Lecture 10: Properties of Least Squares Estimates
STAT 632, Spring 2020

Preliminaries: Let a and b be constants, and X a random variable.

$$\mathrm{E}(aX+b) = a\,E(X)+b$$
$$\mathrm{Var}(aX+b) = a^2\,Var(X)$$

Random Vectors
Let $X_1, X_2, \cdots, X_n$ be random variables. An $n \times 1$ random vector is given by[1]

$$\boldsymbol{X} = \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{pmatrix}$$

The **variance-covariance** matrix of $\boldsymbol{X}$ is the $n \times n$ symmetric matrix

$$\mathrm{Var}(\boldsymbol{X}) = \begin{pmatrix} \mathrm{Var}(X_1) & \mathrm{Cov}(X_1,X_2) & \cdots & \mathrm{Cov}(X_1,X_n) \\ \mathrm{Cov}(X_2,X_1) & \mathrm{Var}(X_2) & \cdots & \mathrm{Cov}(X_2,X_n) \\ \vdots & & & \\ \mathrm{Cov}(X_n,X_1) & \mathrm{Cov}(X_n,X_2) & \cdots & \mathrm{Var}(X_n) \end{pmatrix}$$

$$Cov(X,Y) = E[(X-E(X))(Y-E(Y))] = E(XY)-E(X)E(Y)$$

Sample covariance

$$S_{xy} = \frac{\sum (x_i-\bar{x})(y_i-\bar{y})}{n-1}$$

Note that $\mathrm{Cov}(X_i, X_i) = Var(X_i)$.

Properties of Expectation and Variance for Random Vectors
Let $\boldsymbol{A}$ be an $m \times n$ matrix of constants, $\boldsymbol{b}$ an $m \times 1$ vector of constants, and $\boldsymbol{X}$ an $n \times 1$ random vector, then

R: cov(x,y)

cor(x,y)

$$r = \frac{S_{xy}}{S_x S_y}$$

$$\mathrm{E}(\boldsymbol{AX}+\boldsymbol{b}) = \boldsymbol{A}\mathrm{E}(\boldsymbol{X})+\boldsymbol{b}$$
$$\mathrm{Var}(\boldsymbol{AX}+\boldsymbol{b}) = \boldsymbol{A}\mathrm{Var}(\boldsymbol{X})\boldsymbol{A}'$$

$$E(X) = \begin{pmatrix} E(X_1) \\ E(X_2) \\ \vdots \\ E(X_n) \end{pmatrix}$$

---

[1]Note that the $\boldsymbol{X}$ here is different than the $n \times (p+1)$ model matrix in regression.

Properties of the Multiple Linear Regression Model

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

- $\boldsymbol{Y}$ is an $n \times 1$ response vector
- $\boldsymbol{X}$ is an $n \times (p+1)$ design matrix for the predictors
- $\boldsymbol{\beta}$ is a $(p+1) \times 1$ vector of regression parameters
- $\boldsymbol{e}$ is an $n \times 1$ vector of random errors; assuming $e_i \sim N(0, \sigma^2)$, independently

The variance-covariance matrix for the random errors $\boldsymbol{e}$ is given by

$$\text{Var}(\boldsymbol{e}) = \begin{pmatrix} \sigma^2 & 0 & \cdots & 0 \\ 0 & \sigma^2 & \cdots & 0 \\ \vdots & & & \\ 0 & 0 & \cdots & \sigma^2 \end{pmatrix} = \sigma^2 \boldsymbol{I}_n$$

which follows since $\text{Var}(e_i) = \sigma^2$, and $\text{Cov}(e_i, e_j) = 0$ when $i \neq j$ (by independence).

Use the properties of expectation and variance for random vectors to find $\text{E}(\boldsymbol{Y})$, $\text{Var}(\boldsymbol{Y})$, and the distribution of $\boldsymbol{Y}$.

$$E(Y) = E(\underbrace{X\beta}_{\text{constant}} + e) = X\beta + \underbrace{E(e)}_{=0} = X\beta$$

$$Var(Y) = Var(X\beta + e) = Var(e) = \sigma^2 I_n$$

$$Y \sim N(X\beta, \sigma^2 I_n)$$

Properties of Least Squares Estimates

Recall that the least squares estimates of the population parameters $\boldsymbol{\beta}$ is given by the $(p+1) \times 1$ vector

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

The estimator $\hat{\boldsymbol{\beta}}$ is a random vector since it varies from sample to sample.

Show that $E(\hat{\boldsymbol{\beta}}) = \boldsymbol{\beta}$, the least squares estimates are unbiased.

$$E(\hat{\beta}) = E[(X'X)^{-1}X'Y]$$
$$= (X'X)^{-1}X'E(Y)$$
$$= (X'X)^{-1}X'X\beta$$
$$= \beta \qquad \text{Since } A^{-1}A = I$$

Show that that variance-covariance matrix for the least squares estimates is given by $\mathrm{Var}(\hat{\boldsymbol{\beta}}) = \sigma^2(\boldsymbol{X'X})^{-1}$

$$\mathrm{Var}(\hat{\beta}) = \mathrm{Var}[(X'X)^{-1}X'Y]$$
$$= (X'X)^{-1}X'\,\mathrm{Var}(Y)[(X'X)^{-1}X']'$$
$$= (X'X)^{-1}X'(\sigma^2 I_n)X(X'X)^{-1}$$
$$= \sigma^2(X'X)^{-1}X'X(X'X)^{-1}$$
$$= \sigma^2(X'X)^{-1}$$

since $(A^{-1})' = (A')^{-1}$

$[(X'X)^{-1}]'$

$= [(X'X)']^{-1}$

$= (X'X)^{-1}$

The variance of a particular estimate is given by the diagonal entry:

$$\mathrm{Var}(\hat{\beta}_j) = \sigma^2(\boldsymbol{X'X})^{-1}_{j+1,j+1} \quad \text{for } j = 0, \cdots, p$$

This is how standard errors are calculated when constructing confidence intervals for the parameters. For example, a $1-\alpha$ confidence interval for $\hat{\beta}_j$ is given by

$$\hat{\beta}_j \pm t_{\alpha/2;n-p-1}\mathrm{se}(\hat{\beta}_j)$$

where $\mathrm{se}(\hat{\beta}_j) = \hat{\sigma}\sqrt{(\boldsymbol{X'X})^{-1}_{j+1,j+1}}$, and $\hat{\sigma} = \sqrt{\mathrm{RSS}/(n-p-1)}$

Using the menu pricing data set (lecture 7), the code below demonstrates how to manually compute the least squares estimates, variance-covariance matrix, and standard errors. The results are compared to the output from `lm()`. Note that it is always better to use `lm()` than to do the computations manually. I am just showing this to verify all the formulas.

```
nyc <- read.csv("https://ericwfox.github.io/data/nyc.csv")

# response vector
Y <- matrix(nyc$Price, ncol=1)

# design matrix
X <- cbind(Intercept = 1, nyc[,c('Food', 'Decor', 'East')])
X <- as.matrix(X)
rownames(X) <- nyc$Restaurant
X[1:5,]

##                       Intercept Food Decor East
## Daniella Ristorante           1   22    18    0
## Tello's Ristorante            1   20    19    0
## Biricchino                    1   21    13    0
## Bottino                       1   20    20    0
## Da Umberto                    1   24    19    0

# manually calculate least squares estimates
betaHat <- solve(t(X) %*% X) %*% t(X) %*% Y
betaHat

##                 [,1]
## Intercept -24.026880
## Food        1.536346
## Decor       1.909373
## East        2.067013

# compare with lm()
lm1 <- lm(Price ~ Food + Decor + East, data=nyc)
coef(lm1)

## (Intercept)        Food       Decor        East
##  -24.026880    1.536346    1.909373    2.067013
```

```r
# manually calculate standard errors for least squares estimates
n <- nrow(nyc)
p <- 3
resid <- as.numeric(Y - X %*% betaHat)
sigmaHat2 <- sum(resid^2) / (n-p-1)
covBetaHat <- sigmaHat2 * solve(t(X) %*% X)
covBetaHat

##            Intercept        Food       Decor        East
## Intercept 21.8344883 -0.94964667 -0.12475347  0.19880619
## Food      -0.9496467  0.06926176 -0.02530816 -0.04612459
## Decor     -0.1247535 -0.02530816  0.03610588  0.01149201
## East       0.1988062 -0.04612459  0.01149201  0.86827717

seBetaHat <- sqrt(diag(covBetaHat))
seBetaHat

## Intercept      Food     Decor      East
## 4.6727388 0.2631763 0.1900155 0.9318139

# compare with lm()
summary(lm1)$coef

##              Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) -24.026880  4.6727388 -5.141926 7.669997e-07
## Food          1.536346  0.2631763  5.837705 2.758775e-08
## Decor         1.909373  0.1900155 10.048513 8.201940e-19
## East          2.067013  0.9318139  2.218268 2.791009e-02

vcov(lm1)

##             (Intercept)        Food       Decor        East
## (Intercept)  21.8344883 -0.94964667 -0.12475347  0.19880619
## Food         -0.9496467  0.06926176 -0.02530816 -0.04612459
## Decor        -0.1247535 -0.02530816  0.03610588  0.01149201
## East          0.1988062 -0.04612459  0.01149201  0.86827717
```