Week 13: *Class Review*

EMSE 6035: Marketing Analytics for Design Decisions

John Paul Helveston

November 29, 2023

Analysis

1. Clean data
2. Modeling
   - Simple logit
   - Mixed logit
   - One sub-group model
3. Analysis
   - WTP for key features
   - Market simulation
   - Sensitivity analysis

Report

1. Introduction
2. Survey Design
3. Data Analysis
4. Results (plots / text)
5. Recommendations

Final Presentation

- In class, 12/13 (5:30 - 7:00)
- 10 minutes (strict)
- Slides due on Blackboard by midnight on 12/12

Week 13: *Class Review*

1. Exam Review

BREAK

2. Sensitivity Analysis

Week 13: *Class Review*

1. Exam Review

BREAK

2. Sensitivity Analysis

Things I'm covering

- Data wrangling in R
- Utility models
- Maximum likelihood estimation
- Optimization
- Uncertainty
- Design of experiment
- WTP
- Market simulations
- Sub-group models
- Using R for all of the above
  (e.g., estimating models with `logitr`)

Things I'm **not** covering

- formr.org
- Mixed logit

Data wrangling in R

Steps to importing external data files

1. Create a path to the data

```r
library(here)
path_to_data <- here('data', 'data.csv')
path_to_data
```

```
#> [1] "/Users/jhelvy/gh/teaching/MADD/2023-Fall/class/13-class-review/data/data.csv"
```

2. Import the data

```r
library(tidyverse)
data <- read_csv(path_to_data)
```

Steps to importing external data files

```r
library(tidyverse)

data <- read_csv(here::here('data', 'data.csv'))
```

The main `dplyr` "verbs"

"Verb"	What it does
`select()`	Select columns by name
`filter()`	Keep rows that match criteria
`arrange()`	Sort rows based on column(s)
`mutate()`	Create new columns

Example data frame

```r
beatles <- tibble(
    firstName   = c("John", "Paul", "Ringo", "George"),
    lastName    = c("Lennon", "McCartney", "Starr", "Harrison"),
    instrument  = c("guitar", "bass", "drums", "guitar"),
    yearOfBirth = c(1940, 1942, 1940, 1943),
    deceased    = c(TRUE, FALSE, FALSE, TRUE)
)

beatles
```

```
#> # A tibble: 4 × 5
#>   firstName lastName  instrument yearOfBirth deceased
#>   <chr>     <chr>     <chr>            <dbl> <lgl>
#> 1 John      Lennon    guitar            1940 TRUE
#> 2 Paul      McCartney bass              1942 FALSE
#> 3 Ringo     Starr     drums             1940 FALSE
#> 4 George    Harrison  guitar            1943 TRUE
```

`filter()` and `select()`:

Get the **first & last name** of members born after 1941 & are still living

```
beatles %>%
  filter(yearOfBirth > 1941, deceased == FALSE) %>%
  select(firstName, lastName)
```

```
#> # A tibble: 1 × 2
#>   firstName lastName
#>   <chr>     <chr>
#> 1 Paul      McCartney
```

Create new variables with `mutate()`

Use the `yearOfBirth` variable to compute the age of each band member

```r
beatles %>%
    mutate(age = 2022 - yearOfBirth) %>%
    arrange(age)
```

```
#> # A tibble: 4 × 6
#>   firstName lastName  instrument yearOfBirth deceased   age
#>   <chr>     <chr>     <chr>            <dbl> <lgl>    <dbl>
#> 1 George    Harrison  guitar            1943 TRUE        79
#> 2 Paul      McCartney bass              1942 FALSE       80
#> 3 John      Lennon    guitar            1940 TRUE        82
#> 4 Ringo     Starr     drums             1940 FALSE       82
```

Handling if/else conditions

`ifelse(<condition>, <if TRUE>, <else>)`

```
beatles %>%
    mutate(playsGuitar = ifelse(instrument == "guitar", TRUE, FALSE))
```

```
#> # A tibble: 4 × 6
#>   firstName lastName  instrument yearOfBirth deceased playsGuitar
#>   <chr>     <chr>     <chr>            <dbl> <lgl>    <lgl>
#> 1 John      Lennon    guitar            1940 TRUE     TRUE
#> 2 Paul      McCartney bass              1942 FALSE    FALSE
#> 3 Ringo     Starr     drums             1940 FALSE    FALSE
#> 4 George    Harrison  guitar            1943 TRUE     TRUE
```

Utility models

Random utility model

The utility for alternative j is

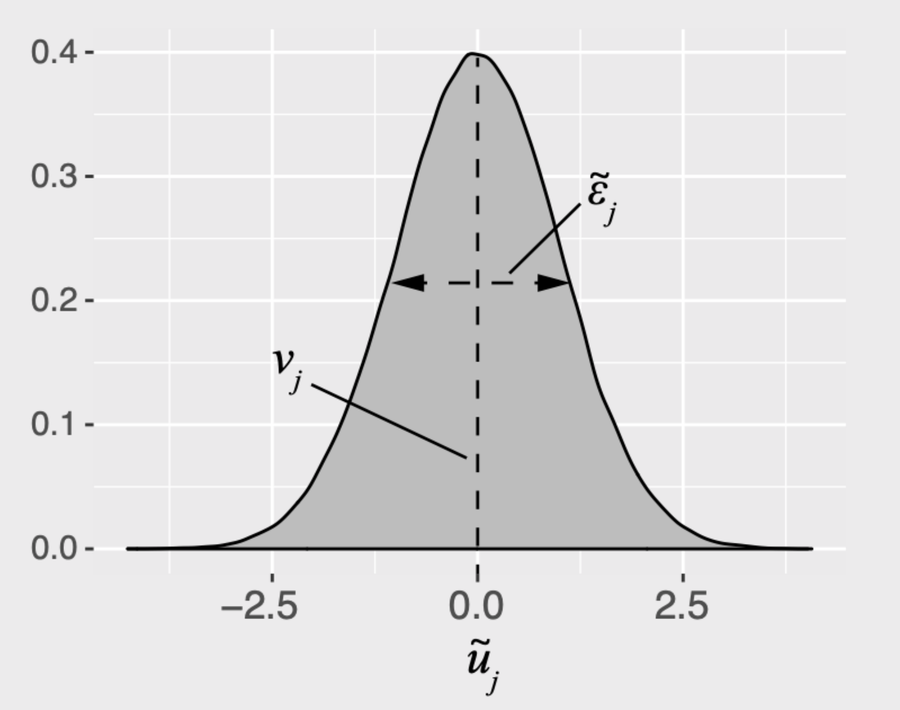
$$\tilde{u}_j = v_j + \tilde{\varepsilon}_j$$

v_j = Things we observe (non-random variables)

$\tilde{\varepsilon}_j$ = Things we *don't* observe (random variable)

Logit model: Assume that $\tilde{\varepsilon}_j$ ~ Gumbel Distribution

$$\tilde{u}_j = v_j + \tilde{\varepsilon}_j$$



Probability of choosing alternative j:

$$P_j = \frac{e^{v_j}}{\sum_k e^{v_k}}$$

Notation Convention

Continuous: x_j

$$u_j = \beta_1 x_j^{\text{price}} + \dots$$

```
#>   price
#> 1     1
#> 2     2
#> 3     3
```

Discrete: δ_j

$$u_j = \beta_1 \delta_j^{\text{ford}} + \beta_2 \delta_j^{\text{gm}} \dots$$

```
#>   brand brand_BMW brand_Ford brand_GM
#> 1  Ford         0          1        0
#> 2    GM         0          0        1
#> 3   BMW         1          0        0
```

Dummy-coded variables

Dummy coding: 1 = "Yes", 0 = "No"

Data frame with one variable: *brand*

```r
data <- data.frame(
    brand = c("Ford", "GM", "BMW"))

data
```

```
#>   brand
#> 1  Ford
#> 2    GM
#> 3   BMW
```

Add dummy columns for each brand

```r
library(fastDummies)

dummy_cols(data, "brand")
```

```
#>   brand brand_BMW brand_Ford brand_GM
#> 1  Ford         0          1        0
#> 2    GM         0          0        1
#> 3   BMW         1          0        0
```

Modeling *continuous* variable

$$v_j = \beta_1 x^{\text{price}}$$

```r
model <- logitr(
    data   = data,
    choice = "choice",
    obsID  = "obsID",
    pars   = "price"
)
```

Coef.	Interpretation
β1	how utility changes with increasing *price*

Modeling *discrete* variable

$$v_j = \beta_1 \delta_j^{\text{ford}} + \beta_2 \delta_j^{\text{gm}}$$

```r
model <- logitr(
    data   = data,
    choice = "choice",
    obsID  = "obsID",
    pars   = c("brand_Ford", "brand_GM")
)
```

Reference level: *BMW*

Coef.	Interpretation
β1	utility for *Ford* relative to *BMW*
β2	utility for *GM* relative to *BMW*

Estimating utility models

1. Open `logitr-cars.Rproj`
2. Open `code/3.1-model-mnl.R`

mnl_dummy

All discrete (dummy-code) variables

```
pars = c(
  "price_20", "price_25",
  "fuelEconomy_25", "fuelEconomy_30",
  "accelTime_7", "accelTime_8",
  "powertrain_Electric")
```

Reference Levels:

- Price: 15
- Fuel Economy: 20
- Accel. Time: 6
- Powertrain: "Gasoline"

mnl_linear

All continuous (linear), except for `powertrain_Electric`

```
pars = c(
  'price', 'fuelEconomy', 'accelTime',
  'powertrain_Electric')
```

Reference Levels:

- Powertrain: "Gasoline"

Practice Question 1

Let's say our utility function is:

$$v_j = \beta_1 x_j^{\text{price}} + \beta_2 x_j^{\text{cacao}} + \beta_3 \delta_j^{\text{hershey}} + \beta_4 \delta_j^{\text{lindt}}$$

And we estimate the following coefficients:

Parameter	Coefficient
β_1	-0.1
β_2	0.1
β_3	-2.0
β_4	-0.1

What are the expected probabilities of choosing each of these bars using a logit model?

Attribute	Bar 1	Bar 2	Bar 3
Price	\$1.20	\$1.50	\$3.00
% Cacao	10%	60%	80%
Brand	Hershey	Lindt	Ghirardelli

Maximum likelihood estimation

Maximum likelihood estimation

$$\tilde{u}_j = \boldsymbol{\beta}'\mathbf{x}_j + \tilde{\varepsilon}_j$$

$$= \boxed{\beta_1} x_{j1} + \boxed{\beta_2} x_{j2} + \ldots + \tilde{\varepsilon}_j$$

Weights that denote the *relative* value of attributes $x_{j1}, x_{j2}, \ldots$

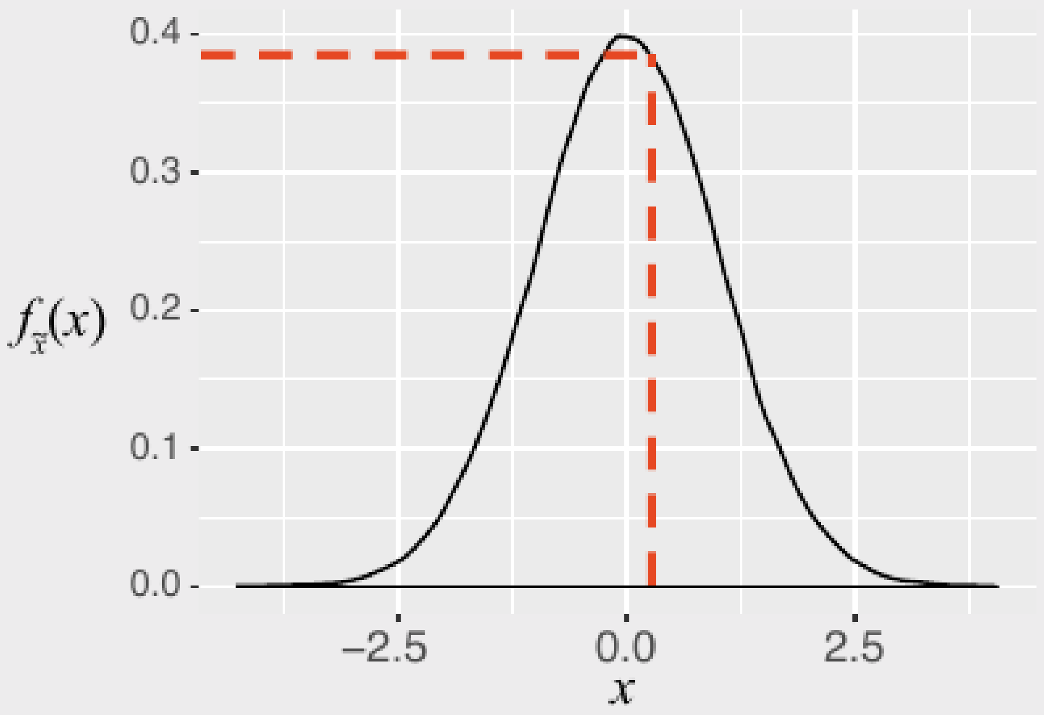
Estimate β_1 , β_2 , … , by minimizing the negative log-likelihood function:

$$\text{minimize} - \ln(\mathcal{L}) = -\sum_{j=1}^{J} y_j \ln\left[P_j(\boldsymbol{\beta}|\mathbf{x})\right]$$

with respect to $\boldsymbol{\beta}$

$y_j = 1$ if alternative j was chosen
$y_j = 0$ if alternative j was not chosen

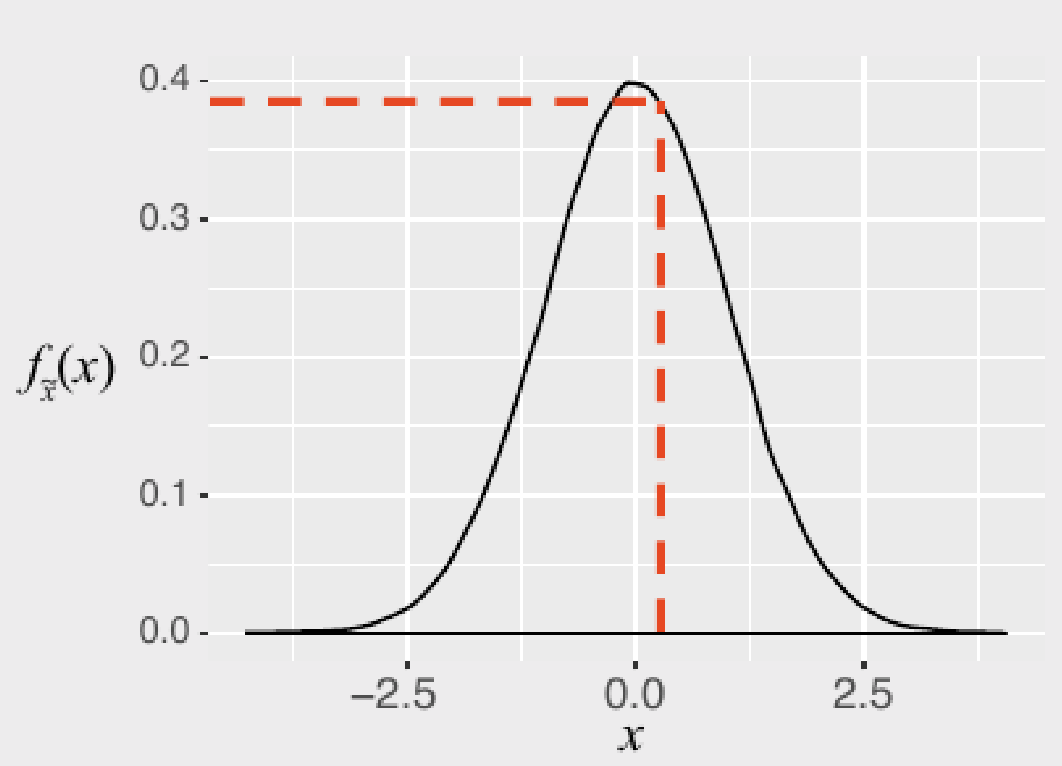
Computing the likelihood



x: an observation

$f(x)$: probability of observing x

Computing the likelihood



x: an observation

$f(x)$: probability of observing x

$\mathcal{L}(\theta|x)$: probability that θ are the true parameters, given that observed x

$$\mathcal{L}(\theta|x) = f(x_1)f(x_2)\ldots f(x_n)$$

Log-likelihood converts multiplication to summation:

$$\ln \mathcal{L}(\theta|x) = \ln f(x_1) + \ln f(x_2)\ldots \ln f(x_n)$$

Practice Question 2

Observations - Height of students (inches):

```
#>  [1] 65 69 66 67 68 72 68 69 63 70
```

a) Let's say we know that the height of students, $\tilde{x}$, in a classroom follows a normal distribution. A professor obtains the above height measurements students in her classroom. What is the log-likelihood that $\tilde{x} \sim \mathcal{N}(68, 4)$? In other words, compute $\ln \mathcal{L}(\mu = 68, \sigma = 4)$.

b) Compute the log-likelihood function using the same standard deviation $(\sigma = 4)$ but with the following different values for the mean, $\mu : 66, 67, 68, 69, 70$. How do the results compare? Which value for μ produces the highest log-likelihood?

Optimization

Optimality conditions

First order necessary condition

x^* is a "stationary point" when

$$\frac{df(x^*)}{dx} = 0$$

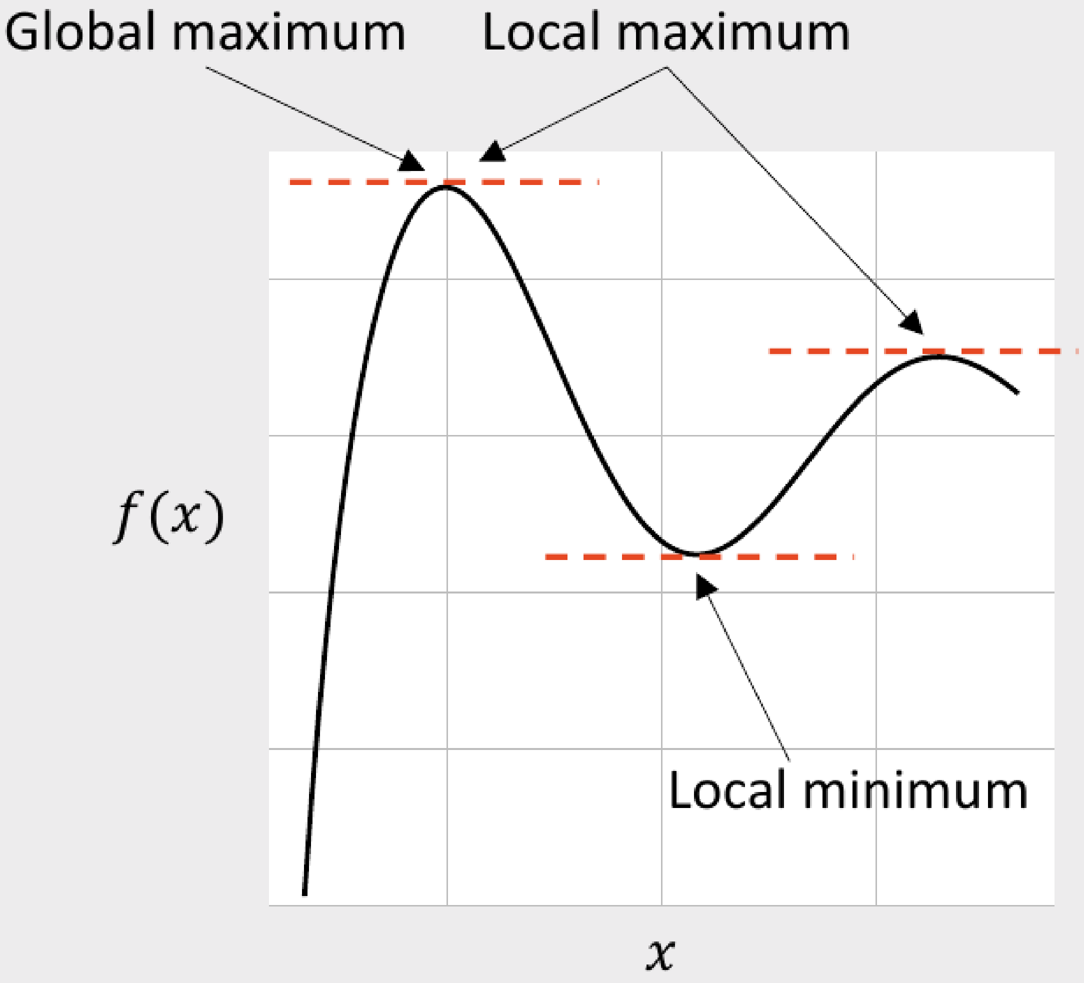
Second order sufficiency condition

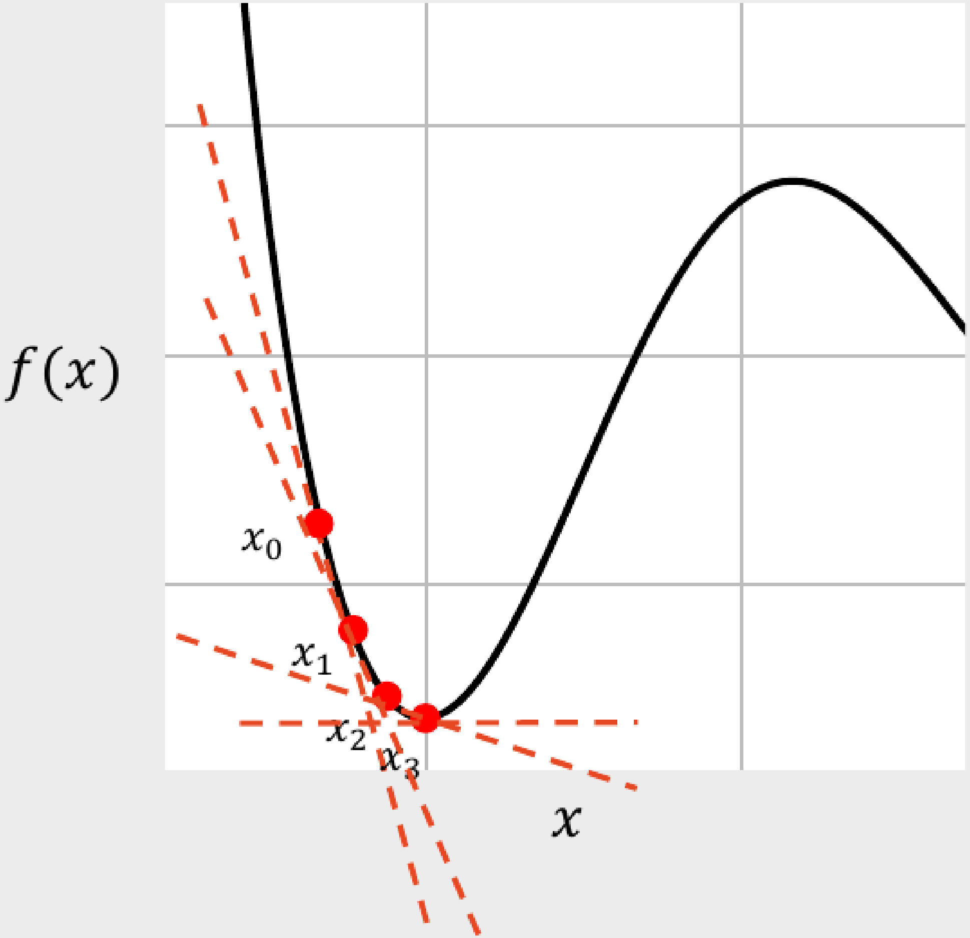
x^* is a local *maximum* when

$$\frac{d^2f(x^*)}{dx^2} < 0$$

x^* is a local *minimum* when

$$\frac{d^2f(x^*)}{dx^2} > 0$$





<u>Gradient Descent Method</u>:

1. Choose a starting point, x_0
2. At that point, compute the gradient, $\nabla f(x_0)$
3. Compute the next point, with a step size γ :

$$x_{n+1} = x_n - \gamma \nabla f(x_n)$$

Very small number

*Stop when $\nabla f(x_n) < \delta$

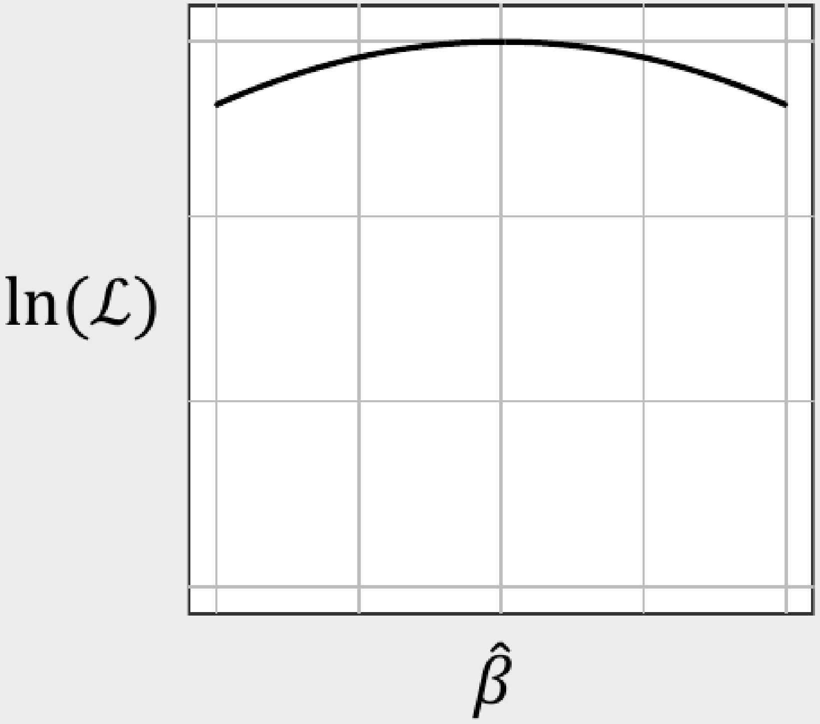
or

*Stop when $(x_{n+1} - x_n) < \delta$

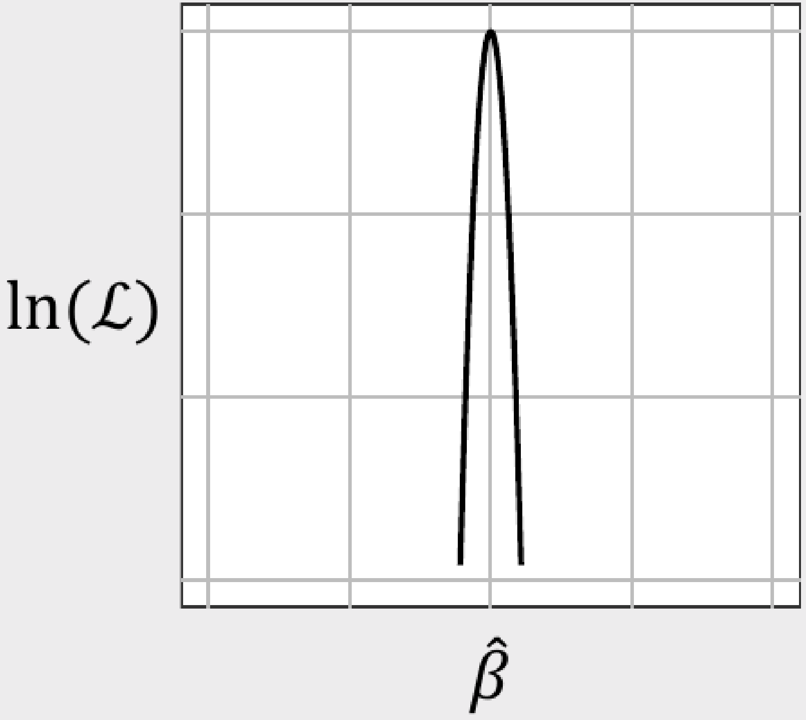
Uncertainty

The certainty of $\hat{\boldsymbol{\beta}}$ is inversely related to the curvature of the log-likelihood function

Greater variance in $\ln(\mathcal{L})$,
Less certainty in $\hat{\beta}$



Less variance in $\ln(\mathcal{L})$,
Greater certainty in $\hat{\beta}$



The *curvature* of the log-likelihood function is inversely related to the hessian

$$\sum\nolimits_{\boldsymbol{\beta}} = -\Big[\overbrace{\nabla^2_{\boldsymbol{\beta}} \ln(\mathcal{L})}^{\text{Hessian}}\Big]^{-1}$$

Covariance of $\widehat{\boldsymbol{\beta}}$

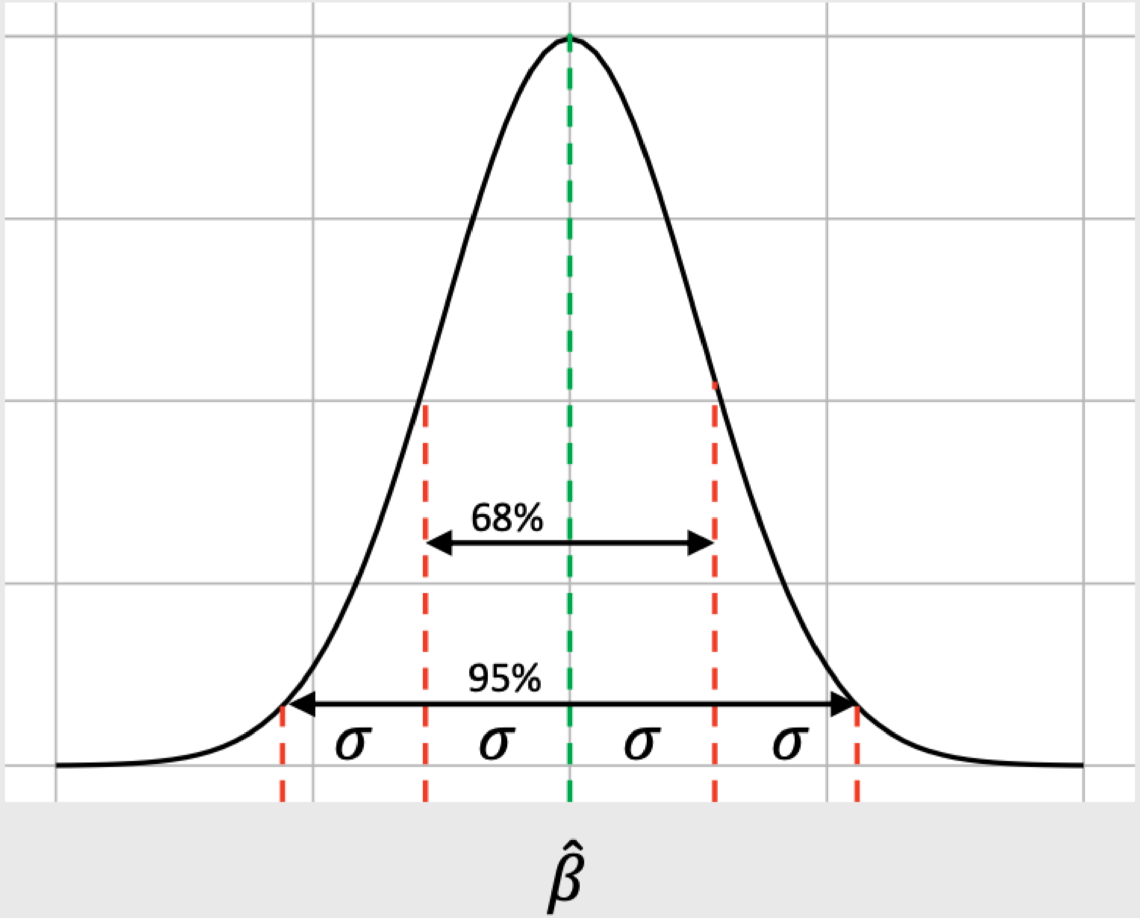
The *curvature* of the log-likelihood function is inversely related to the hessian

$$\sum_{\boldsymbol{\beta}} = -\overbrace{\left[\nabla_{\boldsymbol{\beta}}^2 \ln(\mathcal{L})\right]}^{\text{Hessian}}{}^{-1} = \begin{bmatrix} \sigma_{11}^2 & \cdots & \sigma_{m1}^2 \\ \vdots & \ddots & \vdots \\ \sigma_{1n}^2 & \cdots & \sigma_{mn}^2 \end{bmatrix}$$

Covariance of $\widehat{\boldsymbol{\beta}}$

Usually report parameter uncertainty ("standard errors") with σ values

Est.	Std. Err.
$\hat{\beta}_1$	σ_1
$\hat{\beta}_2$	σ_2
$\vdots$	$\vdots$
$\hat{\beta}_m$	σ_m



A 95% confidence interval is approximately $[\hat{\beta} - 2\sigma, \hat{\beta} + 2\sigma]$

Two approaches for obtaining confidence interval

Using Standard Errors

1. Get coefficients, `beta`
2. Get covariance matrix, `covariance`
3. `se <- sqrt(diag(covariance))`
4. `coef_ci <- c(beta - 2*se, beta + 2*se)`

Using Simulated Draws

1. Get coefficients, `beta`
2. Get covariance matrix, `covariance`
3. `draws <- as.data.frame(MASS::mvrnorm(10^5, beta, covariance))`
4. `coef_ci <- logitr::ci(draws, ci = 0.95)`

In-class example

```r
# 1. Get coefficients
beta <- c(
    price = -0.7, mpg = 0.1, elec = -4.0)

# 2. Get covariance matrix
hessian <- matrix(c(
    -6000,   50,   60,
       50, -700,   50,
       60,   50, -300),
    ncol = 3, byrow = TRUE)

covariance <- -1*solve(hessian)
```

Model from `logitr`

```r
beta <- coef(model)
covariance <- vcov(model)
```

Practice Question 3

Suppose we estimate the following utility model describing preferences for cars:

$$u_j = \alpha p_j + \beta_1 x_j^{mpg} + \beta_2 x_j^{elec} + \varepsilon_j$$

Compute a 95% confidence interval around the coefficients using:

a) Standard errors b) Simulated draws

The estimated model produces the following results:

Parameter	**Coefficient**
α	-0.7
β_1	0.1
β_2	-0.4

Hessian:

$$\begin{bmatrix} -6000 & 50 & 60 \\ 50 & -700 & 50 \\ 60 & 50 & -300 \end{bmatrix}$$

Design of experiment

Wine Pairings Example

meat	wine
fish	white
fish	red
steak	white
steak	red

Main Effects

1. **Fish** or **Steak**?
2. **Red** or **White** wine?

Interaction Effects

1. **Red** or **White** wine *with **Steak***?
2. **Red** or **White** wine *with **Fish***?

"D-optimal" designs maximize **main** effect information but confound **interaction** effect information

$$D = \left(\frac{|\boldsymbol{I}(\boldsymbol{\beta})|}{n^p} \right)^{1/p}$$

where p is the number of coefficients in the model and n is the total sample size

WTP

Willingness to Pay (WTP)

$$\tilde{u}_j = \alpha p_j + \boldsymbol{\beta} x_j + \tilde{\varepsilon}_j$$

$$\boldsymbol{\omega} = \frac{\boldsymbol{\beta}}{-\alpha}$$

Computing WTP with draws

$$\hat{\omega} = \frac{\hat{\beta}}{-\hat{\alpha}}$$

```r
draws_other <- draws[,2:ncol(draws)]
draws_price <- draws[,1]
draws_wtp <- draws_other / (-1*draws_price)
head(draws_wtp)
```

```
#>                [,1]      [,2]
#> [1,] 0.08156866 -5.771992
#> [2,] 0.10238910 -5.875931
#> [3,] 0.12643049 -5.841146
#> [4,] 0.10726155 -5.991838
#> [5,] 0.18738947 -5.695868
#> [6,] 0.14457267 -6.010809
```

Mean WTP with confidence interval

```r
logitr::ci(draws_wtp)
```

```
#>          mean       lower      upper
#> 1   0.1427314  0.03787149  0.2493893
#> 2 -5.7167653 -5.98211335 -5.4602922
```

Willingness to Pay (WTP)

"Preference Space"

$$\tilde{u}_j = \alpha p_j + \boldsymbol{\beta} x_j + \tilde{\varepsilon}_j$$

"WTP Space"

$$\boldsymbol{\omega} = \frac{\boldsymbol{\beta}}{-\alpha}$$

$$\boldsymbol{\lambda} = -\alpha$$

$$\tilde{u}_j = \boldsymbol{\lambda}(\boldsymbol{\omega} x_j - p_j) + \tilde{\varepsilon}_j$$

WTP space models have non-convex log-likelihood functions!

Use multi-start loop with random starting points

Market simulations

Simulate Market Shares

1. Define a market, X
2. Compute shares:

$$\hat{P}_j = \frac{e^{\hat{\boldsymbol{\beta}}' \boldsymbol{X}_j}}{\sum_{k=1}^{J} e^{\hat{\boldsymbol{\beta}}' \boldsymbol{X}_k}}$$

Simulate Market Shares

$$\hat{v} = \widehat{\boldsymbol{\beta}}'\mathbf{x}$$

$$= \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \dots & \vdots \\ x_{J1} & x_{J2} & \dots & x_{Jn} \end{bmatrix} \begin{bmatrix} \hat{\beta}_1 \\ \hat{\beta}_2 \\ \vdots \\ \hat{\beta}_n \end{bmatrix}$$

$$= \begin{bmatrix} \hat{\beta}_1 x_{11} + \hat{\beta}_2 x_{12} + \dots + \hat{\beta}_n x_{1n} \\ \hat{\beta}_1 x_{21} + \hat{\beta}_2 x_{22} + \dots + \hat{\beta}_n x_{2n} \\ \vdots \\ \hat{\beta}_1 x_{J1} + \hat{\beta}_2 x_{J2} + \dots + \hat{\beta}_n x_{Jn} \end{bmatrix} = \begin{bmatrix} \hat{v}_1 \\ \hat{v}_2 \\ \vdots \\ \hat{v}_J \end{bmatrix}$$

Simulate Market Shares

$$\hat{v} = \widehat{\boldsymbol{\beta}}'\mathbf{x}$$

$$= \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \dots & \vdots \\ x_{J1} & x_{J2} & \dots & x_{Jn} \end{bmatrix} \begin{bmatrix} \hat{\beta}_1 \\ \hat{\beta}_2 \\ \vdots \\ \hat{\beta}_n \end{bmatrix}$$

$$= \begin{bmatrix} \hat{\beta}_1 x_{11} + \hat{\beta}_2 x_{12} + \dots + \hat{\beta}_n x_{1n} \\ \hat{\beta}_1 x_{21} + \hat{\beta}_2 x_{22} + \dots + \hat{\beta}_n x_{2n} \\ \vdots \\ \hat{\beta}_1 x_{J1} + \hat{\beta}_2 x_{J2} + \dots + \hat{\beta}_n x_{Jn} \end{bmatrix} = \begin{bmatrix} \hat{v}_1 \\ \hat{v}_2 \\ \vdots \\ \hat{v}_J \end{bmatrix}$$

In R:

```
X %*% beta
```

Simulating Market Shares **with Uncertainty**

Rely on the `predict()` function to compute shares with uncertainty.

Internally, it:

1. Takes draws of $\boldsymbol{\beta}$
2. Computes P_j for each draw
3. Returns mean and confidence interval computed from draws

Review the `logitr-cars` examples

Break

05:00

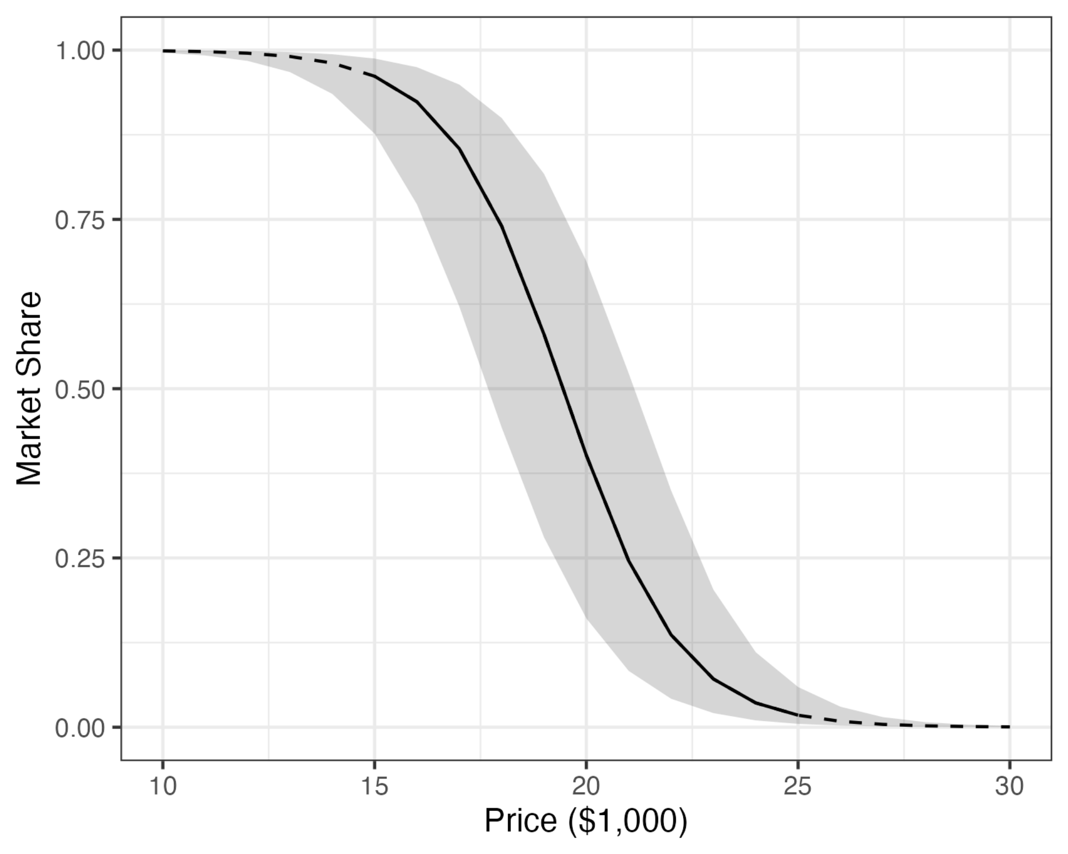
Week 13: *Class Review*

1. Exam Review

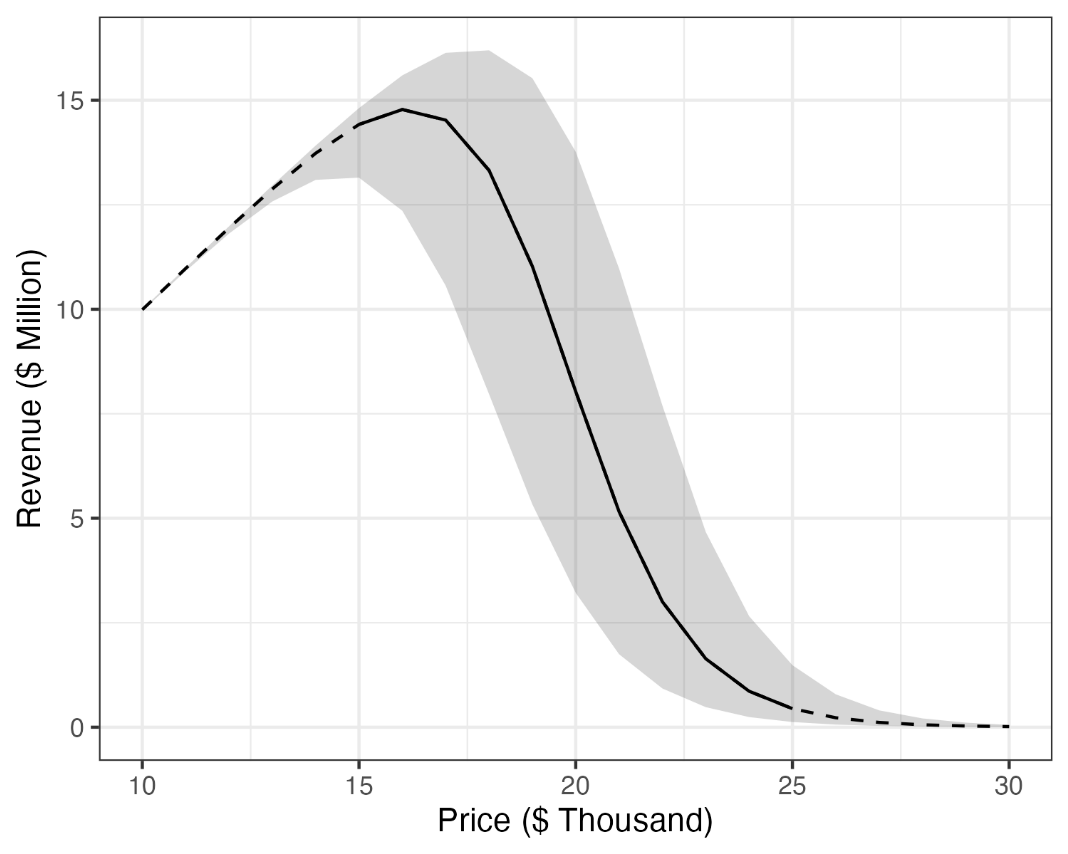
BREAK

2. Sensitivity Analysis

Market share sensitivity to price

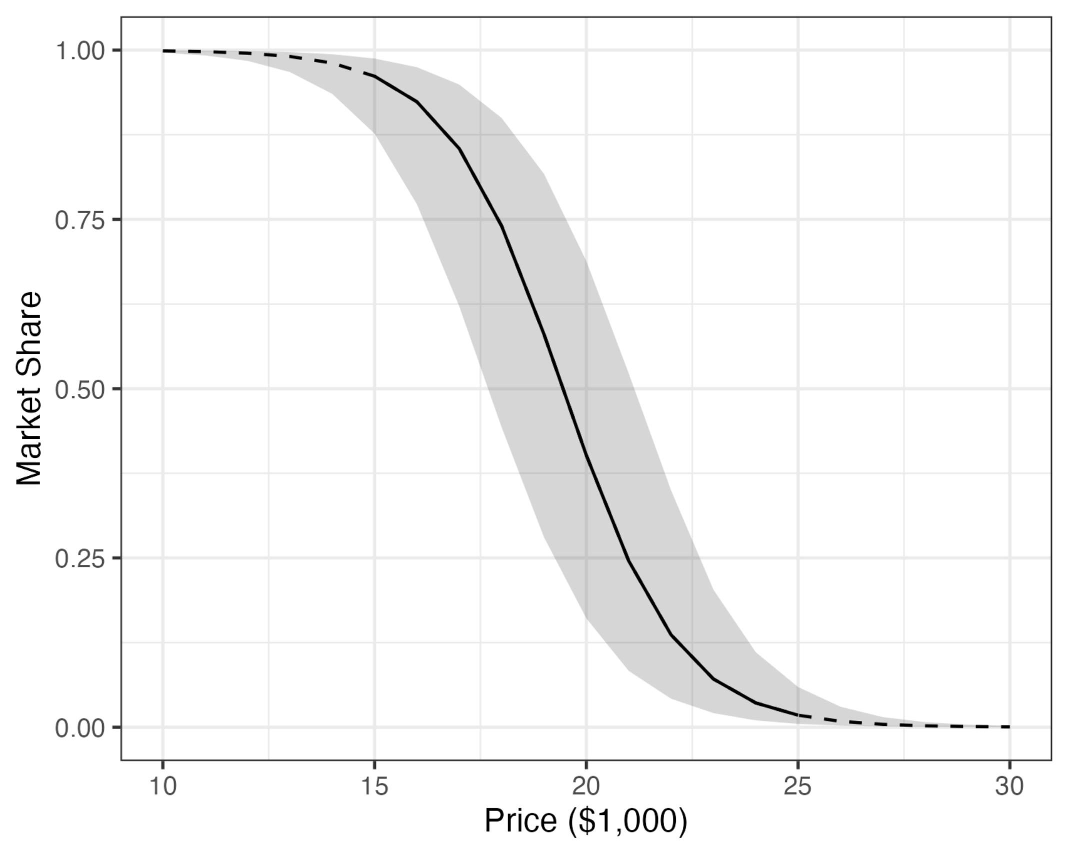


Revenue sensitivity to price



$$R = Q * P$$

Market share sensitivity to price

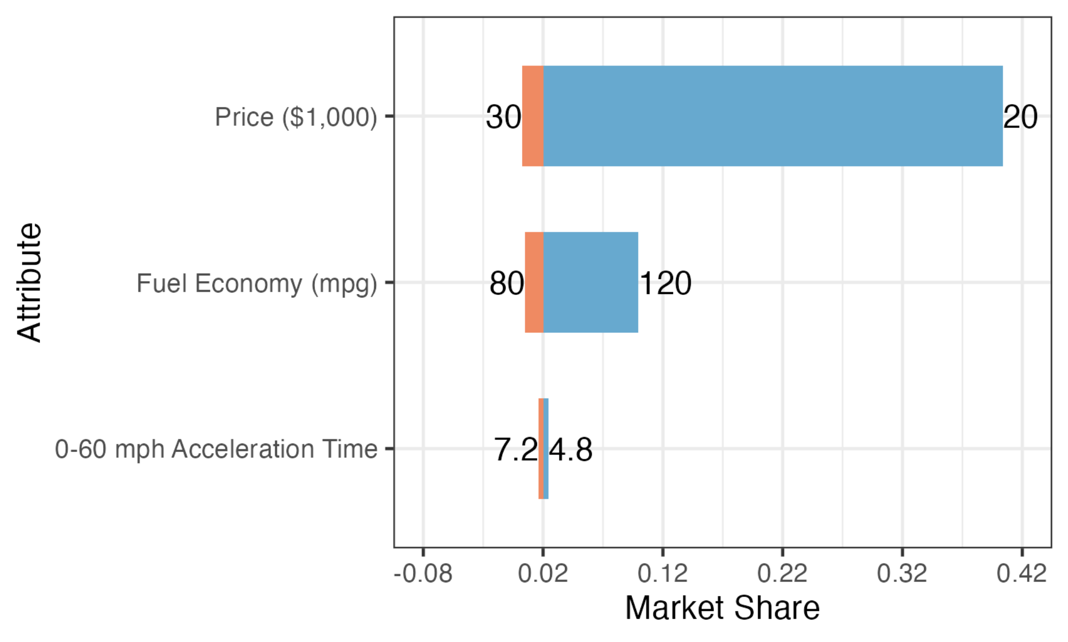


Observations

- Solid line reflects *interpolation* (attribute range in survey)
- Dashed line reflects *extrapolation* (beyond attribute range in survey)
- Ribbon reflects *parameter uncertainty*

Market share sensitivity to all attributes

Market share sensitivity to all attributes



Observations

- Middle point reflects baseline market share:
  - **Price**: $25,000
  - **Fuel Economy**: 100 mpg
  - **0-60 mph Accel. time**: 6 sec
- Boundaries on each attribute should reflect max feasible attribute bounds

Sensitivity analyses

1. Open `logitr-cars`
2. Open `code/9.1-compute-sensitivity.R`
3. Open `code/9.2-plot-sensitivity.R`

Your Turn

15:00

As a team:

- Read in and clean your final data.
- Estimate a baseline model.
- Set your baseline market simulation case.
- Compute sensitivities to price and other attributes.