Day 4: Demand I

In this lecture, we talked about measuring the response to different demand-side policies.

We will replicate some of the results in "Estimating the Elasticity to Real Time Pricing," by Fabra, Rapson, Reguant, and Wang.

Loading **packages**. Here we load RCall, which will allow us to use some libraries in R.

```
begin
    using PlutoUI
    using DataFrames    , Statistics    ,Missings
    using CSV
    using Plots
    using StatsPlots
    using RCall
    using Binscatters
end

```

Data exploration

Loading **data**.

- data_rtp.csv: Smart meter data of a small sample of 100 consumers.

The data is already merged with several other hourly data that can be either consumer specific (weather) or market specific (solar, wind, prices).

	id	rtp	tou	date	y	m	dom	hr	more
1	8	1	0	20563	2016	4	19	2	
2	8	1	0	20563	2016	4	19	3	
3	8	1	0	20563	2016	4	19	4	
4	8	1	0	20563	2016	4	19	5	
5	8	1	0	20563	2016	4	19	6	

```
begin
    mydata = CSV.read("data_rtp.csv", DataFrame)
    mydata = dropmissing(mydata)
    first(mydata, 5)
end
```

We create some Fourier transforms of time.

	id	rtp	tou	date	y	m	dom	hr	more
1	8	1	0	20563	2016	4	19	2	
2	8	1	0	20563	2016	4	19	3	
3	8	1	0	20563	2016	4	19	4	
4	8	1	0	20563	2016	4	19	5	
5	8	1	0	20563	2016	4	19	6	

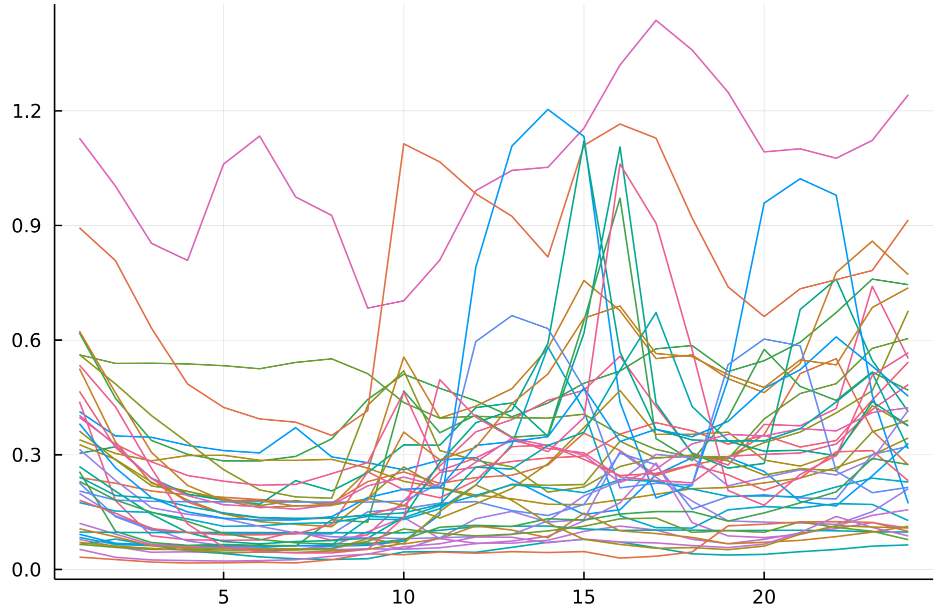
```
begin
    mydata.t = mydata.date + mydata.hr/24 .- 20560.0;
    for x in [1, 7, 365]
        mydata[!, string("tau",x)] = ((mydata.t .+ 0.5)/x) * π * 2.0
        mydata[!, string("tau2",x)] = mydata[!, string("tau",x)].^2.0
        for k = 1:4
            mydata[!, string("cos",k,"tau",x)] = cos.(mydata[!, string("tau",x)]/k)
        end
    end
    select!(mydata,Not(:t));
    first(mydata, 5)
end
```

We rescale variables to avoid problems with Lasso, which is also sensitive to scaling!

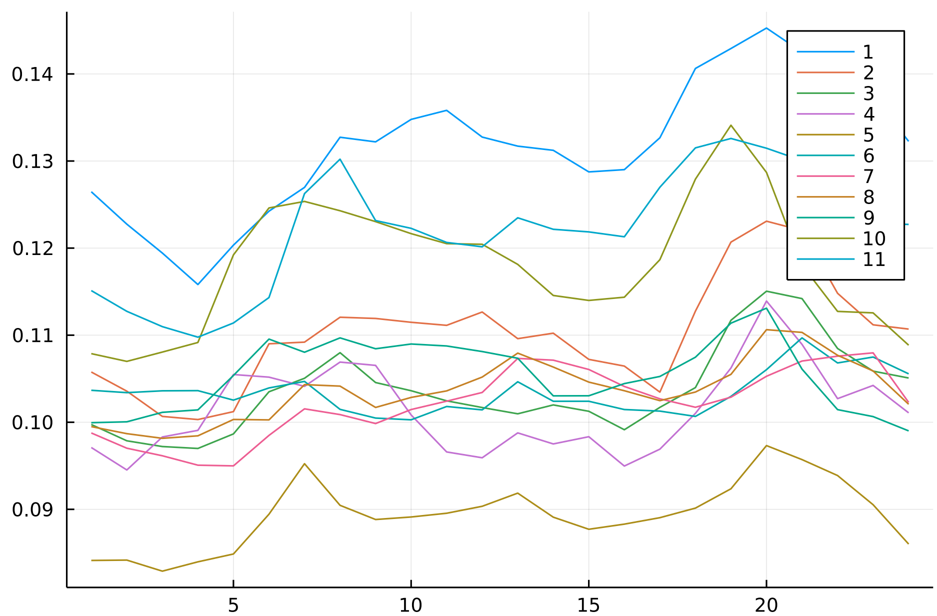
	id	rtp	tou	date	y	m	dom	hr	more
1	8	1	0	20563	2016	4	19	2	
2	8	1	0	20563	2016	4	19	3	
3	8	1	0	20563	2016	4	19	4	
4	8	1	0	20563	2016	4	19	5	
5	8	1	0	20563	2016	4	19	6	

```
begin
    # we rescale some variables as lasso can go "bananas"
    for v in names(mydata[!,Between(:wind_hat,:cos4tau365)])
        mu = mean(mydata[!,v]);
        sigma = std(mydata[!,v]);
        mydata[!,v] = (mydata[!,v] .- mu)/sigma
    end
    first(mydata, 5)
end
```

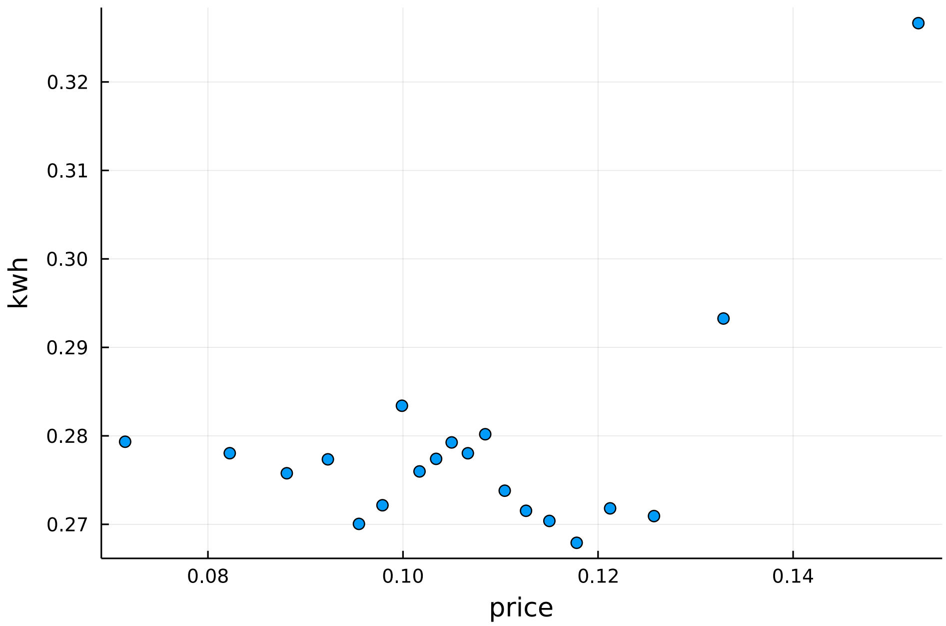
It can be useful to plot the data to examine patterns. We can plot the typical consumption pattern of consumers during the day.



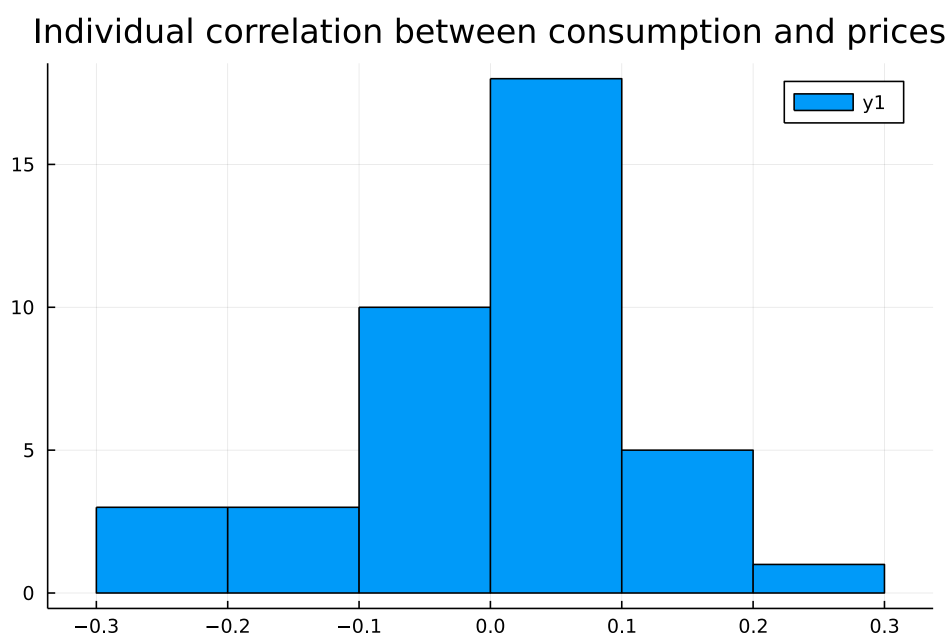

```
let
    # here we can learn about ways of collapsing data in Julia
    # lots going on here! see suggested exercise below

    df_plt = select(mydata,[:id,:hr,:kwh])
    df_plt = combine(groupby(df_plt, [:id,:hr]), :kwh => mean)
    @df df_plt plot(:hr, :kwh_mean, group=:id, legend=false)

end
```




```
let
    # we can also plot prices
    df_plt = select(mydata,[:hr,:price, :m])
    df_plt = combine(groupby(df_plt, [:hr,:m]), :price => median)
    @df df_plt plot(:hr, :price_median, group=:m)

end
```




```
let
    # we can plot the raw correlation of consumption and prices
    # interesting that not particularly correlated with price
    binscatter(mydata,  @formula(kwh ~ price + fe(id)))
end
```




```
let
    # let's look at individual correlations
    ids = unique(mydata.id);
    corrs = [cor(mydata[mydata.id.==i,:].price,
            mydata[mydata.id.==i,:].kwh) for i in ids];
    histogram(corrs, title="Individual correlation between consumption and prices")
end
```

```
let
    # let's look at individual correlations
    dates = unique(mydata.date);
    corrs = [maximum(mydata[mydata.date.==i,:].price)/
        minimum(mydata[mydata.date.==i,:].price) for i in dates];
    #histogram(corrs, title="Individual correlation between consumption and prices")
    scatter(dates, corrs)
    Plots.savefig("scatter_diffs.pdf");
end
```

Estimation of elasticities

Define the identifiers and create vector to store estimation results

```
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0
```

```
begin
    sample = unique(mydata.id)
    beta_Hh=zeros(length(sample))
    se_Hh=zeros(length(sample))
end
```

We will be running the regression in R. Julia allows you to call R packages, as well as have access to R REPL mode.

- Make sure you have the needed libraries installed in R (`hdm` , `dplyr`).
- Use **@rput** to send Julia objects to R.

```
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0
```

```
begin
    @rput sample
    @rput mydata
    @rput beta_Hh
    @rput se_Hh
end
```

```
"solar_actual + temp + temp2 + mwh_dayaheadiberia + tau1 + tau21 + cos1tau1 + cos2tau1 + co
```

```
begin
    controls = join(names(mydata[!,Between(:solar_actual,:cos4tau365)]), " + ");
    @rput controls
end
```

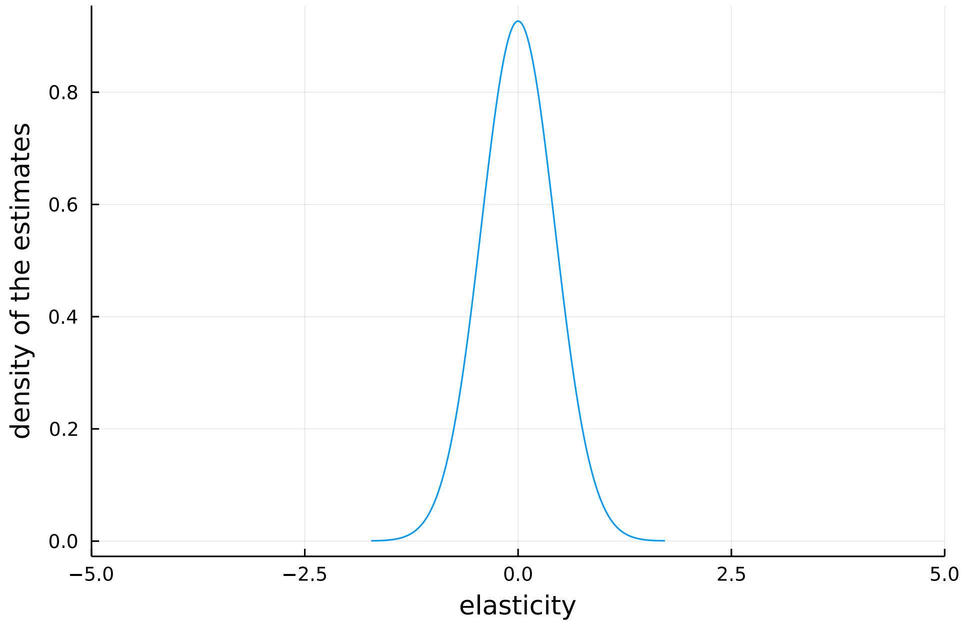
To insert a chunk of R code, use R and triple quotes.

Note that in the following chunk, the loop is defined in Julia, while the regression calls an R package.

We run a separate regression per household.

```
begin
    for i in 1:length(sample)
        @rput i
         R"""
            library(hdm)
            library(dplyr)
            iv.reg = rlassoIV(as.formula(paste0("log(kwh+.01) ~
                log(price+.01) + ", controls, " |
                wind_hat + ", controls)),
                data = filter(mydata,id==sample[i]),
                select.X = TRUE, select.Z = FALSE)
            beta1<-iv.reg$coef
            beta_Hh[i]<-beta1
        """
        @rget beta1
        beta_Hh[i] <- beta1
    end
end
```

We can plot the distribution of our household estimates




```
begin
    density(beta_Hh,legend = false)
    plot!(xlab="elasticity",ylab="density of the estimates",xlim=(-5,5))
end

```

RTP vs Non-RTP elasticities

Next, we will check whether consumers under RTP have different price elasticities.

For that, we split our data between household with and without RTP.

The rest is similar than the previous exercise.

	estimate	rtp
1	0.0	0.0
2	0.0	0.0
3	0.0	0.0
4	0.0	0.0
5	0.0	0.0
6	0.0	0.0
7	0.0	0.0
8	0.0	0.0
9	0.0	0.0
10	0.0	0.0
	more	
40	0.0	1.0

```
begin
    data0=mydata[in([0]).(mydata.rtp), :];
    data1=mydata[in([1]).(mydata.rtp), :];

    sample0=unique(data0.id);
    sample1=unique(data1.id);

    betas=DataFrame(estimate=zeros(length(sample)),
                    rtp=[zeros(length(sample0));ones(length(sample1))]);
end
```

	estimate	rtp
1	0.0	0.0
2	0.0	0.0
3	0.0	0.0
4	0.0	0.0
5	0.0	0.0
6	0.0	0.0
7	0.0	0.0
8	0.0	0.0
9	0.0	0.0
10	0.0	0.0
	more	
40	0.0	1.0

```
begin
    @rput sample0
    @rput data0
    @rput sample1
    @rput data1
    @rput betas
end
```

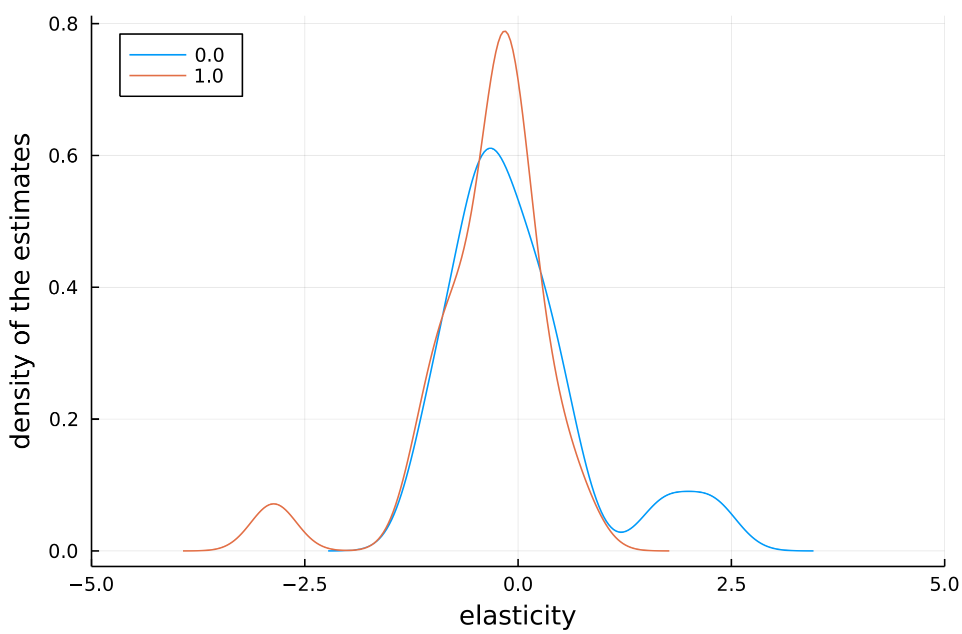
First, we estimate our model for consumers withour RTP (pay attention to the data use in the regression command)

```
begin
    for i in 1:length(sample0)
        @rput i
     R"""
    library(hdm)
    library(dplyr)
        iv.reg =  rlassoIV(as.formula(paste0("log(kwh+.01) ~
        log(price+.01) + ", controls, " |
        wind_hat + ", controls)),
        data = filter(data0,id==sample0[i]),
        select.X = TRUE, select.Z = FALSE)
    beta1<-iv.reg$coef
    """
        @rget beta1
        betas[i,1]=beta1
    end
end
```

We do the same for consumers with RTP=1.

```
begin
    for i in 1:length(sample1)
        @rput i
     R"""
    library(hdm)
    library(dplyr)
        iv.reg =  rlassoIV(as.formula(paste0("log(kwh+.01) ~
        log(price+.01) + ", controls, "  |
        wind_hat + ", controls)),
        data = filter(data1,id==sample1[i]),
        select.X = TRUE, select.Z = FALSE)
    beta1<-iv.reg$coef
    """
        @rget beta1
        betas[(length(sample0)+i),1]=beta1
    end
end
```

Again, we can plot the density of each group of estimates and check whether there are significant differences.




```
begin
    @df betas density(:estimate, group = :rtp,legend = :topleft)
    plot!(xlab="elasticity",ylab="density of the estimates",xlim=(-5,5))
end
```

Follow-up exercises

1. Include the consumption of non-RTP households as a potential control to the Lasso, as in Burlig et al. (2020). What can be some challenges without a pre-treatment period?
2. Explore your ML method of choice as an alternative method to estimate the elasticities.
3. Use the Clustering.jl library we used on day 2 to classify consumers into "typical" profiles. This can be a useful way of reducing the dimensionality of the data. We will do something like this on day 5 as well.