Problem Set 4

Fall 2019

*The assignment is worth **100 points**. There are **16 questions**. You should have the following packages installed:*

```
library(tidyverse)
library(cowplot)
#library(kableExtra)
#library(margins)
```

You may need to update your version of R and follow these steps to download `margins`:

```
if (!require("remotes")) {
  install.packages("remotes")
  library("remotes")
}
install_github("leeper/prediction")
install_github("leeper/margins")
```

In this problem set you will summarize the paper "Imperfect Public Monitoring with Costly Punishment: An Experimental Study" (Ambrus and Greiner, AER 2012) and recreate some of its findings.

Big picture

1. What is the main question asked in this paper?

2. Summarize the experiment design.

3. Summarize the main results of the experiment.

4. Why are these results valuable? What have we learned? Motivate your discussion with a real-world example. In particular discuss the tradeoffs to transparency in groups and how these tradeoffs might be navigated in a firm, or more broadly, a society.

5. If punishment is ineffective under imperfect monitoring, what else can you lean on to ensure people cooperate (at least a little) in a public goods problem?

Theory

Payoffs to agent i are

$$\pi_i = (e_i - x_i) + \alpha \sum_{i=1}^{n} x_i$$

where e_i is the agent's endowment, x_i is her contribution to the public good, α is the marginal per capita return, and n is the group size.

5. Explain α and why in public goods game requires $\frac{1}{n} < \alpha < 1$.

6. Suppose $e_i = e = 20$ (i.e. everyone has 20), $\alpha = 0.4$ and $n = 4$. Show that $x_i = 0$ is a symmetric Nash equilibrium, but $x_i = 20$ is the social optimum. (Recall that in a Nash equilibrium i cannot increase her payoff by changing her contriubtion.)

Replication

Description

Use `theme_classic()` for all plots.

7. Recreate Table 1 and use `kable()` to make a publication-quality table (in HTML).

```
# your code here
```

8. Recreate Figure 1.

```
# your code here
```

9. Recreate Figure 2.

```
# your code here
```

10. Recreate Figure 4.

```
# your code here
```

Inference

Consider the linear model

$$y = \alpha + \beta_1 x_1 + \beta_2 x_2 + \varepsilon$$

11. Write down the marginal effect of x_1.

Now suppose you have a non-linear model

$$y = F(\alpha + \beta_1 x_1 + \beta_2 x_2 + \varepsilon)$$

where $F(\cdot)$ is a "link function" that compresses the inputs so that the output $\hat{y} \in [0, 1]$.

12. Write down the marginal effect of x_1. How does this compare to the marginal effect in the linear model?

13. A probit model uses the Normal CDF Φ as the link function, where $\Phi' = \phi$ is the Normal PDF. Use `glm()` to estimate Model 1 in Table 2 (remember to cluster the standard errors at the group level). Assign the model to the object `m1`. Use `summary()` to view the coefficients. Interpret the coefficients. (For more on the probit model, see the appendix.)

```
# your code here
```

14. Table 2 reports the average marginal effects (AMEs) of the variables on P(contribute). Calculate the AME to the variable `round` as follows:

1. **Use `predict()`to create an object `predictions` that contains the predicted z-scores. (i.e. $\hat{\mathbf{X}\beta}$. Hint: use the option `type="link"` in `predict()`.)**

```
# your code here
```

2. **Use `dnorm()` to calculate the probabilities of the predicted z-scores and store the output in an object called `index`.**

```
# your code here
```

3. **Now calculate the marginal effects by multiplying the predicted probabilities times the estimated coefficient for `round` and store the output in `dydxround`.**

```
# your code here
```

4. **Use `mean()` to calculate the AME.**

```
# your code here
```

15. Verify your calculations with `margins()`, the plot the AMEs. (Note: these will not be exactly the same as those in the paper, since the paper uses an outdated method in Stata.

```
# your code here
```

16. Interpret the AMEs.

Appendix: the probit model

Suppose we have latent response variable

$y^* = \mathbf{X}\beta + \varepsilon$

where $\mathbf{X}$ is a $k \times 1$ vector of features $[x_1\ x_2\ \ldots\ x_k]$ and β is a $1 \times k$ coefficient vector.

The observable binary variable y is defined as

$$
\begin{aligned}
y &= 1 \quad \text{if} \quad y^* > 0 \\
y &= 0 \quad \text{if} \quad y^* \leq 0
\end{aligned}
$$

If we assume that $\varepsilon \sim N(0, 1)$ then

$$
\begin{aligned}
P(y^* > 0) &= P(\mathbf{X}\beta + \varepsilon > 0) \\
&= P(\varepsilon > -\mathbf{X}\beta) \\
&= P(\varepsilon < \mathbf{X}\beta) \quad \text{By symmetry of standard normal} \\
&= \Phi(\mathbf{X}\beta)
\end{aligned}
$$

So $\mathbf{X}\beta$ are z-scores:

$$
\begin{aligned}
P(y = 1) &= P(y^* > 0) = \Phi(z \leq \mathbf{X}\beta) \\
P(y = 0) &= P(y^* \leq 0) = 1 - \Phi(z \leq \mathbf{X}\beta)
\end{aligned}
$$

where Φ is the Standard Normal CDF (e.g. $\Phi(0) = 0.5$; half the standard normal distribution lies below $\mu = 0$).

If we relax the assumption that the error is standard Normal and instead allow it be $\varepsilon \sim N(0, \sigma^2)$, then

$$
\begin{aligned}
P(y^* > 0) &= P(\mathbf{X}\beta + \varepsilon > 0) \\
&= P(\frac{\varepsilon}{\sigma} > \frac{-\mathbf{X}\beta}{\sigma}) \\
&= P(\frac{\varepsilon}{\sigma} < \frac{\mathbf{X}\beta}{\sigma}) \\
&= \Phi(\frac{\mathbf{X}\beta}{\sigma})
\end{aligned}
$$

but we cannot estimate β and σ separately since the probability depends exclusively on their ratio. The standard approach is assume $\sigma = 1$ so ε is a standard normal.