternative hypothesis: true probability of success is not equal to 0.5
95 percent confidence interval:
 0.3605426 0.8088099
sample estimates:
probability of success: 0.6
```

Proportions: Binomial Test Continued

The same proportion with higher N can be significant:

```
# binom.test(12, 20, p=0.5)
binom.test(120, 200, p=0.5)
```

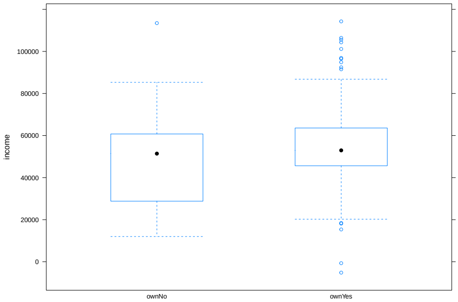
```
Exact binomial test

data:  120 and 200
number of successes = 120, number of trials = 200, p-value =
0.005685
alternative hypothesis: true probability of success is not equal to 0.5
95 percent confidence interval:
 0.5285357 0.6684537
sample estimates:
probability of success: 0.6
```

See book for Agresti-Coull and other methods if your data has small N or is near 0 or 1 proportion.

t-Tests

Does income differ for home owners in our data? A t-test compares the means of two groups, relative to the variance. First let's visualize it:

```
library(lattice)
bwplot(income ~ ownHome, data=seg.df)
```



t.test()

Use formula syntax `t.test(outcomeVar ~ groupingVar)`:

```
t.test(income ~ ownHome, data=seg.df)
```

```

Welch Two Sample t-test

data:  income by ownHome
t = -3.2731, df = 285.25, p-value = 0.001195
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -12080.155  -3007.193
sample estimates:
 mean in group ownNo mean in group ownYes
            47391.01             54934.68
```

Mean income is higher among home owners in our data, $p < .01$.

t.test() for a subset() of Data

`subset()` is an easy way to select portions of a data set. Here's the same t-test but only for the Travelers segment:

```
t.test(income ~ ownHome, data=subset(seg.df, Segment==
    "Travelers"))
```

```
Welch Two Sample t-test

data:  income by ownHome
t = 0.26561, df = 53.833, p-value = 0.7916
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -8508.993 11107.604
sample estimates:
 mean in group ownNo mean in group ownYes
            63188.42             61889.12
```

Mean income is *not* significantly different between home owners and non-owners in the Travelers segment.

ANOVA Basics

ANOVA (analysis of variance) compares the difference in means among two or more groups, relative to their variances.
Recommended procedure: (1) fit a model with `aov()`. (2) use `anova()` on the model object to obtain a typical ANOVA table.
For two groups it is effectively the same as a t-test:

```
seg.aov.own <- aov(income ~ ownHome, data=seg.df)
anova(seg.aov.own)
```

```
Analysis of Variance Table

Response: income
           Df     Sum Sq    Mean Sq F value   Pr(>F)
ownHome     1 4.2527e+09 4252661211  10.832 0.001118 **
Residuals 298 1.1700e+11  392611030
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

ANOVA: Multiple Groups

The same model works for multiple groups. Make sure that the independent variable is a **factor** (use `factor()` to convert if necessary).

```
aggregate(income ~ Segment, mean, data=seg.df)
```

```
      Segment   income
1   Moving up 53090.97
2  Suburb mix 55033.82
3   Travelers 62213.94
4   Urban hip 21681.93
```

ANOVA: Multiple Groups

```
seg.aov.seg <- aov(income ~ Segment, data=seg.df)
anova(seg.aov.seg)
```

```
Analysis of Variance Table

Response: income
           Df     Sum Sq    Mean Sq F value    Pr(>F)
Segment     3 5.4970e+10 1.8323e+10  81.828 < 2.2e-16 ***
Residuals 296 6.6281e+10 2.2392e+08
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Multiple Effect Models: Basic Formulas

Use formula syntax to add variables

Symbol	Meaning
~	RESPONSE ~ PREDICTOR(s)
+	Additional main effect
:	Interaction without main effect
	All main effects and their interactions
.	Shortcut for "all other variables"

Examples:

Model	Formula
income by ownership & segment	income ~ own + segment
income by interaaction of ownership by segment	income ~ own:segment
income with all effects of ownership with segment	income ~ own + segment + own:segment
... or the same but not as clear ...	income ~ own*segment
income as a response to all other variables	income ~ .

ANOVA with Segment + Ownership

```
anova(aov(income ~ Segment + ownHome, data=seg.df))
```

```
Analysis of Variance Table

Response: income
           Df     Sum Sq    Mean Sq F value Pr(>F)
Segment     3 5.4970e+10 1.8323e+10 81.6381 <2e-16 ***
ownHome     1 6.9918e+07 6.9918e+07  0.3115 0.5772
Residuals 295 6.6211e+10 2.2444e+08
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

Mean income differs by *Segment*, but not by *ownership* after Segment is controlled. Model by ownership alone might be misleading:

ANOVA with Segment + Ownership

```
anova(aov(income ~ ownHome, data=seg.df))
```

```
Analysis of Variance Table

Response: income
           Df     Sum Sq    Mean Sq F value   Pr(>F)
ownHome     1 4.2527e+09 4252661211  10.832 0.001118 **
Residuals 298 1.1700e+11  392611030
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

ANOVA with Interaction

```
anova(aov(income ~ Segment * ownHome, data=seg.df))
```

```
Analysis of Variance Table

Response: income
                 Df     Sum Sq    Mean Sq F value Pr(>F)
Segment           3 5.4970e+10 1.8323e+10 81.1305 <2e-16 ***
ownHome           1 6.9918e+07 6.9918e+07  0.3096 0.5784
Segment:ownHome   3 2.6329e+08 8.7762e+07  0.3886 0.7613
Residuals       292 6.5948e+10 2.2585e+08
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Mean income differs by *segment*, not by *ownership*, and not by the *interaction* of ownership with segment.
Recommended: instead of using * , specify all effects directly:

```
anova(aov(income ~ Segment + ownHome
                 + Segment:ownHome, data=seg.df))
```

Model Comparison

The anova() command will also compare the fit of models:

```
anova(aov(income ~ Segment,           data=seg.df),
      aov(income ~ Segment + ownHome, data=seg.df))
```

```
Analysis of Variance Table

Model 1: income ~ Segment
Model 2: income ~ Segment + ownHome
  Res.Df        RSS Df Sum of Sq      F Pr(>F)
1    296 6.6281e+10
2    295 6.6211e+10  1  69918004 0.3115 0.5772
```

In this case, once we model income by segment, adding ownership does not improve the model fit.

Visualization: ANOVA Group Means

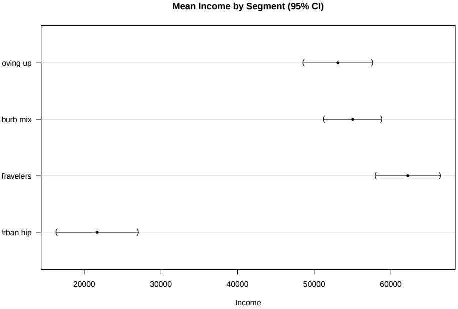
Use **glht()** in **multcomp** package as an easy way to get mean and CI:

```
# install.packages("multcomp")      # if needed
library(multcomp)
```

```
Loading required package: mvtnorm

Loading required package: survival

Loading required package: TH.data

Loading required package: MASS

Attaching package: 'TH.data'

The following object is masked from 'package:MASS':

    geyser
```

Visualization: ANOVA Group Means

```
seg.aov <- aov(income ~ -1 + Segment, data=seg.df)
          # model w/o int.
by.seg  <- glht(seg.aov) # means and CIs
plot(by.seg, xlab="Income", main="Mean Income by
   Segment (95% CI)")
```



Outline

Multiple Linear Regression Model

$$y_t = \beta_0 + \beta_1 x_{1t} + \beta_2 x_{2t} + \dots + \beta_K x_{Kt} + u_t \qquad t = 1, \dots, T$$

Assumptions:

A1: The multiple regression model includes all relevant exogenous variables; those variables, which are included are not irrelevant

A2: The true relationship between $x_{1t}, x_{2t}, \dots, x_{Kt}$ and y_t is linear

A3: The $K+1$ parameters are constant for all T observations

Content of many slides is taken from https://onlinecourses.science.psu.edu/stat501

Multiple Linear Regression Model

B1: $E(u_t) = 0 \quad \forall \quad t = 1, ..., T$

B2: Homoscedasticity
$var(u_t) = \sigma^2 \quad \forall \quad t = 1, ..., T$

B3: No autocorrelation
$cov(u_t, u_s) = 0 \quad \forall \quad t \neq s; t = 1, ..., T; s = 1, ..., T$

B4: The error terms are normally distributed
$u_t \sim N(0, \sigma^2)$

C1: The exogenous variables $x_{1t}, x_{2t}, \ldots, x_{Kt}$
are no random variables, but, like in an experiment, they can be controlled

C2: Freedom of perfect multicollinearity

$\Rightarrow$ If the assumptions are met, OLS is a BLUE (Best Linear Unbiased Estimator).

Quality-Indicators for Estimation Procedures

Unbiasedness: An estimator $\hat{\beta}$ is called unbiased, if the values $\hat{\beta}$, which were estimated on the basis of the repeated random sample hit the true value of β on average, i.e. $E(\hat{\beta}) = \beta$.

Efficiency: An unbiased estimator $\hat{\beta}$ is efficient, if it has the smallest variance within the class of unbiased estimators $\hat{\beta} = E[(\hat{\beta} - \beta)^2]$.

Estimation

$$y_t = \beta_0 + \beta_1 x_{1t} + \beta_2 x_{2t} + ... + \beta_K x_{Kt} + u_t \qquad t = 1, ..., T$$

Let $Y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ . \\ y_T \end{bmatrix}$, $X = \begin{bmatrix} 1 & x_{11} & x_{21} & x_{31} & .. & x_{k1} \\ 1 & x_{12} & .. & .. & .. & .. \\ 1 & .. & .. & .. & .. & .. \\ 1 & .. & .. & .. & .. & .. \\ 1 & x_{1T} & x_{2T} & .. & .. & x_{KT} \end{bmatrix}$, $\beta = \begin{bmatrix} \beta_0 \\ \beta_2 \\ \beta_3 \\ . \\ \beta_K \end{bmatrix}$ and $u = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ . \\ u_T \end{bmatrix}$

Note that Y and X are data and β is the coefficient vector to be estimate.

Estimation

In matrix notation

$$Y = \beta X + u$$

Minimizing the sum of squared errors

$$\min_{\beta} u'u = \min_{\beta} (Y - \beta X)'(Y - \beta X)$$

Using matrix calculus, the solution is

$$\hat{\beta} = \left(X'X\right)^{-1} X'y$$

If the assumptions A1, A2, A3, B1, B2, B3, B4, C1, C2, and C3 are fulfilled then the β is an unbiased, consistent and most efficient linear estimator.

Goodness of Fit

Coefficient of determination:

$$R^2 = 1 - \frac{u'u}{(y-\bar{y})'(y-\bar{y})}$$

Adjusted coefficient of determination:

$$\overline{R}^2 = 1 - \frac{T-1}{T-k-1}\left(1-R^2\right)$$

Hypothesis Testing

$$var(\hat{\beta}) = \hat{\sigma}^2 \cdot Diag\left[(X'X)^{-1}\right]$$

$$\hat{\sigma}^2 = \frac{1}{T-(K+1)} \sum_{t=1}^{T} u_t^2$$

$$se(\hat{\beta}) = \sqrt{\hat{\sigma}^2 \cdot Diag\left[(X'X)^{-1}\right]}$$

$$t = \frac{\hat{\beta}_k - \beta_k}{\sqrt{\hat{\sigma}^2 \cdot Diag\left[(X'X)^{-1}\right]_k}}$$

Hypothesis Testing

Simultaneous test of various linear combinations of parameters

$H_0 : \beta_0 = \beta_1 = \beta_2 = ... = \beta_k = 0 \qquad \forall k = 1, ..., K$

$H_1 : \beta_0 \neq 0$ or $\beta_1 \neq 0$ or $\beta_2 \neq 0...$ or $\beta_k \neq 0 \qquad \forall k = 1, ..., K$

$$F\Big[K+1, T-(K+1)\Big] = \frac{R^2/(K+1)}{(1-R^2)/(T-(K+1))}$$

Determine the critical value F_a with the table for the F-distribution:

$$F_{(K+1, T-K-1)}$$

Hypothesis Testing

Diagnosis and test of alternative specifications

- t-test

$H_0 : \beta_k = 0 \qquad t = \frac{\hat{\beta}_k - 0}{\hat{se}(\hat{\beta}_k)}$

- F-test

$$F = \frac{(S^0_{\hat{u}\hat{u}} - S_{\hat{u}\hat{u}})/L}{S_{\hat{u}\hat{u}}/(T-K-1)}$$

$S^0_{\hat{u}\hat{u}} =$ sum of the squared residuals of the restricted model
$S_{\hat{u}\hat{u}} =$ sum of the squared residuals of the unrestricted model
$L =$ difference in the number of parameters between the unrestricted and the restricted model

Hypothesis Testing

Partial R Squared
Partial R^2 is also called the coefficient of partial determination

What percent of the variation in the response cannot be explained by the predictors in the reduced model, but can be explained by the rest of the predictors in the full model?

$$R^2_{\text{partial}} = \frac{\text{SSE(reduced)} - \text{SSE(full)}}{\text{SSE(reduced)}}$$

Alternative Model Comparison

Information Criteria
To compare regression models one may also use information criterion statistics. For regression models, these statistics combine information about the SSE, number of parameters in the model, and the sample size. A low value, compared to values for other possible models, is good.

Three common information criteria are called Akaike's Information Criterion (AIC), the Bayesian Information Criterion (BIC) (which is sometimes called Schwartz's Bayesian Criterion (SBC)), and Amemiya's Prediction Criterion (APC).

$$
\begin{aligned}
AIC_p &= n\ln(SSE) - n\ln(n) + 2p \\
BIC_p &= n\ln(SSE) - n\ln(n) + p\ln(n) \\
APC_p &= \frac{(n+p)}{n(n-p)} SSE
\end{aligned}
$$

In the formulas, $n =$ sample size and $p =$ number of regression coefficients in the model being evaluated (including the intercept). Notice that the only difference between AIC and BIC is the multiplier of p, the number of parameters. Each of the information criteria is used in a similar way-in comparing two models, the model with the lower value is preferred.

The BIC places a higher penalty on the number of parameters in the model so will tend to reward more parsimonious (smaller) models. This stems from one criticism of AIC in that it tends to overfit models.

Prediction

Confidence Interval for the Mean Response
Confidence interval, called a "t-interval", for the mean response

The standard error of the fit at X_h is given by

$$\mathrm{se}(\hat{y}_h) = \sqrt{\mathrm{MSE}(\mathbf{X}_h^{\mathrm{T}}(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}_h)}$$

The confidence interval is: $\hat{y}_h \pm t_{(\alpha/2,n-p)} \times \mathrm{se}(\hat{y}_h)$

Factors affecting the width of the t-interval for the mean response

- As the mean square error (MSE) decreases, the width of the interval decreases.
- As we decrease the confidence level, the t-multiplier decreases, and hence the width of the interval decreases. In practice, we wouldn't want to set the confidence level below 90.
- As we increase the sample size n, the width of the interval decreases.
- The closer X_h is to the average of the sample's predictor values, the narrower the interval

Prediction

Prediction Interval for a New Response
Compute prediction interval for a new response y_{new}

The prediction interval is: $\hat{y}_h \pm t_{(\alpha/2,n-p)} \times \sqrt{MSE + [\text{se}(\hat{y}_h)]^2}$

with $\sqrt{MSE + [\text{se}(\hat{y}_h)]^2}$ as standard error of the prediction.

Outliers and High Leverage Observations

Distinction

- An outlier is a data point whose response y does not follow the general trend of the rest of the data.
- A data point has high leverage if it has "extreme" predictor x values. With a single predictor, an extreme x value is simply one that is particularly high or low. With multiple predictors, extreme x values may be particularly high or low for one or more predictors, or may be "unusual" combinations of predictor values (e.g., with two predictors that are positively correlated, an unusual combination of predictor values might be a high value of one predictor paired with a low value of the other predictor).

Using Leverages to Help Identify Extreme x Values

Definition and properties of leverages

$$
\begin{aligned}
Y &= X\beta + \varepsilon \\
\hat{y} &= Xb \\
b &= (X'X)^{-1}X'y \\
\hat{y} &= X(X'X)^{-1}X'y \\
H &= X(X'X)^{-1}X' \\
\hat{y} &= Hy \\
\hat{y}_i &= h_{i1}y_1 + h_{i2}y_2 + ... + h_{ii}y_i + ... + h_{in}y_n \quad \text{for } i = 1, ..., n
\end{aligned}
$$

The leverage, h_{ii}, quantifies the influence that the observed response y_i has on its predicted value y^i. That is, if h_{ii} is small, then the observed response y_i plays only a small role in the value of the predicted response y^i. On the other hand, if h_{ii} is large, then the observed response y_i plays a large role in the value of the predicted response y^i. It's for this reason that the h_{ii} are called the "leverages".

Using Leverages to Help Identify Extreme x Values

Identifying data points whose x values are extreme
A common rule is to flag any observation whose leverage value, h_{ii}, is more than 3 times larger than the mean leverage value:

$$\bar{h} \quad = \quad \frac{\sum_{i=1}^{n} h_{ii}}{n} = \frac{p}{n}$$

Rule of thumb:

$$h_{ii} \quad > \quad 3\left(\frac{p}{n}\right)$$

or:

$$h_{ii} \quad > \quad 2\left(\frac{p}{n}\right)$$

Identifying Outliers (Unusual y Values)

Studentized residuals
Studentized residuals (or internally studentized residuals) are defined for each observation, $i = 1, \ldots, n$ as an ordinary residual divided by an estimate of its standard deviation::

$$r_i = \frac{e_i}{s(e_i)} = \frac{e_i}{\sqrt{MSE(1-h_{ii})}}$$

An observation with an internally studentized residual that is larger than 3 (in absolute value) is generally deemed an outlier.

Identifying Influential Data Points

Two Measures

- Difference in fits (DFFITS)
- Cook's distance

The basic idea behind each of these measures is the same, namely to delete the observations one at a time, each time refitting the regression model on the remaining $n-1$ observations. Then, we compare the results using all n observations to the results with the i-th observation deleted to see how much influence the observation has on the analysis.

Identifying Influential Data Points

Difference in Fits (DFFITS)

$$DFFITS_i = \frac{\hat{y}_i - \hat{y}_{(i)}}{\sqrt{MSE_{(i)} h_{ii}}}$$

The numerator measures the difference in the predicted responses obtained when the i-th data point is included and excluded from the analysis. The denominator is the estimated standard deviation of the difference in the predicted responses. Therefore, the difference in fits quantifies the number of standard deviations that the fitted value changes when the i-th data point is omitted.

An observation is deemed influential if the absolute value of its DFFITS value is greater than:

$$2\sqrt{\frac{p+1}{n-p-1}}$$

where n is the number of observations and p is the number of parameters including the intercept.

Identifying Influential Data Points

Cook's distance measure

$$D_i = \frac{(y_i - \hat{y}_i)^2}{p \times MSE}\left[\frac{h_{ii}}{(1-h_{ii})^2}\right]$$

Cook's D_i depends on both the residual, e_i (in the first term), and the leverage, h_{ii}. That is, both the x value and the y value of the data point play a role in the calculation of Cook's distance.

- D_i directly summarizes how much all of the fitted values change when the i-th observation is deleted.
- A data point having a large D_i indicates that the data point strongly influences the fitted values.

Rules of thumb:

- If D_i is greater than 0.5, then the i-th data point is worthy of further investigation as it may be influential.
- If D_i is greater than 1, then the i-th data point is quite likely to be influential.
- If D_i sticks out like a sore thumb from the other D_i values, it is almost certainly influential.

A Strategy for Dealing with Problematic Data Points

Data analysts should use the measures described only as a way of screening the data set for potentially influential data points.

First, check for obvious data errors:

- If the error is just a data entry or data collection error, correct it.
- If the data point is not representative of the intended study population, delete it.
- If the data point is a procedural error and invalidates the measurement, delete it.

Consider the possibility that you might have just misformulated your regression model:

- Did you leave out any important predictors?
- Should you consider adding some interaction terms?
- Is there any nonlinearity that needs to be modeled?

Decide whether or not deleting data points is warranted:

- Do not delete data points just because they do not fit your preconceived regression model.
- You must have a good, objective reason for deleting data points.
- If you delete any data after you've collected it, justify and describe it in your reports.
- If you are not sure what to do about a data point, analyze the data twice — once with and once without the data point — and report the results of both analyses.

Econometric Problems

Incorrect choice of the exogenous variables

$$y_t = \beta_0 + \beta_1 x_{1t} + u'_t$$
$$y_t = \beta_0 + \beta_1 x_{1t} + \beta_2 x_{2t} + u_t \qquad \text{correct model}$$
$$y_t = \beta_0 + \beta_1 x_{1t} + \beta_2 x_{2t} + \beta_3 x_{3t} + u''_t$$

a) Omit some variables

$$u'_t = \beta_2 x_2 + u_t$$
$$E(u'_t) = E(\beta_2 x_{2t} + u_t) = \beta_2 x_{2t} + E(u_t) = \beta_2 x_{2t} + 0 \neq 0$$

Econometric Problems

$$E(\beta_1') = E\left(\hat{\beta}_1 + \hat{\beta}_2 \frac{S_{12}}{S_{11}}\right) = \beta_1 + \beta_2 \frac{S_{12}}{S_{11}}$$

$S_{i,j}$ = covariance of the variables i and j

Consequences for the interval estimator:

$$\left[\hat{\beta}_1' - t_{a/2} \cdot \hat{se}(\hat{\beta}_1'); \hat{\beta}_1' + t_{a/2} \cdot \hat{se}(\hat{\beta}_1')\right] \qquad \Rightarrow \text{biased}$$

Consequences for the hypothesis test:

$$\hat{se}(\hat{\beta}_1') = \sqrt{var(\hat{\beta}_1')} \qquad \Rightarrow \text{biased}$$

Econometric Problems

b) Use of incorrect variables

$$u_t'' = u_t - \beta_3 x_{3t}$$

because of $\beta_3 = 0 \qquad \rightarrow u_t'' = u_t$

$$E(\hat{\beta}_1'') = \beta_1 \qquad E(\hat{\beta}_2'') = \beta_2 \qquad E(\hat{\beta}_3'') = \beta_3 = 0$$

Consequences for the interval estimator:

$\Rightarrow$ unbiased, but inefficient $\qquad u_t'' = u_t$ but $se(\hat{\beta}_1'') > se(\hat{\beta}_1)$

Consequences for the hypothesis test:

$\Rightarrow$ utilizable, but needlessly blurred

Econometric Problems

Non-linear interdependence

Variable parameters

e.g. structural break in t^*

$$y_t = \beta_0 + \gamma D_t + (\beta_1 + \delta D_t) x_t + u_t$$

$$D_t = \begin{cases} 0, & \text{if } t < t^* \\ 0, & \text{else} \end{cases}$$

Assessing the Model Assumptions

- Create a scatterplot with the residuals, ei, on the vertical axis and the fitted values, y^i, on the horizontal axis and visual assess whether:
  - the (vertical) average of the residuals remains close to 0 as we scan the plot from left to right
  - the (vertical) spread of the residuals remains approximately constant as we scan the plot from left to right
  - there are no excessively outlying points
  - violation of any of these three may necessitate remedial action (such as transforming one or more predictors and/or the response variable), depending on the severity of the violation
- If the data observations were collected over time (or space) create a scatterplot with the residuals, on the vertical axis and the time (or space) sequence on the horizontal axis and visual assess whether there is no systematic non-random pattern

Assessing the Model Assumptions

- Create a series of scatterplots with the residuals on the vertical axis and each of the predictors in the model on the horizontal axes and visual assess whether:
  - the (vertical) average of the residuals remains close to 0 as we scan the plot from left to right
  - the (vertical) spread of the residuals remains approximately constant as we scan the plot from left to right
  - violation of either of these for at least one residual plot may suggest the need for transformations of one or more predictors and/or the response variable
- Create a histogram, boxplot, and/or normal probability plot of the residuals to check for approximate normality. (Of these plots, the normal probability plot is generally the most effective.)
- Create a series of scatterplots with the residuals, on the vertical axis and each of the available predictors that have been omitted from the model on the horizontal axes and visual assess whether:
  - there are no strong linear or simple nonlinear trends in the plot;
  - violation may indicate the predictor in question (or a transformation of the predictor) might be usefully added to the model.
  - it can sometimes be helpful to plot functions of predictor variables on the horizontal axis of a residual plot, for example interaction terms consisting of one quantitative predictor multiplied by another quantitative predictor. A strong linear or simple nonlinear trend in the resulting plot may indicate the variable plotted on the horizontal axis might be usefully added to the model.

Test for Error Normality

H_0: the errors follow a normal distribution
H_1: the errors do not follow a normal distribution

Anderson-Darling Test
The Anderson-Darling Test measures the area between a fitted line (based on the chosen distribution) and a nonparametric step function (based on the plot points). The statistic is a squared distance that is weighted more heavily in the tails of the distribution. Smaller Anderson-Darling values indicate that the distribution fits the data better.

$$A^2 = -n - \sum_{i=1}^{n} \frac{2i-1}{n}[\log \mathrm{F}(e_i) + \log(1 - \mathrm{F}(e_{n+1-i}))],$$

where $F(\cdot)$ is the cumulative distribution of the normal distribution. The test statistic is compared against the critical values from a normal distribution in order to determine the p-value.

Test for Error Normality

H_0: the errors follow a normal distribution
H_1: the errors do not follow a normal distribution

Shapiro-Wilk Test
The Shapiro-Wilk Test uses the test statistic

$$W = \frac{\left(\sum_{i=1}^{n} a_i e_{(i)}\right)^2}{\sum_{i=1}^{n}(e_i - \bar{e})^2},$$

where the a_i values are calculated using the means, variances, and covariances of the $e(i)$. W is compared against tabulated values of this statistic's distribution. Small values of W will lead to rejection of the null hypothesis.

Test for Error Normality

H_0: the errors follow a normal distribution
H_1: the errors do not follow a normal distribution

Kolmogorov-Smirnov Test (also known as the Lilliefors Test)
The Kolmogorov-Smirnov Test compares the empirical cumulative distribution function of sample data with the distribution expected if the data were normal. If this observed difference is sufficiently large, the test will reject the null hypothesis of population normality.

$$D = \max(D^+, D^-),$$

where

$$D^+ = \max_i (i/n - \mathrm{F}(e_{(i)}))$$
$$D^- = \max_i (\mathrm{F}(e_{(i)}) - (i-1)/n),$$

where $e(i)$ pertains to the ith largest value of the error terms. The test statistic is compared against the critical values from a normal distribution in order to determine the p-value.

Tests for Constant Error Variance

There are various tests that may be performed on the residuals for testing if they have constant variance. It is usually sufficient to "visually" interpret a residuals versus fitted values plot. However, some tests can provide an added layer of justification to your analysis. Note that some of the following procedures require you to partition the residuals into a certain number of groups, say $g \geq 2$ groups of sizes $n_1, \ldots, _{ng}$ such that $\sum_{i=1}^{g} n_i = n$. For these procedures, the sample variance of group i is given by:

$$s_i^2 = \frac{\sum_{j=1}^{n_i} (e_{i,j} - \bar{e}_{i,\cdot})^2}{n_i - 1},$$

where $e_{i,j}$ is the j-th residual from group i.
Moreover, the pooled variance is given by:

$$s_p^2 = \frac{\sum_{i=1}^{g} (n_i - 1) s_i^2}{n - g}.$$

Tests for Constant Error Variance

F-Test

Suppose we partition the residuals of observations into two groups - one consisting of residuals associated with the lowest predictor values and the other consisting of those belonging to the highest predictor values. Treating these two groups as if they could (potentially) represent two different populations, we can test

$$H_0 : \sigma_1^2 = \sigma_2^2$$
$$H_A : \sigma_1^2 \neq \sigma_2^2$$

using the F-statistic $F^* = s_1^2/s_2^2$. This test statistic is distributed according to a F_{n_1-1,n_2-1} distribution, so if $F^* \geq F_{n_1-1,n_2-1;1-\alpha}$, then reject the null hypothesis and conclude that there is statistically significant evidence that the variance is not constant.

Tests for Constant Error Variance

Modified Levene Test

Another test for nonconstant variance is the modified Levene test (sometimes called the Brown-Forsythe test). This test does not require the error terms to be drawn from a normal distribution and hence it is a nonparametric test. The test is constructed by grouping the residuals into g groups according to the values of the quantity on the horizontal axis of the residual plot. It is typically recommended that each group has at least 25 observations and usually $g = 2$ groups are used.
Begin by letting group 1 consist of the residuals associated with the n_1 lowest values of the predictor. Then, let group 2 consists of the residuals associated with the n_2 highest values of the predictor (so $n_1 + n2_= n$). The objective is to perform the following hypothesis test:

$$H_0 : \text{the variance is constant}$$
$$H_A : \text{the variance is not constant.}$$

The test statistic for the above is computed as follows:

$$\begin{aligned}
d_{i,j} &= |e_{i,j} - \tilde{e}_{i,\cdot}| \text{ where } \tilde{e}_{i,\cdot} \text{is the median of the } j\text{th group of residuals} \\
s_L &= \sqrt{\frac{\sum_{j=1}^{n_1}(d_{1,j} - \bar{d}_1)^2 + \sum_{j=1}^{n_2}(d_{2,j} - \bar{d}_2)^2}{n_1 + n_2 - 2}} \\
L &= \frac{\bar{d}_1 - \bar{d}_2}{s_L\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}
\end{aligned}$$

L is approximately distributed according to a $t_{n_1+n_2-2}$ distribution, or (equivalently) L^2 is approximately distributed according to an F_{1,n_1+n_2-2} distribution.

Tests for Constant Error Variance

Breusch-Pagan Test

The Breusch-Pagan test (also known as the Cook-Weisberg score test) is an alternative to the modified Levene test. The Breusch-Pagan test assumes that the error terms are normally distributed, with $\mathrm{E}(\varepsilon_i) = 0$ and $\mathrm{Var}(\varepsilon_i) = \sigma_i^2$ (i.e., nonconstant variance). The σ_i^2 values depend on the horizontal axis quantity (X_i) values in the following way:

$$\log \sigma_i^2 = \gamma_0 + \gamma_1 X_i.$$

We are interested in testing the null hypothesis of constant variance versus the alternative hypothesis of nonconstant variance. Specifically, the hypothesis test is formulated as:

$$\begin{aligned} H_0 &: \gamma_1 = 0 \\ H_A &: \gamma_1 \neq 0. \end{aligned}$$

This test is carried out by first regressing the squared residuals on the predictor (i.e., regressing e_i^2 on X_i). The sum of squares resulting from this analysis is denoted by SSR^*, which provides a measure of the dependency of the error term on the predictor. The test statistic is given by

$$X^{2*} = \frac{\mathrm{SSR}^*/2}{(\mathrm{SSE}/n)^2},$$

where SSE is from the regression analysis of the response on the predictor. The p-value for this test is found using a χ^2 distribution with 1 degree of freedom (written as χ_1^2).

Multicollinearity

What is multicollinearity? Multicollinearity exists when two or more of the predictors in a regression model are moderately or highly correlated.

- the estimated regression coefficient of any one variable depends on which other predictors are included in the model
- the precision of the estimated regression coefficients decreases as more predictors are added to the model
- the marginal contribution of any one predictor variable in reducing the error sum of squares depends on which other predictors are already in the model
  hypothesis tests for $\beta_k = 0$ may yield different conclusions depending on which predictors are in the model

There are two types of multicollinearity

- **Structural multicollinearity** is a mathematical artifact caused by creating new predictors from other predictors — such as, creating the predictor x^2 from the predictor x.
- **Data-based multicollinearity** is a result of a poorly designed experiment, reliance on purely observational data, or the inability to manipulate the system on which the data are collected.

Detecting Multicollinearity

Some of the common methods used for detecting multicollinearity include:

- The analysis exhibits the signs of multicollinearity — such as, estimates of the coefficients vary from model to model.
- The t-tests for each of the individual slopes are non-significant ($P > 0.05$), but the overall F-test for testing all of the slopes are simultaneously 0 is significant ($P < 0.05$).
- The correlations among pairs of predictor variables are large.

Looking at correlations only among pairs of predictors is limiting. It is possible that the pairwise correlations are small, and yet a linear dependence exists among three or even more variables, for example, if $X_3 = 2X_1 + 5X_2 + error$

Detecting Multicollinearity Using Variance Inflation Factors

The variance inflation factor for the estimated coefficient b_k — denoted VIFk — is just the factor by which the variance is inflated:

$$\begin{aligned} y_i &= \beta_0 + \beta_1 x_{i1} + \cdots + \beta_k x_{ik} + \cdots + \beta_{p-1} x_{i,p-1} + \varepsilon_i \\ Var(b_k) &= \frac{\sigma^2}{\sum_{i=1}^{n} (x_{ik} - \bar{x}_k)^2} \times \frac{1}{1 - R_k^2} \end{aligned}$$

where R_k^2 is the R^2-value obtained by regressing the k-th predictor on the remaining predictors. The greater the linear dependence among the predictor x_k and the other predictors, the larger the R_k^2 value.

Detecting Multicollinearity Using Variance Inflation Factors

The larger the R_k^2 value, the larger the variance of b_k.
How much larger? Take the ratio of the two variances:

$$\begin{aligned} \frac{Var(b_k)}{Var(b_k)_{min}} &= \frac{\left(\frac{\sigma^2}{\sum(x_{ik}-\bar{x}_k)^2} \times \frac{1}{1-R_k^2}\right)}{\left(\frac{\sigma^2}{\sum(x_{ik}-\bar{x}_k)^2}\right)} = \frac{1}{1-R_k^2} \\ VIF_k &= \frac{1}{1-R_k^2} \end{aligned}$$

where R_k^2 is the R^2-value obtained by regressing the k-th predictor on the remaining predictors. Note that a variance inflation factor exists for each of the k predictors in a multiple regression model.
It is a measure of how much the variance of the estimated regression coefficient b_k is "inflated" by the existence of correlation among the predictor variables in the model.
The general rule of thumb:

- VIFs > 4 warrant further investigation
- VIFs > 10 are signs of serious multicollinearity.

Weighted Least Squares

The method of ordinary least squares assumes that there is constant variance in the errors (which is called homoscedasticity). The method of weighted least squares can be used when the ordinary least squares assumption of constant variance in the errors is violated (which is called heteroscedasticity). The model under consideration is:

$$Y = Xb + \varepsilon$$

where now ε is assumed to be (multivariate) normally distributed with mean vector 0 and nonconstant variance-covariance matrix

$$\begin{pmatrix} \sigma_1^2 & 0 & \ldots & 0 \\ 0 & \sigma_2^2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \sigma_n^2 \end{pmatrix}.$$

Weighted Least Squares

If we define the reciprocal of each variance, σ_i^2, as the weight, $w_i = 1/\sigma_i^2$, then let matrix W be a diagonal matrix containing these weights:

$$\mathbf{W} = \begin{pmatrix} w_1 & 0 & \dots & 0 \\ 0 & w_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & w_n \end{pmatrix}.$$

The weighted least squares estimate is then

$$\begin{aligned} \hat{\beta}_{WLS} &= \arg\min_{\beta} \sum_{i=1}^{n} \varepsilon_i^{*2} \\ &= (\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{W}\mathbf{Y}. \end{aligned}$$

With this setting, we can make a few observations:

- Since each weight is inversely proportional to the error variance, it reflects the information in that observation. So, an observation with small error variance has a large weight since it contains relatively more information than an observation with large error variance (small weight).
- The weights have to be known (or more usually estimated) up to a proportionality constant.

Weighted Least Squares

In practice the structure of W is usually unknown, so we have to perform an ordinary least squares (OLS) regression first. Provided the regression function is appropriate, the i-th squared residual from the OLS fit is an estimate of σ_i^2 and the i-th absolute residual is an estimate of σ_i (which tends to be a more useful estimator in the presence of outliers). The residuals are much too variable to be used directly in estimating the weights, w_i, so instead we use either the squared residuals to estimate a variance function or the absolute residuals to estimate a standard deviation function. We then use this variance or standard deviation function to estimate the weights.

Weighted Least Squares

Some possible variance and standard deviation function estimates include:

- If a residual plot against a predictor exhibits a megaphone shape, then regress the absolute values of the residuals against that predictor. The resulting fitted values of this regression are estimates of σ_i. (And remember $w_i = 1/\sigma_i^2$).
- If a residual plot against the fitted values exhibits a megaphone shape, then regress the absolute values of the residuals against the fitted values. The resulting fitted values of this regression are estimates of σ_i.
- If a residual plot of the squared residuals against a predictor exhibits an upward trend, then regress the squared residuals against that predictor. The resulting fitted values of this regression are estimates of σ_i^2. If a residual plot of the squared residuals against the fitted values exhibits an upward trend, then regress the squared residuals against the fitted values. The resulting fitted values of this regression are estimates of σ_i^2.

Weighted Least Squares

After using one of these methods to estimate the weights, w_i, we then use these weights in estimating a weighted least squares regression model. We consider some examples of this approach in the next section.

Some key points regarding weighted least squares are:

- The difficulty, in practice, is determining estimates of the error variances (or standard deviations).
- Weighted least squares estimates of the coefficients will usually be nearly the same as the "ordinary" unweighted estimates. In cases where they differ substantially, the procedure can be iterated until estimated coefficients stabilize (often in no more than one or two iterations); this is called iteratively reweighted least squares.
- In some cases, the values of the weights may be based on theory or prior research.
- In designed experiments with large numbers of replicates, weights can be estimated directly from sample variances of the response variable at each combination of predictor variables.
- Use of weights will (legitimately) impact the widths of statistical intervals.

Regression with Autoregressive Errors

A multiple (time series) regression model can be written as:

$$y_t = \mathbf{X}_t \beta + \varepsilon_t. \tag{1}$$

The errors (ε_t) may be correlated with each other. In other words, we have autocorrelation or a dependency between the errors.
The errors themselves may follow a simple linear regression model that can be written as

$$\varepsilon_t = \rho \varepsilon_{t-1} + \omega_t. \tag{2}$$

Here, $|\rho| < 1$ is called the autocorrelation parameter and the ω_t term is a new error term that follows the usual assumptions that we make about regression errors: $\omega_t \sim_{iid} N(0, \sigma^2)$. So, this model says that the error at time t is predictable from a fraction of the error at time $t-1$ plus some new perturbation ω_t.

Regression with Autoregressive Errors

The error terms ε_t still have mean 0 and constant variance:

$$\mathrm{E}(\varepsilon_t) = 0 \qquad \mathrm{Var}(\varepsilon_t) = \frac{\sigma^2}{1-\rho^2}$$

However, the covariance (a measure of the relationship between two variables) between adjacent error terms is:

$$\mathrm{Cov}(\varepsilon_t, \varepsilon_{t-1}) = \rho \left(\frac{\sigma^2}{1-\rho^2} \right),$$

which implies the coefficient of correlation (a unitless measure of the relationship between two variables) between adjacent error terms is:

$$\mathrm{Corr}(\varepsilon_t, \varepsilon_{t-1}) = \frac{\mathrm{Cov}(\varepsilon_t, \varepsilon_{t-1})}{\sqrt{\mathrm{Var}(\varepsilon_t)\mathrm{Var}(\varepsilon_{t-1})}} = \rho,$$

We can use partial autocorrelation function (PACF) plots to help us assess appropriate lags for the errors in a regression model with autoregressive errors. Specifically, we first fit a multiple linear regression model to our time series data and store the residuals. Then we can look at a plot of the PACF for the residuals versus the lag. Large sample partial autocorrelations that are significantly different from 0 indicate lagged terms of ε that may be useful predictors of ε_t.

Testing and Remedial Measures for Autocorrelation

The Durbin-Watson Test

If we have a first-order autocorrelation with the errors, then the errors are modeled as

$$\varepsilon_t = \rho \varepsilon_{t-1} + \omega_t,$$

where $|\rho| < 1$ and the $\omega_t \sim_{iid} N(0, \sigma^2)$. If we suspect first-order autocorrelation with the errors, then a formal test does exist regarding the parameter ρ. In particular, the Durbin-Watson test is constructed as:

$$H_0 : \rho = 0$$
$$H_A : \rho \neq 0.$$

So the null hypothesis of $\rho = 0$ means that $\varepsilon_t = \omega_t$, or that the error term in one period is not correlated with the error term in the previous period, while the alternative hypothesis of $\rho \neq 0$ means the error term in one period is either positively or negatively correlated with the error term in the previous period. When the researcher has an indication of the direction of the correlation, then the Durbin-Watson test also accommodates the one-sided alternatives $H_A : \rho < 0$ for negative correlations or $H_A : \rho > 0$ for positive correlations.

Testing and Remedial Measures for Autocorrelation

The test statistic for the Durbin-Watson test on a data set of size n is:

$$D = \frac{\sum_{t=2}^{n}(e_t - e_{t-1})^2}{\sum_{t=1}^{n} e_t^2},$$

where $e_t = y_t - \hat{y}_t$ are the residuals from the ordinary least squares fit. Exact critical values are difficult to obtain, but tables (for certain significance values) can be used to make a decision. The tables provide a lower and upper bound, called d_L and d_U, respectively. In testing for positive autocorrelation, if $D < d_L$ then reject H_0, if D>dU then fail to reject H_0, or if $d_L \leq D \leq d_U$, then the test is inconclusive.

Testing and Remedial Measures for Autocorrelation

Ljung-Box Q Test
The Ljung-Box Q test (sometimes called the Portmanteau test) is used to test whether or not observations over time are random and independent. In particular, for a given k, it tests the following:

$$H_0 : \text{the autocorrelations up to lag } k \text{ are all } 0$$
$$H_A : \text{the autocorrelations of one or more lags differ from } 0.$$

The test statistic is calculated as:

$$Q_k = n(n+2)\sum_{j=1}^{k}\frac{r_j^2}{n-j},$$

which is approximately χ_k^2-distributed.

Testing and Remedial Measures for Autocorrelation

Cochrane-Orcutt Procedure
The Cochrane-Orcutt procedure involves an iterative process (after identifying the need for an AR(1) process) to correct for autocorrelation:
Estimate ρ for

$$\varepsilon_t = \rho\varepsilon_{t-1} + \omega_t$$

by performing a regression through the origin. Call this estimate r.
Transform the variables from the multiple regression model

$$y_t = \beta_0 + \beta_1 x_{t,1} + \ldots + \beta_{p-1} x_{t,p-1} + \varepsilon_t$$

by setting $y_t^* = y_t - ry_{t-1}$ and $x_{t,j}^* = x_{t,j} - rx_{t-1,j}$ for $j = 1,\ldots,p-1$.
Regress y_t^* on the transformed predictors using ordinary least squares to obtain estimates $\hat{\beta}_0^*,\ldots,\hat{\beta}_{p-1}^*$. Look at the error terms for this fit and determine if autocorrelation is still present. If autocorrelation is still present, then iterate this procedure. If it appears to be corrected, then transform the estimates back to their original scale by setting $\hat{\beta}_0 = \hat{\beta}_0^*/(1-r)$ and $\hat{\beta}_j = \hat{\beta}_j^*$ for $j = 1,\ldots,p-1$. Notice that only the intercept parameter requires a transformation. Furthermore, the standard errors of the regression estimates for the original scale can also be obtained by setting $\mathrm{s.e.}(\hat{\beta}_0) = \mathrm{s.e.}(\hat{\beta}_0^*)/(1-r)$ and $\mathrm{s.e.}(\hat{\beta}_j) = \mathrm{s.e.}(\hat{\beta}_j^*)$ for $j = 1,\ldots,p-1$.

Testing and Remedial Measures for Autocorrelation

Forecasting Issues

When calculating forecasts for regression with autoregressive errors, it is important to utilize the modeled error structure as part of our process. For example, for AR(1) errors, $\varepsilon_t = \rho\varepsilon_{t-1} + \omega_t$, our fitted regression equation is

$$\hat{y}_t = b_0 + b_1 x_t,$$

with forecasts of y at time t, denoted F_t, computed as:

$$F_t = \hat{y}_t + e_t = \hat{y}_t + re_{t-1}.$$

So, we can compute forecasts of y at time $t+1$, denoted F_{t+1}, iteratively:

- Compute the fitted value for time period t, $\hat{y}_t = b_0 + b_1 x_t$
- Compute the residual for time period t, $e_t = y_t - \hat{y}_t$.
- Compute the fitted value for time period t+1, $\hat{y}_{t+1} = b_0 + b_1 x_{t+1}$.
- Compute the forecast for time period $t+1$, $F_{t+1} = \hat{y}_{t+1} + re_t$.
- Iterate.

Outline

Framework

Framework for estimating demand in marketing

- There are a large number of competing products (large relative to number of distinct producers)
- Products are differentiated; no two products are exactly alike
- Consumers make "discrete choices": that is, they typically choose only one of the competing products
- Examples: soft drinks, pharmaceutical, potato chips, cars, air travel, etc.

Consumer Behavioral Model

- There are J alternatives in market, indexed by $j = 1, \dots, J$. Each purchase occasion, each consumer i divides her/his income y_i on (at most) one of the alter natives, and on an "outside good":

$$\max_{j,z} U_i(x_j, z) \quad \text{s.t. } p_j + p_z z = y_i$$

  where x_j are characteristics of brand j, and p_j the price, z is quantity of outside good, and p_z its price, outside good (denoted $j = 0$) is the non-purchase of any alternative (that is, spending entire income on other types of goods)
- Substitute in the budget constraint ($z = \frac{y - p_j}{p_z}$) to derive conditional indirect utility functions for each brand:

$$U^*_{ij}(x_j, p_j, p_z, y) = U_i(x_j, \frac{y - p_j}{p_z})$$

  If outside good is bought:

$$U^*_{i0}(p_z, y) = U_i(0, \frac{y - p_j}{p_z})$$

- Consumer chooses the brand yielding the highest conditional indirect utility:

$$\max_j U^*_{ij}(x_j, p_j, p_z, y_i)$$

Econometric Model

- U^*_{ij} usually specified as sum of two parts:

$$U^*_{ij}(x_j, p_j, p_z, y_i) = V_{ij}(x_j, p_j, p_z, y_i) + \varepsilon_{ij}$$

  ε_{ij} observed by agent i, not by researcher.
- Because of the $\varepsilon'_{ij}s$, the product that consumer i chooses is random, from the researcher's point of view. Thus this model is known as the "random utility" model.
- Specific assumptions about the $\varepsilon'_{ij}s$ will determine consumer $i's$ choice probabilities, which corresponds to her/his "demand function". Probability that consumer i buys brand j is

$$D_{ij}(p_1 \dots p_J, p_z, y_) \quad = \quad \text{Prob}\{\varepsilon_{i0}, \dots, \varepsilon_{iJ} : U^*_{ij} > U^*_{ij'} \text{ for } j' \neq j\}$$

Examples

Consumer *i* chooses either to buy the good 1, or buy the outside good.
With only two goods, consumer i chooses 1 if

$$V_{i0} + \varepsilon_{i0} \leq V_{i1} + \varepsilon_{i1}$$
$$V_{i0} - V_{i1} \leq \varepsilon_{i1} - \varepsilon_{i0}$$

Probit: If one assumes

$$\eta_i \int \cdots \int \varepsilon_{i1} - \varepsilon i0 \tilde{N}(0,1)$$

The choice probabilities are

$$P_{i1} \quad = \quad Pr(\eta_i \leq V_{i0} - V_{i1}) = 1 - \Phi(V_{i0} - V_{i1})$$

$\Phi(x)$ is the cumulative distribution function (CDF) of a standard normal random variable, i.e.

$$\Phi(x) \quad = \quad Pr(\eta_i < x)$$

Examples contd.

Logit: If $(\varepsilon_{ij}, j = 0, 1)$ are distributed *i.i.d.* type I extreme value across i, with CDF:

$$F(x) = \exp\left[-\exp\left(-\frac{x-\eta}{\mu}\right)\right] = PR[\varepsilon \leq x]$$

with $\eta = 0.577$ (Euler's constant), and the scale parameter (usually) $\mu = 1$. Then choice probabilities are

$$P_{i1} = \frac{\exp(V_{i1})}{\exp(V_{i0}) + \exp(V_{i1})}$$

For J products, the choice probabilities scale up and take the following multinomial logit form

$$P_{ij} = \frac{\exp(V_{ij})}{\sum_{J'=0,\ldots,J} \exp(V_{ij'})}$$

Problems with Multinomial Logit

- Assumes respondents have the same preferences (or that preferences depend on observable characteristics)
- Restrictive implication of multinomial logit: "Odds ratio" (the ratio of two choice probabilities) between any two brands j, k doesn't depend on the number of alternatives available

$$\frac{P_{ij}}{P_{ik}} \quad = \quad \frac{\exp(V_{ij})/\sum_{j'=1}^{J}\exp(V_{ij'})}{\exp(V_{ik})/\sum_{j'=1}^{J}\exp(V_{ij'})} = \frac{\exp(V_{ij)}}{\exp(V_{ik})}$$

  Implication: invariant to introduction (or elimination) of some alternatives.
  Independence of Irrelevant Alternatives (IIA)
- Since D_{ij} is demand function, IIA implies restrictive substitution patterns:

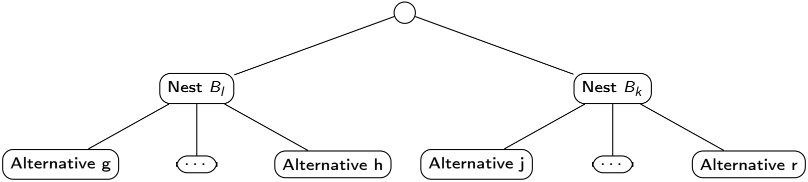
$$\begin{aligned} \varepsilon_{a,c} &= \varepsilon_{b,c} \text{ for all brands } a, b \neq c \\ \text{where} \qquad \varepsilon_{a,c} &= \frac{\partial D_{ia}}{\partial p_c}\frac{p_c}{D_{ia}} \end{aligned}$$

  is the cross-price elasticity of demand between brands c and a.
  If $V_{ij} = \beta_j + \alpha(y_i - p_j)$ then $\varepsilon_{a,c} = \alpha p_c D_c$, for all $c \neq a$. Unrealistic!

Nested Logit Model

In order to overcome the problem of proportional substitution patterns across alternatives coming from IIA in the standard logit model, the nested logit model can be applied.

Here, the set of alternatives *1, . . . , J* is partitioned into K non-overlapping subsets/nests denoted $B_1, B_2, \ldots, B_K$.



The observed component of utility for a person n from product j can be decomposed as:

$$U_{nj} = V_{nj} + \varepsilon_{nj} \equiv W_{nk} + Y_{nj} + \varepsilon_{nj},$$

where W_{nk} - deterministic utility that is constant $\forall j \in B_k$;
Y_{nj} - deterministic utility that varies over all j within B_k

Nested Logit Model

The distribution of the error terms ε_{nj} is a type of generalized extreme value distribution:

$$F(\varepsilon_{nj}) = \exp\left(-\sum_{k=1}^{K} \left(\sum_{j \in B_k} \exp(-\varepsilon_{nj}/\lambda_k) \right)^{\lambda_k} \right)$$

For two alternatives j and r in one nest, ε_{nj} is correlated with ε_{nr}.
The error terms of alternatives from different nests are still uncorrelated (standard logit model).

λ_k reflects the **degree of independence** among the unobserved portions of utility for alternatives in nest B_k ($0 \leqslant \lambda_k \leqslant 1$ with $\lambda_k = 1$ returning a standard logit model) and is usually **fixed** over consumers.

Thus, the choice probability for alternative $j \in B_k$ is

$$P_{nj} = \frac{\exp(V_{nj}/\lambda_k)\left(\sum_{j \in B_k} \exp(V_{ni}/\lambda_k) \right)^{\lambda_k - 1}}{\sum_{l=1}^{K} \left(\sum_{i \in B_l} \exp(V_{ni}/\lambda_l) \right)^{\lambda_k - 1}}$$

where i denotes all other alternatives; l denotes all other nests.

Nested Logit Model

Using the decomposition of utility, the nested logit probability of choosing alternative j can be written as

$$P_{nj} = P_{nj|B_k} \times P_{nB_k}, \text{ with}$$

$$P_{nB_k} = \frac{\exp(W_{nk} + \lambda_k \ln \sum_{i \in B_k} \exp(Y_{ni}/\lambda_k))}{\sum_{l=1}^{K} \exp(W_{nl} + \lambda_l \ln \sum_{i \in B_l} \exp(Y_{ni}/\lambda_l))}$$

$$P_{nj|B_k} = \frac{\exp(Y_{nj}/\lambda_k)}{\sum_{i \in B_k} \exp(Y_{ni}/\lambda_k))}$$

where $P_{nj|B_k}$ - conditional probability of choosing alternative j given its nest B_k is chosen; P_{nB_k} - marginal probability of choosing nest B_k.

$\ln \sum_{i \in B_k} \exp(Y_{ni}/\lambda_k) =:$ "inclusive value"

Source: Train (2009) *Discrete Choice Methods with Simulation*, Chapter 4

The Mixed Logit Model

- The mixed logit model overcomes these limitations by allowing the coefficients in the model to vary across decision makers
- The mixed logit choice probability is given by:

$$P_{ni} \quad = \quad \int \frac{\exp(x\beta'_{ni})}{\sum_{j=1}^{J} exp(x'_{ni}\beta})f(\beta|\theta)d\beta$$

  where $f(\beta|\theta)$ is the density function of β
- Allowing the coefficients to vary implies that we allow for the fact that different decision makers may have different preferences
- It can also be seen that the IIA property no longer holds

The Mixed Logit Model with Panel Data

- If we observe an individual making several choices this can be taken into account in the analysis
- The probability of a particular sequence of choices is given by:

$$S_n = \int \prod_{t=1}^{T} \prod_{j=1}^{J} \left[\frac{\exp(x'_{njt}\beta)}{\sum_{j=1}^{J} \exp(x'_{njt}\beta)} \right]^{y_{njt}} f(\beta|\theta) d\beta$$

where $y_{njt} = 1$ if the individual chose alternative j in choice situation t and 0 otherwise.

Another Problems with Multinomial Logit: Price Endogeneity?

- We made the implicit assumption that the distribution of the $\varepsilon_{ij}'s$ are independent of the prices . this analogous to assuming that prices are exogenous
- If this assumption is violated then we may estimate a positive price coefficient
- Possible explanation for a positive price coefficient: In differentiated product markets, where each product is valued on the basis of its characteristics, brands with highly-desired characteristics (higher quality) may command higher prices. If any of these characteristics are not observed, and hence not controlled for, we can have endogeneity problems. i.e. $E(p\varepsilon) \neq 0$.

Aggregate Logit Model (Berry 1994)

- data are at least market-level observations on prices, quantities and characteristics of the products
- consumers are modeled in a discrete choice framework
- utility of consumer i for product j depends on observed and unobserved characteristics (y, ν) of the consumer and observed and unobserved characteristics of the product (x, ξ)

Include Unobserved Characteristic ξ_j

$$
\begin{aligned}
u_{ijt} &= x_{jt}\beta - \alpha p_{jt} + \xi_{jt} + \varepsilon_{ijt} \\
&= \delta_{jt} + \varepsilon_{ijt}
\end{aligned}
$$

where

$$
\begin{aligned}
\beta &= \text{homogeneous taste parameter for product characteristics} \\
\alpha &= \text{price parameter, invariant across consumers} \\
\xi_{jt} &= \text{mean of consumers' valuations of an unobserved} \\
&\quad\ \text{product characteristic such as product quality}
\end{aligned}
$$

Assumption: ξ, as unobserved quality, is correlated with price and potentially other demand shifters
(e.g. product characteristics) $\rightarrow$ endogeneity problem
All else equal, consumers are more willing to pay for brands for which ξ_{jt} is high

Aggregate Logit Model (Berry 1994)

Make multinomial logit assumption that ε_{ij} is i.i.d type extreme value distributed, across consumers i and brands j

Define choice indicator

$$y_{ij} = \begin{cases} 1 & \text{if } n \text{ chooses brand } j \\ 0 & \text{otherwise} \end{cases}$$

Given these assumptions, choice probabilities take multinomial logit form:

$$Pr(y_{ij} = 1 | \beta, x_{j'}, \xi_{j'}, j' = 1, \ldots, J) \quad = \quad \frac{(\delta_j)}{\sum_{j'=1}^{J} (\delta_{j'})}$$

Aggregate market shares are (for $j = 0, \ldots, J$)

$$\begin{aligned} s_j &= \frac{1}{M}[M \cdot Pr(y_{ij} = 1 | \beta, x_{j'}, \xi_{j'}, j' = 1, \ldots, J)] = \frac{\exp(\delta_j)}{\sum_{j'=1}^{J} \exp(\delta_{j'})} \\ &\equiv \tilde{s}_j(\delta_0, \ldots, \delta_J) \end{aligned}$$

$\tilde{s}$ is the 'predicted share" function, for fixed values of the parameters α and β and the unobservables $\xi_1, \ldots, \xi_J$

Homogeneous Logit with Aggregaqted Data

If we have market share data, s_{jt}

$$s_{jt} = \frac{exp(\delta_{jt})}{\sum_{k=0}^{J} exp(\delta_{kt})}, \quad s_{0t} = \frac{1}{\sum_{k=0}^{J} exp(\delta_{kt})}$$

Normalize $\delta_{0t} = 0$

$$\begin{aligned} \frac{s_{jt}}{s_{0t}} &= exp(\delta_{jt}) \\ ln(s_{jt}) - ln(s_{0t}) &= \delta_{jt} = x_{jt}\beta - \alpha p_{jt} + \xi_{jt} \end{aligned}$$

With homogeneous logit, invert shares to get mean utility δ_{jt}

Homogeneous Logit with Aggregated Data: 2SLS

$$ln(s_{jt}) - ln(s_{0t}) \quad = \quad \delta_{jt} = x_{jt}\beta - \alpha p_{jt} + \xi_{jt}$$

- If no endogeneity, OLS can be used
- Given endogeneity of price, one can instrument for price and use 2SLS

Homogeneous Logit with Aggregated Data: GMM

- Alternatively, Berry (1994) suggest a GMM approach with a set of instruments Z
- Step 1: Compute $\hat{\delta}_{jt} = ln(s_{jt}) - ln(s_{0t})$
  - Let $\xi_{jt}(\theta) = \hat{\delta}_{jt} - x_{jt}\beta - \alpha p_{jt}$ where $\theta = (\beta, \alpha)$
- Step 2: GMM with moment conditions: $E(\xi(\theta)'Z) = 0$
  - $Min \xi(\theta)' ZWZ' \xi(\theta)$ where $W = (E(Z'\xi\xi'Z))^{-1}$
  - then $\theta = (XZWZ'X)^{-1}(X'ZWZ'\delta)$
  - where $X = [x \quad p]$
  - Start with $W = I$ or $W = (Z'Z)^{-1}$ to get initial estimate of θ
    - Re-compute $W = (E(Z'\xi\xi'Z))^{-1}$ for new estimate of θ

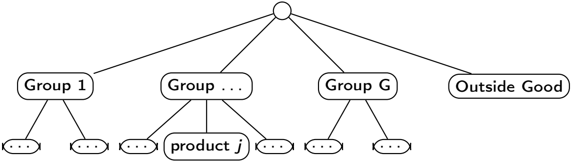
Problems of the Homogeneous Logit Demand System

- Own elasticity: $\eta_j = \frac{\partial s_j}{\partial p_j} \frac{p_j}{s_j} = \alpha p_j (1 - s_j)$
- Cross elasticity: $\eta_{jk} = \frac{\partial s_j}{\partial p_k} \frac{p_k}{s_j} = \alpha p_k s_k$
- Bad properties:
  - Own elasticities proportional to price, so conditional on share more more expensive products tend to be more price elastic
  - Cross-elasticity of product j, w.r.t. price of product k, is dependent only on product k's price and share

Nested Logit with Aggregate Data

Consumer tastes have an extreme value distribution, but consumer tastes are correlated (in a restricted fashion) across product j

$$
\begin{aligned}
G+1 &= \text{exhaustive and mutually exclusive sets, } g=0,1,\ldots G \\
j=0 &= \text{outside good is the only member of group 0} \\
J_g &= \text{set of products in group } g
\end{aligned}
$$



For product $j \in J_g$

$$
\begin{aligned}
u_{ij} &= \delta_j + \zeta_{ig} + (1-\sigma)\varepsilon_{ij} \\
\delta_j &= x_j\beta - \alpha p_j + \xi_j \\
\varepsilon_{ij} &= \text{IID extreme values} \\
\zeta_{ig} &= \text{for a consumer } i\text{, the variable is common to all products in group } g
\end{aligned}
$$

Nested Logit with Aggregate Data

$0 \leq \sigma < 1$ represent the within group correlation of taste among products
$\sigma \to 1$: correlation of tastes between products in the same nest get higher
$\sigma \to 0$: independent tastes for products (standard Logit)

Possible interpretation:
The nested logit model is a random coefficient model involving random coefficients ζ_{ig} only on group specific dummy variables

d_{jg} : dummy variable $= 1$ if $j \in J_g$

$$u_{ij} \quad = \delta_j + \sum_g \left[d_{jg} \zeta_{ig} \right] + (1 - \sigma) \varepsilon_{ij}$$

$\to$ the nested logit allows us to model correlations between groups of similar products
$\to$ correlation pattern depends only on grouping of products that are determined prior to estimation

Nested Logit with Aggregate Data

If product j is in group g

$$
\begin{aligned}
s_{j/g}(\delta,\sigma) &= [e^{\delta_j/(1-\sigma)}]/D_g \\
D_g &\equiv \sum_{j\in J_g} e^{\delta_j/(1-\sigma)} \\
s_g(\delta,\sigma) &= \frac{D_g^{(1-\sigma)}}{\left[\sum_g D_g^{(1-\sigma)}\right]} \\
s_j(\delta,\sigma) &= s_{j/g}(\delta,\sigma)s_g(\delta,\sigma) = \frac{e^{\delta_j/(1-\sigma)}}{D_g^{\sigma}\left[\sum_g D_g^{(1-\sigma)}\right]} \\
s_0(\delta,\sigma) &= \frac{1}{\left[\sum_g D_g^{(1-\sigma)}\right]}
\end{aligned}
$$

Nested Logit with Aggregate Data

Simple analytic expression for mean utility levels

$$\begin{aligned}
\ln(s_j) - \ln(s_0) &= \delta_j/(1-\sigma) - \sigma \ln(D_g) \\
\ln(D_g) &= \frac{\ln(s_g) - \ln(s_0)}{1-\sigma} \\
\delta_j(s,\sigma) &= \ln(s_j) - \sigma \ln(s_{j/g}) - \ln(s_0) \\
\ln(s_j) - \ln(s_0) &= x_j\beta - \alpha p_j + \sigma \ln(s_{j/g}) + \xi_j
\end{aligned}$$

α, β, σ are parameters to estimate
$s_{j/g}$ is a within group share of product j and is an endogenous variable

Source: Berry, Steven T. (1994), *Estimating discrete-choice models of product differentiation*, RAND Journal of Economics, Vol.25, No.3

Nested Logit with Aggregate Data

Nested logit provides more flexible elasticity patterns

- **Own elasticity:** $\eta_j = \frac{\partial s_j}{\partial p_j} \frac{p_j}{s_j} = \frac{1}{1-\sigma}\left[1-\sigma s_{j/g} - (1-\sigma)s_j\right]\alpha p_j$
- **Cross elasticity:** $\eta_{jk} = -\frac{\partial s_j}{\partial p_k} \frac{p_k}{s_j} = -\frac{1}{1-\sigma}\left[\sigma s_{k/g} + (1-\sigma)s_k\right]\alpha p_k$ if j and k are in the same subgroup, else $\eta_{jk} = -\alpha p_k s_k$

Remarks on price elasticities:

- The own price-elasticity for alternatives in a nest are greater than the corresponding own elasticity for alternatives not in a nest; this is consistent with the idea that products in a nest face a more competitive choice context
- The cross-elasticities between pairs of products in a common nest are greater than the corresponding cross-elasticities for products not in the same nest
- $\sigma \to 1$: cross-price elasticities between two products in the same nest get higher
- $\sigma \to 0$: standard Logit cross-price elasticities

Source: Koppelman and Wen (1998), *Alternative nested logit models*, Transpn Res. B, Vol.32, No.5, p.289-298

Nested Logit with Aggregate Data

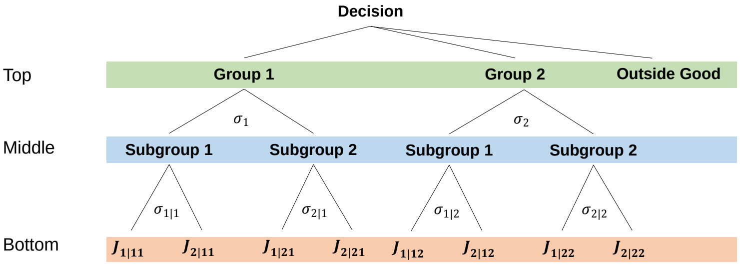
IIA for Nested Logit Model:

- IIA holds within each nest of alternatives but not across nests.
- The elasticity reduces to multinomial logit elasticity, if the alternative does not share a nest with any other alternative.
- There is a proportional substitution across alternatives within a nest but not across nests.
- The ratio of probabilities of two alternatives in the same nest is independent of the characteristics of alternatives in other nests but depends on characteristics of alternatives in the same nest (*"Independence from Irrelevant Nests"*).

Further properties:

- Even if no price endogeneity is present, an instrumental variable estimator is necessary, since additional moments are needed to estimate σ with GMM
- The 2SLS or GMM approach will still work, as long as we have two or more instruments to create enough identifying restrictions: one each for α and σ

Three Levels - Nested Logit Model



Level 1: G groups

Level 2: H subgroups

Level 3: J alternatives

Notation:

σ_1 - parameter for group 1 (σ_g)

$\sigma_{1|1}$ - parameter for subgroup 1, group 1 ($\sigma_{h|g}$)

$J_{1|11}$ - alternative 1 in subgroup 1, group 1 ($J_{j|hg}$)

Three Levels - Nested Logit Model

Utility specification:

$$u_{ij} = \delta_j + \zeta_{ig} + (1-\sigma_g)\varepsilon_{ihg} + (1-\sigma_{h|g})\varepsilon_{ij}$$

where for a consumer i,

$$\begin{aligned}
\zeta_{ig} &=: \text{ the utility from group } g \\
\varepsilon_{ihg} &=: \text{ the utility from a subgroup } h \text{ in group } g \\
\varepsilon_{ij} &=: \text{ preferences for the chosen product } j \\
\sigma_{h|g} &=: \text{ the coefficient of taste correlation for products in one subgroup} \\
\sigma_g &=: \text{ the coefficient of taste correlation for products in one group}
\end{aligned}$$

Source: Verboven, Frank (1996), *International price discrimination in the European car market*, RAND Journal of Economics, Vol.27, No.2

Three Levels - Nested Logit Model

General rules:

- A nested logit model is appropriate when there is correlation for unobserved reasons between the alternatives in each nest but no correlation between alternatives in different nests.
- Choice probabilities in the nested logit model are expressed as a series of logits. The top model describes the choice of nest; the middle model describe the choice of subnest within each nest; and the bottom models describe the choice of alternative within each subnest.
- The top model includes the inclusive value for each nest that represents the expected utility from the subnests within each nest. Similarly, the middle models include an inclusive value for each subnest that is the expected utility from the alternatives within the subnests.
- Each layer of a nesting in a nested logit introduces parameters that represent the degree of correlation among alternatives within the nests, which values increase as one goes down the tree to be consistent with utility maximization, $0 < \sigma_g < \sigma_{h|g} < 1$

Three Levels - Nested Logit Model

The share for a product j, belonging to a subgroup h of a group g, is given by:

$$S_j(\delta, \sigma_{h|g}, \sigma_g) = S_{j|h_g} \times S_{h|g} \times S_g$$

where,

$$S_{j|h_g} = \frac{\exp\left(\frac{\delta_j}{1-\sigma_{h|g}}\right)}{D_h} \qquad S_{h|g} = \frac{D_h^{\left(\frac{1-\sigma_{h|g}}{1-\sigma_g}\right)}}{\sum_{h \in B_g} D_h^{\left(\frac{1-\sigma_{h|g}}{1-\sigma_g}\right)}} \qquad S_g = \frac{\left[\sum_{h \in B_g} D_h^{\left(\frac{1-\sigma_{h|g}}{1-\sigma_g}\right)}\right]^{(1-\sigma_g)}}{\sum_g \left[\sum_{h \in B_g} D_h^{\left(\frac{1-\sigma_{h|g}}{1-\sigma_g}\right)}\right]^{(1-\sigma_g)}}$$

with

$$D_h =: \sum_{l \in B_{h_g}} \exp\left(\delta_l / (1 - \sigma_{h|g})\right)$$

Three Levels - Nested Logit Model

$$
\begin{aligned}
I_{h_g} &= (1-\sigma_{h|g})\ln\left(\sum_{l\in B_{h_g}} \exp\left(\delta_l/(1-\sigma_{h|g})\right)\right) \\
I_g &= (1-\sigma_g)\ln\left(\sum_{h\in B_g} \exp\left(I_{h_g}/(1-\sigma_g)\right)\right)
\end{aligned}
$$

- I_{h_g} is the **"inclusive value"** for each subgroup that is the **expected utility** from the alternatives within the subgroups (included at the "middle model")
- I_g is the **"inclusive value"** for each group that represents the **expected utility** from the subgroups within each group (included at the "top model")
- B_{h_g} is a set of products in subgroup h of group g
- B_g is a set of subgroups in group g
- $(1-\sigma_{h|g})$ is the coefficient of the inclusive value in the middle model
- $(1-\sigma_g)$ is the coefficient of the inclusive value in the top model

Three Levels - Nested Logit Model

With the **outside good** as the only member of group zero and with $\delta_0 = 0$ and $S_{0|h0} = S_{h0|g0} = 1$,

$$S_0(\delta, \sigma_{h|g}, \sigma_g) = \frac{1}{\sum_g \left[\sum_{h \in B_g} \left[\sum_{j \in B_{hg}} \exp\left(\frac{\delta_j}{1 - \sigma_{h|g}} \right) \right]^{\left(\frac{1 - \sigma_{h|g}}{1 - \sigma_g} \right)} \right]^{(1 - \sigma_g)}}$$

It implies that the logit error of the outside good is uncorrelated with those of the products for every individual.

Three Levels - Nested Logit Model

A simple analytic expression for mean utility levels results from

$$\ln(S_j) - \ln(S_0) = \frac{\delta_j}{(1-\sigma_{h|g})} + \frac{\sigma_g - \sigma_{h|g}}{1-\sigma_g} \ln(D_h) - \sigma_g \ln(D_g) \tag{3}$$

where,

$$D_h =: \sum_{l \in B_{h_g}} \exp\left(\delta_l / (1-\sigma_{h|g})\right)$$

$$D_g =: \sum_{h \in B_{h_g}} \left[\sum_{l \in B_{h_g}} \exp\left(\delta_l / (1-\sigma_{h|g})\right)\right]^{\left(\frac{1-\sigma_{h|g}}{1-\sigma_g}\right)}$$

$$\ln(D_g) = \frac{\ln(S_g) - \ln(S_0)}{1-\sigma_g} = \frac{\ln(S_j) - \ln(S_{j|h_g}) - \ln(S_{h|g}) - \ln(S_0)}{1-\sigma_g} \tag{4}$$

$$\ln(D_h) = \frac{\delta_j}{1-\sigma_{h|g}} - \ln(S_{j|h_g}) \tag{5}$$

Three Levels - Nested Logit Model

Substituting (2) and (3) into (1),

$$\delta_j = \ln(S_j) - \ln(S_0) - \sigma_{h|g} \ln(S_{j|h_g}) - \sigma_g \ln(S_{h|g})$$

$\Rightarrow$ Estimation is based on

$$\ln(S_j) - \ln(S_0) \quad = \quad X_j \beta + \alpha P_j + \xi_j + \sigma_{h|g} \ln(S_{j|h_g}) + \sigma_g \ln(S_{h|g})$$

- $\alpha, \beta, \sigma_{h|g}, \sigma_g$ parameters to be estimated with $\alpha < 0$ and $0 < \sigma_g < \sigma_{h|g} < 1$
- $S_{j|h_g}$ is a share of the product j in the subgroup h of group g
- $S_{h|g}$ is a share of the subgroup h in the group g
- S_g is a share of the group g in the set of groups G
- $S_{j|h_g}$ and $S_{h|g}$ are endogenous variables

Three Levels - Nested Logit Model

Interpretation of coefficients for taste correlation:

- $0 < \sigma_g < \sigma_{h|g} < 1$ implies also that the variance of the random utilities is the smallest at the lowest level of the tree, and it cannot decrease as we move from a low to a higher level
- If $\sigma_{h|g} = 0$ and $\sigma_g = 0$, an individual's preferences are uncorrelated across all products, resulting in the simple logit model with symmetric competition
- If $\sigma_{h|g} > 0$ and $\sigma_g = 0$, individual preferences are only correlated across products from the same subgroup
- If $\sigma_{h|g} > 0$ and $\sigma_g > 0$, individual preferences are correlated across products from the same subgroup and across products from different subgroup within the same group
- If $\sigma_g \to \sigma_{h|g}$, preferences are equally correlated across all products from the same group
- If $\sigma_{h|g} \to 1$, products in the same subgroup become perfect substitutes
- If $\sigma_g \to 1$, products in the same group become perfect substitutes

The researcher can constrain the $\sigma_{h|g}$ and/or σ_g to be the same for all (or some) sub-/nests, indicating that the correlation is the same in each of these nests.

Three Levels - Nested Logit Model

Price elasticities:

$$\eta_{jj} = \frac{\partial S_j}{\partial P_j}\frac{P_j}{S_j} = \left[\frac{1}{1-\sigma_{h|g}} - \left(\frac{1}{1-\sigma_{h|g}} - \frac{1}{1-\sigma_g}\right) S_{j|hg} - \left(\frac{\sigma_g}{1-\sigma_g}\right) S_{j|hg} S_{h|g} - S_j\right]\alpha P_j$$

$$\eta_{jk} = \frac{\partial S_k}{\partial P_j}\frac{P_j}{S_k} = -\left[\left(\frac{1}{1-\sigma_{h|g}} - \frac{1}{1-\sigma_g}\right) S_{j|hg} + \left(\frac{\sigma_g}{1-\sigma_g}\right) S_{j|hg} S_{h|g} + S_j\right]\alpha P_j$$

$$\eta_{jk'} = \frac{\partial S_{k'}}{\partial P_j}\frac{P_j}{S_{k'}} = -\left[\left(\frac{\sigma_g}{1-\sigma_g}\right) S_{j|hg} S_{h|g} + S_j\right]\alpha P_j$$

$$\eta_{jk''} = \frac{\partial S_{k''}}{\partial P_j}\frac{P_j}{S_{k''}} = -S_j \alpha P_j$$

where k, k', k'' index products that respectively belong to the same subgroup, to a different subgroup within the same group, and to a different group as product j.

Source: Verboven, Frank (1996), *International price discrimination in the European car market*, RAND Journal of Economics, Vol.27, No.2, p.265

Challenges when Estimating Aggregate Logit or Nested-Logit Demand Models

- Heterogeneity
  - Marketes seek to differentiate products that appeal differentially to different segments to reduce competition and increase margins
- Endogeneity
  - Researchers typically do not know (or have data) about all factors that firms offer and consumers value in a product at a given time or market
  - Firms account for this in setting marketing mix
  - This creates a potential endogeneity problem

The BLP-Model

Why is BLP demand estimation so popular in marketing?

- Berry, Levinsohn and Pakes (1995) address these issues:
  - Estimates differentiated product demand systems with aggregate data
  - Uses discrete choice models with random coefficients (accounts for heterogeneity)
  - Accounts for researcher unobservables that affect consumer choice, and firm's marketing mix choice (accounts for endogeneity)

Utility Function

The BLP-Model:

$$u_{ij} = \boldsymbol{x_j}\beta - \alpha p_j + \varepsilon_{ij}$$

with households (hh) $i = 1, \ldots, I$ and products $j = 1, \ldots, J$

p_j: product price and x_j: product characteristics (e.g. noname brand(yes/no), packaging, promotion)

BLP-Model has random coefficients $\alpha_{\boldsymbol{i}}$ and $\beta_{\boldsymbol{i}}$:

$$u_{ij} = \alpha_i f(y_i - p_j) + \boldsymbol{x_j}\beta_{\boldsymbol{i}} + \xi_j + \varepsilon_{ij},$$

and an additional error term ξ_j that represents the unknown product characteristics and y_i income

Function f($\cdot$): identity function or the logarithm log($\cdot$) of income y_i minus product price p_j

Random Coefficients

Random coefficients are individual- or cluster-specific random effects of variables, that are distributed accordingly to an appropriate probability distribution, e.g. $\beta_{\boldsymbol{i}} \overset{i.i.d.}{\sim} N(\beta, \boldsymbol{\Sigma}_\beta) \quad \forall\ i = 1, \dots, I$

$$\begin{aligned} \begin{pmatrix} \alpha_i \\ \beta_{\boldsymbol{i}} \end{pmatrix} &= \begin{pmatrix} \alpha \\ \beta \end{pmatrix} + \boldsymbol{\Pi} \boldsymbol{D_i} + \boldsymbol{\Sigma} \nu_{\boldsymbol{i}} \\ &= \begin{pmatrix} \alpha \\ \beta \end{pmatrix} + \begin{pmatrix} \boldsymbol{\Pi}_\alpha \\ \boldsymbol{\Pi}_\beta \end{pmatrix} \boldsymbol{D_i} + \begin{pmatrix} \boldsymbol{\Sigma}_\alpha \\ \boldsymbol{\Sigma}_\beta \end{pmatrix} (\nu_{i\alpha}, \nu_{i\beta}) \end{aligned}$$

$$\text{and} \quad \nu_{\boldsymbol{i}} \sim P^*_\nu(\nu), \;\; \boldsymbol{D_i} \sim \hat{P}^*_{\boldsymbol{D}}(\boldsymbol{D})$$

Denote as $\theta_{\mathbf{1}} = (\alpha, \beta)$ and the vector $\theta_{\mathbf{2}}$ represents all parameters from $\boldsymbol{\Pi}$ and $\boldsymbol{\Sigma}$. Further $\theta = (\theta_{\mathbf{1}}, \theta_{\mathbf{2}})$

Random Coefficients

$\boldsymbol{D_i}$: vector of observed variables of hh (e.g. income and age of head of hh)
ν_i: surrogate of all unobserved, but relevant, variables of hh (e.g. state of health, education, ethnicity,...)

Interpretation of parameters:

- α and β: mean value of influence variables
- $\mathbf{\Pi}_\alpha$ and $\mathbf{\Pi}_\beta$: matrix of standard deviation of interaction of product variables and demographic characteristics (e.g. how does income impact preferences)
- $\mathbf{\Sigma}_\alpha$ and $\mathbf{\Sigma}_\beta$: matrix of standard deviation of interaction of product variables and surrogate of unknown demographic characteristics

For simplicity assume: $\boldsymbol{D}$, ε and ν are independent of each other

Random Coefficients

Add index of markets and/or time periods: $t = 1, \ldots, T$

$$
\begin{aligned}
u_{ijt} &= \alpha_i(y_i - p_{jt}) + \boldsymbol{x_{jt}}\beta_{\boldsymbol{i}} + \xi_{jt} + \varepsilon_{ijt} \\
&= \alpha_i y_i - (\alpha + \boldsymbol{\Pi}_\alpha \boldsymbol{D_i} + \boldsymbol{\Sigma}_\alpha \nu_{\boldsymbol{i}\alpha}) p_{jt} + \boldsymbol{x_{jt}}(\beta + \boldsymbol{\Pi}_\beta \boldsymbol{D_i} + \boldsymbol{\Sigma}_\beta \nu_{\boldsymbol{i}\beta}) + \xi_{jt} + \varepsilon_{ijt} \\
&= \alpha_i y_i + (-\alpha p_{jt} + \boldsymbol{x_{jt}}\beta + \xi_{jt}) + (-p_{jt}, \boldsymbol{x_{jt}})(\boldsymbol{\Pi} \boldsymbol{D_i} + \boldsymbol{\Sigma}\nu_{\boldsymbol{i}}) + \varepsilon_{ijt} \\
&= \alpha_i y_i + \delta_{jt} + \mu_{ijt} + \varepsilon_{ijt}
\end{aligned}
$$

where

$$
\begin{aligned}
\delta_{jt} &\equiv (-\alpha p_{jt} + \boldsymbol{x_{jt}}\beta + \xi_{jt}) \\
\mu_{ijt} &\equiv (-p_{jt}, \boldsymbol{x_{jt}})(\boldsymbol{\Pi} \boldsymbol{D_i} + \boldsymbol{\Sigma}\nu_{\boldsymbol{i}}) \\
i &= 1, \ldots, I_t \qquad j = 1, \ldots, J_t \qquad t = 1, \ldots, T
\end{aligned}
$$

δ_{jt}: mean utility for all hh of product (on market t)
μ_{ijt}: mean deviation of utility for hh i

$\rightarrow$ Analytical inversion of δ_{jt} no longer feasible

Market Share

On market t household i acquires product j that provides greatest utility

Two restrictions:

1. $\sum_{k=0}^{J} q_{ikt} = 1$
2. $\sum_{k=0}^{J} p_k \, q_{ikt} \leq y_i$

q_{ijt}: quantity of product j bought form household i on market t

Total amount bought of product j:

$$A_{jt}(\boldsymbol{x_{J_t}}, \boldsymbol{p_{J_t}}, \delta_{\boldsymbol{J_t}}; \theta_2) = \{\boldsymbol{D_i}, \nu_i, \varepsilon_{\boldsymbol{it}} | u_{ijt} \geq u_{ikt}, \forall\ k \ \in \ \{0, 1, ..., J_t\}\}$$

$\boldsymbol{x_{J_t}} = (\boldsymbol{x_{1t}}, ..., \boldsymbol{x_{J_t t}})', \boldsymbol{p_{J_t}} = (p_{1t}, ..., p_{J_t t})'$: vectors of product characteristics and prices
$\delta_{\boldsymbol{J_t}} = (\delta_{1t}, ..., \delta_{J_t t})'$ mean utility for all J_t products on market t

Market Share

$$
\begin{aligned}
s_{jt} &= \int_{A_{jt}} dP^*(\boldsymbol{D}, \varepsilon, \nu) \\
&= \int_{\boldsymbol{D}} \int_{\nu} \frac{exp\left((-\alpha p_{jt} + \boldsymbol{x_{jt}}\beta + \xi_{jt}) + (-p_{jt}, \boldsymbol{x_{jt}})(\boldsymbol{\Pi D_i} + \boldsymbol{\Sigma} \nu_{\boldsymbol{i}})\right)}{1 + \sum_{k=\boldsymbol{1}}^{J} exp\left((-\alpha p_{kt} + \boldsymbol{x_{kt}}\beta + \xi_{jk}) + (-p_{kt}, \boldsymbol{x_{kt}})(\boldsymbol{\Pi D_i} + \boldsymbol{\Sigma} \nu_{\boldsymbol{i}})\right)} d\hat{P}^*_{\boldsymbol{D}}(\boldsymbol{D}) dP^*_{\nu}(\nu) \\
&= \int_{\boldsymbol{D}} \int_{\nu} s_{ijt} \, d\hat{P}^*_{\boldsymbol{D}}(\boldsymbol{D}) dP^*_{\nu}(\nu)
\end{aligned}
$$

Assume: ε is maximum-extreme value distributed

Market Share Approximation

$$
\begin{aligned}
\hat{s}_{jt} &= \frac{1}{ns} \sum_{i=1}^{ns} \frac{exp\left(\delta_{jt} + \mu_{ijt}\right)}{1 + \sum_{m=1}^{J} exp\left(\delta_{mt} + \mu_{imt}\right)} \\
&= \frac{1}{ns} \sum_{i=1}^{ns} \hat{s}_{ijt}
\end{aligned}
$$

With *ns* (number of simulation) random draws of (ν_i, D_i)

Outside Good

Outside good (og): reference category, surrogate for all products, that are not listed.

Utility of outside good: set all influence variables to zero:

$$u_{i0t} = \alpha_i y_i + \beta_{i0} + \xi_{0t} + \varepsilon_{i0t}$$

or

$$u_{i0t} = \alpha_i y_i + \beta_0 + \sigma_0 \nu_{i0} + \xi_{0t} + \varepsilon_{i0t}$$

Utility of og is same for all other alternatives for hh i on market t. Utility can only be identified except scale and level.

Price Elasticity of Demand: Homogeneous Response Parameter

$$\begin{aligned} \eta_{jkt} &= \frac{\partial s_{jt}}{\partial p_{kt}} \frac{p_{kt}}{s_{jt}} \\ &= \begin{cases} -\alpha p_{jt}(1-s_{jt}) & \text{, if } j=k \\ \alpha p_{kt} s_{kt} & \text{, else} \end{cases} \end{aligned}$$

Problems:

- cross-price elasticities only proportional to market share; since individual preferences for product characteristics only accounted for in ε_{ijt}
- for small market shares: $\alpha p_{jt}(1-s_{jt}) \rightarrow$ high price, high elasticity and vice versa

Price Elasticity of Demand: BLP-Model

$$
\begin{aligned}
\eta_{jkt} &= \frac{\frac{\partial s_{jt}}{s_{jt}}}{\frac{\partial p_{kt}}{p_{kt}}} = \frac{\partial s_{jt} p_{kt}}{\partial p_{kt} s_{jt}} \\
&= \begin{cases} -\frac{p_{kt}}{s_{jt}} \int_{\boldsymbol{D}} \int_{\nu} \alpha_i s_{ijt}(1 - s_{ijt}) d\hat{P}^*_{\boldsymbol{D}}(\boldsymbol{D}) dP^*_{\nu}(\nu) & \text{, if } j = k \\ \frac{p_{kt}}{s_{jt}} \int_{\boldsymbol{D}} \int_{\nu} \alpha_i s_{ijt} s_{ikt} d\hat{P}^*_{\boldsymbol{D}}(\boldsymbol{D}) dP^*_{\nu}(\nu) & \text{, else} \end{cases}
\end{aligned}
$$

After integration of ε_{ijt}, elasticities still depend on household i through μ_{ijt}

Price Elasticity of Demand Approximation

$$\eta_{jkt} \approx \begin{cases} -\frac{p_{kt}}{s_{jt}} \frac{1}{ns} \sum_{i=1}^{ns} \alpha_i(\nu_i, D_i)\hat{s}_{ijt}(1-\hat{s}_{ijt}) & \text{, if } j = k \\ \frac{p_{kt}}{s_{jt}} \frac{1}{ns} \sum_{i=1}^{ns} \alpha_i(\nu_i, D_i)\hat{s}_{ijt}\hat{s}_{ijt} & \text{, else} \end{cases}$$

Good Properties:

- Higher priced products more likely purchased by low α_i customers — can have lower elasticities
- Cross elasticities vary across products — e.g. price cut on Honda Civic induces more switching from Toyota Corolla than from BMW 3series

Estimation Results from Berry, Levinsohn & Pakes, 1995

TABLE III
RESULTS WITH LOGIT DEMAND AND MARGINAL COST PRICING
(2217 OBSERVATIONS)

Variable	OLS Logit Demand	IV Logit Demand	OLS ln (*price*) on *w*
Constant	−10.068	−9.273	1.882
	(0.253)	(0.493)	(0.119)
*HP/Weight**	−0.121	1.965	0.520
	(0.277)	(0.909)	(0.035)
Air	−0.035	1.289	0.680
	(0.073)	(0.248)	(0.019)
MP$	0.263	0.052	—
	(0.043)	(0.086)	
*MPG**	—	—	−0.471
			(0.049)
*Size**	2.341	2.355	0.125
	(0.125)	(0.247)	(0.063)
Trend	—	—	0.013
			(0.002)
Price	−0.089	−0.216	—
	(0.004)	(0.123)	
No. Inelastic Demands	1494	22	*n.a.*
(+/−2 *s.e.'s*)	(1429–1617)	(7–101)	
R^2	0.387	*n.a.*	.656

Notes: The standard errors are reported in parentheses.
* The continuous product characteristics—hp/weight, size, and fuel efficiency (*MP$* or *MPG*)—enter the demand equations in levels, but enter the column 3 price regression in natural logs.

Estimation Results from Berry, Levinsohn & Pakes, 1995

TABLE VI

A SAMPLE FROM 1990 OF ESTIMATED OWN- AND CROSS-PRICE SEMI-ELASTICITIES: BASED ON TABLE IV (CRTS) ESTIMATES

	Mazda 323	Nissan Sentra	Ford Escort	Chevy Cavalier	Honda Accord	Ford Taurus	Buick Century	Nissan Maxima	Acura Legend	Lincoln Town Car	Cadillac Seville	Lexus LS400	BMW 735i
323	−125.933	1.518	8.954	9.680	2.185	0.852	0.485	0.056	0.009	0.012	0.002	0.002	0.000
Sentra	0.705	−115.319	8.024	8.435	2.473	0.909	0.516	0.093	0.015	0.019	0.003	0.003	0.000
Escort	0.713	1.375	−106.497	7.570	2.298	0.708	0.445	0.082	0.015	0.015	0.003	0.003	0.000
Cavalier	0.754	1.414	7.406	−110.972	2.291	1.083	0.646	0.087	0.015	0.023	0.004	0.003	0.000
Accord	0.120	0.293	1.590	1.621	−51.637	1.532	0.463	0.310	0.095	0.169	0.034	0.030	0.005
Taurus	0.063	0.144	0.653	1.020	2.041	−43.634	0.335	0.245	0.091	0.291	0.045	0.024	0.006
Century	0.099	0.228	1.146	1.700	1.722	0.937	−66.635	0.773	0.152	0.278	0.039	0.029	0.005
Maxima	0.013	0.046	0.236	0.256	1.293	0.768	0.866	−35.378	0.271	0.579	0.116	0.115	0.020
Legend	0.004	0.014	0.083	0.084	0.736	0.532	0.318	0.506	−21.820	0.775	0.183	0.210	0.043
TownCar	0.002	0.006	0.029	0.046	0.475	0.614	0.210	0.389	0.280	−20.175	0.226	0.168	0.048
Seville	0.001	0.005	0.026	0.035	0.425	0.420	0.131	0.351	0.296	1.011	−16.313	0.263	0.068
LS400	0.001	0.003	0.018	0.019	0.302	0.185	0.079	0.280	0.274	0.606	0.212	−11.199	0.086
735*i*	0.000	0.002	0.009	0.012	0.203	0.176	0.050	0.190	0.223	0.685	0.215	0.336	−9.376

Note: Cell entries i, j, where i indexes row and j column, give the percentage change in market share of i with a \$1000 change in the price of j.

Estimation Results from Berry, Levinsohn & Pakes, 1995

TABLE VII

SUBSTITUTION TO THE OUTSIDE GOOD

Model	Given a price increase, the percentage who substitute to the outside good (as a percentage of all who substitute away.) Logit	BLP
Mazda 323	90.870	27.123
Nissan Sentra	90.843	26.133
Ford Escort	90.592	27.996
Chevy Cavalier	90.585	26.389
Honda Accord	90.458	21.839
Ford Taurus	90.566	25.214
Buick Century	90.777	25.402
Nissan Maxima	90.790	21.738
Acura Legend	90.838	20.786
Lincoln Town Car	90.739	20.309
Cadillac Seville	90.860	16.734
Lexus LS400	90.851	10.090
BMW 735*i*	90.883	10.101

Choosing Instruments

- The BLP estimation procedure is based on instruments Z that satisfy the moment condition $E(\xi_{jt}|Z) = 0$
- IV's are needed for:
  - Moment conditions to identify θ_2 (Heterogeneity): Recall nested logit needed instruments even if price were not endogenous
  - Correcting for price and other marketing mix endogeneity: IV should be correlated with price but not with ξ_{jt}

Choosing Instruments: Product Characteristics

- Own product characteristics (almost all papers)
  - These can just identify the linear parameters associated with these characteristics in mean utility
- Other product characteristics (BLP)
  - Sum of characteristics of other products produced by firm
  - Sum of characteristics of competitor products
- Intuition for instrument validity: other product characteristics have no direct impact on consumer utility for product, but through competition impacts prices
- Key assumption: Characteristics are chosen before ξ_{jt} is known
- Widely used because it is generally available

Choosing Instruments: Cost Shifters

- Characteristics entering cost, but not demand
  - Generally hard to find
  - BLP use scale economies argument to use total production as a cost instrument
- Input factor prices
  - Affects costs and thus prices, but not directly demand
  - Often used to explain price differentials across time, but often does not vary across brands (e.g. market wages)
- If we know production is in different states or countries, we can get brand specific variation in factor prices

Choosing Instruments: Prices in Other Markets

- Prices of products in other markets (Nevo 2001, Hausman 1996)
  - If there are common cost shocks across markets, then price in other markets can be a valid instrument
  - But how to justify no common demand shocks? (e.g. national advertising, seasonality)

Choosing Instruments: Lagged Characteristics

- When current characteristics are simultaneously related to the unobservables, one may motivate use of lagged characteristics similar to the dynamic panel data literature
- Example: Sweeting (Ecta, 2012) assumes an AR(1) process on the unobservables $\xi_{jt} = \rho\xi_{jt-1} + \eta_{jt}$, where $E(\eta_{jt}|x_{t-1}) = 0$ to justify the moment condition $E(\xi_{jt} - \rho\xi_{jt-1}|x_{t-1}) = 0$
- Can lagged prices be a valid instrument?
    - Not if last week's promotions drives this weeks's unobservable (e.g., due to stockpiling, which is unobserved)

Variation in Aggregate Data that allows Identification

- Some of the identification is due to functional form
  - But we also need enough variation in product characteristics and prices and see shares changing in response across different demographics for identification
- Variation across markets
  - Demographics
  - Choice sets (possibly)
- Variation across time
  - Demographics (possibly)
  - Choice sets (possibly)
- To help further identification
  - Add micro data
  - Add supply model

Variation in some Well-Cited Papers

- BLP (1995)
  - National market over time (10 years)
  - Demographics hardly changes, but choice sets (characteristics and prices) change
  - Identification due to changes in shares due to choice sets
- Nevo (2001)
  - Many local markets, over many weeks
  - Demographics different across markets, product characteristics virtually identical, except prices
  - Identification comes from changes in shares across choice sets and across demographics

Caveats about Cross-Sectional Variation Across Markets

- Selection problem: Is ξ_{jm} affected by market characteristics?
  - E.g., less fresh vegetables sold at lower prices in poor neighborhoods
  - may need to model selection
- Can distribution of v_i be systematically different across markets?
  - E.g., if richer cities have better cycling paths for bikes (a market unobservable), distribution of random coefficients for bike characteristics may differ across markets (by income)
  - Allowing for heterogeneity in distributions across markets may be necessary
- Caution: Identification of heterogeneity is tough with aggregate data, so we should not get carried away in demanding more complexity in modeling

Implementation

A straightforward approach to the estimation

Solve $MIN||s(x, p, \delta(x, p, \xi; \theta_1); \theta_2) - S||$

But:

- All parameters enter the minimization in a nonlinear fashion
- Often one has to estimate a large number of parameters (brand and market dummies)
- The error term is not defined as the difference between the observed and predicted market shares, rather it is defined as the structural error ξ_{jt}
- The structural error term and prices are correlated
- The standard nonlinear simultaneous-equations model allows both parameters and variables to enter in a nonlinear way, but requires a separable additive error term (not true in the minimization above)

Implementation

Berry (1994) showed a way to solve this problem

- Express the error term as an explicit function of the parameters of the model and the data
- The error term, ξ_{jt}, only enters the mean utility level, $\delta(\cdot)$
- The mean utility level is a linear function of ξ_{jt}

$\Rightarrow$ To obtain an expression for the error term we need to express the mean utility as a linear function of the variables and parameters of the model

We solve

$$s(\delta_{\cdot t}, \theta_2) \quad = \quad S_{\cdot t} \qquad t = 1, \ldots, T$$

where $s(\cdot)$ are the estimated markets shares and S are the observed market shares

Overview of Steps for Implementation

Four main Overview of steps for parameter estimation in BLP-model:

Step (0) Preparation: draw unknown data, ν, and known demographic characteristics, $\boldsymbol{D}$, from assumed or empirical distribution.

Step (1) Estimation of market share s_{jt} given θ_2 and mean utility δ.

Step (2) Computation of δ, with "Contraction Mapping", so that s_{jt} from step (1) equals empirical market share $\bar{s}_{jt}$.

Step (3) Compute ξ and ω with 2SLS (given θ_2 and δ). Compute GMM form.

Step (4) Search for θ_2, that minimizes GMM form from step (3).

Steps for Implementation: Step (0)

- $\boldsymbol{D}$: draw from empirical distribution
(e.g. ns $=$ 100, different hh in different shop-chains (place), but same for different time periods; changes in demand are NOT explained through changes in demographic characteristics).

- ν: e.g. $\nu \sim P_{\nu}^{*}(\nu) = N(\mathbf{0}, \boldsymbol{I_n})$

Steps for Implementation: Step (1)

Given θ_2 and mean utility δ use the simple approximation of market shares:

$\hat{s}_{jt}(p_{J_t}, x_{J_t}, \delta_{J_t}, P_{ns}; \theta_2) =$

$$\frac{1}{ns}\sum_{i=1}^{ns}\frac{exp\Big(\delta_{jt}+\mu_{ijt}(\theta_2)\Big)}{1+\sum_{m=1}^{J} exp\Big(\delta_{mt}+\mu_{imt}(\theta_2)\Big)}$$

with v_i and D_i, i = 1,...,ns, form step (0)

Notation:
$\boldsymbol{x_{J_t}} = (\boldsymbol{x_{1t}}, ..., \boldsymbol{x_{J_t t}})', \boldsymbol{p_{J_t}} = (p_{1t}, ..., p_{J_t t})'$ and $\delta_{\boldsymbol{J_t}} = (\delta_{1t}, ..., \delta_{J_t t})'$ vectors of product-characteristics, prices and mean utility for products of market t.

Steps for Implementation: Step (2) Contraction Mapping

Given θ_2: Find mean utility δ, so that $\boldsymbol{s}(\boldsymbol{p}, \boldsymbol{x}, \delta, P_{ns}; \theta_2)$ equals empirical market share $\bar{\boldsymbol{s}}$ sufficiently accurate:

$$\delta^{\boldsymbol{h}+1} = \delta^{\boldsymbol{h}} + ln(\bar{\boldsymbol{s}}) - ln(\boldsymbol{s}(\boldsymbol{p}, \boldsymbol{x}, \delta^{\boldsymbol{h}}, P_{ns}; \theta_2))$$

$h = 0, \ldots, H$ for h-th iteration and $h = H$ until a convergence criterium is satisfied

Vector notation of quantities for all markets T:
$\boldsymbol{p} = (\boldsymbol{p}_{\boldsymbol{J}_1}{}', \ldots, \boldsymbol{p}_{\boldsymbol{J}_T}{}')'$
$\boldsymbol{x}$, δ,$\boldsymbol{s}$, $\bar{\boldsymbol{s}}$ analog

Steps for Implementation: Step (3)

Solve for error-terms/ unknown variables:

$$\begin{aligned} \xi_{jt} &= \delta_{jt} - \alpha p_{jt} + \boldsymbol{x_{jt}}\beta \\ \omega_{jt} &= ln(p_{jt} - b(\theta)) - \boldsymbol{w_{jt}}\gamma = ln(mc_j(\theta)) - \boldsymbol{w_{jt}}\gamma \end{aligned}$$

Linear in $\theta_{\mathbf{1}} = (\alpha, \beta)$ and $\gamma \Rightarrow$ use 2SLS-Method

Instrument variable matrix $\boldsymbol{Z}$:

$$E(\boldsymbol{Z}'\psi\boldsymbol{(}\theta,\gamma\boldsymbol{)}) = 0$$

where $\psi(\theta,\gamma) = (\psi_{\boldsymbol{J_1}}, ..., \psi_{\boldsymbol{J_T}})'$ with $\psi_{\boldsymbol{J_t}} = (\xi_{1t}, \omega_{1t}, ..., \xi_{J_t t}, \omega_{J_t t})'$

Steps for Implementation: Step (3)

GMM-Form:

$$(\hat{\theta}, \hat{\gamma}) = argmin_{\theta,\gamma} \;\; \psi(\theta,\gamma)' \boldsymbol{Z} \boldsymbol{W} \boldsymbol{Z}' \psi(\theta,\gamma)$$

for overidentified situation use consistent estimator with optimal weighting matrix $\boldsymbol{W}$: $E(\boldsymbol{Z}'\psi\psi'\boldsymbol{Z})^{-1}$

Different methods, two are:

- homoscedastic error-terms: $\hat{\boldsymbol{W}} = (\boldsymbol{Z}'\boldsymbol{Z})^{-1}$
- two stage estimation:

  (0) use $\hat{\boldsymbol{W}} = (\boldsymbol{Z}'\boldsymbol{Z})^{-1}$, to estimate $(\hat{\theta}, \hat{\gamma})$
  (1) use $(\hat{\theta}, \hat{\gamma})$ to compute $\hat{\boldsymbol{W}} = (\boldsymbol{Z}'\psi(\hat{\theta},\hat{\gamma})\psi(\hat{\theta},\hat{\gamma})'\boldsymbol{Z})^{-1}$
  (2) use weighting matrix $\hat{\boldsymbol{W}}$ in GMM-form to estimate $(\hat{\theta}, \hat{\gamma})$

Steps for Implementation: Step (4)

Minimizing GMM-Form according to $\theta_2 = (\mathbf{\Pi}, \mathbf{\Sigma})$
(Instead of all parameters $\theta = (\theta_1, \theta_2)$ and γ)
(If appropriate use different matrix of influence variables $\boldsymbol{X}_1$ for mean utility δ_{jt} and $\boldsymbol{X}_2$ for household specific mean deviation μ_{ijt})

2SLS-Estimators for other parameters:
Demand side, mean utility:

$$\hat{\theta}_1 = (X_1' Z_d W_d Z_d' X_1)^{-1} X_1' Z_d W_d Z_d' \delta(\hat{\theta}_2)$$

Supply side:

$$\hat{\gamma} = (\mathscr{W}' Z_s W_s Z_s' \mathscr{W})^{-1} \mathscr{W}' Z_s W_s Z_s' mc(\hat{\theta}_2)$$

Notation:
$\mathscr{W}$ matrix of influence variables for marginal costs, indices "d" for supply and "s" indicate (possible) different instruments for demand and supply.

Steps for Implementation: Step (4)

Minimization search algorithm :

- Non-derivative Simplex method (slow, but robust)
- Quasi-Newton method (gradient-method; fast, but sensitive to starting values), needs analytic derivative of GMM

Possible way of minimization:

Demand side only:

- use non-derivative Simplex method to obtain starting values for
- use Quasi-Newton method

Add supply side:

- slows down convergence
- use non-derivative Simplex method to obtain final estimators

Homogeneous Consumers

Expected utility of the category (in the logit model):

$$V_{Buy} = E(max(U_j)) = \ln(\sum_{j \in C} \exp(V_j))$$

where V_j is their expected utility for the j-th brand considered (in set C) and $V_0 = 0$ is the constrained utility not to buy

Estimation of MNL-Model with Homogeneous Consumers

→ Estimation of the MNL-Model with Maximum Likelihood

- N is the sample size and $y_{in} = \begin{cases} 1 & \text{if consumer } n \text{ chooses alternative } i \\ 0 & \text{else} \end{cases}$
- Assumption: Identical choice sets for all individuals: $C_n = C$, $|C| = J$
- Likelihood function: $L = \prod_{n=1}^{N} \prod_{i=1}^{J} P_n(i)^{y_{in}}$
- Loglikelihood function: $\ell = \sum_{n=1}^{N} \sum_{i=1}^{J} y_{in} \log P_n(i)$

where $P_n(i) = \frac{exp(x_i \beta)}{\sum\limits_{j=\mathbf{1}}^{J} exp(x_j \beta)}$

Measures of Goodness of Fit

1. Likelihood-Ratio-Test:

$$-2(\ell(\hat{\beta}_R) - \ell(\hat{\beta}_U))$$

R: restricted model; U: unrestricted model

2. Likelihood-Ratio-Index:

$$\rho^2 = 1 - \frac{l(x)}{l(c)}$$

where $l(x)$ is the Loglikelihood of the chosen model, $l(c)$ is the Loglikelihood of the null model, $l(c) = \sum_{j=1}^{J} N_j \ln(N_j/N)$ where N_j is the share of the brand in the sample and N is the sample size

Measures of Goodness of Fit

3. Corrected Likelihood Ratio Index:

$$\bar{\rho}_2 = 1 - \frac{\ell(x) - \eta}{\ell(c)}$$

where η is an additional parameter

4. $R^2 = 1 - \exp(\ell(c) - \ell(x))^{2/N}$

5. pseudo – $R^2 = \frac{1 - \exp[2/N\ell(c) - \ell(x)]}{1 - \exp 2/N\ell(c)}$

6. Forecast index:

$$f_i = \sum_{j=1}^{J} \left(\frac{N_{jj}}{N_{..}} - (\frac{N.j}{N_{..}})^2 \right)$$

f_i is between 0 and $1 - \sum_{j=1}^{J} (\frac{N.j}{N_{..}})^2$

Calculation of Elasticities

Conditional Logit Model

- For consumer n:

$$\eta_{in} = \frac{\partial P_n(i)}{\partial X_{in}} \frac{X_{in}}{P_n(i)} = \beta' X_{in}(1 - P_n(i))$$

- Aggregated over all consumers:

$$\eta_i = \frac{\partial P(i)}{\partial X_i} \frac{X_i}{P(i)} = \frac{\sum_n \eta_{in}}{\#consumers}$$

Nested Logit Model - Estimation

(i) **Simultaneous estimation by traditional Maximum Likelihood:**

- Log-likelihood function: $LL(\beta) = \sum_{n=1}^{N} \sum_j y_{nj} \ln(P_{nj}) \rightarrow \max_{\beta}$,
  with P_{nj} - nested logit choice probability; y_{nj} - dummy variable for choice j; and β - parameters to estimate
- Estimates are consistent and efficient

(ii) **Sequential Estimation ("bottom-up"):**

Step 1: Estimate the lower model (Choice of alternatives within a nest)
Step 2: Estimate the upper model (Choice of nest), using the estimated coefficients from step 1.

- Estimates are consistent BUT not efficient
- Estimation difficulties:
  - $\rightarrow$ Standard errors of the upper-model parameters are biased downwards
  - $\rightarrow$ Some common parameters appear in several submodels. Estimating various models provides, however, separate estimates.

Nested Logit Model - IIA property

- For alternatives from the same nest ($j \in B_k$ and $r \in B_k$):

$$\frac{P_{nj}}{P_{nr}} = \frac{\exp(V_{nj}/\lambda_k)(\sum_{j \in B_k} \exp(Y_{ni}/\lambda_k))^{\lambda_k - 1}}{\exp(V_{nr}/\lambda_k)(\sum_{r \in B_k} \exp(Y_{ni}/\lambda_k))^{\lambda_k - 1}} = \frac{\exp(V_{nj}/\lambda_k)}{\exp(V_{nr}/\lambda_k)}$$

$\Rightarrow$ IIA holds within each nest as odds are independent of all other alternatives

- For alternatives from different nests ($j \in B_k$ and $g \in B_l$):

$$\frac{P_{nj}}{P_{ng}} = \frac{\exp(V_{nj}/\lambda_k)(\sum_{j \in B_k} \exp(Y_{ni}/\lambda_k))^{\lambda_k - 1}}{\exp(V_{ng}/\lambda_l)(\sum_{g \in B_l} \exp(Y_{ni}/\lambda_l))^{\lambda_l - 1}}$$

$\Rightarrow$ IIA does not hold for alternatives from different nests as odds depend on the attributes of all alternatives in the compared nests

Nested Logit Model - Elasticities

Own-price effects (for $\forall n$):

- For alternatives not in a nest (as a standard Logit):

$$\eta_{jj} = \frac{\partial P_{nj}}{\partial X_{nj}} \frac{X_{nj}}{P_{nj}} = \beta' X_{nj}(1 - P_{nj})$$

- For alternatives in a nest k:

$$\eta_{jj} = \beta' X_{nj}\left((1 - P_{nj}) + (\frac{1 - \lambda_k}{\lambda_k})(1 - P_{nj|B_k})\right)$$

Cross-price effects (for $\forall n$):

- Changes in alternative from a different nest (as a standard Logit):

$$\eta_{jg} = -\beta' X_{ng} P_{ng}$$

- Changes in alternative from the same nest:

$$\eta_{jr} = -\beta' X_{nr}\left(P_{nr} + (\frac{1 - \lambda_k}{\lambda_k}) P_{nr|B_k}\right)$$

Estimation of the Mixed Logit Model

Maximum Simulated Likelihood

- The θ parameters can be estimated by maximizing the simulated log-likelihood function

$$SLL = \sum_{n=1}^{N} \ln \left\{ \frac{1}{R} \sum_{r=1}^{R} \prod_{t=1}^{T} \prod_{j=1}^{J} \left[\frac{\exp(x'_{njt}\beta_n^{[r]})}{\sum_{j=1}^{J} exp(x'_{njt}\beta_n^{[r]}})\right]^{y_{njt}} \right\}$$

  where $\beta_n^{[r]}$ is the r-th draw for individual n from the distribution of β

- In R-software language, the parameters in the bottom panel of the output are the elements of the lower-triangular matrix L, where the covariance matrix for the random coefficients is given by $\Sigma = LL'$

Estimation of the Mixed Logit Model

Individual-level coefficients

- The mixed logit model can be used to estimate individual-level coefficients
- the expected value of β conditional on a given response pattern y_n and a set of alternatives characterized by x_n is given by

$$E[\beta|y_n, x_n] = \frac{\int \beta \prod_{t=1}^{T} \prod_{j=1}^{J} \left[\frac{\exp(x'_{njt}\beta)}{\sum_{j=1}^{J} exp(x'_{njt}\beta}) \right]^{y_{njt}} f(\beta|\theta) d\beta}{\int \prod_{t=1}^{T} \prod_{j=1}^{J} \left[\frac{\exp(x'_{njt}\beta)}{\sum_{j=1}^{J} exp(x'_{njt}\beta}) \right]^{y_{njt}} f(\beta|\theta) d\beta}$$

- Intuitively this can be thought of as the conditional mean of the coefficient distribution for the sub-group of individuals who face the same alternatives and make the same choices

Estimation of the Mixed Logit Model

Compute individual-level coefficients

- Revelt and Train (2000) suggest approximating $E[\beta|y_n, x_n]$ using simulation

$$\hat{\beta}_n = \frac{\frac{1}{R}\sum_{r=1}^{R} \beta_n^{[r]} \prod_{t=1}^{T} \prod_{j=1}^{J} \left[\frac{\exp(x'_{njt}\beta)}{\sum_{j=1}^{J} exp(x'_{njt}\beta_n^{[r]}}) \right]^{y_{njt}}}{\frac{1}{R}\sum_{r=1}^{R} \prod_{t=1}^{T} \prod_{j=1}^{J} \left[\frac{\exp(x'_{njt}\beta)}{\sum_{j=1}^{J} exp(x'_{njt}\beta_n[r]}) \right]^{y_{njt}}}$$

where $\beta_n^{[r]}$ is the r-th draw for individual n from the estimated distribution of β

Estimation of the Mixed Logit Model

Discrete parameter distributions

- So far we have assumed that the distribution of the coefficients in the model is continuous
- Alternatively the coefficients may be discrete, which leads to the **latent class** model
- Each respondent is assumed to belong to a class q, where preferences vary across, but not within, class
- In this case the probability of a particular sequence of choices is given by:

$$S_n = \sum_{q=1}^{Q} H_{nq} \prod_{t=1}^{T} \prod_{j=1}^{J} \left[\frac{\exp(x'_{njt}\beta_q)}{\sum_{j=1}^{J} \exp(x'_{njt}\beta_q)} \right]^{y_{njt}}$$

Estimation of the Mixed Logit Model

Discrete parameter distributions

- The probability of belonging to class q, H_{nq}, is typically specified as

$$H_{nq} = \frac{\exp(Z_n'\gamma_q}{\sum_{q=1}^{Q} \exp(Z_n'\gamma_q}$$

  where $\gamma_q = 0$

- The log-likelihood for this model is

$$S_n = \sum_{n=1}^{N} (ln) \left\{ \sum_{q=1}^{Q} H_{nq} \prod_{t=1}^{T} \prod_{j=1}^{J} \left[\frac{\exp(x_{njt}'\beta_q)}{\sum_{j=1}^{J} \exp(x_{njt}'\beta_q)} \right]^{y_{njt}} \right\}$$

- Determine the number of classes using the AIC, CAIC and/or BIC information criteria after estimating models with different number of classes, say 2-10

Estimation of the Mixed Logit Model

Discrete parameter distributions: Posterior class membership probabilities

- The class membership probability is give by

$$H_{nq} = \frac{\exp(Z_n'\gamma_q}{\sum_{q=1}^{Q}\exp(Z_n'\gamma_q}$$

- This is the **prior** class membership probability
- The **posterior** class membership probability is given by

$$G_{nq} = \frac{H_{nq}\prod_{t=1}^{T}\prod_{j=1}^{J}\left[\frac{\exp(x_{njt}'\beta_q)}{\sum_{j=1}^{J}\exp(x_{njt}'\beta_q)}\right]^{y_{njt}}}{\sum\limits_{q=1}^{Q} H_{nq}\prod_{t=1}^{T}\prod_{j=1}^{J}\left[\frac{\exp(x_{njt}'\beta_q)}{\sum_{j=1}^{J}\exp(x_{njt}'\beta_q)}\right]^{y_{njt}}}$$

- Determine the number of classes using the AIC, CAIC and/or BIC information criteria after estimating models with different number of classes, say 2-10

Estimation of the Mixed Logit Model

Discrete parameter distributions: Individual-level parameters

- The latent class model can be used to estimate individual-level coefficients
- The expected value of β conditional on a given response pattern y_n and a set of alternatives characterised by x_n is given by:

$$E[\beta|y_n, x_n] = \sum_{q=1}^{Q} \beta_q G_{nq}$$

- We can obtain an estimate of β_n by plugging in our estimates of β_q and G_{nq} into this formulae

An alternative to MSL: Bayesian Estimation

- We have seen that the parameters in the mixed logit model can be estimated using maximum simulated likelihood
- Alternatively we can use Bayesian procedures to obtain the estimates
- The results can be interpreted in the same way as if they were maximum likelihood estimates
- The Bayesian approach can therefore be viewed as an alternative algorithm to obtain the estimates

An alternative to MSL: Bayesian estimation

- By Bayes rule, we have that the posterior parameter distribution is given by

$$K(\theta|Y) = \quad = \frac{L(Y|\theta)k(\theta)}{L(Y)}$$

  where $L(Y|\theta)$ is the likelihood function (the probability of observing the data given the parameters), $k(\theta)$ is the prior parameter distribution and $L(Y) = \int L(Y|\theta)k(\theta)d(\theta)$
- The mean of the posterior distribution can be shown to have the same asymptotic properties as the maximum likelihood estimator
- The Bayesian approach involves taking many draws from the posterior distribution and averaging these draws

An alternative to MSL: Bayesian Estimation

- Train (2009) describes an algorithm for taking draws from the posterior distribution of the coefficients in a mixed logit with normally distributed coefficients
- The mean and variance of β as well as the individual-level parameters, β_n, are treated as parameters to be estimated
- The values from the algorithm converges to draws from the posterior distribution
- The iterations prior to convergence are called the "burn-in"
- Even after convergence the draws are correlated, so only a portion of the draws are kept

An Example with Scanner Panel Data

Product category: Cola sales in 2 Scanner Panel markets in the US (Eau Claire and Pittsfiled)

- IRI Academic Data Set on Cola Purchases (Broonenberg et al. 2001)
- 403 households with cola purchases in 2002 until 2006
- 15481 cola purchases of those households (min 7 per year)
- 53916 purchase trips of those households
- Products/Brands: Classic Coke, Diet Coke, Pepsi, Diet Pepsi
- Sales in 144oz (12 cans of 12oz) and 288oz (24 cans of 12oz) measurement units
- Observed variables: Price, Display, Feature

An Example with Scanner Panel Data

	Choices	Within Share	Share	Mean Price	Mean Display(Major)	Mean Display(Minor)	Mean Feature
COKE CLASSIC	4595	29.682%	8.523%	4.078	0.565	0.260	0.320
DIET COKE	3555	22.964%	6.594%	4.057	0.554	0.258	0.273
PEPSI	4544	29.353%	7.138%	4.018	0.552	0.282	0.349
DIET PEPSI	2787	18.002%	5.169%	3.997	0.548	0.272	0.3087
No Choice	38435		71.287%				

Estimation Results of the Conditional Logit Model

```
mlogit(formula = decision ~ 0 + b1 + b2 + b3 + b4 + logprice +
    display_minor + display_major + feature, data = data_ml_ccc,
    method = "nr", print.level = 0)

Frequencies of alternatives:
       1        2        3        4        5
0.085225 0.065936 0.051692 0.084279 0.712868

nr method
6 iterations, 0h:0m:9s
g'(-H)^-1g = 0.00389
successive function values within tolerance limits

Coefficients :
               Estimate Std. Error  t-value  Pr(>|t|)
b1            -0.347124   0.080495  -4.3124 1.615e-05 ***
b2            -0.592556   0.080629  -7.3492 1.994e-13 ***
b3            -0.877229   0.079650 -11.0136 < 2.2e-16 ***
b4            -0.372595   0.080238  -4.6436 3.423e-06 ***
logprice      -1.659451   0.047636 -34.8359 < 2.2e-16 ***
display_minor  0.633088   0.032954  19.2111 < 2.2e-16 ***
display_major  0.692561   0.030123  22.9914 < 2.2e-16 ***
feature        0.301805   0.022616  13.3449 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '' 1

Log-Likelihood: -51323
```

Estimation Results of the Conditional Logit Model

```
Elasticities for the first choice set
            1.1         1.2         1.3         1.4         1.5
1.1 -1.59488014  0.06457039  0.06457039  0.06457039  0.06457039
1.2  0.05051775 -1.60893278  0.05051775  0.05051775  0.05051775
1.3  0.07596039  0.07596039 -1.58349014  0.07596039  0.07596039
1.4  0.12458416  0.12458416  0.12458416 -1.53486638  0.12458416
1.5  1.34381784  1.34381784  1.34381784  1.34381784 -0.31563269


Average elasticities over all 53916 choice set
            1.1         1.2        1.3         1.4         1.5
1.1 -1.51802357  0.14142696  0.1414270  0.14142696  0.14142696
1.2  0.10941736 -1.55003317  0.1094174  0.10941736  0.10941736
1.3  0.08577952  0.08577952 -1.5736710  0.08577952  0.08577952
1.4  0.13985726  0.13985726  0.1398573 -1.51959327  0.13985726
1.5  1.18296943  1.18296943  1.1829694  1.18296943 -0.47648111


Elasticities based on average choice set (over all 53916 choice sets)
            1.1         1.2        1.3         1.4         1.5
1.1 -1.53007035  0.12938019  0.1293802  0.12938019  0.12938019
1.2  0.10005751 -1.55939302  0.1000575  0.10005751  0.10005751
1.3  0.07679803  0.07679803 -1.5826525  0.07679803  0.07679803
1.4  0.12664478  0.12664478  0.1266448 -1.53280575  0.12664478
1.5  1.22657002  1.22657002  1.2265700  1.22657002 -0.43288052
```

Estimation Results of the Mixed Logit Model

```
mlogit(formula = decision ~ 0 + b1 + b2 + b3 + b4 + logprice +
    display_minor + display_major + feature, data = data_ml_ccc,
    rpar = mnl_ccc.rpar, R = 250, correlation = FALSE, panel = TRUE,
    print.level = 2)
Coefficients :
                   Estimate Std. Error  t-value  Pr(>|t|)
b1                -1.0244324  0.0961748 -10.6518 < 2.2e-16 ***
b2                -2.1904201  0.1037805 -21.1063 < 2.2e-16 ***
b3                -1.8859695  0.1011255 -18.6498 < 2.2e-16 ***
b4                -1.3504151  0.0964817 -13.9966 < 2.2e-16 ***
logprice          -1.9417732  0.0554235 -35.0352 < 2.2e-16 ***
display_minor      0.5520318  0.0364719  15.1358 < 2.2e-16 ***
display_major      0.7725219  0.0331286  23.3188 < 2.2e-16 ***
feature            0.4049725  0.0257337  15.7371 < 2.2e-16 ***
sd.b1              1.5130605  0.0267444  56.5748 < 2.2e-16 ***
sd.b2              2.6721659  0.0435812  61.3147 < 2.2e-16 ***
sd.b3              2.5756806  0.0430208  59.8706 < 2.2e-16 ***
sd.b4              2.4479237  0.0363702  67.3058 < 2.2e-16 ***
sd.logprice        0.2385191  0.0088327  27.0041 < 2.2e-16 ***
sd.display_minor   0.1498420  0.0246836   6.0705 1.275e-09 ***
sd.display_major  -0.1197110  0.0185442  -6.4554 1.079e-10 ***
sd.feature         0.2514937  0.0240506  10.4569 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '' 1

Log-Likelihood: -37443
```

Estimation Results of the Mixed Logit Model

```
random coefficients
              Min.    1st Qu.     Median       Mean      3rd Qu. Max.
b1            -Inf -2.0449762 -1.0244324 -1.0244324 -0.003888654  Inf
b2            -Inf -3.9927686 -2.1904201 -2.1904201 -0.388071578  Inf
b3            -Inf -3.6232396 -1.8859695 -1.8859695 -0.148699330  Inf
b4            -Inf -3.0015145 -1.3504151 -1.3504151  0.300684369  Inf
logprice      -Inf -2.1026519 -1.9417732 -1.9417732 -1.780894480  Inf
display_minor -Inf  0.4509649  0.5520318  0.5520318  0.653098652  Inf
display_major -Inf  0.6917781  0.7725219  0.7725219  0.853265742  Inf
feature       -Inf  0.2353426  0.4049725  0.4049725  0.574602469  Inf
```

Estimation Results of the Mixed Logit Model

```
Elasticities for the first choice set
            1.1         1.2         1.3         1.4         1.5
1.1 -1.55891657  0.04798127  0.03757386  0.03720022  0.05461944
1.2  0.06107126 -1.29835720  0.03864306  0.05556371  0.05624074
1.3  0.09530535  0.07700837 -1.08033150  0.08523983  0.09705612
1.4  0.10627537  0.12471343  0.09600595 -1.14395910  0.11627805
1.5  1.29626459  1.04865413  0.90810863  0.96595534 -0.32419435


Average elasticities over all 53916 choice set
            1.1         1.2         1.3         1.4         1.5
1.1 -1.44478267  0.08423999  0.07201369  0.07826874  0.11122955
1.2  0.08972462 -1.17566675  0.06330548  0.08028362  0.09082809
1.3  0.09332622  0.07939351 -1.07055878  0.08884693  0.10342279
1.4  0.10894705  0.11134212  0.09989192 -1.14415780  0.12319616
1.5  1.15278479  0.90069112  0.83534770  0.89675851 -0.42867659


Elasticities based on average choice set (over all 53916 choice sets)
            1.1         1.2         1.3         1.4         1.5
1.1 -1.45863089  0.07894508  0.06682018  0.07095215  0.10255147
1.2  0.08726218 -1.17439942  0.06219110  0.07922628  0.08800012
1.3  0.08915107  0.07506651 -1.07975814  0.08645962  0.09931639
1.4  0.10680050  0.10788870  0.09754434 -1.14759377  0.11963406
1.5  1.17541714  0.91249912  0.85320252  0.91095572 -0.40950204
```

Estimation Results of the Mixed Logit Model

```
Coefficients (with full covariance matrix):

                     Estimate Std. Error  t-value  Pr(>|t|)
b1                  -1.177823   0.098206 -11.9934 < 2.2e-16 ***
b2                  -2.135662   0.104253 -20.4853 < 2.2e-16 ***
b3                  -2.578595   0.107525 -23.9813 < 2.2e-16 ***
b4                  -1.412417   0.098785 -14.2978 < 2.2e-16 ***
logprice            -1.852173   0.056894 -32.5549 < 2.2e-16 ***
display_minor        0.626875   0.037081  16.9057 < 2.2e-16 ***
display_major        0.792902   0.033996  23.3235 < 2.2e-16 ***
feature              0.380461   0.027003  14.0897 < 2.2e-16 ***
b1.b1                1.907687   0.099247  19.2216 < 2.2e-16 ***
b1.b2                0.695163   0.097603   7.1223 1.061e-12 ***
b1.b3               -0.069235   0.097651  -0.7090 0.4783248
b1.b4                0.726544   0.098159   7.4017 1.343e-13 ***
b1.logprice         -0.342847   0.058746  -5.8361 5.345e-09 ***
b1.display_minor     0.031848   0.038965   0.8173 0.4137358
b1.display_major     0.080622   0.035623   2.2632 0.0236220 *
b1.feature          -0.041836   0.028516  -1.4671 0.1423486
b2.b2                2.524224   0.041022  61.5331 < 2.2e-16 ***
b2.b3                1.351809   0.036215  37.3277 < 2.2e-16 ***
b2.b4               -0.182994   0.028159  -6.4986 8.106e-11 ***
b2.logprice         -0.304476   0.018102 -16.8203 < 2.2e-16 ***
b2.display_minor    -0.031674   0.029991  -1.0561 0.2909104
b2.display_major     0.004742   0.027550   0.1721 0.8633437
b2.feature          -0.019405   0.024171  -0.8028 0.4220837
```

Estimation Results of the Mixed Logit Model

```
b3.b3                        2.579813   0.042943  60.0751 < 2.2e-16 ***
b3.b4                        1.157909   0.028595  40.4930 < 2.2e-16 ***
b3.logprice                 -0.223993   0.016990 -13.1842 < 2.2e-16 ***
b3.display_minor            -0.117355   0.030861  -3.8027 0.0001431 ***
b3.display_major            -0.041711   0.028044  -1.4874 0.1369222
b3.feature                  -0.131909   0.024423  -5.4009 6.630e-08 ***
b4.b4                        2.057169   0.033437  61.5246 < 2.2e-16 ***
b4.logprice                 -0.221727   0.016951 -13.0807 < 2.2e-16 ***
b4.display_minor             0.193460   0.032022   6.0414 1.528e-09 ***
b4.display_major             0.110449   0.029000   3.8085 0.0001398 ***
b4.feature                  -0.061958   0.024685  -2.5099 0.0120755 *
logprice.logprice            0.481277   0.020396  23.5964 < 2.2e-16 ***
logprice.display_minor      -0.527299   0.039171 -13.4615 < 2.2e-16 ***
logprice.display_major      -0.462332   0.036050 -12.8249 < 2.2e-16 ***
logprice.feature            -0.141964   0.028596  -4.9645 6.887e-07 ***
display_minor.display_minor  0.275927   0.025703  10.7354 < 2.2e-16 ***
display_minor.display_major  0.254108   0.019876  12.7848 < 2.2e-16 ***
display_minor.feature       -0.084993   0.027668  -3.0719 0.0021268 **
display_major.display_major  0.098467   0.018670   5.2741 1.334e-07 ***
display_major.feature        0.060846   0.025237   2.4110 0.0159093 *
feature.feature              0.153006   0.025910   5.9053 3.519e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '' 1

Log-Likelihood: -37179
```

Estimation Results of the Mixed Logit Model

```
random coefficients
              Min.    1st Qu.     Median       Mean    3rd Qu. Max.
b1            -Inf -2.4645381 -1.1778230 -1.1778230  0.1088921  Inf
b2            -Inf -3.9016088 -2.1356616 -2.1356616 -0.3697143  Inf
b3            -Inf -4.5436209 -2.5785952 -2.5785952 -0.6135695  Inf
b4            -Inf -3.0829263 -1.4124167 -1.4124167  0.2580928  Inf
logprice      -Inf -2.3483759 -1.8521726 -1.8521726 -1.3559693  Inf
display_minor -Inf  0.1963650  0.6268754  0.6268754  1.0573858  Inf
display_major -Inf  0.4182847  0.7929018  0.7929018  1.1675189  Inf
feature       -Inf  0.1922600  0.3804607  0.3804607  0.5686615  Inf
```

Estimation Results of the Mixed Logit Model

```
Elasticities for the first choice set
            1.1         1.2         1.3         1.4         1.5
1.1 -1.25588051  0.06813343  0.01866049  0.04451648  0.06176563
1.2  0.04854148 -1.51802328  0.06026216  0.01698245  0.05210281
1.3  0.02240130  0.10154115 -1.30333915  0.07138030  0.07490158
1.4  0.08426137  0.04511857  0.11254752 -1.24982987  0.11863941
1.5  1.10067636  1.30323013  1.11186897  1.11695064 -0.30740943


Average elasticities over all 53916 choice set
            1.1         1.2         1.3         1.4         1.5
1.1 -1.27387761  0.11835052  0.03800397  0.09870358  0.12289335
1.2  0.08661643 -1.45937147  0.11210713  0.03854078  0.09818842
1.3  0.02688004  0.10783238 -1.26216616  0.07827460  0.08108696
1.4  0.10653286  0.05936035  0.12120755 -1.21187101  0.12556068
1.5  1.05384828  1.17382821  0.99084751  0.99635205 -0.42772942


Elasticities based on average choice set (over all 53916 choice sets)
            1.1         1.2         1.3         1.4         1.5
1.1 -1.31543644  0.10905081  0.03433517  0.08993580  0.11486125
1.2  0.08002586 -1.50453865  0.10900275  0.03624891  0.09357044
1.3  0.02518326  0.10894538 -1.28095967  0.07417823  0.07776794
1.4  0.10151399  0.05575539  0.11415553 -1.23287965  0.12073819
1.5  1.10871332  1.23078708  1.02346621  1.03251671 -0.40693782
```