Linear Regression

Computing the slope and intercept

Jenine Harris
Brown School

Importing and merging data sources

```
# distance to syringe program data
dist.ssp <- read.csv(file = "/Users/harrisj/Box/teaching/Teaching/Fall20

# summary
summary(object = dist.ssp)
```

```
##     county           STATEABBREVIATION     dist_SSP       HIVprevalence
##  Length:500         Length:500         Min.   :  0.00   Min.   :  -1.00
##  Class :character   Class :character   1st Qu.: 35.12   1st Qu.:  52.98
##  Mode  :character   Mode  :character   Median : 75.94   Median : 101.15
##                                        Mean   :107.74   Mean   : 165.75
##                                        3rd Qu.:163.83   3rd Qu.: 210.35
##                                        Max.   :510.00   Max.   :2150.70
##  opioid_RxRate       pctunins        metro
##  Min.   :  0.20   Min.   : 3.00   Length:500
##  1st Qu.: 45.12   1st Qu.: 8.60   Class :character
##  Median : 62.40   Median :11.70   Mode  :character
##  Mean   : 68.33   Mean   :12.18
##  3rd Qu.: 89.95   3rd Qu.:15.00
##  Max.   :345.10   Max.   :35.90
```

Codebook

Leslie looked through the variables and the codebook and determined that the variables had the following meanings:

- county: the county name
- STATEABBREVIATION: the two-letter abbreviation for the state the county is in
- dist_SSP: distance in miles to the nearest syringe services program
- HIVprevalence: people age 13 and older living with diagnosed HIV per 100,000
- opioid_RxRate: number of opioid prescriptions per 100 people
- pctunins: percentage of the civilian noninstitutionalized population with no health insurance coverage
- metro: county is non-metro, which includes open countryside, rural towns, or smaller cities with up to 49,999 people, or metro

Computing the slope and intercept in a simple linear regression

- For simple linear regression, *simple* does not mean *easy*, instead it is the term used for a regression model with *one predictor*.
- For example, a simple linear regression model could be used to examine the relationship between the percentage of people without health insurance and the distance to a syringe program for a county.
- Perhaps lack of insurance is related to SES and that counties with poorer residents were likely to be further from health resources like needle exchange programs.
- To understand this relationship between one predictor and an outcome, use a *simple linear regression model*.
- Like the t-test and chi-squared, linear regression is appropriate for examining relationships in a sample to understand what is happening in the population sampled.

Exploring the relationship with a scatterplot

- The line through the figure was the simple linear regression line that they would be estimating and interpreting for the relationship between percentage of people without health insurance and distance to a syringe program for a county.

```
dist.ssp %>%
  ggplot(aes(x = pctunins, y = dist_SSP)) +
  geom_point(aes(size = "County"), color = "#7463AC", alpha = .6) +
  geom_smooth(aes(linetype = "Linear fit line"), method = "lm", se = FAL
  theme_minimal() +
  labs(x = "Percent uninsured", y = "Miles to syringe program") +
  scale_size_manual(values = 2, name = "") +
  scale_linetype_manual(values = 1, name = "")
```

Computing the slope of the line

- The equation for the line with the independent variable x being the percentage of uninsured people in a county and the outcome variable y being the distance in miles to a needle exchange program:
  - $distance = b_0 + b_1 * uninsured + error$
- The formula to compute the slope uses the difference between the x and y for each observation and the overall mean values of x and y .
  - $b_1 = \frac{\sum_{i=1}^{n}(x_i - m_x)(y_i - m_y)}{\sum_{i=1}^{n}(x_i - m_x)^2}$

Where:

- i is an individual observation, in this case a county
- n is the sample size, in this case 500
- x_i is the value of `pctunins` for i
- m_x is the mean value of `pctunins` for the sample
- y_i is the value of `dist_SSP` for i
- m_y is the mean value of `dist_SSP` for the sample
- σ is the symbol for sum
- b is the slope

Using the slope to find the intercept

- Once the slope is computed, the intercept can be computed by putting the slope and the values of m_x and m_y into the equation for the line: $m_y = b_0 + b_1 \cdot m_x$ and solving it for b_0 , which is the y-intercept.

- Because this method of computing the slope and intercept relies on the squared differences and works to minimize the residuals overall, it is often called **Ordinary Least Squares** or **OLS** regression.

Estimating the linear regression model in R

- If the electricity went off, the slope and intercept of a line could still be calculated by hand using the OLS method without too much trouble, depending on the sample size.
- With electricity, R can do the work using the `lm()` function; `lm` stands for *linear model*.
- The `lm()` function takes two arguments, `formula =` and `data =`.
- There is also an `na.action =` option to deal with missing values even though these data do not have any.

```
# linear regression of distance to syringe program by percent uninsured
dist.by.unins <- lm(formula = dist_SSP ~ pctunins,
                    data = dist.ssp, na.action = na.exclude)
summary(dist.by.unins)
```

```
##
## Call:
## lm(formula = dist_SSP ~ pctunins, data = dist.ssp, na.action = na.exclude)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -217.71  -60.86  -21.61   47.73  290.77
##
```

Navigating the linear regression output

```
##
## Call:
## lm(formula = dist_SSP ~ pctunins, data = dist.ssp, na.action = na.exclude)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -217.71  -60.86  -21.61   47.73  290.77
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  12.4798    10.1757   1.226    0.221
## pctunins      7.8190     0.7734  10.110   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 85.91 on 498 degrees of freedom
## Multiple R-squared:  0.1703,    Adjusted R-squared:  0.1686
## F-statistic: 102.2 on 1 and 498 DF,  p-value: < 2.2e-16
```

- $distance = 12.48 + 7.82 * uninsured$
- Use the regression model to predict distance to syringe program for a county with 10% of the residents uninsured.

Understanding residuals

- The linear fit line is based on the values of the intercept and slope that were the best at minimizing the distances between all the points and the regression line.
- These distances are called *residuals* and are the leftover information that the line does not explain.

Interpreting residuals

- The top left graph is the ideal since it would predict every observation perfectly, but this is not realistic in most research where there are many potential sources of error.
- In the remaining three graphs, all the points stayed in the same place, but the line was different.
- The graph on the top right is the best of the non-deterministic fit lines because it minimizes the total error.
- This is how OLS works. OLS minimizes those distances, it minimizes the *residuals*.

Viewing residuals in messy data

```
# add predicted values to the data
dist.ssp$predicted <- predict(lm(dist_SSP ~ pctunins,
                                 data = dist.ssp,
                                 na.action = na.exclude))

# use geom_segment to draw lines between observed (purple) and
# predicted values for each county, these are residuals
dist.ssp %>%
  ggplot(aes(x = pctunins, y = dist_SSP)) +
  geom_segment(aes(xend = pctunins, yend = predicted, linetype = "Residu
               key_glyph = draw_key_vpath) +
  geom_point(aes(color = "County"), size = 2, alpha = .6) +
  geom_smooth(aes(size = "Linear model"), method = "lm", se = FALSE,
              color = "gray60", linetype = 2) +
  scale_linetype_manual(values = 1, name = "") +
  scale_color_manual(values = "#7463AC", name = "") +
  scale_size_manual(values = .5, name = "") +
  theme_minimal() +
  labs(y = "Miles to syringe program", x = "Percent uninsured",
       title = "Relationship between percentage without health insurance
```

Viewing residuals in messy data