Problem Set for Intro to Linear Models and OLS

[Put your name here]

Instructions

This problem set consists of a few exercises to help familiarize you with linear regression and how to perform regression analysis with R. When you've finished, knit the document and upload to **Moodle**.

This problem set consists of two sections. For the first, you'll answer some open-ended questions about linear regression. These will test your knowledge on some of the basics of linear models and OLS. No math will be required for this section; it will only be a test of your conceptual understanding.

The second section will focus on an applied application of linear regression in R. The goal of this exercise is to introduce some of the basics of doing regression analysis and provide you with some example code to play with and use to analyze an example dataset.

Concepts

Question 1. You have a friend who tells you about an "OLS model" they want to estimate for a research paper. What would you say to ***politely*** correct your friend's misunderstanding of the difference between "OLS" and "models"?

[Delete this text and write your answer here.]

Question 2. What are **two** conditions for OLS estimates to be consistent and unbaised? Can we "prove" whether these conditions hold? Explain.

[Delete this text and write your answer here.]

Question 3. Suppose we have some data on voter turnout and we want to estimate the relationship between income and whether someone voted in an election. To ensure we don't get a biased estimate of this relationship, we need to "control for" an individual's education level in our analysis. What does it mean to "control for" a variable in our analysis? And how should we interpret our estimate of the relationship between income and turnout when we control for education?

[Delete this text and write your answer here.]

Regression analysis in R

The code chunk at the beginning of this document (not visible in the rendered pdf of this document), sets up a dataset called `ds`. Here's what it looks like:

```
head(ds, 5) # shows first 5 rows
```

```
## # A tibble: 5 x 7
##   code  country              year population   gdp human_capital polity
##   <chr> <chr>               <dbl>      <dbl> <dbl>         <dbl>  <dbl>
## 1 AGO   Angola               2017       3.39  12.2          1.47     -2
## 2 ALB   Albania              2017       1.08  10.4          2.95      9
```

```
## 3 ARE   United Arab Emirates  2017       2.24  13.5          2.74     -8
## 4 ARG   Argentina             2017       3.79  13.5          3.04      9
## 5 ARM   Armenia               2017       1.08  10.4          3.13      5
```

For 136 countries, this dataset has the population and GDP of each (log-transformed), their measure of human capital (the Human Capital Index) and their polity 2 score.

Suppose we want to estimate the following model of a country's GDP:

$$\text{GDP}_i = \beta_0 + \beta_1 \text{population}_i + \beta_2 \text{human capital}_i + \beta_3 \text{polity}_i + \beta_4 \text{polity}_i^2 + \epsilon_i.$$

With our dataset, we can easily do this in R by writting:

```
ols_fit <- lm(gdp ~ population + human_capital + polity + I(polity^2),
              data = ds)
```

The above code estimates the model via OLS. We can get a summary of statistical inferences for this model using the `summary()` function on our fitted model object:

```
summary(ols_fit)
```

```
##
## Call:
## lm(formula = gdp ~ population + human_capital + polity + I(polity^2),
##     data = ds)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.3282 -0.3106  0.0369  0.3477  1.7980
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    5.52499    0.24383   22.66  < 2e-16 ***
## population     1.02530    0.03726   27.52  < 2e-16 ***
## human_capital  1.20431    0.09231   13.05  < 2e-16 ***
## polity        -0.05352    0.01089   -4.91  2.6e-06 ***
## I(polity^2)    0.01336    0.00211    6.33  3.7e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.62 on 131 degrees of freedom
## Multiple R-squared:  0.884,  Adjusted R-squared:  0.881
## F-statistic:  250 on 4 and 131 DF,  p-value: <2e-16
```

This gives us the standard errors for our esimates, t-values, and p-values. It also provides some goodness of fit (GOF) metrics.

A problem with the above summary, however, is that it returns classic OLS standard errors. As we discussed in the lecture, these are not robust to *iid* violations.

Thankfully, we can use another version of the `lm` function from the `estimatr` package that will let us easily estimate a model with robust standard errors:

```
robust_fit <- lm_robust(
  gdp ~ population + human_capital + polity + I(polity^2),
  data = ds,
  se_type = "HC1" # This tells the function to return HC1 errors
)
```

We can now get an updated summary of the model:

```
summary(robust_fit)
```

```
##
## Call:
## lm_robust(formula = gdp ~ population + human_capital + polity +
##     I(polity^2), data = ds, se_type = "HC1")
##
## Standard error type:  HC1
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|) CI Lower CI Upper  DF
## (Intercept)     5.5250    0.20976   26.34 3.47e-54  5.11003   5.9399 131
## population      1.0253    0.02960   34.64 8.18e-68  0.96675   1.0839 131
## human_capital   1.2043    0.08418   14.31 1.53e-28  1.03777   1.3708 131
## polity         -0.0535    0.01657   -3.23 1.57e-03 -0.08630  -0.0207 131
## I(polity^2)     0.0134    0.00255    5.24 6.25e-07  0.00831   0.0184 131
##
## Multiple R-squared:  0.884 , Adjusted R-squared:  0.881
## F-statistic:  413 on 4 and 131 DF,  p-value: <2e-16
```

If we want a tydier way to summarize our models, we can use `screenreg` from the `texreg` package:

```
screenreg(
  robust_fit,
  include.ci = FALSE
)
```

```
##
## ==========================
##                 Model 1
## --------------------------
## (Intercept)      5.52 ***
##                 (0.21)
## population       1.03 ***
##                 (0.03)
## human_capital    1.20 ***
##                 (0.08)
## polity          -0.05 **
##                 (0.02)
## polity^2         0.01 ***
##                 (0.00)
## --------------------------
## R^2              0.88
## Adj. R^2         0.88
## Num. obs.      136
## RMSE             0.62
## ==========================
## *** p < 0.001; ** p < 0.01; * p < 0.05
```

We can also produce side-by-side summaries of models with `screenreg`:

```
screenreg(
  list("Classic Fit" = ols_fit,
       "Robust S.E.s" = robust_fit),
  include.ci = FALSE
)
```

```
##
## ========================================
##                Classic Fit  Robust S.E.s
## ----------------------------------------
## (Intercept)      5.52 ***     5.52 ***
##                 (0.24)       (0.21)
## population       1.03 ***     1.03 ***
##                 (0.04)       (0.03)
## human_capital    1.20 ***     1.20 ***
##                 (0.09)       (0.08)
## polity          -0.05 ***    -0.05 **
##                 (0.01)       (0.02)
## polity^2         0.01 ***     0.01 ***
##                 (0.00)       (0.00)
## ----------------------------------------
## R^2              0.88         0.88
## Adj. R^2         0.88         0.88
## Num. obs.      136          136
## RMSE                          0.62
## ========================================
## *** p < 0.001; ** p < 0.01; * p < 0.05
```

Instructions: While polity might be predictive of GDP, it's also possible that GDP determines how democratic (or autocratic) a country is. Estimate a model, or set of models, of democracy using variables in `ds`. Report your results and interpret the coefficients from your model(s). If you estimate more than one model, use your goodness of fit statistics to make an informed choice about which of your models is best. You can use and modify the code above to do your analysis.

```
# Here's a code chunk you can use.
# To produce more code chunks, just type
#   For windows: "ctrl + alt + I"
#   For mac: "Cmd + Option + I"
```