

# Estimating Models with {logitr}



👤 John Paul Helveston

🏛️ The George Washington University |  
Dept. of Engineering Management and  
Systems Engineering

📅 June 15, 2023

# Many FOSS options model estimation

R packages:

- `{logitr}`: Fastest, mixed logit, WTP space.
- `{apollo}`: Most flexible, great documentation.
- `{mlogit}`: The OG R package.
- `{gmnl}`: Generalized logit model (though slow).
- `{mixl}`: Good for big datasets (uses C for speed).

Python packages:

- `{xlogit}`: Basically Python version of `{logitr}`.

`Stan`: For the Bayesians.

# Many FOSS options model estimation

R packages:

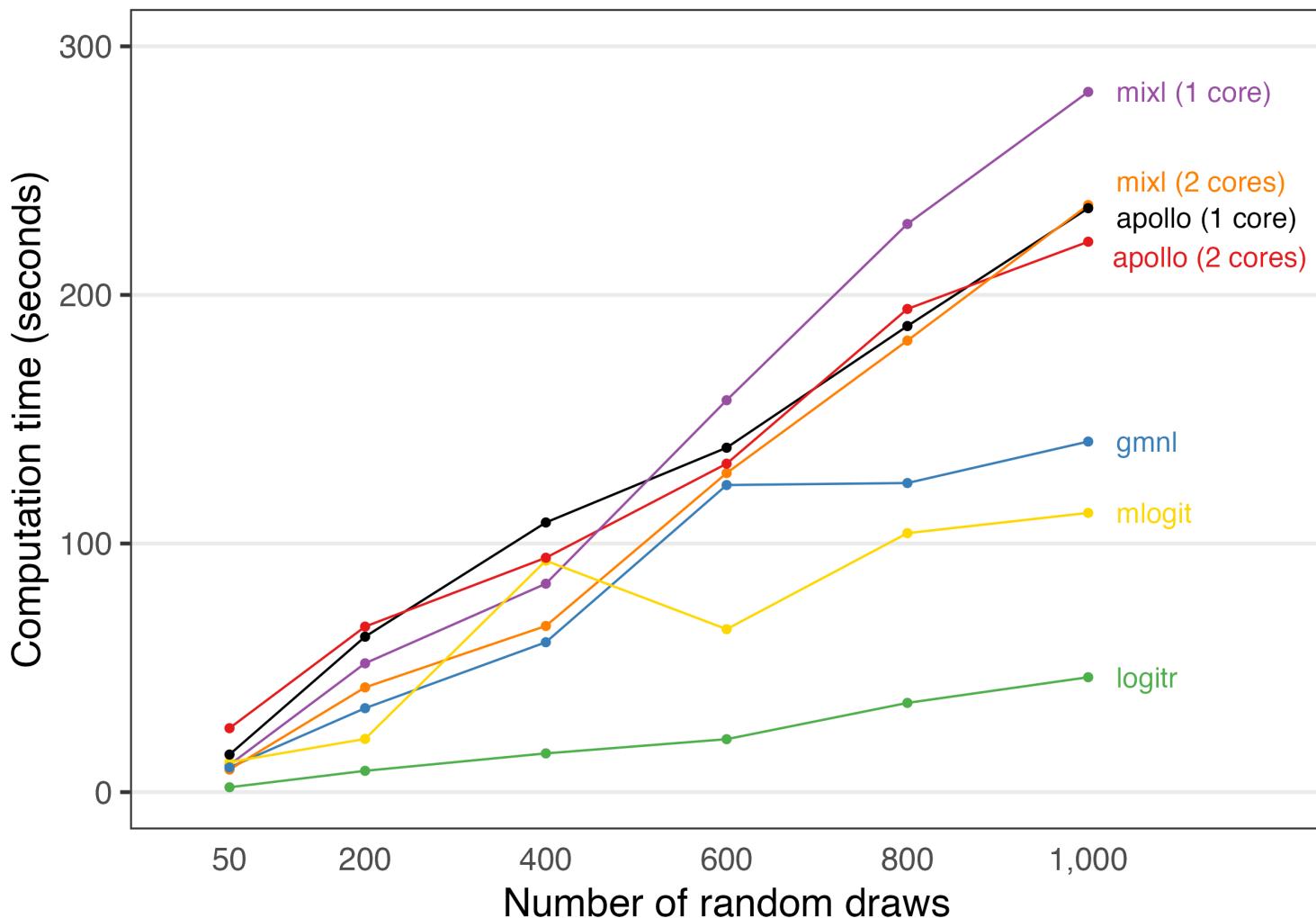
- `{logitr}`: Fastest, mixed logit, WTP space. ← I wrote this one, so I'm showcasing it!
- `{apollo}`: Most flexible, great documentation.
- `{mlogit}`: The OG R package.
- `{gmnl}`: Generalized logit model (though slow).
- `{mixl}`: Good for big datasets (uses C for speed).

Python packages:

- `{xlogit}`: Basically Python version of `{logitr}`.

`Stan`: For the Bayesians.

# {logitr} is fast!



{logitr} supports two common forms of utility models

Preference Space

WTP Space

$$u_j = \boldsymbol{\beta}' \mathbf{x}_j + \alpha p_j + \varepsilon_j \quad u_j = \lambda (\boldsymbol{\omega}' \mathbf{x}_j - p_j) + \varepsilon_j$$

# {logitr} has a similar UI with {cbcTools}

({cbcTools} uses {logitr} to simulate choices and assess power)

## {cbcTools}

```
power <- cbc_power(  
    nbreaks = 10,  
    n_q      = 6,  
    data     = data,  
    obsID   = "obsID",  
    outcome = "choice",  
    pars    = c("price", "type", "freshness"  
)
```

## {logitr}

```
model <- logitr(  
    data     = data,  
    obsID   = "obsID",  
    outcome = "choice",  
    pars    = c("price", "type", "freshness")
```

# Utility model refresher

# Which would you choose?

\$2.49



\$2.99



\$1.99



\$3.99



# Estimate marginal utilities

$$u_j = \beta' \mathbf{x}_j + \alpha p_j + \varepsilon_j, \quad \varepsilon_j \sim \text{Gumbel} \left(0, \frac{\pi^2}{6}\right)$$

```
#>           Estimate Std. Error   z-value Pr(>|z|) 
#> price      -0.3886257 0.02426923 -16.01311    0  
#> brandhiland -3.1167063 0.14496806 -21.49926    0  
#> brandyoplait  1.4463603 0.08869767  16.30663    0  
#> branddannon   0.6440868 0.05435965  11.84862    0
```

# Convert marginal *utilities* to marginal WTPs

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

```
#>              Estimate Std. Error z-value Pr(>|z|) 
#> brandhiland -8.01982   0.46038 -17.4199 < 2.2e-16 ***
#> brandyoplait  3.72173   0.15923  23.3728 < 2.2e-16 ***
#> branddannon    1.65734   0.16748   9.8959 < 2.2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

# Alternative approach: **Estimate a WTP-Space Model**

Substitutions:

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

$$\lambda = -\alpha$$

"Preference Space"

$$u_j = \beta' \mathbf{x}_j + \alpha p_j + \varepsilon_j$$

"WTP Space"

$$u_j = \lambda (\omega' \mathbf{x}_j - p_j) + \varepsilon_j$$

# What's the difference?

Preference Space

WTP Space

$$u_j = \beta' \mathbf{x}_j + \alpha p_j + \varepsilon_j$$



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

$$u_j = \lambda (\omega' \mathbf{x}_j - p_j) + \varepsilon_j$$

## Mixed logit:

Unreasonably large WTP variance across population

$$u_j = \beta' \mathbf{x}_j + \alpha p_j + \varepsilon_j$$

$$\hat{\beta} \sim \mathcal{N}(\hat{\mu}, \hat{\Sigma})$$



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

$$\hat{\alpha} \sim \mathcal{N}(\hat{\mu}, \hat{\sigma}^2)$$

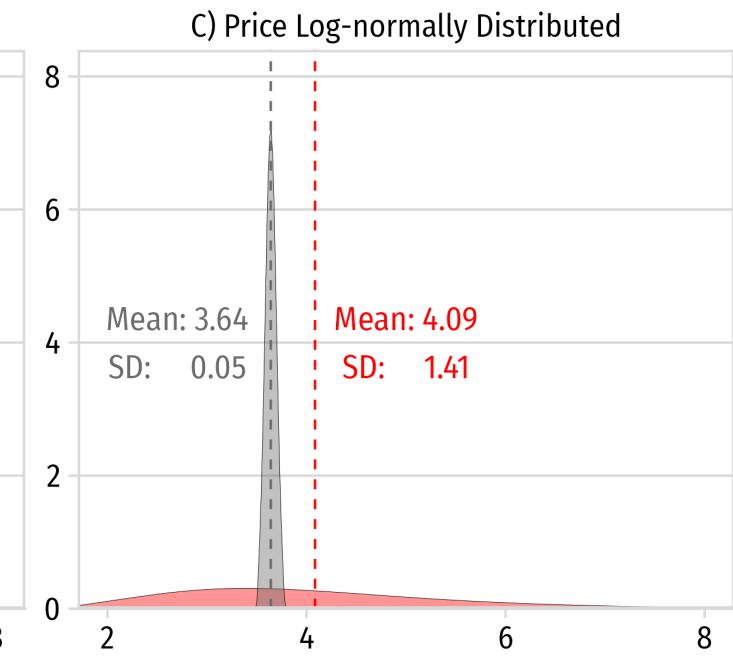
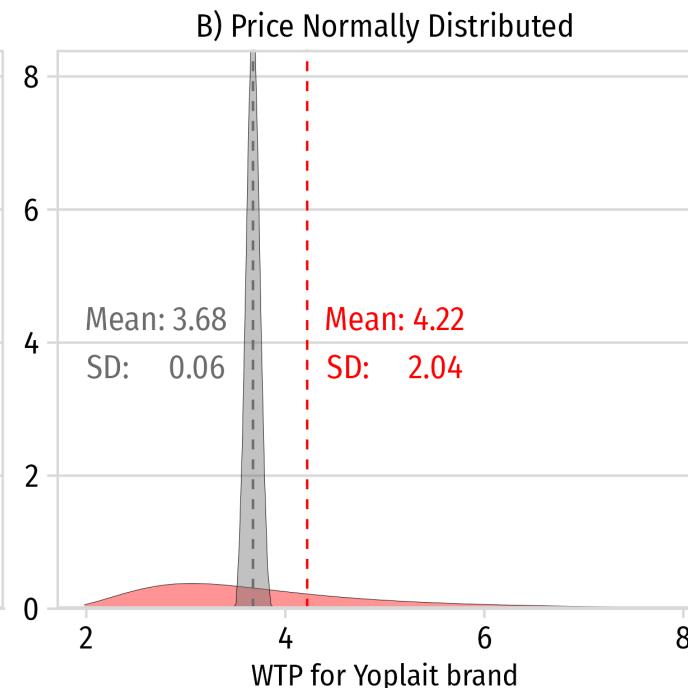
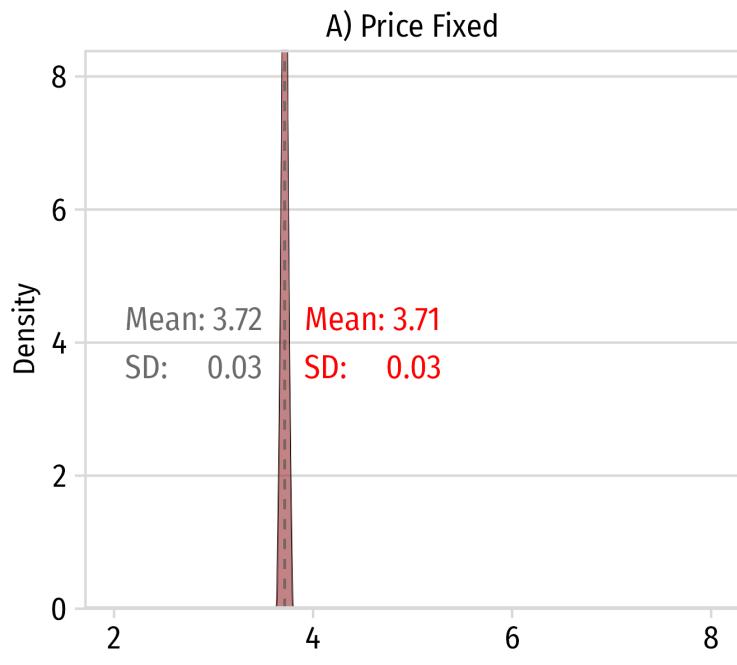
# Preference space model produces unreasonably large variance in WTP

**Preference Space**

$$\hat{\beta} \sim \mathcal{N}(\hat{\mu}, \hat{\Sigma})$$

**WTP Space**

$$\hat{\omega} \sim \mathcal{N}(\hat{\mu}, \hat{\Sigma})$$



Model space:  WTP  Preference

# *Practical Considerations*

# *Practical Considerations*

WTP space models produce immediately interpretable results

Unit: "Utility" (relative)

$$u_j = \beta' \mathbf{x}_j + \alpha p_j + \varepsilon_j$$

```
#>           Estimate Std. Error   z-value Pr(>|z|)  
#> price      -0.3886257 0.02426923 -16.01311    0  
#> brandhiland -3.1167063 0.14496806 -21.49926    0  
#> brandyoplait  1.4463603 0.08869767  16.30663    0  
#> branddannon   0.6440868 0.05435965  11.84862    0
```

Units: \$ (absolute)

$$u_j = \lambda (\omega' \mathbf{x}_j - p_j) + \varepsilon_j$$

```
#>           Estimate Std. Error   z-value Pr(>|z|)  
#> scalePar      0.388626  0.024265  16.0156 < 2.2e-16 ***  
#> brandhiland   -8.019815  0.460382 -17.4199 < 2.2e-16 ***  
#> brandyoplait   3.721731  0.159234  23.3728 < 2.2e-16 ***  
#> branddannon    1.657345  0.167479   9.8959 < 2.2e-16 ***  
#> ---  
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

# *Practical Considerations*

WTPs can be directly compared across different models  
(even estimates from different data sets)

$$u_j^* = \boldsymbol{\beta}^{*'} \mathbf{x}_j + \alpha^* p_j + \varepsilon_j^*, \quad \varepsilon_j^* \sim \text{Gumbel}\left(0, \sigma^2 \frac{\pi^2}{6}\right)$$

## **Preference Space**

Parameters proportional to  $\sigma$

$$\left(\frac{u_j^*}{\sigma}\right) = \left(\frac{\boldsymbol{\beta}^*}{\sigma}\right)' \mathbf{x}_j + \left(\frac{\alpha^*}{\sigma}\right) p_j + \left(\frac{\varepsilon_j^*}{\sigma}\right)$$

$$u_j = \boldsymbol{\beta}' \mathbf{x}_j + \alpha p_j + \varepsilon_j$$

## **WTP Space**

Parameters independent of  $\sigma$

$$\left(\frac{u_j^*}{-\alpha^*}\right) = \left(\frac{\boldsymbol{\beta}^*}{-\alpha^*}\right)' \mathbf{x}_j + \left(\frac{\alpha^*}{-\alpha^*}\right) p_j + \left(\frac{\varepsilon_j^*}{-\alpha^*}\right)$$

$$u_j = \lambda (\boldsymbol{\omega}' \mathbf{x}_j - p_j) + \varepsilon_j$$

## *Practical Considerations*

Neither space systematically predicts choice better

- **Train and Weeks (2005)** and **Sonnier et al. (2007)** found preference space model fit data better.
- **Das et al. (2009)** found nearly identical model fit on out-of-sample predictions with each model specification.

...but most software is built for

$$u_j = \beta' \mathbf{x}_j + \alpha p_j + \varepsilon_j$$

not

$$u_j = \lambda (\omega' \mathbf{x}_j - p_j) + \varepsilon_j$$

`logitr` to the rescue!



# The logitr Package

Estimation of multinomial and mixed logit models in with "Preference" space or "Willingness-to-pay" (WTP) space utility parameterizations.



- Multinomial logit (MNL) models
- Mixed logit (MXL) models with normal and log-normal parameter distributions.
- Preference space and WTP space utility parameterizations.
- Weighted models to differentially weight individual observations.
- Uncorrelated or correlated heterogeneity covariances for mixed logit models.
- Functions for computing WTP from preference space models.
- Functions for predicting expected probabilities and outcomes for sets of alternatives based on an estimated model.
- A parallelized multistart optimization loop that uses different random starting points in each iteration to search for different local minima (useful for non-convex problems like MXL models or models with WTP space parameterizations).

# Data format

Data must be arranged in a "long" format:

- Each row is an alternative from a choice observation.
- Choice observations do *not* have to be symmetric.

Required variables:

- `outcome`: A dummy variable for the chosen alternative (`1` or `0`).
- `obsID`: A sequence of repeated numbers identifying each unique choice observation, e.g. `1, 1, 2, 2, 3, 3`.
- `pars`: Any other variables to use as model covariates.

# Data format

```
head(yogurt, 10)
```

```
#>   choice obsID alt price   brand
#> 1      0     1   1   8.1 dannon
#> 2      0     1   2   6.1 hiland
#> 3      1     1   3   7.9 weight
#> 4      0     1   4  10.8 yoplait
#> 5      1     2   1   9.8 dannon
#> 6      0     2   2   6.4 hiland
#> 7      0     2   3   7.5 weight
#> 8      0     2   4  10.8 yoplait
#> 9      1     3   1   9.8 dannon
#> 10     0     3   2   6.1 hiland
```

- `outcome = "choice"`
- `obsID = "obsID"`
- `pars = c("price", "brand")`

# Multinomial logit in Preference Space

```
mnl_pref <- logitr(  
  data      = yogurt,  
  outcome   = "choice",  
  obsID     = "obsID",  
  pars      = c("price", "brand")  
)  
  
summary(mnl_pref)
```

$$u_j = \beta' \mathbf{x}_j + \alpha p_j + \varepsilon_j$$

```
#> =====  
#>  
#> Model estimated on: Fri Jun 09 10:12:27 2023  
#>  
#> Using logitr version: 1.1.0  
#>  
#> Call:  
#> logitr(data = yogurt, outcome = "choice", obsID = "obsID", pars = c("p  
#>     "brand"))  
#>  
#> Frequencies of alternatives:  
#>          1         2         3         4  
#> 0.402156 0.029436 0.229270 0.339138  
#>  
#> Exit Status: 3, Optimization stopped because ftol_rel or ftol_abs was  
#>  
#> Model Type: Multinomial Logit  
#> Model Space: Preference  
#> Model Run: 1 of 1  
#> Iterations: 20  
#> Elapsed Time: 0h:0m:0.01s  
#> Algorithm: NLOPT_LD_LBFGS  
#> Weights Used?: FALSE  
#> Robust? FALSE  
#>  
#> Model Coefficients:  
#>             Estimate Std. Error z-value Pr(>|z|)  
#> price       -0.388626  0.024269 -16.013 < 2.2e-16 ***  
#> brandhiland -3.116706  0.144968 -21.499 < 2.2e-16 ***  
#> brandyoplait  1.446360  0.088698  16.307 < 2.2e-16 ***  
#> branddannon   0.644087  0.054360  11.849 < 2.2e-16 ***  
#> ---  
#> Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
#>
```

# Multinomial logit in WTP Space

```
library(logitr)

mnl_wtp <- logitr(
  data      = yogurt,
  outcome   = "choice",
  obsID     = "obsID",
  pars      = "brand",
  scalePar = "price"
)
summary(mnl_wtp)
```

$$u_j = \lambda (\boldsymbol{\omega}' \mathbf{x}_j - p_j) + \varepsilon_j$$

```
#> =====
#>
#> Model estimated on: Fri Jun 09 10:12:41 2023
#>
#> Using logitr version: 1.1.0
#>
#> Call:
#> logitr(data = yogurt, outcome = "choice", obsID = "obsID", pars =
#>         scalePar = "price")
#>
#> Frequencies of alternatives:
#>           1          2          3          4
#> 0.402156 0.029436 0.229270 0.339138
#>
#> Exit Status: 3, Optimization stopped because ftol_rel or ftol_abs was
#>
#> Model Type: Multinomial Logit
#> Model Space: Willingness-to-Pay
#> Model Run: 1 of 1
#> Iterations: 40
#> Elapsed Time: 0h:0m:0.02s
#> Algorithm: NLOPT_LD_LBFGS
#> Weights Used?: FALSE
#> Robust? FALSE
#>
#> Model Coefficients:
#>             Estimate Std. Error z-value Pr(>|z|)
#> scalePar    0.388633  0.024269 16.013 < 2.2e-16 ***
#> brandhiland -8.019717  0.455549 -17.605 < 2.2e-16 ***
#> brandyoplait 3.721711  0.157655 23.607 < 2.2e-16 ***
#> branddannon  1.657290  0.165712 10.001 < 2.2e-16 ***
#> ---
#> Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

## Caution

Log-likelihood function for WTP space models is  
**non-convex** 😔

# Use a Multistart

```
mnl_wtp <- logitr(  
  data      = yogurt,  
  outcome   = "choice",  
  obsID     = "obsID",  
  pars      = "brand",  
  scalePar  = "price",  
  numMultiStarts = 10  
)
```

```
summary(mnl_wtp)
```

$$u_j = \lambda (\boldsymbol{\omega}' \mathbf{x}_j - p_j) + \varepsilon_j$$

```
#> =====  
#>  
#> Model estimated on: Fri Jun 09 10:13:43 2023  
#>  
#> Using logitr version: 1.1.0  
#>  
#> Call:  
#> logitr(data = yogurt, outcome = "choice", obsID = "obsID", pars = "bra  
#>   scalePar = "price", numMultiStarts = 10)  
#>  
#> Frequencies of alternatives:  
#>   1       2       3       4  
#> 0.402156 0.029436 0.229270 0.339138  
#>  
#> Summary Of Multistart Runs:  
#>   Log Likelihood Iterations Exit Status  
#> 1       -2665.11      40      3  
#> 2       -2665.11      39      3  
#> 3       -2665.11      43      3  
#> 4       -2665.11      47      3  
#> 5       -2665.11      54      3  
#> 6       -2665.11      42      3  
#> 7       -2665.11      39      3  
#> 8       -2665.11      44      3  
#> 9       -2665.11      39      3  
#> 10      -2665.11      38      3  
#>  
#> Use statusCodes() to view the meaning of each status code  
#>  
#> Exit Status: 3, Optimization stopped because ftol_rel or ftol_abs was  
#>  
#> Model Type:    Multinomial Logit  
#> Model Space:   Willingness-to-Pay  
#> Model Run:      10 of 10
```

# Mixed logit in Preference Space

```
mxl_pref <- logitr(  
  data      = yogurt,  
  outcome   = "choice",  
  obsID     = "obsID",  
  pars      = c("price", "brand"),  
  randPars  = c(brand = "n"),  
  numMultiStarts = 10  
)
```

$$u_j = \beta' \mathbf{x}_j + \alpha p_j + \varepsilon_j$$

$$\hat{\beta} \sim \mathcal{N}(\hat{\mu}, \hat{\Sigma})$$

# Mixed logit in WTP Space

```
mxl_wtp <- logitr(  
  data      = yogurt,  
  outcome   = "choice",  
  obsID     = "obsID",  
  pars      = "brand",  
  scalePar = "price",  
  randPars  = c(brand = "n"),  
  randScale = "ln",  
  numMultiStarts = 10  
)
```

$$u_j = \lambda (\omega' \mathbf{x}_j - p_j) + \varepsilon_j$$

$$\hat{\omega} \sim \mathcal{N}(\hat{\mu}, \hat{\Sigma})$$

# *Convenient helper functions*

# `predict()`: Expected shares for a set of alternatives

Define a set of alternatives

```
data <- subset(  
  yogurt, obsID == 42,  
  select = c('price', 'brand', 'obsID'))  
  
data
```

```
#>   price   brand obsID  
#> 1   6.3    dannon  42  
#> 2   6.1    hiland  42  
#> 3   7.9    weight  42  
#> 4  11.5    yoplait 42
```

Predict probabilities

```
predict(  
  mnl_pref,  
  newdata = data,  
  obsID = "obsID",  
  returnData = TRUE  
)
```

```
#>   obsID predicted_prob price   brand  
#> 1     42      0.62391435  6.3    dannon  
#> 2     42      0.01568877  6.1    hiland  
#> 3     42      0.17593683  7.9    weight  
#> 4     42      0.18446005 11.5   yoplait
```

# Your turn

- Download the practice zip file for this section.
- Open the `estimating-models.Rproj` file to open RStudio.
- In RStudio, open the `practice.R` file.
- Experiment with estimating different models (use either one of the example datasets included in the package, or simulate your own data!)

{logitr} documentation:  
<https://jhelvy.github.io/logitr/>

Back to workshop website:  
<https://jhelvy.github.io/2023-qux-conf-conjoint/>

@JohnHelveston   
@jhelvy   
@jhelvy   
jhelvy.com   
jph@gwu.edu 