Julia

Mizuhiro Suzuki

Table of contents

1 Julia

This is a Quarto website.

To learn more about Quarto websites visit https://quarto.org/docs/websites.

2 1

```
using CSV
using DataFrames
using StringEncodings
using FixedEffectModels
using RegressionTables
using Plots
using Optim
using Printf
using GLM
```

```
data = CSV.File(
    open(read, "data/demand_estimation/CleanData_20180222.csv", enc"shift-jis"),
    missingstring = ["NA", ""],
    ) |> DataFrame
first(select(data, Not([:base_color, :option_color])), 5)
```

	Maker	Type	Name	Year	Sales	comment	Model	Year_true	
	String15	String7	String31	Int64	Int64	String?	String	Int64	
1	Audi	Foreign	A1	2011	4206	*missing*	1.4 TFSI	2011	...
2	Audi	Foreign	A1	2012	4502	*missing*	1.4 TFSI	2012	...
3	Audi	Foreign	A1	2013	5071	*missing*	1.4 TFSI	2012	...
4	Audi	Foreign	A3	2006	4830	*missing*		2006	...
5	Audi	Foreign	A3	2007	3874	*missing*		2007	...

```
dataHH = CSV.read("data/demand_estimation/HHsize.csv", DataFrame)
dataHH[!, :HH] = parse.(Int, replace.(dataHH.HH, "," => ""))
first(dataHH, 5)
```

	year	HH
	Int64	Int64
1	1975	33310006
2	1976	33911052
3	1977	34380314
4	1978	34858696
5	1979	35350173

```
dataCPI = CSV.File(
    open(read, "data/demand_estimation/zni2015s.csv", enc"shift-jis"),
    select = 1:2,
    skipto = 7
    ) |> DataFrame
rename!(dataCPI, "  " => "year", " " => "CPI")
first(dataCPI, 5)
```

	year	CPI
	Int64	Float64
1	1970	31.5
2	1971	33.5
3	1972	35.2
4	1973	39.3
5	1974	48.4

```
data = data[!, [
        :Maker, :Type, :Name, :Year, :Sales,
        :Model, :price, :kata, :weight, :FuelEfficiency,
        :HorsePower, :overall_length, :overall_width, :overall_height
        ]]
rename!(data, "Year" => "year")
data = leftjoin(data, dataHH, on = :year)
data = leftjoin(data, dataCPI, on = :year)
first(data, 5)
```

	Maker	Type	Name	year	Sales	Model	price	kata	
	String15	String7	String31	Int64	Int64	String	Float64	String15	
1	Audi	Foreign	A1	2011	4206	1.4 TFSI	289.0	DBA-8XCAX	...
2	Audi	Foreign	A1	2012	4502	1.4 TFSI	273.0	DBA-8XCAX	...
3	Audi	Foreign	A1	2013	5071	1.4 TFSI	273.0	DBA-8XCAX	...
4	Audi	Foreign	A3	2006	4830		284.0	GH-8PBSE	...
5	Audi	Foreign	A3	2007	3874		286.0	GH-8PBSE	...

```
dropmissing!(data, :FuelEfficiency);
```

```
cpi2016 = dataCPI[dataCPI.year .== 2016, "CPI"][1]
data[!, :price] = data.price ./ (data.CPI / cpi2016) / 100;
```

```
data[!, :size] = (data[:, :overall_length] / 1000) .* (data[:, :overall_width] / 1000) .*
data[!, :hppw] = data[:, :HorsePower] ./ data[:, :weight];

unique_name = unique(data[!, [:Name]])
unique_name[!, :NameID] = rownumber.(eachrow(unique_name))
data = leftjoin(data, unique_name, on = :Name);

data = transform(
    groupby(data, :year),
    :Sales => sum => :inside_total
);
data[!, :outside_total] = data.HH .- data.inside_total;
data[!, :share] = data.Sales ./ data.HH;
data[!, :share0] = data.outside_total ./ data.HH;
```

```
transform!(
    groupby(data, [:year, :Maker]),
    [:hppw, :FuelEfficiency, :size] .=> sum .=> [:hppw_sum_own, :FuelEfficiency_sum_own, :
    [:hppw, :FuelEfficiency, :size] .=> (x -> sum(x.^2)) .=> [:hppw_sqr_sum_own, :FuelEffi
    nrow => "group_n"
);
transform!(
    groupby(data, [:year]),
    [:hppw, :FuelEfficiency, :size] .=> sum .=> [:hppw_sum_mkt, :FuelEfficiency_sum_mkt, :
    [:hppw, :FuelEfficiency, :size] .=> (x -> sum(x.^2)) .=> [:hppw_sqr_sum_mkt, :FuelEffi
    nrow => "mkt_n"
);
```

```
data[!, :iv_BLP_own_hppw]             = data[:, :hppw_sum_own]           .- data[:, :hppw]
data[!, :iv_BLP_own_FuelEfficiency]   = data[:, :FuelEfficiency_sum_own] .- data[:, :FuelE
data[!, :iv_BLP_own_size]             = data[:, :size_sum_own]           .- data[:, :size]
data[!, :iv_BLP_other_hppw]           = data[:, :hppw_sum_mkt]           .- data[:, :hppw_
data[!, :iv_BLP_other_FuelEfficiency] = data[:, :FuelEfficiency_sum_mkt] .- data[:, :FuelE
data[!, :iv_BLP_other_size]           = data[:, :size_sum_mkt]           .- data[:, :size_
```

```
data[!, :iv_GH_own_hppw]             = (
    (data[:, :group_n] .- 1) .* data[:, :hppw].^2 .+
    (data[:, :hppw_sqr_sum_own] .- data[:, :hppw].^2) .-
    2 .* data[:, :hppw] .* (data[:, :hppw_sum_own] .- data[:, :hppw])
);
data[!, :iv_GH_own_FuelEfficiency]   = (
    (data[:, :group_n] .- 1) .* data[:, :FuelEfficiency].^2 .+
    (data[:, :FuelEfficiency_sqr_sum_own] .- data[:, :FuelEfficiency].^2) .-
    2 .* data[:, :FuelEfficiency] .* (data[:, :FuelEfficiency_sum_own] .- data[:, :FuelEff
);
data[!, :iv_GH_own_size]             = (
    (data[:, :group_n] .- 1) .* data[:, :size].^2 .+
    (data[:, :size_sqr_sum_own] .- data[:, :size].^2) .-
    2 .* data[:, :size] .* (data[:, :size_sum_own] .- data[:, :size])
);
data[!, :iv_GH_other_hppw]           = (
    (data[:, :mkt_n] .- data[:, :group_n]) .* data[:, :hppw].^2 .+
    (data[:, :hppw_sqr_sum_mkt] .- data[:, :hppw_sqr_sum_own]) .-
    2 .* data[:, :hppw] .* (data[:, :hppw_sum_mkt] .- data[:, :hppw_sum_own])
);
data[!, :iv_GH_other_FuelEfficiency] = (
    (data[:, :mkt_n] .- data[:, :group_n]) .* data[:, :FuelEfficiency].^2 .+
    (data[:, :FuelEfficiency_sqr_sum_mkt] .- data[:, :FuelEfficiency_sqr_sum_own]) .-
    2 .* data[:, :FuelEfficiency] .* (data[:, :FuelEfficiency_sum_mkt] .- data[:, :FuelEff
);
data[!, :iv_GH_other_size]           = (
    (data[:, :mkt_n] .- data[:, :group_n]) .* data[:, :size].^2 .+
    (data[:, :size_sqr_sum_mkt] .- data[:, :size_sqr_sum_own]) .-
    2 .* data[:, :size] .* (data[:, :size_sum_mkt] .- data[:, :size_sum_own])
);
```

```
NIPPYOautoIDvec = [
    260, 4, 76, 104, 64, 54, 152, 153, 71, 197,
    42, 45, 114, 208, 209, 77, 236, 58, 127, 187,
    79, 175, 19, 117, 216, 112, 256, 119, 37, 158
];
```

```
data_NIPPYO = data[in(NIPPYOautoIDvec).(data[:, :NameID]), [:Sales, :price, :hppw, :FuelEf
data_NIPPYO[!, :log_sales] = log.(data_NIPPYO[:, :Sales]);
data_NIPPYO[!, :log_price] = log.(data_NIPPYO[:, :price]);
```

```
data_NIPPYO[!, :log10_sales] = log10.(data_NIPPYO[:, :Sales]);
data_NIPPYO[!, :log10_price] = log10.(data_NIPPYO[:, :price]);
```

```
ols_intro = reg(
    data_NIPPYO,
    @formula(log_sales ~ log_price + hppw + FuelEfficiency + size),
    Vcov.robust()
)
```

```
                                FixedEffectModel
================================================================================
Number of obs:                          196  Converged:                      true
dof (model):                              4  dof (residuals):                190
R²:                                   0.217  R² adjusted:                   0.201
F-statistic:                        19.1148  P-value:                       0.000
================================================================================
                  Estimate  Std. Error    t-stat  Pr(>|t|)    Lower 95%  Upper 95%

log_price       -1.24828     0.309545   -4.03262    <1e-04   -1.85886    -0.63769
hppw            -4.19685     3.97622    -1.05549    0.2925  -12.0401      3.64635
FuelEfficiency   0.0886034   0.0283887   3.12108    0.0021    0.0326059   0.144601
size             0.306896    0.0588927   5.2111     <1e-06    0.190728    0.423063
(Intercept)      5.76211     1.06127     5.42946    <1e-06    3.66873     7.85549
================================================================================
```

```
fit_line = predict(
    lm(@formula(log10_sales ~ log10_price), data_NIPPYO)
    );
plot(
    data_NIPPYO.price,
    data_NIPPYO.Sales,
    seriestype = :scatter,
    legend = false,
    xscale = :log10,
    yscale = :log10,
    xlabel = "Price",
    ylabel = "Sales",
    xticks = ([1, 3, 10], [1, 3, 10]),
)
plot!(
```