GOV 2000 Section 6: Random Samples and Descriptive Inference (Regression)

Konstantin Kashin[1]
Harvard University

October 10, 2012

[1]These notes and accompanying code draw on the notes from Molly Roberts, Maya Sen, Iain Osgood, Brandon Stewart, and TF's from previous years

Outline

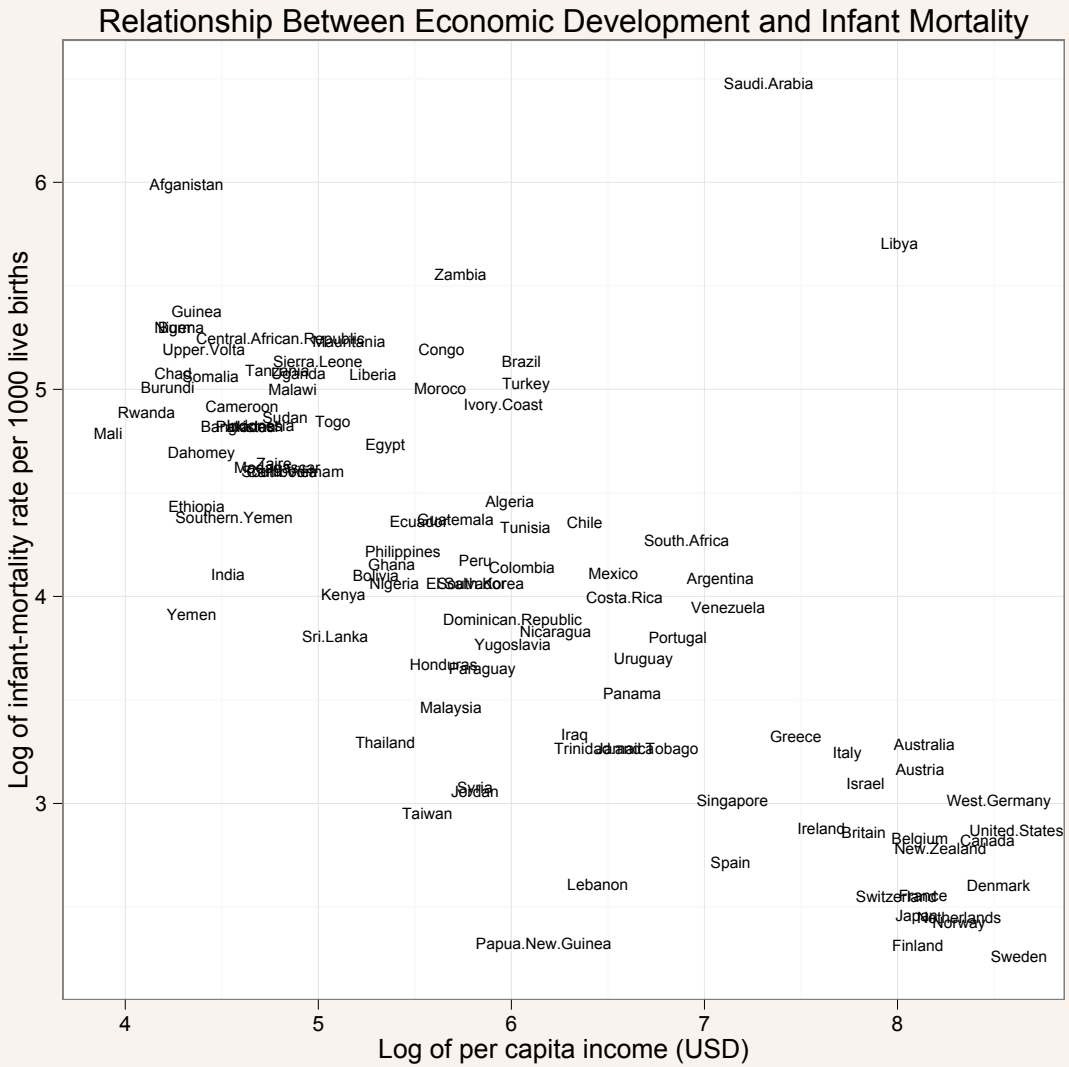
Data

We are going to work with `Leinhardt` dataset from `Leinhardt.RData`.

- `lincome`: Log of per-capita income in U. S. dollars.
- `linfant`: Log of infant mortality rate per 1000 live births.
- `region`: A factor with levels: Africa; Americas; Asia, Asia and Oceania; Europe.
- `oil`: Oil-exporting country. A factor with levels: no, yes.

We want to regress log of infant mortality rate on log of per-capita income.

Scatterplot of the Data



Population Linear Conditional Expectation Function

$$\mathrm{E}[Y|X = x] = \beta_0 + \beta_1 x$$

In R:

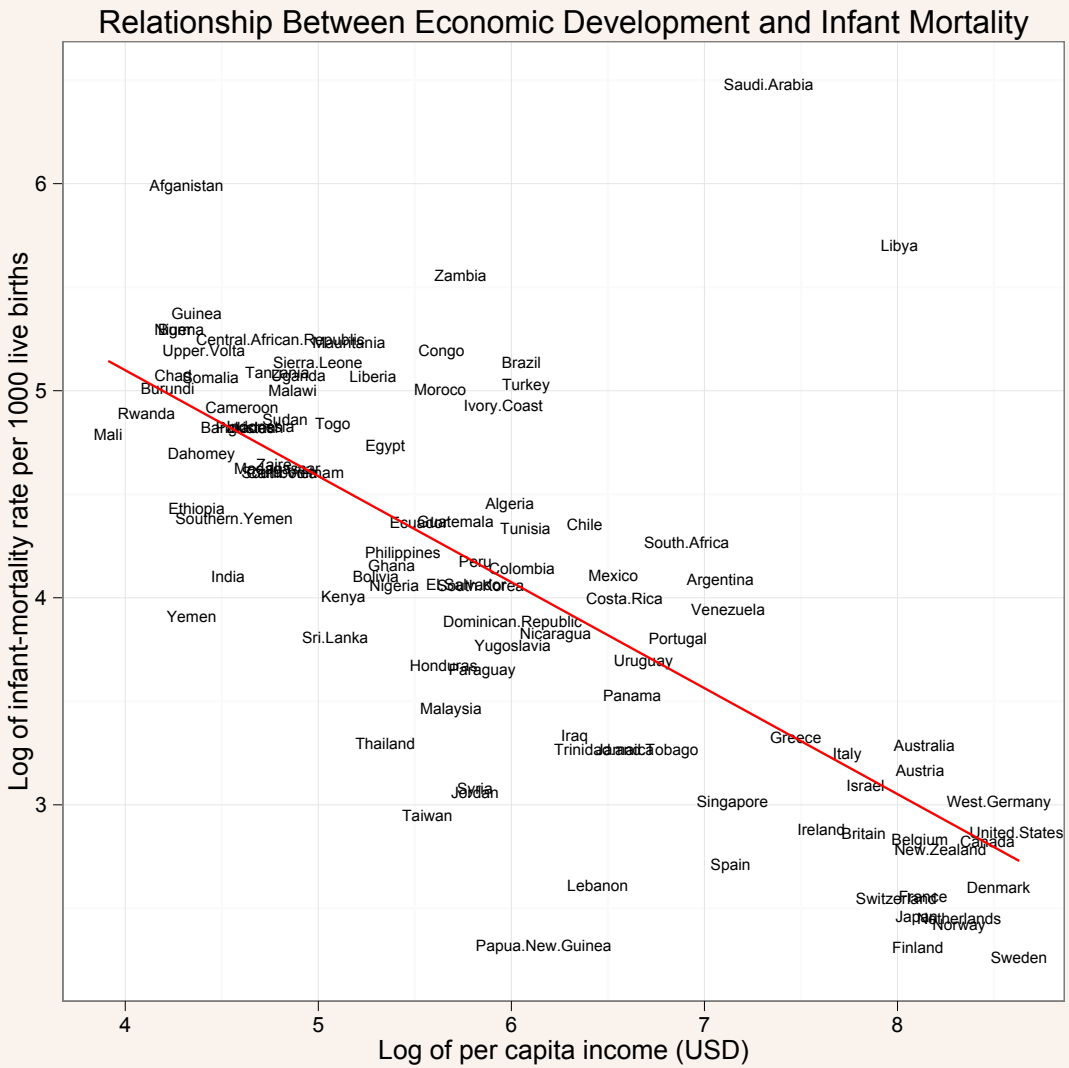
```
lm(linfant ~ lincome, data=Leinhardt)
```

True population parameters:

$\beta_0 = 7.1458$

$\beta_1 = -0.5118$

Population Linear Conditional Expectation Function



Outline

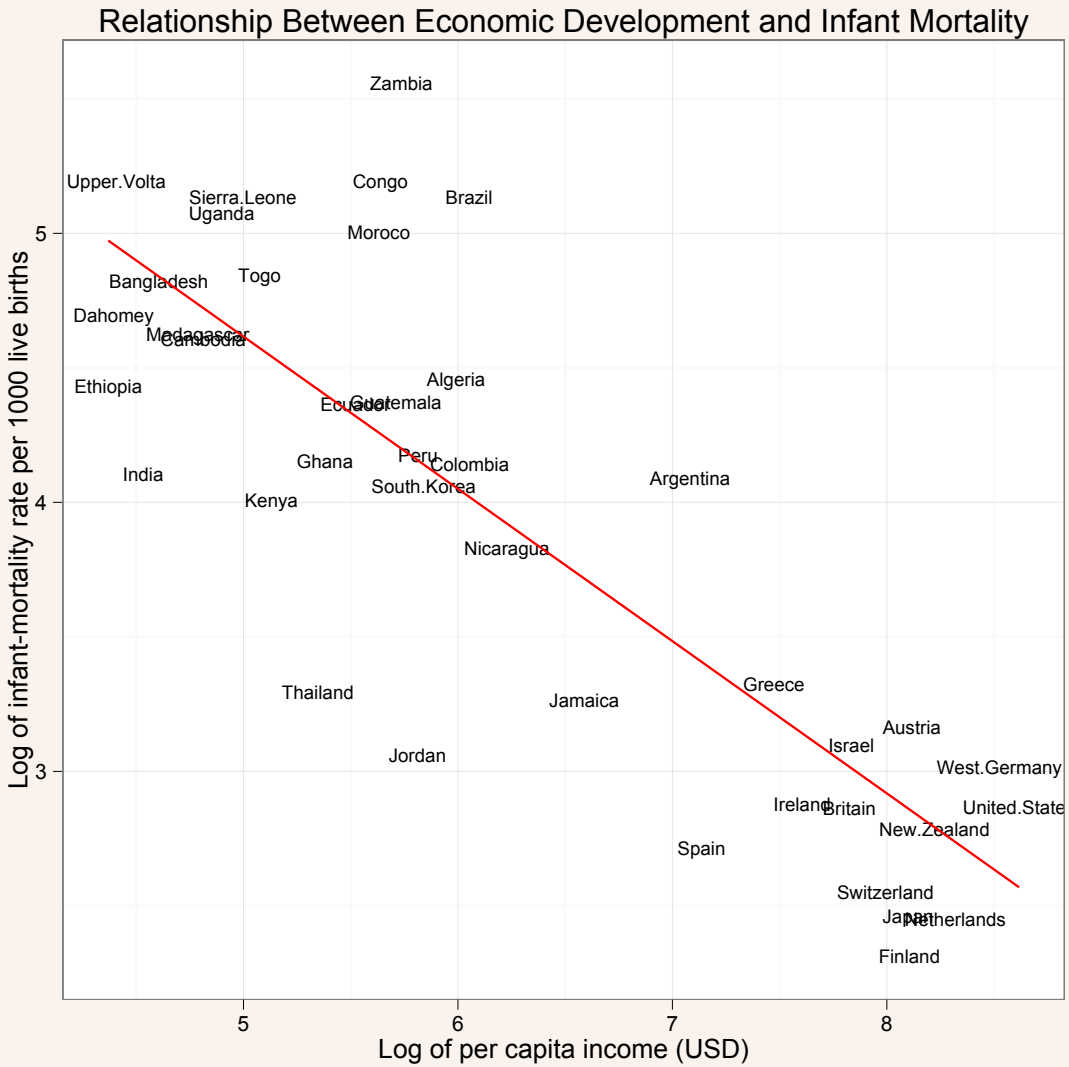
Our Sample

Let's take one sample of size 40 (without replacement) and run regression on it.

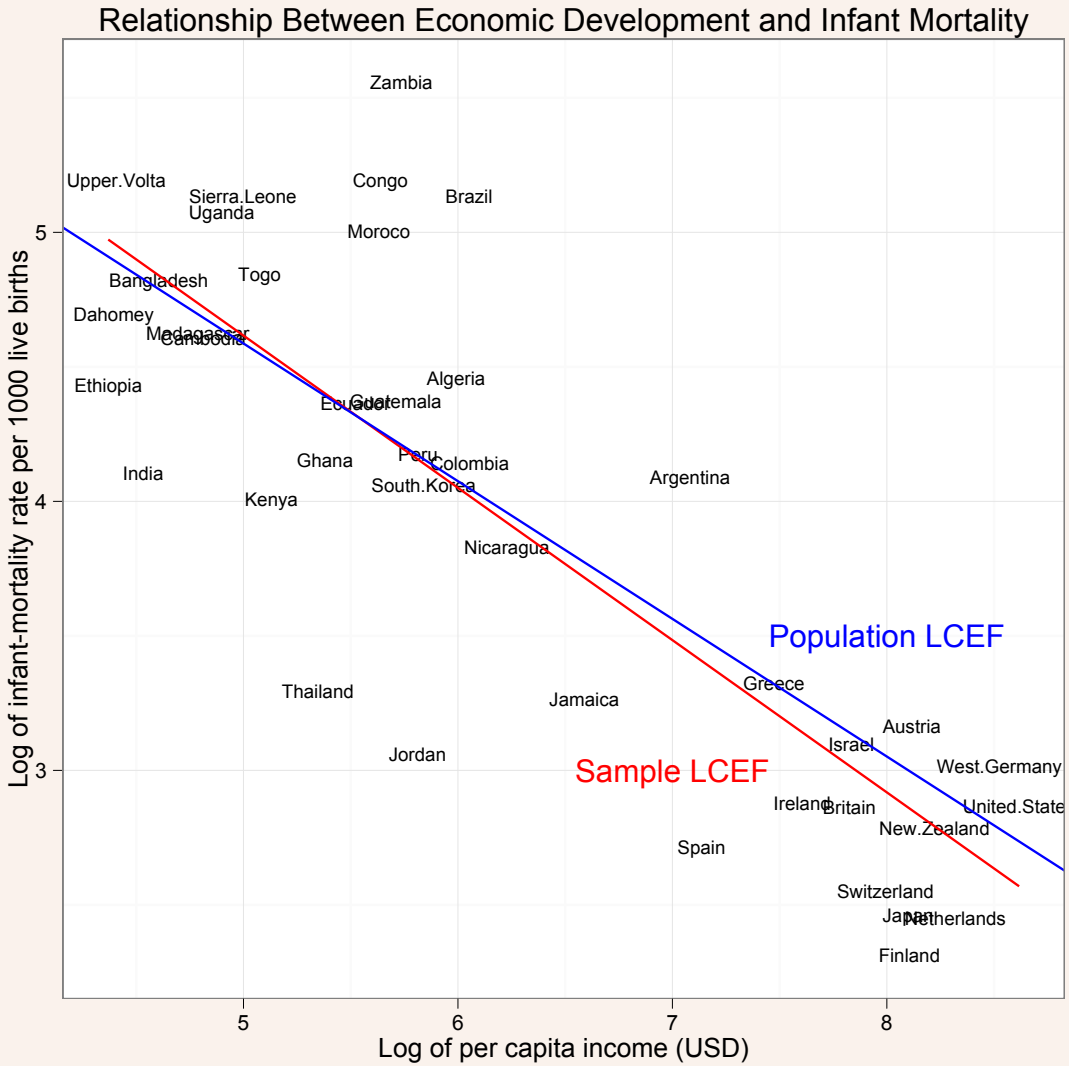
```
set.seed(01238)
my.samp <- Leinhardt[sample(nrow(Leinhardt),size=40,
    replace=FALSE),]
lm.samp <- lm(linfant ~ lincome, data=my.samp)
```

Estimated regression coefficients:
$\hat{\beta}_0 = 7.4462$
$\hat{\beta}_1 = -0.5660$

Estimated Linear Conditional Expectation Function



Estimated Linear Conditional Expectation Function



Regression Output

This is what we get as regression output:

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.44624    0.41046  18.141  < 2e-16 ***
lincome     -0.56600    0.06412  -8.827 9.72e-11 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

Let's focus on the standard errors for now...

Standard Errors of Regression Coefficients

- What do the standard errors of the regression coefficients represent?
  - Variability of sampling distributions of regression coefficients

  How can we estimate the standard errors of the regression coefficients?
  - Theory
  - Boostrapping (resampling sample)

Outline

Repeated Sampling

We can think of our sample as one of many possible samples we can draw from our population:

	Samples from Pop.			
	1	**2**	**3**	⋯
X_1, Y_1	(4.997, 5.136)	$(x_{1,2}, y_{1,2})$	$(x_{1,3}, y_{1,3})$	⋯
X_2, Y_2	(4.812, 4.605)	$(x_{2,2}, y_{2,2})$	$(x_{2,3}, y_{2,3})$	⋯
⋮	⋮	⋮	⋮	⋯
X_{40}, Y_{40}	(7.834, 3.096)	$(x_{40,2}, y_{40,2})$	$(x_{40,3}, y_{40,3})$	⋯
$(\bar{X}_{40}, \bar{Y}_{40})$	(6.258, 3.904)	$(\bar{x}_2, \bar{y}_2)$	$(\bar{x}_3, \bar{y}_3)$	⋯
$(\bar{S^2_X}, \bar{Y^2_X})$	(1.862,0.888)	(s^2_{x2}, s^2_{y2})	(s^2_{x3}, s^2_{y3})	⋯
$\hat{\beta}_0$	7.446	$\hat{\beta}_{0,2}$	$\hat{\beta}_{0,3}$	⋯
$\hat{\beta}_1$	-0.566	$\hat{\beta}_{1,2}$	$\hat{\beta}_{1,3}$	⋯

Repeated Sampling

Now let's take 10000 samples, each of size $n = 40$. For each sample, we will run a linear regression, as before. How do we do this in R?

First, create matrix that will hold output:

```
holder <- matrix(data = NA, ncol = 2, nrow = sims)
colnames(holder) <- c("intercept", "slope")
```

Now, using for loops:

```
sims <- 10000
set.seed(02138)
for(i in 1:sims){
my.samp <- Leinhardt[sample(nrow(Leinhardt),size=40,
    replace=FALSE),]
samp.lm <- lm(linfant ~ lincome, data=my.samp)
holder[i,1] <- samp.lm$coefficients[1]
holder[i,2] <- samp.lm$coefficients[2] }
```

More Efficient Repeated Sampling

But we know for loops are not the most efficient for this task, so how can we do this using `replicate()`?
First, define function that we will replicate:

```
sample.lm.fxn <- function() {
my.samp <- Leinhardt[sample(nrow(Leinhardt),size=40,
    replace=FALSE),]
samp.lm <- lm(linfant ~ lincome, data=my.samp)
holder <- samp.lm$coefficients[1]
holder[2] <- samp.lm$coefficients[2]
return(holder)
}
```

Now, tell `replicate()` to repeat the function above 10000 times:

```
set.seed(02138)
results <- replicate(10000, sample.lm.fxn())
```

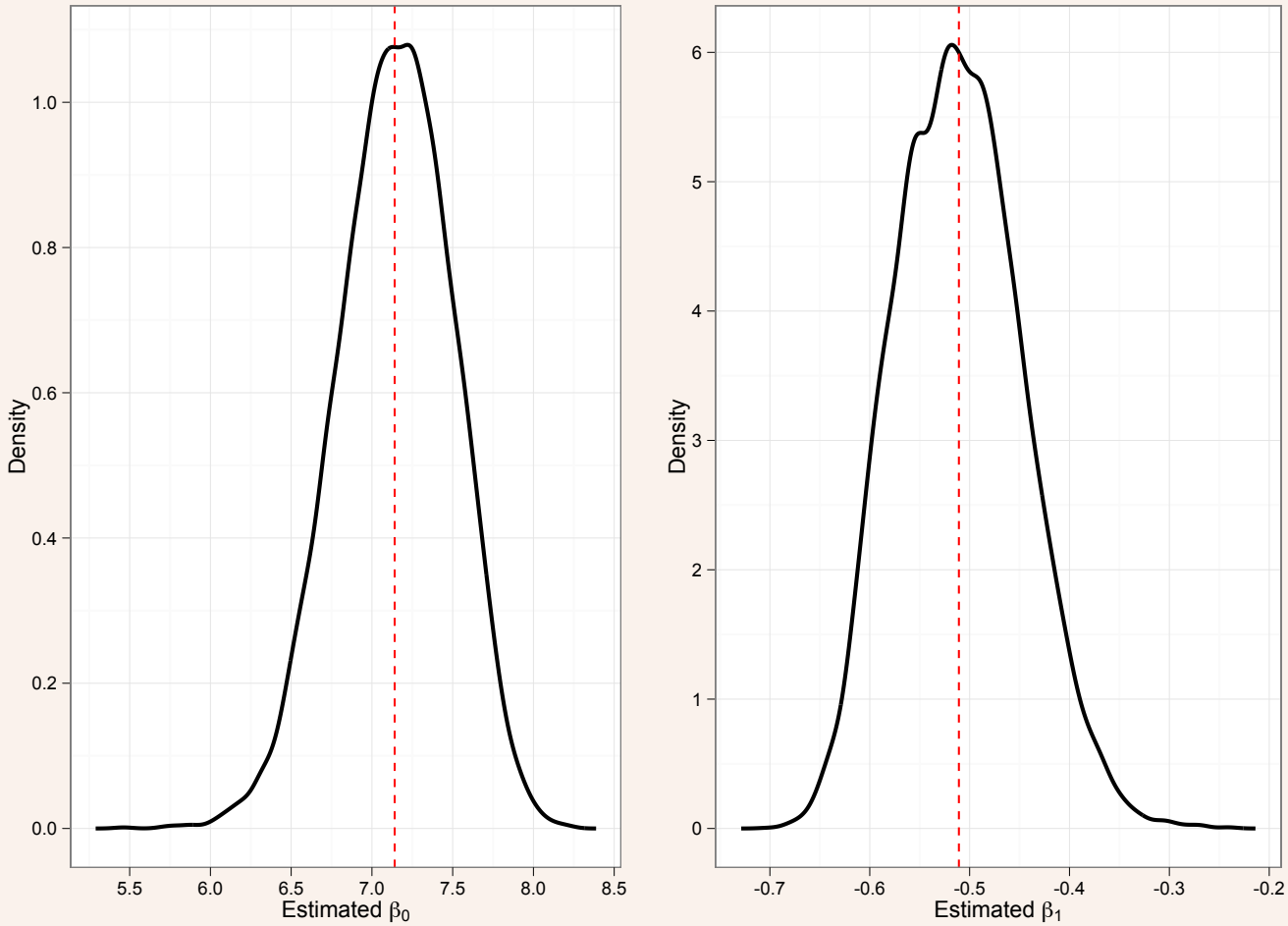
Sampling Distributions for Regression Coefficients

```
holder <- t(results)

par(mfrow = c(1,2))
plot(density(holder[,1]), col = "black",
main = "Sampling Distribution for Intercept", xlab=
    expression(beta[0]))
abline(v = mean(holder[,1]), col="red",lwd=2)

plot(density(holder[,2]), col = "black",
main = "Sampling Distribution for Slope", xlab=
    expression(beta[1]))
abline(v = mean(holder[,2]), col="red", lwd=2)
```

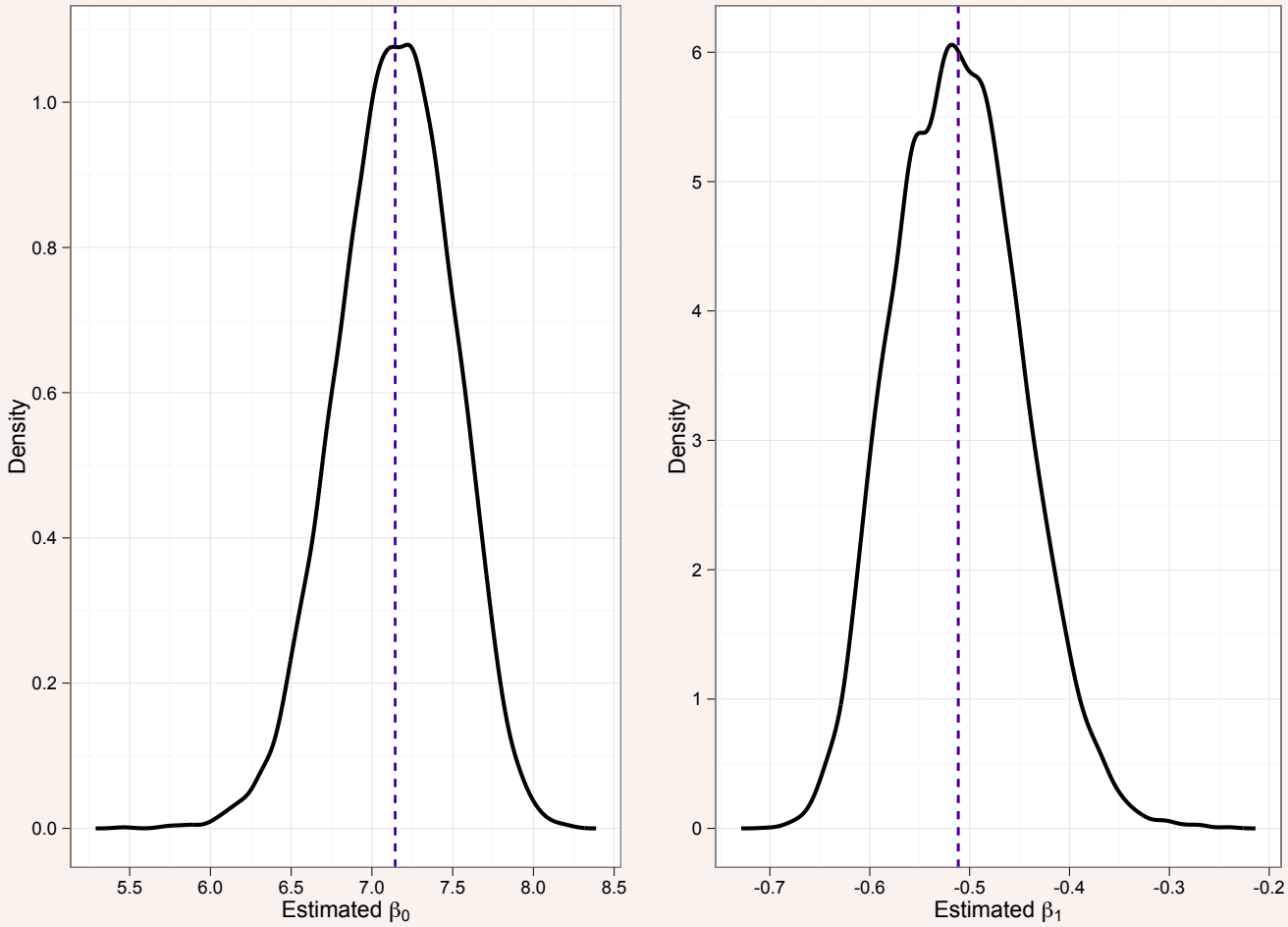
Sampling Distributions for Regression Coefficients



Means of the sampling distributions are plotted in red.

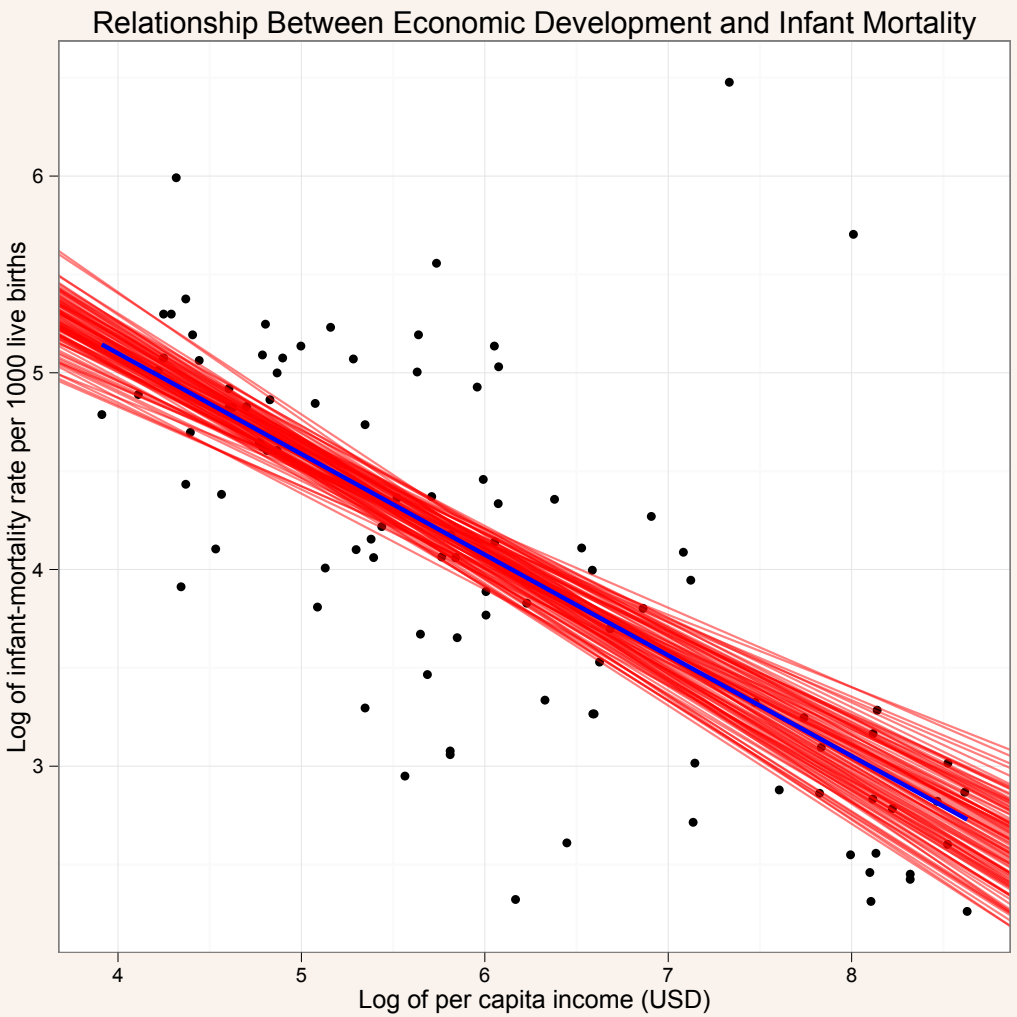
Sampling Distributions for Regression Coefficients

Now add the true, population regression parameters in blue:



Sampling Distributions for Regression Coefficients

Let's plot the first 100 regression lines (in red) and population regression line (in blue):



Sampling Distribution for $\hat{\beta}_1$

We know, from theory, that the true sampling distribution of $\hat{\beta}_1$ is:

$$\hat{\beta}_1 \sim \mathcal{N}\left(\beta_1, \frac{\sigma^2}{\sum_i (x_i - \bar{x})^2}\right)$$

Why might the standard error of the simulated sampling distribution not match the theoretical standard error?

From theory: $SE(\beta_1) = 0.0507$
From simulation: $SE(\beta_1) = 0.0623$

Possible explanations:

- Sampling without replacement
- Heteroskedasticity / non-constant variance

Estimating Sampling Distribution for $\hat{\beta}_1$

We can estimate sampling distribution for $\hat{\beta}_1$ as:

$$\hat{\beta}_1 \sim \mathcal{N}\left(\hat{\beta}_1, \frac{\hat{\sigma}^2}{\sum_i (x_i - \bar{x})^2}\right)$$

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^{n} u_i^2}{n-2} = \frac{SSR}{n-2}$$

To get estimated standard error of $\hat{\beta}_1$ in R:

```
sigma2.hat <- sum(residuals(samp.lm)^2)/(samp.lm$df)
denom <- sum((my.samp$lincome-mean(my.samp$lincome))
    ^2)
sqrt(sigma2.hat/denom)
```

Outline

Regression Output

This is what we get as regression output:

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.44624    0.41046  18.141  < 2e-16 ***
lincome     -0.56600    0.06412  -8.827 9.72e-11 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

How do we get the test statistic and the p-value?

Regression Output: Test Statistic

The test statistic for the null hypothesis that $\beta_1 = 0$ is just given by:

$$T = \frac{\hat{\beta}_1 - \beta_{1,H_0}}{\widehat{SE}(\hat{\beta}_1)} = \frac{\hat{\beta}_1 - 0}{\widehat{SE}(\hat{\beta}_1)}$$

Moreover, we know that $T \sim t_{n-2}$ under the null.

In R:

```
t.stat <- (summary(samp.lm)$coefficients[2,1]-0)/
    summary(samp.lm)$coefficients[2,2]
```

$T_{obs} = -8.8269$

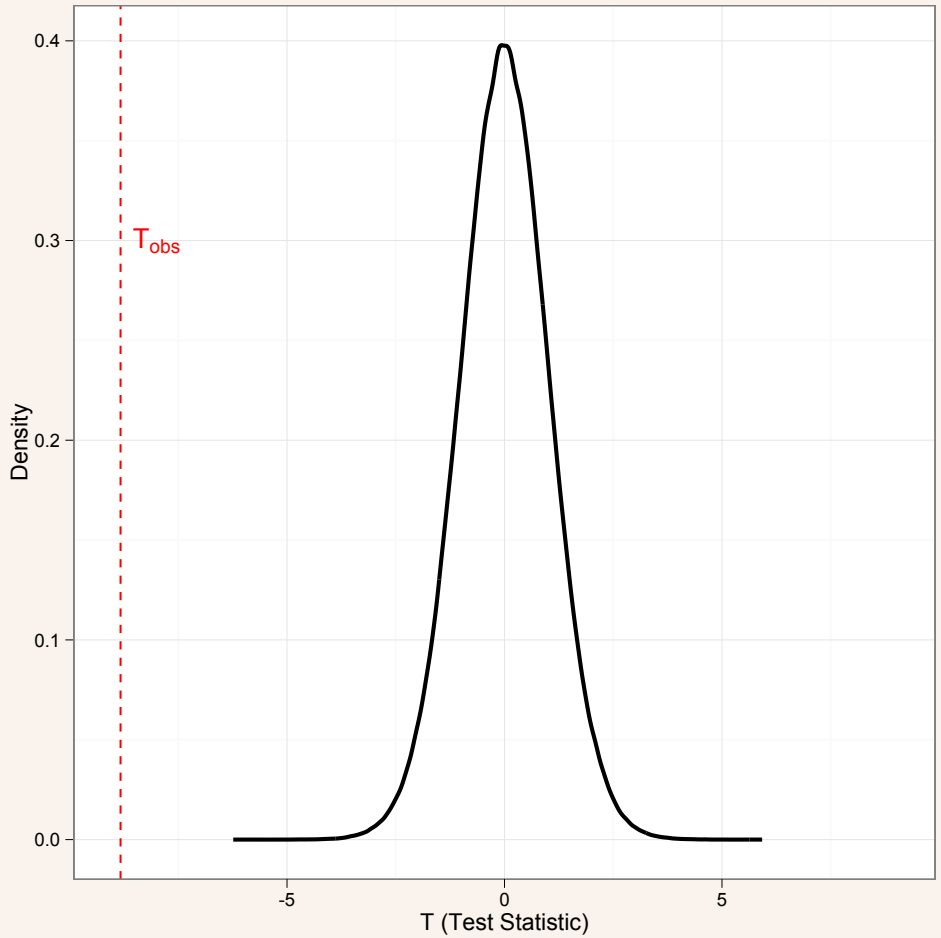
Regression Output

This is what we get as regression output:

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.44624    0.41046  18.141  < 2e-16 ***
lincome     -0.56600    0.06412  -8.827 9.72e-11 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

Regression Output: Test Statistic

We can plot the observed test statistic on top of the sampling distribution of the test statistic, which is a t-distribution with $n - 2$ degrees of freedom:



Regression Output: P-Value

The definition of the p-value is the probability of obtaining a value of the test statistic *at least as extreme* as the one you observed:

$$p = P(|T| \geq |T_{obs}|) = 2 \cdot P(T \geq |T_{obs}|)$$

In R:

```
2*pt(-abs(t.stat), df=38)
```

p-value = $9.7185e - 11$

Regression Output

This is what we get as regression output:

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.44624    0.41046  18.141  < 2e-16 ***
lincome     -0.56600    0.06412  -8.827 9.72e-11 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

p-value is less than $\alpha = 0.001$, so we can reject at $\alpha = 0.001$ level!

Regression Output: Confidence Intervals

Suppose we wanted a two-sided 95% confidence interval for β_1:

$$\hat{\beta}_1 \pm t_{\alpha/2} \cdot \widehat{SE}[\hat{\beta}_1]$$

In R:

```
coef(samp.lm)[2] + qt(0.025,df=38)*summary(samp.lm)$
    coefficients[2,2]
coef(samp.lm)[2] - qt(0.025,df=38)*summary(samp.lm)$
    coefficients[2,2]
```

95% CI: $[-0.6958, -0.4362]$