Unit IV

ECON 3406

Dr. Josh Martin

Introduction to linear regression

$$y = \beta_0 + \beta_1 x + \epsilon$$

- Linear regression is the statistical method for fitting a line to data where the relationship between two variables, x and y, can be modeled by a straight line with some error
  - β_0 = y-intercept
  - β_1 = slope
  - ϵ = error term (a.k.a residual)

Introduction to linear regression

- When we use x to predict y, we usually call:
  - x the explanatory, predictor or independent variable,
  - y the response or dependent variable
- As we move forward in this chapter, we will learn about:
  - criteria for line-fitting
  - the uncertainty associated with estimates of model parameters

Introduction to linear regression

Introduction to linear regression

World's most useless regression

Possums...?

```r
x <- read.csv("possum.csv")
head(x)
```

```
  site pop sex age head_l skull_w total_l tail_l
1    1 Vic   m   8   94.1    60.4    89.0   36.0
2    1 Vic   f   6   92.5    57.6    91.5   36.5
3    1 Vic   f   6   94.0    60.0    95.5   39.0
4    1 Vic   f   6   93.2    57.1    92.0   38.0
5    1 Vic   f   2   91.5    56.3    85.5   36.0
6    1 Vic   f   1   93.1    54.8    90.5   35.5
```

World's most useless regression

```
lm1 <- lm(head_l ~ total_l, x)
coefficients(lm1)
```

```
(Intercept)     total_l
 42.7097931   0.5729013
```

World's most useless regression

```r
summary(x$total_l)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  75.00   84.00   88.00   87.09   90.00   96.50 
```

```r
xs <- seq(70, 100, 1)

b0 <- coefficients(lm1)[1]
b1 <- coefficients(lm1)[2]
y_hat <- b0 + b1 * xs

head(
  data.frame(
  x = xs,
  y_hat = round(y_hat, 1)
  )
)
```

```
   x y_hat
1 70  82.8
2 71  83.4
3 72  84.0
4 73  84.5
5 74  85.1
6 75  85.7
```

World's most useless regression

Residuals

$$Data = Fit + Residual$$

- Residuals are the leftover variation after accounting fo the model fit
- The residual for the i^{th} observation (x_i, y_i) is the difference of the observed response (y_i) and the response that we would predict based on the model fit $(\hat{y}_i)$

$$\epsilon_i = y_i - \hat{y}_i$$

Residuals

```r
p <- data.frame(
  x = x$total_l,
  y = x$head_l,
  y_hat = predict(lm1)
)
p$residual <- p$y_hat - p$y
head(p)
```

```
     x    y    y_hat   residual
1 89.0 94.1 93.69801 -0.4019925
2 91.5 92.5 95.13026  2.6302607
3 95.5 94.0 97.42187  3.4218658
4 92.0 93.2 95.41671  2.2167113
5 85.5 91.5 91.69285  0.1928530
6 90.5 93.1 94.55736  1.4573594
```

Residuals

Residuals

Residuals

Correlation

$$R = \frac{1}{n-1} \sum_{i=1}^{n} \frac{x_i - \bar{x}}{s_x} \frac{y_i - \bar{y}}{s_y}$$

- Correlation describes the strength of the linear relationship between two variables
  - We denote the correlation by R
  - Always takes values between -1 and 1

Correlation

```r
zx <- (x$total_l - mean(x$total_l)) / sd(x$total_l)
zy <- (x$head_l - mean(x$head_l)) / sd(x$head_l)
sum(zx * zy)/ (nrow(x) - 1)
```

```
[1] 0.6910937
```

```r
cor(x$head_l, x$total_l)
```

```
[1] 0.6910937
```

Correlation

Correlation

Least squares regression

"Line of best fit"

- We begin by thinking about what we mean by "best"
  - Mathematically, we want a line that minimizes the magnitude of residuals
  - Most commonly, this is done by minimizing the sum of the squared residuals

$$minarg? = e_1^2 + e_2^2 + \ldots + e_n^2$$

Least squares regression

Conditions for the least squares line

- **Linearity**: The data should show a linear trend
- **Normal residuals**: Generally, the residuals must be nearly normal. When this condition is found to be unreasonable, it is usually because of outliers or concerns about influential points
- **Constant variability**: The variability of points around the least squares line remains roughly constant
- **Independent observations**: Be cautious about applying regression to time series data, which are sequential observations in time such as a stock price each day

Least squares regression

Conditions for the least squares line

Least squares regression

New example: Elmhurst College

```
x <- read.csv("elmhurst.csv")
head(x)
```

```
  family_income gift_aid price_paid
1        92.922    21.72      14.28
2         0.250    27.47       8.53
3        53.092    27.75      14.25
4        50.200    27.22       8.78
5       137.613    18.00      24.00
6        47.957    18.52      23.48
```

```
lm2 <- lm(gift_aid ~ family_income, x)
coefficients(lm2)
```

```
  (Intercept) family_income
  24.31932901   -0.04307165
```

```
b0 <- coefficients(lm2)[1]
b1 <- coefficients(lm2)[2]
```

Least squares regression

Scatterplot

Least squares regression

Residuals

```r
y <- data.frame(
  x = x$family_income,
  y = x$gift_aid,
  y_hat = predict(lm2)
)
y$residuals <- y$y - y$y_hat
head(y)
```

```
        x     y    y_hat  residuals
1  92.922 21.72 20.31702  1.4029751
2   0.250 27.47 24.30856  3.1614389
3  53.092 27.75 22.03257  5.7174311
4  50.200 27.22 22.15713  5.0628679
5 137.613 18.00 18.39211 -0.3921097
6  47.957 18.52 22.25374 -3.7337418
```

```r
head(y$residuals)
```

```
[1]  1.4029751  3.1614389  5.7174311  5.0628679 -0.3921097 -3.7337418
```

```r
head(residuals(lm2))
```

```
         1          2          3          4          5          6
 1.4029751  3.1614389  5.7174311  5.0628679 -0.3921097 -3.7337418
```

Least squares regression

Residuals

Least squares regression

Slope

$$b_1 = \frac{s_y}{s_x} R$$

```
mean_y <- round(mean(x$gift_aid)*1000)
sd_y <- round(sd(x$gift_aid)*1000)
mean_x <- round(mean(x$family_income)*1000, 0)
sd_x <- round(sd(x$family_income)*1000, 0)
r <- round(cor(x$gift_aid, x$family_income), 3)
r
```

```
[1] -0.499
```

```
sd_y / sd_x * r
```

```
[1] -0.04311361
```

```
coefficients(lm2)[2]; b1
```

```
family_income
  -0.04307165
```

```
family_income
  -0.04307165
```

Least squares regression

Intercept

$$y - y_0 = m \times (x - x_0)$$

- You might recall the point-slope form of a line from math class, which we can use to find the model fit, including the estimate of β_0

$$y - \bar{y} = beta_1 \times (x - \bar{x})$$

- To find the y-intercept, set $x = 0$

$$b_0 = \bar{y} - beta_1 \bar{x}$$

Least squares regression

Intercept

```
1 (mean_y - b1*mean_x)/1000
```

```
family_income
     24.31979
```

```
1 coefficients(lm2)[1]; b0
```

```
(Intercept)
   24.31933
```

```
(Intercept)
   24.31933
```

Least squares regression

Interpretation: Slope

- What do these regression coefficients mean?
  - The slope describes the estimated difference in the y variable if the explanatory variable x for a case happened to be **one unit** larger.
  - For each additional \$1,000 of family income, we would expect a student to receive a net difference of \$1,000×(−0.0431) = −\$43.10 in aid on average, i.e. \$43.10 less.

Least squares regression

Interpretation: Slope

- What do these regression coefficients mean?
  - The intercept describes the average outcome of y if $x = 0$ and the linear model is valid all the way to $x = 0$, which in many applications is not the case.
  - The estimated intercept $\beta_0 = 24,319$ describes the average aid if a student's family had no income.
- We must be cautious in this interpretation: while there is a real association, we cannot interpret a **causal** connection between the variables.

Least squares regression

Extrapolation

- Applying a model estimate to values outside of the realm of the original data is called **extrapolation**.
  - If we extrapolate, we are making an unreliable bet that the approximate linear relationship will be valid in places where it has not been analyzed.

```
1 as.numeric(b0 + b1*1000)*1000
```

```
[1] -18752.32
```

Least squares regression

Extrapolation

Least squares regression

Strength of a fit: R-squared

- We evaluated the strength of the linear relationship between two variables earlier using the correlation, R.
  - However, it is more common to explain the strength of a linear fit using R^2
  - If provided with a linear model, we might like to describe how closely the data cluster around the linear fit.

Least squares regression

Strength of a fit: R-squared

$$R^2 = \frac{s_y^2 - s_\epsilon^2}{s_y^2}$$

```
sy2 <- sd(y$y)^2
se2 <- sd(y$residuals)^2

(sy2 - se2) / sy2
```

```
[1] 0.2485582
```

```
summary(lm2)$r.squared
```

```
[1] 0.2485582
```

- About 25% of the variation in gift aid can be explained by differences in family income.

Least squares regression

Example 3: Categorical Predictors

```r
x <- read.csv("mariokart.csv")
y <- data.frame(
  new = ifelse(x$cond == "new", 1, 0),
  price = x$total_pr
)
summary(y)
```

```
      new              price
 Min.   :0.0000   Min.   : 28.98
 1st Qu.:0.0000   1st Qu.: 41.17
 Median :0.0000   Median : 46.50
 Mean   :0.4126   Mean   : 49.88
 3rd Qu.:1.0000   3rd Qu.: 53.99
 Max.   :1.0000   Max.   :326.51
```

```r
z <- (y$price - mean(y$price)) / sd(y$price)
y <- y[z <= 2.576,]

lm3 <- lm(price ~ new, y)
coefficients(lm3)
```

```
(Intercept)         new
   42.87110    10.89958
```

Least squares regression

Example 3: Categorical Predictors

Least squares regression

Example 3: Categorical Predictors

- The estimated intercept is the value of the response variable for the first category.
  - The intercept is the estimated price when new takes value 0, i.e. when the game is in used condition.
  - That is, the average selling price of a used version of the game is $42.87.
- The estimated slope is the average change in the response variable between the two categories.
  - The slope indicates that, on average, new games sell for about $10.90 more than used games.

Least squares regression

Highly senstive to outliers

Least squares regression

Highly senstive to outliers

Inference for linear regression

Example: Midterm elections & unemployment

- **Context:** U.S. House elections occur every two years; those held **midway through a presidential term** are called **midterms**.
- **Theory:** When **unemployment is high**, the **President's party performs worse** in midterms.
- **Approach:** Use **historical data (1898–2018)**—excluding **Great Depression** years—to test whether **unemployment predicts midterm losses**.

Inference for linear regression

Example: Midterm elections & unemployment

Inference for linear regression

Example: Midterm elections & unemployment

```
x <- read.csv("midterms_house.csv")
head(x)
```

```
  year               potus      party unemp house_change
1 1899    William McKinley Republican 11.62    -9.223301
2 1903  Theodore Roosevelt Republican  4.30    -4.275907
3 1907  Theodore Roosevelt Republican  3.29   -12.291499
4 1911 William Howard Taft Republican  5.86   -26.590640
5 1915      Woodrow Wilson   Democrat  6.63   -20.962199
6 1919      Woodrow Wilson   Democrat  3.38   -10.280374
```

```
z <- (x$unemp - mean(x$unemp)) / sd(x$unemp)
x <- x[z <= 2.576,]

lm4 <- lm(house_change ~ unemp, x)
coefficients(lm4)
```

```
(Intercept)       unemp
 -7.3644063  -0.8897261
```

Inference for linear regression

Example: Midterm elections & unemployment

Inference for linear regression

Example: Midterm elections & unemployment

- We might wonder, is this convincing evidence that the "true" linear model has a negative slope?
  - That is, do the data provide strong evidence that the political theory is accurate, where the unemployment rate is a useful predictor of the midterm election?
- $H_0 : \beta_1 = 0$
- $H_0 : \beta_1 \neq 0$

Inference for linear regression

Example: Midterm elections & unemployment

```r
sd_x <- sd(x$unemp)
sd_y <- sd(x$house_change)
n <- nrow(x)
z <- list()
for(i in 1:10000){
  set.seed(i)
  y <- data.frame(
    x = rnorm(n, mean = 0, sd = sd_x),
    y = rnorm(n, mean = 0, sd = sd_y)
  )
  z[[length(z)+1]] <- coefficients(lm(y ~ x, y))[2]
}
z <- unlist(z)
dz <- density(z)
```

Inference for linear regression

Example: Midterm elections & unemployment

Inference for linear regression

Example: Midterm elections & unemployment

```
b1 <- as.numeric(coefficients(lm4)[2])

mean(ifelse(b1 >= z, 1, 0))*2
```

```
[1] 0.2972
```

```
summary(lm4)
```

```
Call:
lm(formula = house_change ~ unemp, data = x)

Residuals:
     Min       1Q   Median       3Q      Max
-14.0124  -7.6989   0.0913   7.2974  16.1447

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -7.3644     5.1553  -1.429    0.165
unemp        -0.8897     0.8350  -1.066    0.296

Residual standard error: 8.913 on 27 degrees of freedom
Multiple R-squared:  0.04035,   Adjusted R-squared:  0.004812
F-statistic: 1.135 on 1 and 27 DF,  p-value: 0.2961
```

Inference for linear regression

- We usually rely on statistical software to identify point estimates, standard errors, test statistics, and p-values in practice.
- However, be aware that software will not generally check whether the method is appropriate, meaning we must still verify conditions are met.

Inference for linear regression

Elmhurst College example

```r
x <- read.csv("elmhurst.csv")
sd_x <- sd(x$family_income)
sd_y <- sd(x$gift_aid)
n <- nrow(x)
z <- list()
for(i in 1:10000){
  set.seed(i)
  y <- data.frame(
    x = rnorm(n, mean = 0, sd = sd_x),
    y = rnorm(n, mean = 0, sd = sd_y)
  )
  z[[length(z)+1]] <- coefficients(lm(y ~ x, y))[2]
}
z <- unlist(z)
dz <- density(z)
```

Inference for linear regression

Elmhurst College example

Inference for linear regression

Elmhurst College example

```
lm2 <- lm(gift_aid ~ family_income, x)
b1 <- as.numeric(coefficients(lm2)[2])

mean(ifelse(b1 >= z, 1, 0))*2
```

```
[1] 0.001
```

```
summary(lm2)
```

```
Call:
lm(formula = gift_aid ~ family_income, data = x)

Residuals:
     Min       1Q   Median       3Q      Max
-10.1128  -3.6234  -0.2161   3.1587  11.5707

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   24.31933    1.29145  18.831  < 2e-16 ***
family_income -0.04307    0.01081  -3.985 0.000229 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.783 on 48 degrees of freedom
```