**Empirical exercise** – The GMM objective function and gradient of it

The code in the function file is reproduced at the end of this file. The second line defines the error vector `e`.

Using the error vector `e` and the instrument vector `Z`, the third line constructs the sample moments. In particular, `m_bar` contains a set of sample moments. It contains `N_mom` sample moments, and hence it is a `N_mom` × `1` vector. The first sample moment is the zero mean moment for the error term. The other sample moments result from the requirement that each of the `N_mom` variables in `Z` is uncorrelated with the error `e`. Note that we have `N_par` parameters to estimate in `B_true`. If the number of moments is larger than the number of parameters to estimate, the model is over-identified.

Using the sample moments, the fourth line constructs the GMM criterion function. We consider a quadratic form of `m_bar`. This is our objective function. The unique `B_hat` that solves this function is the GMM estimator. We ask `fminunc` to evaluate this function at `B_true`. `Obj` takes a scalar value at the minimising `B_hat`.

`G_bar` is the derivative of the sample moment vector with respect to parameter vector `B_true`.

`GradObj` is the gradient of the objective function. The gradient is the partial derivatives of the objective function with respect to each element of `B_true`. That is, the component $i$ of the gradient is the partial derivative of the function with respect to the component $i$ of `B_true`. `B_true` has `N_par` elements, and therefore the gradient has `N_par` elements. The gradient is evaluated at `B_true` because `G_bar` is evaluated at `B_true`.

Given the arguments defined above, we can now instruct MATLAB that our function is `exercisegmmfunction`, and it accepts `y`, `X`, `Z`, `B_true`, `W_hat`, `N_obs`, and `N_par` as input arguments, and returns `Obj` and `GradObj` as output arguments.

---

```
function [Obj,GradObj] = exercisegmmfunction(y,X,Z,B_true,W_hat,N_obs,N_par)
e = y-exp(X*B_true);
m_bar = (1/N_obs)*Z'*e;
Obj = m_bar'*W_hat*m_bar;
G_bar = -(1/N_obs)*Z'*(X.*repmat(exp(X*B_true),[1 N_par]));
GradObj = 2*G_bar'*W_hat*m_bar;
end
```