Lab Notes

2017-4-28

Computing OLS Estimates

Thus far, we've used R functions to compute our OLS estimates. Today, we'll look under the hood to help clarify how these estimates are being produced.

Betas

Recall that the least squares estimator is one that minimizes the sum of the squared residuals:

$$SSR(\boldsymbol{\beta}) = \sum_{i=1}^{n}(y_i - \boldsymbol{x}_i'\boldsymbol{\beta})^2$$

This has the solution:

$$\begin{aligned} SSR(\boldsymbol{\beta}) &= \sum_{i=1}^{n}(y_i - \boldsymbol{x}_i'\boldsymbol{\beta})(y_i - \boldsymbol{x}_i'\boldsymbol{\beta}) \\ &= \sum_{i=1}^{n} y_i^2 - 2\boldsymbol{\beta}'\sum_{i=1}^{n}\boldsymbol{x}_i y_i + \boldsymbol{\beta}'\sum_{i=1}^{n}\boldsymbol{x_i x_i'\beta} \end{aligned}$$

$$\begin{aligned} 0 &= \frac{\partial}{\partial\boldsymbol{\beta}} SSR(\hat{\beta}) \\ &= -2\sum_{i=1}^{n}\boldsymbol{x_i} y_i + 2\sum_{i=1}^{n}\boldsymbol{x_i x_i'}\hat{\boldsymbol{\beta}} \end{aligned}$$

$$\hat{\boldsymbol{\beta}} = \left(\sum_{i=1}^{n}\boldsymbol{x_i x_i'}\right)^{-1}\left(\sum_{i=1}^{n}\boldsymbol{x}_i y_i\right)$$

Now let's compute this manually and check to see if it matches our results using the lm() function

```r
library(car)
attach(Duncan)

M2 <- lm(income ~ education + prestige)

x <- cbind(1, education, prestige) # our x matrix is 45 x 3
y <- income # our y vector is 45 x 1

sumxx <- t(x)%*%x # we multiply the transpose of x by x to get a 3 x 3 matrix
invsumxx <- solve(sumxx) # we then find the inverse of this

sumxy <- apply(x*y, 2, sum) # we multiply each value of x by y to get a 45 x 3 matrix,
```

```r
# then sum the columns to get a 1 x 3 matrix

Betas <- invsumxx%*%sumxy # we then multiply the two together to get our Beta estimates
Betas
```

```
##                   [,1]
##            10.42636103
## education   0.03226315
## prestige    0.62372386
```

```r
coefficients(M2)
```

```
## (Intercept)   education    prestige
## 10.42636103  0.03226315  0.62372386
```

We've therefore solved for the $\hat{\boldsymbol{\beta}}$ that minimizes the sum of the squared residuals.

We can alternatively express this in matrix notation:

$$y_i = \boldsymbol{x}_i'\boldsymbol{\beta} + e_i$$

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{e}$$

When using mathematical notation, it is conventional to treat bold lower case letters as vectors and bolded upper case letters as matrices. Thus, y_i and e_i lose their subscripts and are bolded, which is the same thing, and $\boldsymbol{x}_i$ loses its subscript and is made uppercase, which is the same thing.

Our sample sums can also be written in matrix notation.

$$\sum_{i=1}^{n} \boldsymbol{x_i x_i'} = \boldsymbol{X'X}$$

$$\sum_{i=1}^{n} \boldsymbol{x}_i y_i = \boldsymbol{X'y}$$

Therefore the least squares estimator can be alternatively written as

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X'X})^{-1}(\boldsymbol{X'y})$$

```r
X <- cbind(1, education, prestige)
y <- income

xprimex <- t(x)%*%x
xprimexinv <- solve(t(x)%*%x)
xprimey <- (t(x)%*%y)

BetaMat <- xprimexinv%*%xprimey
BetaMat
```

```
##                   [,1]
##            10.42636103
## education   0.03226315
## prestige    0.62372386
```

```r
coefficients(M2)
```

```
## (Intercept)   education    prestige
## 10.42636103  0.03226315  0.62372386
```

Finally, recall the formula for $\hat{\beta}$ with simple linear regression:

$$\frac{Cov(x, y)}{Var(x)}$$

You may have memorized this formula by now, but where do we get this from?

The simple linear regression formula is as follows:

$$y_i = \beta_0 + x_i\beta_1 + e_i$$

Since we only have one non-constant regressor, we can more easily solve for $\hat{\beta}_0$ and $\hat{\beta}_1$ by minimizing the sum of the squared residuals:

$$SSR(\boldsymbol{\beta}) = \sum_{i=1}^{n}(y_i - \beta_0 - x_i\beta_1)^2$$

This has the solutions

$$\frac{\partial SSR}{\partial \beta_0} = \bar{y}_i - \hat{\beta}_1\bar{x}_i$$

$$\frac{\partial SSR}{\partial \beta_1} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} = \frac{Cov(x_i, y_i)}{Var(x)}$$

We can easily check to see that this formula produces the $\hat{\beta}$ using least squares.

```r
M1 <- lm(income ~ education)
coefficients(M1)
```

```
## (Intercept)   education
##  10.6034983   0.5948594
```

```r
Beta1 <- cov(education, income)/var(education)
Beta1
```

```
## [1] 0.5948594
```

```r
Intercept <- mean(income) - Beta1*(mean(education))
Intercept
```

```
## [1] 10.6035
```

This works for simple linear regression, but to compute the $\hat{\beta}$ in multiple regression, we subtstite $\frac{Cov(x_i,y)}{Var(x_i)}$ with $\frac{Cov(\tilde{x}_i,y)}{Var(x_i)}$, where $\tilde{x}_i$ are the residuals from a regression of x_i on all the other covariates.

Sigmas

Now that we've computed $\hat{\boldsymbol{\beta}}$, we'll turn to $\hat{\boldsymbol{\sigma}}^2$. The $\hat{\boldsymbol{\sigma}}^2$ are the estimates of the variances of $\hat{\boldsymbol{\beta}}$. To estimate these variances, we need to familiarize ourselves with the variance-covariance matrix or just the covariance matrix. As implied by the name, a covariance matrix is a matrix that shows us the covariances of two or more variables. Recall that the sample covariance is defined as follows:

$$Cov(x, y) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{n-1} = E\big((x_i - \bar{x})(y_i - \bar{y})\big)$$

It tells us how two random variables jointly vary with one another. We are also familiar with the Pearson correlation:

$$\rho_{xy} = \frac{Cov(x,y)}{\sigma_x \sigma_y}$$

Since the covariance of any variable with itself is its variance, the diagonal elements of the covariance matrix are the variances of each variable. This is because the sample variance is defined as follows:

$$Var(x) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}$$

The other elements of the covariance matrix are populated with the covariances between the variables that correspond with each row and column. We can extract the covariance matrix of $\hat{\boldsymbol{\beta}}$ from a linear regression in R very easily.

```
M2
```

```
##
## Call:
## lm(formula = income ~ education + prestige)
##
## Coefficients:
## (Intercept)    education     prestige
##    10.42636      0.03226      0.62372
```

```
vcov(M2)
```

```
##              (Intercept)   education     prestige
## (Intercept) 17.335172941 -0.24711213 -0.004413545
## education   -0.247112129  0.01742151 -0.014017625
## prestige    -0.004413545 -0.01401763  0.015540677
```

For example, the entry in the 3rd row and 2nd column shows us the covariance between $\hat{\beta}_{prestige}$ and $\hat{\beta}_{education}$. Since the covariance between education and prestige and prestige and education are the same, the matrix is symmetric, and the entry in the 2nd row and 3rd column shows us the same thing. The diagonal elements show us the variances of each $\hat{\beta}$. The square root of these diagonals therefore gives us the standard errors.

```
sqrt(diag(vcov(M2)))
```

```
## (Intercept)   education    prestige
##   4.1635529   0.1319906   0.1246623
```

```
summary(M2)$coefficients[,2]
```

```
## (Intercept)   education    prestige
##   4.1635529   0.1319906   0.1246623
```

But how do we compute the covariance matrix manually?

Recall that we derived the following:

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X'X})^{-1}(\boldsymbol{X'y})$$

We can $\boldsymbol{y}$ for $\boldsymbol{X\beta} + e$, since $\boldsymbol{y} = \boldsymbol{X\beta} + e$ to obtain the following:

$$\begin{aligned}
\hat{\boldsymbol{\beta}} &= (\boldsymbol{X'X})^{-1}\boldsymbol{X'y} \\
&= (\boldsymbol{X'X})^{-1}\boldsymbol{X'(X\beta + e)} \\
&= (\boldsymbol{X'X})^{-1}\boldsymbol{X'X\beta} + (\boldsymbol{X'X})^{-1}\boldsymbol{X'e} \\
&= \boldsymbol{\beta} + (\boldsymbol{X'X})^{-1}\boldsymbol{X'e}
\end{aligned}$$

As an aside, we can see that $\hat{\boldsymbol{\beta}} = \boldsymbol{\beta}$ when $E(\boldsymbol{X'e}) = 0$. This means that the explanatory variables are uncorrelated with the error term, when there is no omitted variable bias.

Using the formula for covariance, the covariance matrix of $\hat{\boldsymbol{\beta}}$ can be defined as follows:

$$E\bigg((\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})\bigg)$$

This can be expanded to

$$\begin{aligned} E\bigg((\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})\bigg) &= E\bigg(\big((\boldsymbol{X'X})^{-1}\boldsymbol{X'e}\big)\big((\boldsymbol{X'X})^{-1}\boldsymbol{X'e}\big)'\bigg) \\ &= (\boldsymbol{X'X})^{-1}\boldsymbol{X}'E(\boldsymbol{ee}')\boldsymbol{X}(\boldsymbol{X'X})^{-1} \end{aligned}$$

This is our formula for the covariance matrix of $\hat{\boldsymbol{\beta}}$. It is sometimes called a sandwich estimator because we see that $\boldsymbol{X'}E(\boldsymbol{ee}')\boldsymbol{X}$ is sandwiched by $(\boldsymbol{X'X})^{-1}$. This is what we obtain when we compute the covariance matrix of a linear model in R.

But as we found in the last homework, there are two distinct cases that need to be considered: one when the conditional variance of the errors is the same (homoskedastic), and another when the conditional variance of the error varies (heteroskedastic).

This determines how we compute the "meat" of the sandwich or $\boldsymbol{X'}E(\boldsymbol{ee}')\boldsymbol{X}$. When the conditional variance of the errors stays the same, we have

$$Var(\boldsymbol{e}|\boldsymbol{x}) = E((\boldsymbol{e} - E(e)^2|\boldsymbol{x}) = E(\boldsymbol{e}^2|\boldsymbol{x}) = \boldsymbol{\sigma}^2 = \sigma^2$$

The sandwich estimator then becomes

$$\begin{gathered} (\boldsymbol{X'X})^{-1}\boldsymbol{X}'E(\boldsymbol{ee}')\boldsymbol{X}(\boldsymbol{X'X})^{-1} \\ (\boldsymbol{X'X})^{-1}\boldsymbol{X}'\boldsymbol{X}\sigma^2(\boldsymbol{X'X})^{-1} \\ (\boldsymbol{X'X})^{-1}\sigma^2 \end{gathered}$$

where σ^2 is the variance of the errors. We can estimate this using the sample variance of the residuals:

$$s^2 = \frac{\sum_{i=1}^n \hat{e_i^2}}{n-k}$$

This gives us

$$(\boldsymbol{X'X})^{-1}s^2$$

Let's try computing this using R now.

```r
residuals <- resid(M2) # First obtain the residuals from our regression

n <- nobs(M2)
k <- length(M2$coefficients)
s2 <- (sum(residuals^2))/(n-k) # Compute the sample variance of the residuals

Sigma2 <- solve(t(x)%*%x)*s2
Sigma2 # This is the variance covariance matrix
```

```
##                           education     prestige
##           17.335172941 -0.24711213 -0.004413545
## education -0.247112129  0.01742151 -0.014017625
## prestige  -0.004413545 -0.01401763  0.015540677
```

```r
sqrt(diag(Sigma2)) # The square root of the diagonal gives us the standard errors
```

```
##           education  prestige
## 4.1635529 0.1319906 0.1246623
```

```r
summary(M2)$coefficients[,2]
```

```
## (Intercept)   education    prestige
##   4.1635529   0.1319906   0.1246623
```

In the heteroskedastic case, however, we cannot simply replace $\boldsymbol{X'}E(\boldsymbol{ee'})\boldsymbol{X}$ with σ^2. Instead, we substitute the estimated residuals into the formula.

$$(\boldsymbol{X'X})^{-1}\boldsymbol{X'}E(\boldsymbol{ee'})\boldsymbol{X}(\boldsymbol{X'X})^{-1}$$

$$(\boldsymbol{X'X})^{-1}\left(\sum_{i=1}^{n}\boldsymbol{x_i x_i'}\hat{e}_i^2\right)(\boldsymbol{X'X})^{-1}$$

Finally, we know that $\hat{e}_i^2$ is biased toward zero, so we make a scale adjustment to the estimator and obtain:

$$\hat{\boldsymbol{V}}_{\hat{\boldsymbol{\beta}}} = \frac{n}{n-k}(\boldsymbol{X'X})^{-1}\left(\sum_{i=1}^{n}\boldsymbol{x_i x_i'}\hat{e}_i^2\right)(\boldsymbol{X'X})^{-1}$$

These produce what we call robust or heteroskedasticity-consistent standard errors. Let's try to reproduce them in R.

```r
sumxx <- t(x)%*%x

bread <- solve(sumxx) # Compute the bread portions of the sandwich

meat <- t(x)%*%diag(residuals^2)%*%x # Compute the meat portion of the sandwich

adj <- n/(n-k) # Compute the degrees of freedom adjustment

vcovRobust <- adj*bread%*%meat%*%bread
seRobust <- sqrt(diag(adj*bread%*%meat%*%bread))

library(sandwich)
M2vcov <- vcovHC(M2, "HC1") # There are several different variations, we choose HC1
M2SE <- sqrt(diag(M2vcov))

vcovRobust
```

```
##                         education    prestige
##           14.4033769 -0.38936221  0.19820167
## education -0.3893622  0.02411131 -0.01984759
## prestige   0.1982017 -0.01984759  0.01963197
```

```r
M2vcov
```

```
##             (Intercept)   education    prestige
## (Intercept)  14.4033769 -0.38936221  0.19820167
## education    -0.3893622  0.02411131 -0.01984759
```

```
## prestige      0.1982017 -0.01984759  0.01963197
```

```r
seRobust
```

```
##           education  prestige
## 3.7951781 0.1552782 0.1401141
```

```r
M2SE
```

```
## (Intercept)   education    prestige
##   3.7951781   0.1552782   0.1401141
```

Assumptions of the Linear Regression Model

Now that we've covered mechanics of linear regression in more detail, it's worthwhile to go back and discuss its underlying assumptions to put the parts back together. These assumptions enable us to assess whether or not linear regression is an appropriate tool for estimation.

Assumption 1: Linearity

y is linearly related to x through the β parameters. Linear regression is linear in parameters but not in variables. NWe've covered this in the functional form section of last homework. Nonlinear relationships between y and x are possible with the inclusion of transformed variables.

Assumption 2: No Perfect Collinearity

The x's are linearly independent, meaning that none of the x variables are a linear combination of the remaining x variables. When this occurs, the $(\boldsymbol{X'X})$ cannot be inverted, and as we saw least squares regression is then impossible. Put another way, if an additional x variable adds no new information, we cannot estimate its effect on y

This matrix is also sometimes called the design matrix because in an experimental setting the researcher can control this by manipulating the distribution of the x variables. It is okay to have correlated regressors. But as we discussed before, if our x variables are highly correlated, then there is little to distinguish then, and this leads to large standard errors.

Assumption 3: Conditional Mean Zero

This is the key assumption that says that the error term, e_i is unrelated to $\boldsymbol{x}_i$. It is also expressed as $E(e_i|\boldsymbol{x}_i) = 0$. It means that for a given set of x values the error is expected to be zero. It is stronger than saying e_i and $\boldsymbol{x}_i$ are uncorrelated. This is because e_i and $\boldsymbol{x}_i$ must also not be related in a nonlinear way. There must be no systematic relationship.

Since the error term includes anything that determines y_i that is not measured and included in the regression, omitted variable bias is a violation of this assumption.

Note that by construction, the residuals always sum to zero, and the correlation between the residuals and x will almost always be zero. This does not mean that the errors are uncorrelated with x. But if you examine your residuals by plotting them against an x and observe any pattern, this could be a sign of omitted variable bias.

Controlled experiments and quasi-experiments are a way to address this by making sure this assumption holds.

Assumption 4: $(\boldsymbol{x}_i, y_i)$ are I.I.D.

I.I.D. stands for independently and indentically distributed. This assumptions says that the data are a random draw from the actualy population. Sometimes this is misinterpreted. It does not mean that y_i and $\boldsymbol{x}_i$ are independent of one another. It also does not mean that our x was assigned randomly. It means that observation i is independent of observation j. The random sampling framework is necessary for the application of statistical methods of inference.

Serial correlation is an example of when this assumption is violated. In this case our errors are correlated with one another, and our standard errors are biased.

Assumption 5: Homoskedasticity

This assumptions says that the variance of the error term does not change with $\boldsymbol{x}_i$.

$$Var(e_i|\boldsymbol{x}_i) = \sigma^2 \text{ for all } i$$

As we discussed, when our errors are heteroskedastic, and this assumption is violated, our conventional standard errors are biased. It implies that the diagonal elements of the covariance matrix are distinct from one another.

Assumption 6: Normality of errors

This is the least important assumption of linear regression, as it does not influence our estimation of the regression line. This is often embedded in the idea that the errors are a combined effect of many small factors. If this assumption fails, then the standard errors will be biased unless n is large.

Gauss-Markov Theorem

The Gauss-Markov Theorem tells us that when Assumptions 1-3 hold, then $\boldsymbol{\beta}_{LS}$ is linear and unbiased.

If $\boldsymbol{y} = \boldsymbol{X\beta} + \boldsymbol{e}$, then recall:

$$\begin{aligned}
\hat{\boldsymbol{\beta}} &= (\boldsymbol{X'X})^{-1}\boldsymbol{X'y} \\
&= (\boldsymbol{X'X})^{-1}\boldsymbol{X'}(\boldsymbol{X\beta} + \boldsymbol{e}) \\
&= (\boldsymbol{X'X})^{-1}\boldsymbol{X'X\beta} + (\boldsymbol{X'X})^{-1}\boldsymbol{X'e} \\
&= \boldsymbol{\beta} + (\boldsymbol{X'X})^{-1}\boldsymbol{X'e}
\end{aligned}$$

This means that if Assumption 2 holds and $(\boldsymbol{X'X})$ is invertible. And Assumption 3 holds and $\boldsymbol{X'e} = 0$, then $\hat{\boldsymbol{\beta}} = \boldsymbol{\beta}$.

When Assumptions 1-5 hold, then we can make a stronger claim. When there is no serial correlation, no heteroskedasticity, no endogeneity, and no perfect collinearity, then the Gauss-Markov theorem holds that least squares is the best linear unbiased estimator (BLUE). This means that among the linear estimators that are unbiased, $\hat{\boldsymbol{\beta}}_{LS}$ has the least variance.

When Assumptions 1-6 hold, then Gauss-Markov holds that least squares is Minimum Variance Unbiased (MVU). This means that among al estimators that are unbiased, $\hat{\boldsymbol{\beta}}_{LS}$ has the least variance.

Instrumental Variables

The following replicates Table 6 from The Colonial Origins of Comparative Development: An Empirical Investigation by Daron Acemoglu, Simon Johnson, and James A. Robinson.

```r
setwd("~/desktop")
library(AER)
```

```
## Loading required package: lmtest

## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: survival
```

```r
data <- read.csv("maketable6.csv", header=T)
colnames(data)
```

```
##  [1] "shortnam" "avelf"    "lat_abst" "temp1"    "temp2"    "temp3"
##  [7] "temp4"    "temp5"    "humid1"   "humid2"   "humid3"   "humid4"
## [13] "steplow"  "deslow"   "stepmid"  "desmid"   "drystep"  "drywint"
## [19] "edes1975" "avexpr"   "logpgp95" "landlock" "goldm"    "iron"
## [25] "silv"     "zinc"     "oilres"   "logem4"   "baseco"
```

```r
#--Panel C, OLS Regressions

#--Columns 1 and 2 (Temperature and humidity controls)
reg1 <- lm(logpgp95 ~ avexpr + temp1 + temp2 + temp3 + temp4 + temp5 + humid1 + humid2 + humid3 + humid4
summary(reg1)
```

```
##
## Call:
## lm(formula = logpgp95 ~ avexpr + temp1 + temp2 + temp3 + temp4 +
##     temp5 + humid1 + humid2 + humid3 + humid4, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9389 -0.3157  0.1121  0.3868  1.3479
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.498671   1.307838   3.440 0.000854 ***
## avexpr       0.471875   0.049946   9.448 1.75e-15 ***
## temp1        0.110506   0.076670   1.441 0.152649
## temp2       -0.024901   0.024616  -1.012 0.314208
## temp3       -0.020484   0.033438  -0.613 0.541554
## temp4       -0.048432   0.029903  -1.620 0.108499
## temp5       -0.019377   0.020278  -0.956 0.341614
## humid1      -0.009548   0.011648  -0.820 0.414363
## humid2       0.018735   0.019436   0.964 0.337415
## humid3       0.034833   0.011998   2.903 0.004553 **
## humid4      -0.036743   0.014339  -2.562 0.011901 *
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6761 on 99 degrees of freedom
##   (53 observations deleted due to missingness)
## Multiple R-squared:  0.6829, Adjusted R-squared:  0.6509
## F-statistic: 21.32 on 10 and 99 DF,  p-value: < 2.2e-16
```

```r
reg2 <- lm(logpgp95 ~ lat_abst + avexpr + temp1 + temp2 + temp3 + temp4 + temp5 + humid1 + humid2 + humi
summary(reg2)
```

```
##
## Call:
## lm(formula = logpgp95 ~ lat_abst + avexpr + temp1 + temp2 + temp3 +
##     temp4 + temp5 + humid1 + humid2 + humid3 + humid4, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.90349 -0.37607  0.06229  0.40515  1.32091
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.350274   1.282940   3.391  0.00101 **
## lat_abst     1.815579   0.800188   2.269  0.02546 *
## avexpr       0.412962   0.055394   7.455 3.62e-11 ***
## temp1        0.076560   0.076588   1.000  0.31995
## temp2       -0.005688   0.025559  -0.223  0.82434
## temp3       -0.019090   0.032764  -0.583  0.56148
## temp4       -0.030107   0.030389  -0.991  0.32426
## temp5       -0.010965   0.020209  -0.543  0.58865
## humid1      -0.008033   0.011431  -0.703  0.48388
## humid2       0.022005   0.019095   1.152  0.25198
## humid3       0.033816   0.011762   2.875  0.00496 **
## humid4      -0.040147   0.014128  -2.842  0.00546 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6624 on 98 degrees of freedom
##   (53 observations deleted due to missingness)
## Multiple R-squared:  0.6987, Adjusted R-squared:  0.6649
## F-statistic: 20.66 on 11 and 98 DF,  p-value: < 2.2e-16
```

```r
#--Columns 3 and 4 (Control for percent of European descent in 1975)
reg3 <- lm(logpgp95 ~ avexpr + edes1975, data=data)
summary(reg3)
```

```
##
## Call:
## lm(formula = logpgp95 ~ avexpr + edes1975, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8022 -0.3522  0.0768  0.4169  1.5626
##
## Coefficients:
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 5.054554   0.356673  14.171  < 2e-16 ***
## avexpr      0.452276   0.054494   8.300 3.26e-13 ***
## edes1975    0.004650   0.002164   2.148   0.0339 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7063 on 108 degrees of freedom
##   (52 observations deleted due to missingness)
## Multiple R-squared:  0.6273, Adjusted R-squared:  0.6204
## F-statistic: 90.87 on 2 and 108 DF,  p-value: < 2.2e-16
```

```
reg4 <- lm(logpgp95 ~ lat_abst + avexpr + edes1975, data=data)
summary(reg4)
```

```
##
## Call:
## lm(formula = logpgp95 ~ lat_abst + avexpr + edes1975, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.75180 -0.35509  0.05831  0.41510  1.50861
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 5.078424   0.358911  14.150  < 2e-16 ***
## lat_abst    0.427727   0.582444   0.734    0.464
## avexpr      0.436790   0.058541   7.461 2.39e-11 ***
## edes1975    0.003595   0.002601   1.382    0.170
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7078 on 107 degrees of freedom
##   (52 observations deleted due to missingness)
## Multiple R-squared:  0.6291, Adjusted R-squared:  0.6187
## F-statistic:  60.5 on 3 and 107 DF,  p-value: < 2.2e-16
```

```
#--Columns 5 and 6 (Controls for soil quality, natural resources, and landlocked)

reg5 <- lm(logpgp95 ~ avexpr + steplow + deslow + stepmid + desmid + drystep + drywint + goldm + iron +
summary(reg5)
```

```
##
## Call:
## lm(formula = logpgp95 ~ avexpr + steplow + deslow + stepmid +
##     desmid + drystep + drywint + goldm + iron + silv + zinc +
##     oilres + landlock, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9095 -0.3981  0.1010  0.4339  1.3636
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.692e+00  3.462e-01  13.551   <2e-16 ***
```

```
## avexpr       5.231e-01  4.487e-02  11.658   <2e-16 ***
## steplow     -2.969e-01  1.778e-01  -1.670   0.0983 .
## deslow       3.800e-01  1.897e-01   2.003   0.0480 *
## stepmid     -5.167e-01  4.110e-01  -1.257   0.2117
## desmid       6.144e-01  5.448e-01   1.128   0.2623
## drystep     -3.884e-01  3.126e-01  -1.243   0.2171
## drywint     -1.566e+00  4.207e+00  -0.372   0.7105
## goldm        5.545e-02  8.923e-02   0.621   0.5358
## iron        -1.980e-02  6.386e-02  -0.310   0.7572
## silv         4.433e-02  5.539e-02   0.800   0.4256
## zinc        -2.992e-02  7.961e-02  -0.376   0.7079
## oilres       4.377e-08  4.027e-08   1.087   0.2797
## landlock    -1.333e-01  2.058e-01  -0.648   0.5187
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6937 on 95 degrees of freedom
##   (54 observations deleted due to missingness)
## Multiple R-squared:  0.6797, Adjusted R-squared:  0.6359
## F-statistic: 15.51 on 13 and 95 DF,  p-value: < 2.2e-16
```

```r
reg6 <- lm(logpgp95 ~ lat_abst + avexpr + steplow + deslow + stepmid + desmid + drystep + drywint + gol
summary(reg6)
```

```
##
## Call:
## lm(formula = logpgp95 ~ lat_abst + avexpr + steplow + deslow +
##     stepmid + desmid + drystep + drywint + goldm + iron + silv +
##     zinc + oilres + landlock, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.75852 -0.36259  0.06409  0.44993  1.31275
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.014e+00  3.758e-01  13.342  < 2e-16 ***
## lat_abst     1.019e+00  5.012e-01   2.033   0.0448 *
## avexpr       4.415e-01  5.967e-02   7.399 5.71e-11 ***
## steplow     -2.728e-01  1.754e-01  -1.555   0.1232
## deslow       3.069e-01  1.901e-01   1.615   0.1097
## stepmid     -5.566e-01  4.049e-01  -1.375   0.1724
## desmid       4.737e-01  5.405e-01   0.876   0.3831
## drystep     -3.623e-01  3.078e-01  -1.177   0.2421
## drywint     -2.049e+00  4.146e+00  -0.494   0.6223
## goldm        6.269e-02  8.786e-02   0.714   0.4773
## iron        -4.056e-03  6.331e-02  -0.064   0.9490
## silv         5.367e-02  5.470e-02   0.981   0.3290
## zinc        -4.765e-02  7.881e-02  -0.605   0.5469
## oilres       4.976e-08  3.973e-08   1.252   0.2135
## landlock    -2.015e-01  2.052e-01  -0.982   0.3287
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6825 on 94 degrees of freedom
```

```
##   (54 observations deleted due to missingness)
## Multiple R-squared:  0.6932, Adjusted R-squared:  0.6475
## F-statistic: 15.17 on 14 and 94 DF,  p-value: < 2.2e-16
```

```r
#*--Columns 7 and 8 (Control for ethnolinguistic fragmentation)

reg7 <- lm(logpgp95 ~ avexpr + avelf, data=data)
summary(reg7)
```

```
##
## Call:
## lm(formula = logpgp95 ~ avexpr + avelf, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.71664 -0.26432  0.05585  0.38794  1.22976
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.43857    0.32761  16.601  < 2e-16 ***
## avexpr       0.46165    0.03926  11.759  < 2e-16 ***
## avelf       -1.09294    0.22326  -4.895 3.82e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6164 on 99 degrees of freedom
##   (61 observations deleted due to missingness)
## Multiple R-squared:  0.7226, Adjusted R-squared:  0.717
## F-statistic: 128.9 on 2 and 99 DF,  p-value: < 2.2e-16
```

```r
reg8 <- lm(logpgp95 ~ lat_abst + avexpr + avelf, data=data)
summary(reg8)
```

```
##
## Call:
## lm(formula = logpgp95 ~ lat_abst + avexpr + avelf, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.71775 -0.26481  0.05495  0.38758  1.22988
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.437612   0.333442  16.308  < 2e-16 ***
## lat_abst    -0.008886   0.489389  -0.018    0.986
## avexpr       0.462215   0.050119   9.222  5.9e-15 ***
## avelf       -1.094809   0.246812  -4.436  2.4e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6196 on 98 degrees of freedom
##   (61 observations deleted due to missingness)
## Multiple R-squared:  0.7226, Adjusted R-squared:  0.7141
## F-statistic: 85.09 on 3 and 98 DF,  p-value: < 2.2e-16
```

```r
#--Column 9 (All Controls)

reg9 <- lm(logpgp95 ~ lat_abst + avexpr + temp1 + temp2 + temp3 + temp4 + temp5 + humid1 + humid2 + humi
summary(reg9)
```

```
##
## Call:
## lm(formula = logpgp95 ~ lat_abst + avexpr + temp1 + temp2 + temp3 +
##     temp4 + temp5 + humid1 + humid2 + humid3 + humid4 + edes1975 +
##     avelf + steplow + deslow + stepmid + desmid + drystep + drywint +
##     goldm + iron + silv + zinc + oilres + landlock, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.49774 -0.25609 -0.00436  0.28665  1.37159
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.698e+00  1.262e+00   3.722 0.000377 ***
## lat_abst    -1.367e-02  8.937e-01  -0.015 0.987833
## avexpr       3.948e-01  5.374e-02   7.347 1.94e-10 ***
## temp1        1.109e-01  7.797e-02   1.423 0.158955
## temp2        3.276e-04  2.516e-02   0.013 0.989644
## temp3       -3.880e-02  3.412e-02  -1.137 0.259065
## temp4       -3.830e-02  3.045e-02  -1.258 0.212359
## temp5       -2.887e-02  2.109e-02  -1.369 0.174979
## humid1       9.689e-03  1.185e-02   0.818 0.416014
## humid2       7.935e-05  1.931e-02   0.004 0.996731
## humid3       1.012e-02  1.219e-02   0.830 0.409148
## humid4      -7.863e-03  1.472e-02  -0.534 0.594712
## edes1975     5.159e-03  2.715e-03   1.900 0.061220 .
## avelf       -1.023e+00  2.760e-01  -3.706 0.000397 ***
## steplow     -2.852e-01  1.626e-01  -1.754 0.083444 .
## deslow       5.088e-01  2.164e-01   2.352 0.021270 *
## stepmid     -2.068e-01  3.991e-01  -0.518 0.605915
## desmid       4.805e-01  4.593e-01   1.046 0.298793
## drystep     -7.988e-02  2.741e-01  -0.291 0.771537
## drywint      1.694e+00  3.805e+00   0.445 0.657447
## goldm       -7.162e-03  8.032e-02  -0.089 0.929182
## iron        -3.246e-02  5.688e-02  -0.571 0.569885
## silv        -1.403e-02  5.028e-02  -0.279 0.780897
## zinc         5.772e-02  6.957e-02   0.830 0.409322
## oilres       7.195e-08  1.650e-07   0.436 0.663977
## landlock     2.190e-01  2.009e-01   1.090 0.278954
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.562 on 76 degrees of freedom
##   (61 observations deleted due to missingness)
## Multiple R-squared:  0.823,  Adjusted R-squared:  0.7648
## F-statistic: 14.13 on 25 and 76 DF,  p-value: < 2.2e-16
```

```r
#--Panels A and B, IV Regressions
```

```r
#--Columns 1 and 2 (Temperature and humidity controls)

ivreg1 <- ivreg(logpgp95 ~ avexpr + temp1 + temp2 + temp3 + temp4 + temp5 + humid1 + humid2 + humid3 + h
summary(ivreg1)
```

```
##
## Call:
## ivreg(formula = logpgp95 ~ avexpr + temp1 + temp2 + temp3 + temp4 +
##     temp5 + humid1 + humid2 + humid3 + humid4 | logem4 + temp1 +
##     temp2 + temp3 + temp4 + temp5 + humid1 + humid2 + humid3 +
##     humid4, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.97895 -0.59986 -0.06071  0.69951  1.83014
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.955978   2.525727   0.774    0.442
## avexpr       0.895321   0.193448   4.628 2.07e-05 ***
## temp1        0.008773   0.164562   0.053    0.958
## temp2        0.030101   0.048571   0.620    0.538
## temp3       -0.011487   0.073669  -0.156    0.877
## temp4        0.001892   0.068429   0.028    0.978
## temp5       -0.034917   0.035822  -0.975    0.334
## humid1      -0.021611   0.022182  -0.974    0.334
## humid2       0.029156   0.035531   0.821    0.415
## humid3       0.029773   0.022229   1.339    0.186
## humid4      -0.032895   0.026524  -1.240    0.220
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9267 on 59 degrees of freedom
## Multiple R-Squared: 0.3456,  Adjusted R-squared: 0.2347
## Wald test: 6.081 on 10 and 59 DF,  p-value: 2.773e-06
```

```r
ivreg2 <- ivreg(logpgp95 ~ avexpr + lat_abst + temp1 + temp2 + temp3 + temp4 + temp5 + humid1 + humid2 +
summary(ivreg2)
```

```
##
## Call:
## ivreg(formula = logpgp95 ~ avexpr + lat_abst + temp1 + temp2 +
##     temp3 + temp4 + temp5 + humid1 + humid2 + humid3 + humid4 |
##     logem4 + lat_abst + temp1 + temp2 + temp3 + temp4 + temp5 +
##         humid1 + humid2 + humid3 + humid4, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.04317 -0.66984 -0.05906  0.70845  1.73452
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.875937   2.706316   0.693 0.490969
## avexpr       0.937114   0.254140   3.687 0.000501 ***
```

```
## lat_abst    -0.850979   1.850704  -0.460 0.647369
## temp1        0.021642   0.170219   0.127 0.899270
## temp2        0.023539   0.050533   0.466 0.643099
## temp3       -0.008724   0.078649  -0.111 0.912061
## temp4       -0.004177   0.070512  -0.059 0.952965
## temp5       -0.039086   0.038194  -1.023 0.310384
## humid1      -0.021500   0.023315  -0.922 0.360282
## humid2       0.026879   0.037305   0.721 0.474101
## humid3       0.028971   0.023460   1.235 0.221849
## humid4      -0.031607   0.027942  -1.131 0.262647
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9751 on 58 degrees of freedom
## Multiple R-Squared: 0.2878,  Adjusted R-squared: 0.1527
## Wald test:  5.18 on 11 and 58 DF,  p-value: 1.306e-05
```

```r
#--Columns 3 and 4 (Control for percent of European descent in 1975)

ivreg3 <- ivreg(logpgp95 ~ avexpr + edes1975 | logem4 + edes1975, data=data)
summary(ivreg3)
```

```
##
## Call:
## ivreg(formula = logpgp95 ~ avexpr + edes1975 | logem4 + edes1975,
##     data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.39891 -0.56812  0.03487  0.69490  1.63718
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.096985   1.601353    1.31 0.194835
## avexpr       0.915577   0.260130    3.52 0.000782 ***
## edes1975    -0.002084   0.007728   -0.27 0.788228
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.942 on 67 degrees of freedom
## Multiple R-Squared: 0.2321,  Adjusted R-squared: 0.2092
## Wald test: 25.34 on 2 and 67 DF,  p-value: 6.392e-09
```

```r
ivreg4 <- ivreg(logpgp95 ~ avexpr + lat_abst + edes1975 | logem4  + lat_abst + edes1975, data=data)
summary(ivreg4)
```

```
##
## Call:
## ivreg(formula = logpgp95 ~ avexpr + lat_abst + edes1975 | logem4 +
##     lat_abst + edes1975, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.60321 -0.75981  0.02085  0.79854  2.01643
##
```

```
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.7137225  1.9230530   0.891  0.37609
## avexpr       1.0187543  0.3328557   3.061  0.00319 **
## lat_abst    -1.8026206  1.4509772  -1.242  0.21850
## edes1975     0.0005289  0.0078624   0.067  0.94657
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.036 on 66 degrees of freedom
## Multiple R-Squared: 0.08598, Adjusted R-squared: 0.04443
## Wald test:    14 on 3 and 66 DF,  p-value: 3.642e-07
```

```
#--Columns 5 and 6 (Controls for soil quality, natural resources, and landlocked)

ivreg5 <- ivreg(logpgp95 ~ avexpr + steplow + deslow + stepmid + desmid + drystep + drywint + goldm + i
summary(ivreg5)
```

```
##
## Call:
## ivreg(formula = logpgp95 ~ avexpr + steplow + deslow + stepmid +
##     desmid + drystep + drywint + goldm + iron + silv + zinc +
##     oilres + landlock | logem4 + steplow + deslow + stepmid +
##     desmid + drystep + drywint + goldm + iron + silv + zinc +
##     oilres + landlock, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.53468 -0.62490 -0.06596  0.69691  2.06321
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.210e+00  1.748e+00   0.693 0.491450
## avexpr       1.039e+00  2.565e-01   4.052 0.000158 ***
## steplow     -2.264e-01  3.365e-01  -0.673 0.503876
## deslow       3.939e-01  4.184e-01   0.942 0.350474
## stepmid     -4.485e-01  8.000e-01  -0.561 0.577287
## desmid       6.090e-01  8.753e-01   0.696 0.489478
## drystep      1.693e-01  5.442e-01   0.311 0.756955
## drywint      4.050e+00  7.667e+00   0.528 0.599421
## goldm       -5.855e-02  1.615e-01  -0.362 0.718386
## iron        -1.270e-01  1.385e-01  -0.917 0.363286
## silv         2.917e-02  1.059e-01   0.275 0.784010
## zinc        -1.026e-02  1.732e-01  -0.059 0.952975
## oilres       1.185e-07  3.762e-07   0.315 0.753986
## landlock     5.645e-01  5.672e-01   0.995 0.323870
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.07 on 56 degrees of freedom
## Multiple R-Squared: 0.1716,  Adjusted R-squared: -0.02077
## Wald test: 3.079 on 13 and 56 DF,  p-value: 0.001723
```

```
ivreg6 <- ivreg(logpgp95 ~ avexpr + lat_abst + steplow + deslow + stepmid + desmid + drystep + drywint +
summary(ivreg6)
```

```
##
## Call:
## ivreg(formula = logpgp95 ~ avexpr + lat_abst + steplow + deslow +
##     stepmid + desmid + drystep + drywint + goldm + iron + silv +
##     zinc + oilres + landlock | logem4 + lat_abst + steplow +
##     deslow + stepmid + desmid + drystep + drywint + goldm + iron +
##     silv + zinc + oilres + landlock, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.17940 -0.89079 -0.01283  0.92238  3.31506
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -8.053e-01  4.031e+00  -0.200   0.8424
## avexpr       1.414e+00  6.679e-01   2.118   0.0387 *
## lat_abst    -3.007e+00  3.214e+00  -0.936   0.3536
## steplow     -1.326e-01  4.770e-01  -0.278   0.7821
## deslow       7.076e-01  7.176e-01   0.986   0.3284
## stepmid     -4.669e-01  1.088e+00  -0.429   0.6695
## desmid       1.214e+00  1.377e+00   0.881   0.3819
## drystep      2.886e-01  7.800e-01   0.370   0.7128
## drywint      7.256e+00  1.172e+01   0.619   0.5384
## goldm       -1.152e-01  2.408e-01  -0.478   0.6344
## iron        -2.280e-01  2.436e-01  -0.936   0.3534
## silv         1.595e-03  1.512e-01   0.011   0.9916
## zinc         7.657e-02  2.654e-01   0.289   0.7740
## oilres      -1.241e-07  5.989e-07  -0.207   0.8366
## landlock     9.978e-01  1.015e+00   0.983   0.3298
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.453 on 55 degrees of freedom
## Multiple R-Squared: -0.4998, Adjusted R-squared: -0.8816
## Wald test: 1.624 on 14 and 55 DF,  p-value: 0.1017
```

```r
#--Columns 7 and 8 (Control for ethnolinguistic fragmentation)

ivreg7 <- ivreg(logpgp95 ~ avexpr + avelf| logem4 + avelf, data=data)
summary(ivreg7)
```

```
##
## Call:
## ivreg(formula = logpgp95 ~ avexpr + avelf | logem4 + avelf, data = data)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -1.765814 -0.483480 -0.004687  0.409363  1.587484
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.6122     0.8916   4.051 0.000135 ***
## avexpr        0.7300     0.1218   5.994 9.09e-08 ***
## avelf        -0.8095     0.3353  -2.414 0.018506 *
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7371 on 67 degrees of freedom
## Multiple R-Squared: 0.5299,  Adjusted R-squared: 0.5158
## Wald test: 39.67 on 2 and 67 DF,  p-value: 4.304e-12
```

```r
ivreg8 <- ivreg(logpgp95 ~ avexpr + lat_abst + avelf | logem4 + lat_abst + avelf, data=data)
summary(ivreg8)
```

```
##
## Call:
## ivreg(formula = logpgp95 ~ avexpr + lat_abst + avelf | logem4 +
##     lat_abst + avelf, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.97311 -0.49298 -0.06279  0.56562  1.64093
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.1835     1.1693   2.723  0.00828 **
## avexpr        0.8659     0.1953   4.433  3.6e-05 ***
## lat_abst     -1.9508     1.2431  -1.569  0.12137
## avelf        -1.0420     0.3738  -2.788  0.00693 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8404 on 66 degrees of freedom
## Multiple R-Squared: 0.398,   Adjusted R-squared: 0.3706
## Wald test: 20.71 on 3 and 66 DF,  p-value: 1.443e-09
```

```r
#--Column 9 (All Controls)
ivreg9 <- ivreg(logpgp95 ~ avexpr + lat_abst + steplow + deslow + stepmid + desmid + drystep + drywint +
summary(ivreg9)
```

```
##
## Call:
## ivreg(formula = logpgp95 ~ avexpr + lat_abst + steplow + deslow +
##     stepmid + desmid + drystep + drywint + goldm + iron + silv +
##     zinc + oilres + landlock + edes1975 + avelf | logem4 + lat_abst +
##     steplow + deslow + stepmid + desmid + drystep + drywint +
##     goldm + iron + silv + zinc + oilres + landlock + edes1975 +
##     avelf, data = data)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -1.700444 -0.555066  0.005045  0.465983  2.149232
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.701e+00  2.577e+00   1.048   0.2994
## avexpr       9.840e-01  4.614e-01   2.133   0.0376 *
## lat_abst    -3.538e+00  2.258e+00  -1.567   0.1230
## steplow     -8.608e-02  3.148e-01  -0.273   0.7856
## deslow       5.902e-01  4.620e-01   1.277   0.2070
```

```
## stepmid     -3.120e-01  6.983e-01  -0.447   0.6568
## desmid       8.624e-01  9.090e-01   0.949   0.3471
## drystep      4.617e-01  5.659e-01   0.816   0.4182
## drywint      8.119e+00  7.879e+00   1.030   0.3075
## goldm       -1.180e-01  1.585e-01  -0.744   0.4600
## iron        -2.303e-01  1.695e-01  -1.358   0.1801
## silv        -4.189e-02  1.015e-01  -0.413   0.6816
## zinc         1.855e-01  1.955e-01   0.949   0.3469
## oilres      -1.091e-07  4.060e-07  -0.269   0.7892
## landlock     8.313e-01  7.460e-01   1.114   0.2702
## edes1975     4.239e-03  6.855e-03   0.618   0.5390
## avelf       -1.725e+00  6.659e-01  -2.590   0.0124 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9178 on 53 degrees of freedom
## Multiple R-Squared: 0.4235,  Adjusted R-squared: 0.2494
## Wald test: 3.856 on 16 and 53 DF,  p-value: 0.0001032
```