Hint: the following packages will be useful in solving this prolem set.

In [164...
```
using Optim
using Statistics
using ForwardDiff
using Plots
using LinearAlgebra
```

Problem Set 2

Due: May 3, 2021 (in class; subject to change if COVID restrictions apply)

A binary response is a variable that takes on only two values, customarily 0 and 1, which can be thought of as codes for whether or not a condisiton is satisfied. For example, 0=drive to work, 1=take the bus. Often the observed binary variable, say y, is related to an unobserved (latent) continuous varable, say y^*. We would like to know the effect of covariates, x, on y. The model can be represented as

$$\begin{aligned} y^* &= g(x) - \varepsilon \\ y &= 1(y^* > 0) \\ Pr(y=1) &= F_\varepsilon[g(x)] \\ &\equiv p(x,\theta) \end{aligned}$$

For the logit model, the probability has the specific form

$$p(x,\theta) = \frac{1}{1+\exp(-x'\theta)}$$

Problem 1 (MLE)

We will consider maximum likelihood estimation of the logit model for binary 0/1 dependent variables. We will use the BFGS algorithm to find the MLE.

The log-likelihood function is

$$s_n(\theta) = \frac{1}{n}\sum_{i=1}^{n}\left(y_i \ln p(x_i,\theta) + (1-y_i)\ln[1-p(x_i,\theta)]\right)$$

The following code generates that follow a logit model with given θ:

In [191...
```
function LogitDGP(n, theta)
    k = size(theta,1)
    x = ones(n,1)
    if k>1
        x = [x  randn(n,k-1)]
    end
    y = Float64.((1.0 ./ (1.0 .+ exp.(-x*theta)) .> rand(n,1)))
    return y, x
end
```

```
LogitDGP (generic function with 1 method)
```

Out[191…

Let us estimate $\hat{\theta}$ from the dataset with 100 points ($n = 100$) and generated from the true θ (θ_0) value of $[0.5, 0.5]$.

In [192…

```
n=100
theta = [0.75,0.25]
(y,x) = LogitDGP(n,theta)
```

Out[192…

```
([0.0; 1.0; … ; 0.0; 0.0], [1.0 -1.1194714896206426; 1.0 0.7907570846532597; … ; 1.0 -0.
18735590485531636; 1.0 -1.340239374172393])
```

(1. a) Estimate $\hat{\theta}$.

Hint:

1. Refer to Nerlove lecture notes for an example code for mle estimation.
2. Code for the log likelihood function

$$s_n(\theta) = \frac{1}{n}\sum_{i=1}^{n}(y_i \ln p(x_i,\theta) + (1-y_i)\ln[1-p(x_i,\theta)])$$

   is written as below:

(1. b) Empirically prove consistency of $\hat{\theta}$ by increasing the number of n in DGP and re-estimate.

Hint: Refer to GMM lecture notes for an example code for empirically proving consistency.

(1. c) Empirically prove asymptotic normality of $\hat{\theta}$ by repeatedly generate data.

Hint: Refer to GMM lecture notes for an example code for empirically proving asymptotic normality.

Problem 2 (GMM)

Recall from GMM lecture notes:

Suppose the model is

$$\begin{aligned} y_t^* &= \alpha + \rho y_{t-1}^* + \beta x_t + \epsilon_t \\ y_t &= y_t^* + \upsilon_t \end{aligned}$$

where ϵ_t and υ_t are independent Gaussian white noise errors. Suppose that y_t^* is not observed, and instead we observe y_t. If we estimate the equation

$$y_t = \alpha + \rho y_{t-1} + \beta x_t + \nu_t$$

this the estimator is biased and inconsistent.

What about using the GIV estimator?

Consider using as instruments $Z = [1\ x_t\ x_{t-1}\ x_{t-2}]$. The lags of x_t are correlated with y_{t-1} as long as β is different from zero, and by assumption x_t and its lags are uncorrelated with ϵ_t and υ_t (and thus they're also uncorrelated with ν_t). Thus, these are legitimate instruments. As we have 4 instruments and 3 parameters, this is an overidentified situation.

In [238…

```
function lag(x::Array{Float64,2},p::Int64)
        n,k = size(x)
        lagged_x = [ones(p,k); x[1:n-p,:]]
end

function lag(x::Array{Float64,1},p::Int64)
        n = size(x,1)
        lagged_x = [ones(p); x[1:n-p]]
end


function  lags(x::Array{Float64,2},p)
        n, k = size(x)
        lagged_x = zeros(eltype(x),n,p*k)
        for i = 1:p
                lagged_x[:,i*k-k+1:i*k] = lag(x,i)
        end
    return lagged_x
end

function  lags(x::Array{Float64,1},p)
        n = size(x,1)
        lagged_x = zeros(eltype(x), n,p)
        for i = 1:p
                lagged_x[:,i] = lag(x,i)
        end
    return lagged_x
end
```

Out[238…

```
lags (generic function with 2 methods)
```

Given $[\alpha_0, \rho_0, \beta_0] = [0, 0.9, 1]$, let us generate data using the pre-defined lag function above:

In [280…

```
n = 100
sig = 1

x = randn(n) # an exogenous regressor
e = randn(n) # the error term
ystar = zeros(n)
# generate the dep var
for t = 2:n
  ystar[t] = 0.0 + 0.9*ystar[t-1] + 1.0*x[t] + e[t]
end
# add measurement error
y = ystar + sig*randn(n)
ylag = lag(y,1)
data = [y ylag x];
data = data[2:end,:] # drop first obs, missing due to lag
theta = [0, 0.9, 1]
```

Out[280…

```
3-element Array{Float64,1}:
 0.0
 0.9
 1.0
```

(2. a) Given the following GIVmoments function, write down the moment conditions for each data point. In other words, write down $m_t(\theta)$ $\forall t$ where t is an index for each data points and

$\bar{m}_n(\theta)$.

In [281…

```
# moment condition
function GIVmoments(theta, data)
        data = [data lags(data,2)]
    data = data[3:end,:] # get rid of missings
        n = size(data,1)
        y = data[:,1]
        ylag = data[:,2]
        x = data[:,3]
        xlag = data[:,6]
        xlag2 = data[:,9]
        X = [ones(n,1) ylag x]
        e = y - X*theta
        Z = [ones(n,1) x xlag xlag2]
        m = e.*Z
end
```

Out[281…

```
GIVmoments (generic function with 1 method)
```

(2. b) Calculate $\hat{\theta}$ using two step GMM.

Hint: use the following code to generate moment.

Compare the estimates with the true parameter value: $[\alpha_0, \rho_0, \beta_0] = [0, 0.9, 1]$

(2. c) Empirically prove consistency of $\hat{\theta}$ from two step GMM by increasing the number of n in DGP and re-estimate.