Economics 217 - Applied Econometrics II

- Professor: Alan Spearot
  - Email: aspearot@ucsc.edu
  - Office Hours: 10AM-12PM Monday, 459 Engineering 2 (on Wednesday during the first week)
- TA: Jijian Fan
  - Email: jifan@ucsc.edu
  - Section times: Wednesday 1:20-2:25pm, and Friday 10:40-11:45am, Earth & Marine B210
  - Office Hours: TBD
- Take-home coding projects
  - Project 1: February 7th and 8th
  - Project 2: March 14th and 15th

Economics 217 - Applied Econometrics II

- Topics covered in this course
  - Generalized Linear and Non-linear Models
  - Non-parametric models
  - Resampling techniques
  - Learning models and Data Mining
  - Time series econometrics
- Books and course materials:
  - Asteriou and Hall for time series (same book as 216)
  - Online handouts and vignettes
- Programing
  - R will be the backbone of the course (www.r-project.org)

Economics 217 - Applied Econometrics II

- Basic course expectations
  - There will be some derivations. I will try to keep the course as applied as possible but the course is meant to be more than just programming.
  - Too many people just load data into a computer and look at the results without thinking about where it came from or what the results mean.
  - Code will be presented in class. It is not expected that you sit here with your computer open running code with me (that is for 294A). However, you will need it for the homework and beyond so it is a critical part of the course.
- Assignments
  - Timed take home coding projects will be assigned at two points during the quarter. You are expected to work alone on these.
  - You may work in groups to work on assignments, but you must turn in original work for homework answers. If I see identical work being turned in, we'll have a chat in my office.
  - Explicit cheating will be dealt with harshly.

Linear models to generalized linear models

- In 216, the primary focus was estimation and inference using *linear models*

$$y_i = \mathbf{x}_i^T \beta + u_i$$

  - y_i: dependent variable
  - $\mathbf{x}_i$: $p \times 1$ vector of independent variables
  - β: $p \times 1$ coefficients on the independent variables
  - u_i: error term for individual i.

- An similar representation that will be useful going forward is

$$\mu_i = \mathbf{x}_i^T \beta, \quad where \quad Y_i \sim F(\mu_i)$$

- That is, $\mathbf{x}_i^T \beta$ represents the average of another variable Y_i, where Y_i is random around this average according to the distribution function $f()$

- For the most part, we will assume that observations are independent from one another.

Linear models to generalized linear models (cont)

- There are two ways to estimate the canonical linear model
- Linear regression
  - the command "lm" in R estimates the linear model using linear regression
  - Standard errors assume homoskedasticity, though there are ways to allow for clustering and heteroskedasticity
- Maximum likelihood, also known as normal regression
  - the command "glm" in R estimates the normal regression as a default
  - Why this command is called "glm" is the purpose of the next few lectures.
- What were some of the positives and negatives of linear models, either linear regression or normal regression?

Two ways of estimating the linear model in R

- In R, everything (usually) starts with something called a "data frame"
- For example, if I want to load a CSV file of data and call it "classdata", I write:

classdata<-read.csv(filename,header=TRUE)

  - "filename" is the location of the file on your hard drive, server, website, whatever...
  - "<-" is the assignment operator. The thing on the right is assigned to the thing on the left. This works for a variety of data types in R.
  - "header=TRUE" indicates that the first row of the file identifies column names. This obviously won't be used if there are no headers, in which case you can assign variable names later.

- After loading what is called the "foreign" library, one can load stata .dta files in R

classdata<-read.dta(filename)

- There are a variety of ways to load other data formats - see online documentation.

Current Population Survey

- The CPS is a widely used and influential dataset from the US Bureau of Labor Statistics
- The Center for Economics and Policy Research (CEPR) has cleaned this data for use in a way which is consistent over time
  - http://ceprdata.org/cps-uniform-data-extracts/
  - Outgoing Rotational Group: 1979-2013.
  - Data on Wages, Demographics, locations, Industries, Occupations, Labor Force Participation, Education, etc...
  - We'll combine surveys from 1983, 1988,...2008, 2013 to a "pooled cross-section".
- On the course website, this pooled cross-section is listed as "Org.data"
- To load it in R, I write:

```
d<-read.dta("/Users/acspearot/Data/CPSDWS/org_example.dta")
```

Summarizing Data in R

- Though much of these techniques are better learned in lab or on your own, I will be going over some basic commands in R to summarize and manipulate data.

- The first lists the contents and variable types within the dataset, and is called "str":

```
str(d)
```

- The second is appropriately named "summary":

```
summary(d)
```

- Summary can be used on individual variables:

```
summary(d$rw)
```

  - The $ picks off a particular variable within the data frame
  - "rw" is the real wage in the org dataset

Two ways of estimating the linear model in R (cont)

- First, let's run the linear regression using the command:

```
wage.lm<-lm(log(rw)~age,d))
```

- Here, log(rw)~age is the regression equation.
  - *Age* is in years
  - *log*(*rw*) is the natural log of the real wage
- To time the procedure, you can instead run the following:

```
ptm <- proc.time()
wage.lm<-lm(log(rw)~age,d)
proc.time() - ptm
```

  - *proc.time*() reports the time when this function is called
  - *proc.time*() − *ptm* subtracts the previous time from the current time.

Two ways of estimating the linear model in R (cont)

- Next, let's run the normal regression using the **glm** command (using the timing command from earlier):

```
ptm <- proc.time()
wage.lm<-glm(log(rw)~age,d,family=gaussian))
proc.time() - ptm
```

- Again, log(rw)~age is the regression equation.
- family=gaussian indicates that we are using a gaussian (normal) distribution to model the error term
- GLM is estimated by maximum likelihood (as we'll see).
- What are the differences in estimates, t-statistics, standard errors, and regression performance?

Linear models to generalized linear models

- In many cases, as we discussed earlier, the canonical linear model is too simple.
- Thankfully, there is a parsimonious and powerful extension of the linear model called the *generalized linear model*

$$g(\mu_i) = \mathbf{x}_i^T \beta$$

- $g()$ is called the **link function**. Naturally, it links the mean of the observed variable to the linear function of covariates.
  - This is a crucial step in enriching our models. $g()$ bounds the link between observable data and outcomes.
  - What technique from 216 does this remind you of?
- Where this all becomes "generalized" is when we introduce a wide class of distributions that have nice properties within this setup.

The exponential family of distributions

- Generalized linear models are based (in part) on the *exponential family of distributions*
- The canonical exponential family is any distribution that can be arranged as.

$$f(y;\theta) = \exp(yb(\theta) + c(\theta) + d(y))$$

  - y is the random variable
  - θ is some parameter of interest.

- A nice property is that the exponential family is log-additive. That is, take logs of $f(y;\theta)$

$$\log(f(y;\theta)) = yb(\theta) + c(\theta) + d(y)$$

- This makes manipulating the Log-likelihood function particularly useful.
- The generalized exponentional family is written as:

$$f(y;\theta) = \exp(g(y)b(\theta) + c(\theta) + d(y))$$

The exponential family: Gaussian

- Many distributions are of the exponential family.
- Consider the normal distribution

$$f(y;\mu) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left[-\frac{1}{2\sigma^2}(y-\mu)^2\right]$$

- Simplify by expanding $(y-\mu)^2$

$$f(y;\mu) = \exp\left[-\frac{y^2}{2\sigma^2} + \frac{y\mu}{\sigma^2} - \frac{\mu^2}{2\sigma^2} - \frac{1}{2}\log\left(2\pi\sigma^2\right)\right]$$

- σ^2 is considered a nuisance parameter, and the focus is on μ.

$$b(\mu) = \frac{\mu}{\sigma^2} \quad , \quad c(\mu) = -\frac{\mu^2}{2\sigma^2} - \frac{1}{2}\log\left(2\pi\sigma^2\right) \quad , \quad d(y) = -\frac{y^2}{2\sigma^2}$$

- These functions will be used later when estimating equations based around these distributions.

The exponential family: Gaussian in R

- In R, there are two ways to view distributions of random variables.
- The first is to use the function and plot the function using a sequence of numbers.
- For the Gaussian distribution, this is done with the following commands:

1. Create a vector of length 1000 evenly spaced between -10 and 10.

```
x<-seq(-10,10,length.out=1000)
```

2. Insert this vector into the standard normal distribution

```
pdf<-dnorm(x,mean = 0, sd = 1, log = FALSE)
```

3. Plot the PDF as a function of x.

```
plot(pdf x, ylab="pdf",xlab="x",main="PDF of Standard normal")
```

The exponential family: Gaussian in R

- For my tastes, the best way to view distributions is to generate a large vector of random variables, and then view it with a histogram or a kernel density.
- For the Gaussian distribution, first generate a random vector of length 1000 using the command 'rnorm'

```
x<-rnorm(1000,mean = 0, sd = 1)
```

- Then, plot the density using the plot command

```
plot(density(x),xlab='x',ylab='pdf of x')
```

- We'll learn about the technical details of kernel densities during the *non-parametric* section of the course, but basically we are estimating a smooth relationship between one variable (pdf) and the other (x).
- Why do you think this estimate looks a tad off? What can be done about it?

The exponential family: Binomial

- As you learned last quarter with the probit and logit, not all variables are continuous and in some cases they are dichotomous.

- The binomial distribution represents such variables, where p is the probability some event occurs:

$$f(y;p) = \binom{n}{y} p^y (1-p)^{n-y}$$

- y is the number of times the event has occurred after n trials.

- As a reminder, n "choose" y is $\binom{n}{y} = \frac{n!}{y!(n-y)!}$.

- p is the parameter of interest. Rearranging, we get:

$$f(y;p) = \exp\left[y\log(p) + \log(1-p)(n-y) + \log\binom{n}{y}\right]$$

- Collecting y's

$$f(y;p) = \exp\left[y\log\left(\frac{p}{1-p}\right) + n\log(1-p) + \log\binom{n}{y}\right]$$

- Here, $b(p) = \log\left(\frac{p}{1-p}\right)$, $c(p) = n\log(1-p)$, and $d(y) = \log\binom{n}{y}$

The exponential family: Binomial in R

- To visualize this distribution, like before, generate a random vector of length 1000 but instead using the command 'rbinom'

```
x<-rbinom(n=1000, size=100,p=0.5)
```

- Three components of this function
  - size=100 is the number of trials per observation
  - n=1000 represents the number of observations (length of the random vector)
  - p=0.5 is the probability of the event occuring
- Then, plot the density using the plot command

```
plot(density(x),xlab='x',ylab='pdf of x')
```

- What does this distribution look like?

The exponential family: Poisson

- The Poisson distribution is another extremely important distribution, commonly used to represent count variables.
  - It can also be used with continuous data to bound variables below at zero.
- Assume for now that y is a discrete random variable taking on values 0 and higher. The poisson distribution is written as:

$$f(y;\theta) = \frac{\theta^y \exp[-\theta]}{y!}$$

- Quick simplification yields:

$$f(y;\theta) = \exp[y\log(\theta) - \theta - \log(y!)]$$

- To generate this in R, use

```
x<-rpois(n,theta)
```

  where n is the length of the random vector.

- Then, plot as before (note R calls theta "lambda").

Logit and Probit in GLM

- You learned (last quarter) how to estimate the logit and probit models. Both can be structured using the GLM framework
- Both Probit and Logit are of the binomial family.
  - With the binomial distribution, p is the probability that an event occurs, and p is also the mean of the dichotomous variable, μ.
- The "link" function tells use how we link μ_i to the vector of explanatory variables
- For the logit model, use the logit function for the link:

$$g(\mu) = \log\left(\frac{\mu_i}{1-\mu_i}\right) = \mathbf{x}_i^T\beta$$

- Exponentiating both sides:

$$\left(\frac{\mu_i}{1-\mu_i}\right) = \exp\left(\mathbf{x}_i^T\beta\right)$$

- Rearranging for μ

$$\mu_i = \frac{\exp\left(\mathbf{x}_i^T\beta\right)}{1+\exp\left(\mathbf{x}_i^T\beta\right)}$$

  which is the familiar logit formula.

R Examples - Logit and Probit

- For probit, use the inverse of the standard normal for the link:

$$\begin{aligned} & g(\mu_i) &=& \Phi^{-1}(\mu_i) \\ \Rightarrow \quad & \mu_i &=& \Phi\left(\mathbf{x}_i^T \beta\right) \end{aligned}$$

- To estimate a logit model in R, use

```
glm(y~x,classdata,family=binomial(link="logit"))
```

- This looks like any other regression with the exception of the family/link component
  - family=binomial() indicates that we have a dichotomous variable, one or zero
  - the "link" function tells use how we link the mean to the vector of explanatory variables
  - If we are using profit, replace "logit" with "probit" in code

Estimating Logit and Probit

- Let's now estimate a logit model using the Org dataset.
- In the Org dataset, *nilf* is an indicator variable taking on the value of 1 if the respondent is not in the labor force
  - This is crucial of estimates of unemployment, since to be unemployed you must be in the labor force
  - Also, *nilf* will also include "discouraged workers", in the sense that fluctuations in this variable will represent cyclical changes in willingness to search for work.
- To estimate the Logit model, use:

```
glm_logit<-glm(nilf~age,d,family=binomial(link="logit"))
```

- For Probit:

```
glm_probit<-glm(nilf~age,d,family=binomial(link="probit"))
```

- R provides the estimates for β's. Use "summary" to report standard errors and such.
- To compute marginal effects, additional commands should be used.

Aside on estimating Logit and Probit with factor variables

- So far, our regressions with the Org dataset have been pretty simple
  - No dummy variables other other factors to absorb variation in the dependent variable or test other hypotheses
  - Not using panel data to examine variation within individuals
- In R, *factor variables* can be used to easily estimate fixed effects
  - We're going to ignore one major problem for a while (what is called "incidental parameters") when estimating fixed factors using maximum likelihood, and instead focus on syntax
- In Org, *educ* is a factor variable identifying maximum education level as less than high school, high school degree, some college, college degree, and advanced
- We can add education to our model using the following

```
glm_logit_ed<-glm(nilf~age+educ,d,family=binomial(link="logit"))
glm_probit_ed<-glm(nilf~age+educ,d,family=binomial(link="probit"))
```

GLM Extensions: Poisson Regression

- Similar to the logit-probit models, we often wish to constrain the response to explanatory variables to feasible values
- Many variables have a distribution that is non-negative $y \in [0, \infty)$
  - wages cannot be non-negative (unless you're a musician)
  - Time or durations are non-negative
  - Any count data is non-negative
- The Poisson regression is (sometimes) suitable to estimate these types of models
- Poisson has mostly been used to estimate count and simple duration models, but recently has been applied to any model with a non-negative dependent variable.
  - Informally, dollars and cents are count variables, right?
  - More formally, there is a technique called "Pseudo-Poisson Maximum Likelihood" that we will cover if there is time.

GLM Extensions: Poisson Regression

- Recall that the Poisson distribution is written by:

$$f(y;\theta) = \exp[y\log(\theta) - \theta - (\log y!)]$$

- We will show later that the mean of the Poisson distribution is:

$$\mathrm{E}(y) = \theta$$

- So, the distribution can be rewritten as:

$$f(y;\mu) = \exp[y\log(\mu) - \mu - (\log y!)]$$

- We now need a link function. The log-linear model is the typical choice:

$$g(\mu) = \log(\mu) = x\beta$$

- Thus, the mean of the Poisson distribution is bounded by the exponential function:

$$\mathrm{E}[y|x] = \mu = \exp(x\beta)$$

GLM Extensions: Poisson Regression

- To estimate the Poisson regression, the typical syntax is

```
glm(y~x,classdata,family=poission(link="log"))
```

- Within the Org dataset, we will predict hours worked (*hourslw*) with age and the level of education for all individuals in the labor force (but not necessarily working).

- To make estimation go a bit more quickly, we first restrict the sample to include:
  - Only residents from California
  - Only respondents from the 2013 survey
  - Only individuals in the labor force

- This is done with:

```
subd<-subset(d,nilf==0&year==2013&state=="CA")
```

- "subd" is the smaller dataframe we will work with.

GLM Extensions: Poisson Regression

- Next, note that some individuals have "NA" listed as hours worked. We want to change these to zero using the following command

```
sub$hourslw<-ifelse(is.na(subd$hourslw),0,subd$hourslw)
```

- Since hours worked is bounded below at zero, Poisson is a reasonable approach to estimate the determinants of hours worked.

- Estimate the Poisson model, and compare to the normal regression

```
normalreg<-glm(hourslw~age+educ,subd,family=gaussian)
summary(normalreg)
poissonreg<-glm(hourslw~age+educ,subd,family=poisson(link="log"))
summary(poissonreg)
```

- What are the differences in residuals and regression diagnostics?