ECON-320-Lab-4-5

Sonan Memon

Introduction

- As economists, we run a lot of regressions; it is our bread and butter.
- By default, we estimate OLS, which is a linear regression with appealing properties.
- Our goal is to conduct causal inference.

Packages

```r
library(tinytex)

library(tidyverse)

library(dslabs)
library(dplyr)

library(ggplot2)

library(tibble)
library(modelsummary)

library(broom)
```

OLS 1:

$$y = \beta_0 + \beta_1 * x + e,\ \hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

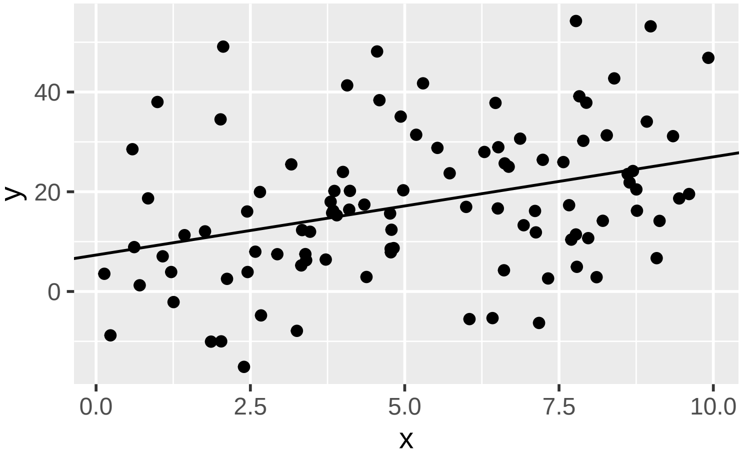
$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

OLS 1:

```
set.seed(1)
data <- tibble(x = runif(100,0, 10),
               e = rnorm(100,0, 15),
               y = 10 + 1.5*x + e)

ols_coefficients <- function(x, y) {
  x_bar <- mean(x)
  y_bar <- mean(y)
  beta_1 <- sum((x - x_bar)*(y - y_bar))/sum((x - x_bar)^2)
  beta_0 <- y_bar - beta_1*x_bar
  return(list(intercept = beta_0, slope = beta_1))
}

result <- ols_coefficients(data$x, data$y)

intercept <- result$intercept
slope <- result$slope
```

OLS 1: Scatterplot

```
ggplot(data, aes(x = x, y = y)) +
  geom_point() +
  geom_abline(slope = slope, intercept = intercept)
```



OLS 2:

- lm function: for univariate model.

```
result <- lm(data = data, y ~ x)

result$coefficients
```

```
(Intercept)           x
   7.310118    1.968515
```

```
#summary(result)
```

OLS 3:

- lm function: for multivariate case.

```
n = 100
set.seed(1)

data <- tibble(x = runif(n,0, 10),
               z = runif(n, 5, 15),
               e = rnorm(n,0, 15),
               y = 10 + 1.5*x - 3*z + e)


model1 <- lm(data = data, y ~ x)

model2 <- lm(data = data, y ~ x + z)
```

Regression Table

Table 1: Results From Simulated Data

	Regression 1	Regression 2
(Intercept)	-20.286***	9.752
	(3.625)	(6.275)
x	1.345*	1.396*
	(0.622)	(0.545)
z		-2.978***
		(0.537)

+ $p < 0.1$, * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

Figure 2: Simulated Data-1

Standardisation of Data

```
n = 100
set.seed(1)

# Generate data in a tibble
data = tibble(
e = rnorm(n, sd = 2),
v = rnorm(n, sd = 1),
x = runif(n, min = 0, max = 10),
y = 8 - 3*x + e,
z = 20 - 0.3*y + 3*x + v,
)

data_sn <- data %>% mutate(
x.sn = (x - mean(x))/sd(x),
y.sn = (y - mean(y))/sd(y)
)
```

```
ggplot()+
geom_density(data = data, aes(x = x), color = "blue")+
geom_density(data = data_sn, aes(x = x.sn), color = "red")
```

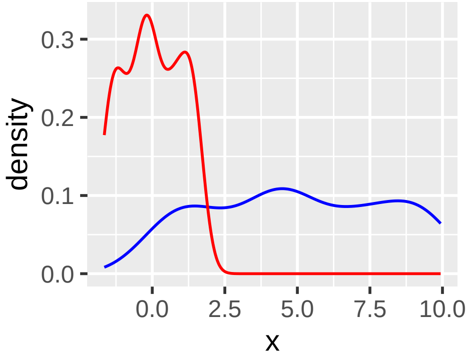


Figure 3: Densities

Confidence Interval

```
lm1 <-lm(data = data, z~y)
lm2 <- lm(data = data, z~y + x)

results <- rbind(tidy(lm1), tidy(lm2)) %>%
filter(term == "y")
results

results <- results %>% mutate(
lower_bound = estimate - 1.96*std.error,
upper_bound = estimate + 1.96*std.error,
model = c("without x", "with x")
)
results
```

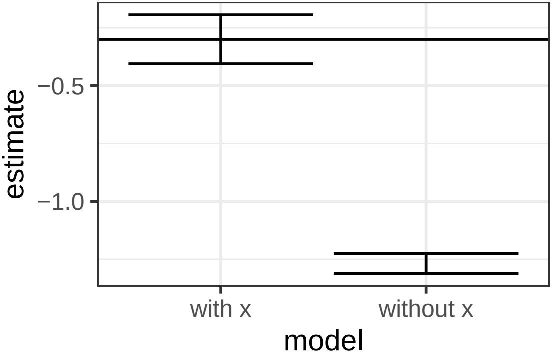


Figure 4: Confidence Bands

Hypothesis Testing:

- Null hypothesis verus alternative hypothesis.
- $$H_0 : \beta = 0.5 \;\; H_a : \beta \neq 0.5$$
- Test statistic:
$$\frac{\hat{\beta} - \beta}{s.e}$$

Hypothesis Testing:

```
beta1 = results$estimate[1]
st.err1 = results$std.error[1]

test1 = (beta1 - (-0.3))/st.err1
test1
```

```
[1] -44.40833
```

```
beta2 = results$estimate[2]
st.err2 = results$std.error[2]

test2 = (beta2 - (-0.3))/st.err2
test2
```

```
[1] 0.001330759
```

Ifelse Again

```r
ifelse(abs(test1) > 1.96,
"Reject the null hypothesis",
"Fail to reject the null hypothesis")
```

```
[1] "Reject the null hypothesis"
```