SOC542 Statistical Methods in Sociology II

Ordinary Least Squares Regression I

Dr. Thomas Davidson

Rutgers University

January 31, 2022

Plan

- Course updates
- Bivariate statistics review
- Ordinary least squares regression
- Lab: Simple regression in R / Github

Course updates

Homework dates

- Syllabus updated with due dates for each homework assignment

Course updates

Homework 1

- Homework 1 will be released on Wednesday, due next Friday 2/11
  - Statistics review
  - Simple OLS regression
- Download and submit using Github Classroom

Expected mean and variance of two random variables

- The expected mean of the sum of two random variables is

$$E[x + y] = E[x] + E[y] = \mu_x + \mu_y$$

- The expected variance is the sum of the variances plus twice their covariance

$$var(x + y) = var(x) + var(y) + 2cov(x, y)$$

- If x and y are independent then $cov(x, y) = 0$ and $var(x + y) = var(x) + var(y)$

Covariance

- Covariance is the a measure of the joint variability of two random variables
- The expectation of the covariance between x and y is

$$cov(x, y) = E[xy] - E[x]E[y]$$

- For a population, the covariance is

$$cov(x, y) = \frac{1}{N}\Sigma(x_i - \mu_x)(y_i - \mu_y)$$

- Sample covariance is defined as

$$cov(x, y)_s = \frac{1}{n-1}\Sigma(x_i - \bar{x})(y_i - \bar{y})$$

Correlation

- Correlation is a scaled version of covariance. We divide the covariance by the product of the standard deviations.

$$\rho(x,y) = \frac{\frac{1}{n-1}\Sigma(x_i - \bar{x})(y_i - \bar{y})}{\sigma_x \sigma_y} = \frac{cov(x,y)}{\sigma_x \sigma_y}$$

- The letter ρ is typically used to refer to correlation. The correlation coefficient ranges from -1 to 1.
- The sample correlation is also a consisent estimator of the population correlation.

Generating correlated variables

We can use `mvrnorm` to generate a set of variables defined by their means and a variance-covariance matrix Σ. In this case, $\mu_x = 20$ and $\mu_y = 5$ and

$$\Sigma = \begin{Bmatrix} var(x) & cov(x,y) \\ cov(y,x) & var(y) \end{Bmatrix}$$

where the diagonal entries denote variance and the off-diagonals denote covariance.

```
n <- 1000
mu <- c(4,1) # vector of means, x and y
sigma <- rbind(c(4, 1), # variance of x, covariance of x and y
               c(1, 1)) # covariance of y and x, variance of y
M <- mvrnorm(n=n, mu=mu, Sigma = sigma)
```

Unlike `rnorm` where we specify a random variable using a mean and standard deviation, `mvrnorm` uses the mean and variance.

Sample statistics

The sample is large so the sample means and variances are close to the population values.

```
df <- as.data.frame(M)
colnames(df) <- c("x", "y")
print(mean(df$x)) # sample mean of x
```

```
## [1] 3.964342
```

```
print(var(df$x)) # sample variance of x
```

```
## [1] 3.781983
```

```
print(mean(df$y)) # sample mean of y
```

```
## [1] 0.9901981
```

```
print(var(df$y)) # sample variance of y
```

```
## [1] 0.9907679
```

Calculating covariance

We can calculate the sample covariance using the formula above. I verify the calculating by comparing it to the output of the built-in cov function.

```
covariance <- (1/(n-1))*sum((df$x-mean(df$x))*(df$y-mean(df$y)))
print(covariance)
```

```
## [1] 0.9228733
```

```
round(covariance,3) == round(cov(df$x,df$y),3)
```

```
## [1] TRUE
```

Calculating correlation

We can do the same for correlation. Note here that I use the `cov` function in the numerator.

```
correlation <- cov(df$x, df$y) / (sd(df$x)*sd(df$y))
print(correlation)
```

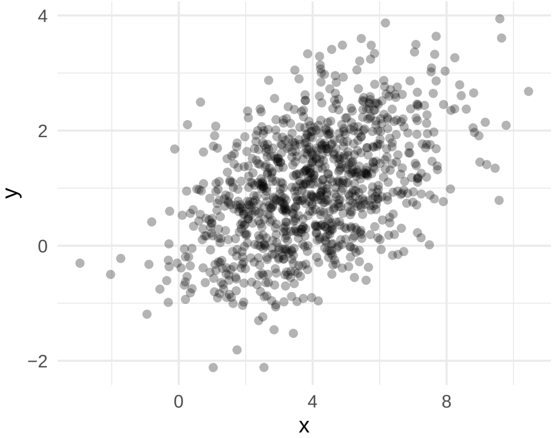
```
## [1] 0.4767561
```

```
round(correlation,3) == round(cor(df$x, df$y),3)
```

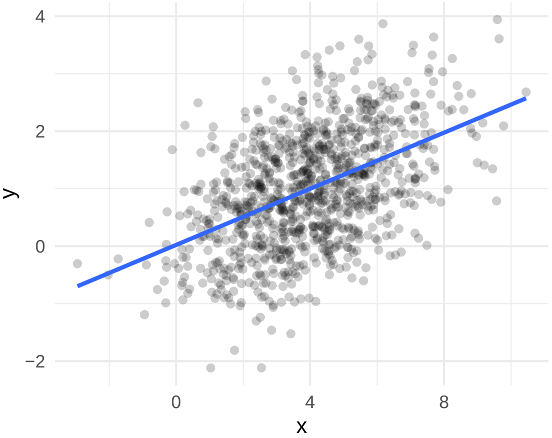
```
## [1] TRUE
```

Plotting the relationship

```
ggplot(data = df, aes(x = x, y = y)) + geom_point(alpha = 0.3) +
    theme_minimal()
```



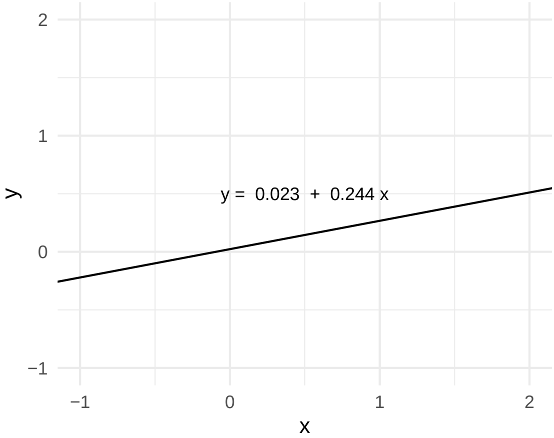
Adding regression line $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x + \hat{u}$.



Properties of the regression line

- The population regression line $y = \beta_0 + \beta_1 x + u$ is defined by two parameters, the slope and intercept.
  - β_0 and β_1 are known as **coefficients**.

Plotting the regression line



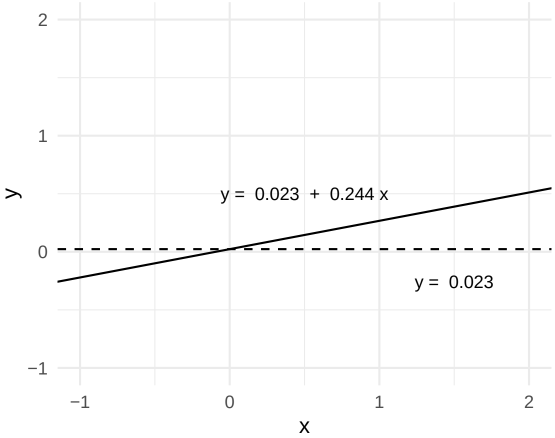
Interpreting the intercept

- The intercept defines the value of y when $x = 0$.
- Where $x = 0$, $\beta_0 x = \beta_1 0 = 0$, thus

$$y = \beta_0 + 0 = \beta_0$$

- Hence, the intercept is a *constant*.

Plotting the intercept



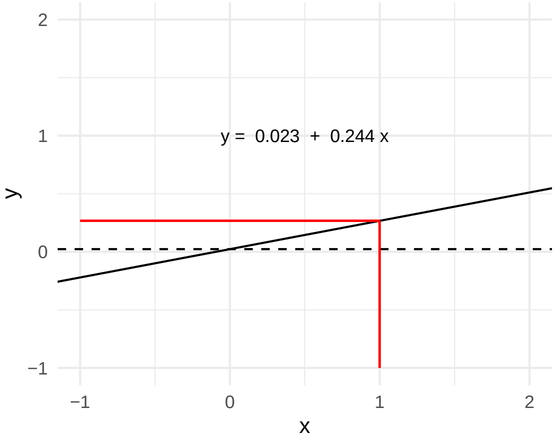
Interpreting the slope

- The slope defines the relationship between change in x and y, where Δ is used to denote change:

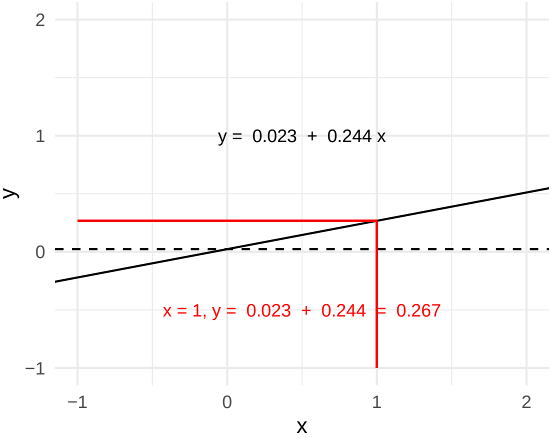
$$\beta_1 x = \frac{\Delta x}{\Delta y}$$

- β_1 denotes the expected *change* in y following a 1-unit change in x
  - e.g. What effect does an additional year of education have one lifetime income?
- If $\beta_1 < 0$ then the relationship is negative (y decreases as x increases)

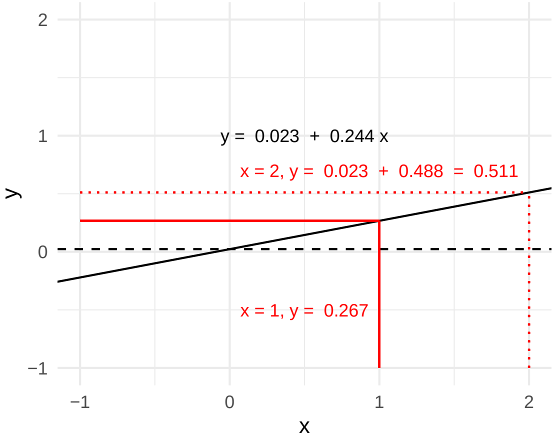
Interpreting the slope



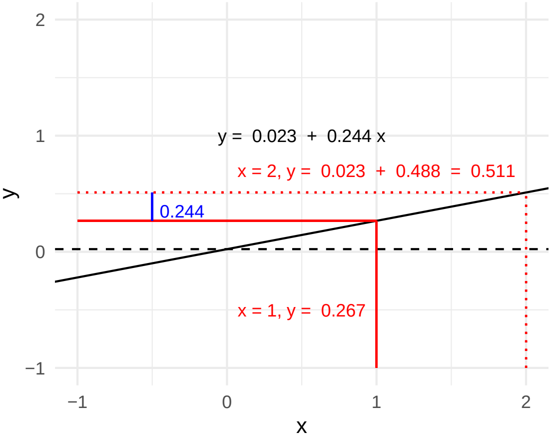
Interpreting the slope



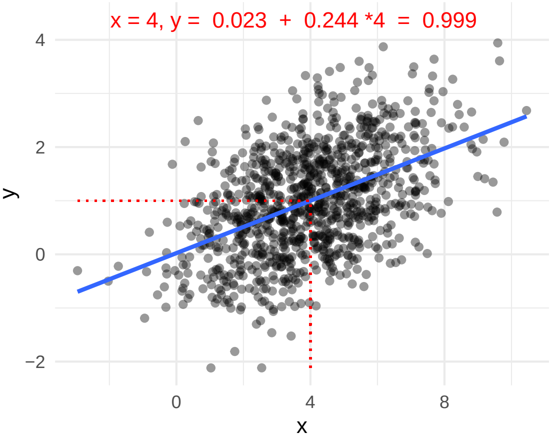
Slope as a comparison: a unit change in x



Slope as a comparison: a unit change in x



Reading the regression line



Ordinary least squares regression (Population model)

- The population ordinary least squares (OLS) regression equation is defined as:

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

- We can also write this as an expectation

$$E[y|x] = \beta_0 + \beta_1 x$$

- u is known as the error term and captures all factors that affect y but are not accounted for by x.

Ordinary least squares regression (sample model)

- The sample analogue is

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x + \hat{u}$$

- The ˆ symbol (pronounced "hat") is used to denote an **estimate**. We use the observed data from x and y to calculate estimates of underlying population quantities.

Defining the coefficients β_1 and β_0

- The OLS estimator of β_1 is

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} = \frac{cov(x, y)}{\sigma^2(x)}$$

- The estimator of the intercept β_0 can be derived from $\hat{\beta}_1$:

$$\beta_0 = \bar{y} - \beta_1\bar{x}$$

Predicted values and residuals

- x and y are vectors where x_i and y_i correspond to the i^{th} elements of each vector.
- We can use the regression equation to calculate the **predicted value** of y_i as a linear function of x_i:

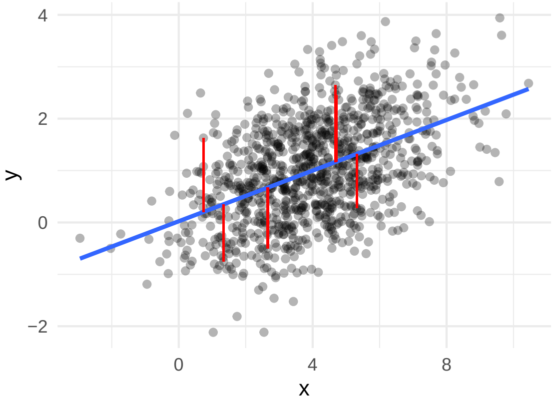
$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$$

- The **residual** is the difference between the observed value of y_i and the predicted value. It measures variation in y_i that is not explained by x.

$$\hat{u}_i = y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i = y_i - \hat{y}_i$$

- Thus, $y_i = \hat{y}_i + \hat{u}_i$.

Visualizing residuals



Red lines show difference between observed y and fitted value $\hat{y}$

Least squares

- This model is know as **least squares** regression because it minimizes the sum of the squared residuals.

$$SSR = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} \hat{u}_i^{\,2}$$

$\hat{x}$ is the least squares estimator of μ_x

- Consider a random variable x. For each value of x, $x_i - \alpha$ is the prediction error.

$$\sum_{i=1}^{n}(x_i - \alpha)^2$$

- The sample average $\bar{x}$ is the estimator α that minimizes the **sum of squared errors (SSE)**.

$\hat{x}$ is the least squares estimator of μ_x

Let's generate a random variable and calculate the SSE using $\alpha = \bar{x}$

```
x <- rnorm(n=100, mean = 5, sd = 1)
xbar <- mean(x)
print(xbar)
```

```
## [1] 5.009309
```

```
print(sum((x-xbar)^2))
```

```
## [1] 106.7624
```

$\hat{x}$ is the least squares estimator of μ_x

Now let's compare the results when alternative values of α are used.

```
## [1] "alpha = xbar =  5.009 , SSE =  106.762"
```

```
## [1] "alpha =  3 , SSE =  510.495"
## [1] "alpha =  4 , SSE =  208.633"
## [1] "alpha =  5 , SSE =  106.771"
## [1] "alpha =  6 , SSE =  204.909"
## [1] "alpha =  7 , SSE =  503.048"
```

β_0 and β_1 minimize the SSR

- For a single sample, $\bar{y}$ is the least squares **estimator** of μ_y.
- For two variables, $\hat{y}$ is the least squares **estimator** of y because it minimizes the **sum of the squared residuals (SSR)**:

$$SSR = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} \hat{u}^2$$

- By substitution,

$$SSR = \sum_{i=1}^{n}(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2$$

Minimizing the sum of the squared residuals

Let's simulate the residuals using some other possible coefficients.

```
coefs <- c(S-0.3, S-0.2, S-0.1, S, S+0.1, S+0.2, S+0.3)

i <- 1
results <- c()
for (s in coefs) {
    u <- df$y - I - s*df$x
    ssr <- round(sum(u^2), 2)
    results[i] <- ssr
    i <- i + 1
}
```

Minimizing the sum of the squared residuals

```
##    coefs results
## 1 -0.056 2519.08
## 2  0.044 1544.44
## 3  0.144  959.68
## 4  0.244  764.80
## 5  0.344  959.81
## 6  0.444 1544.71
## 7  0.544 2519.48
```

Model fit and R^2

- R^2 is a measure of the ratio of the variance of $\hat{y}$ to the variance of y_i

$$R^2 = \frac{\sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} = \frac{ESS}{TSS}$$

- We can also write it as a fraction of the unexplained variance:

$$R^2 = 1 - \frac{SSR}{TSS}$$

- R^2 has a range of [0,1] where higher values indicate more variance explained. It is often common to have models with very low values of R^2.

Mean squared error

- An alternative measure of fit is the **mean squared error (MSE)**, defined as

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

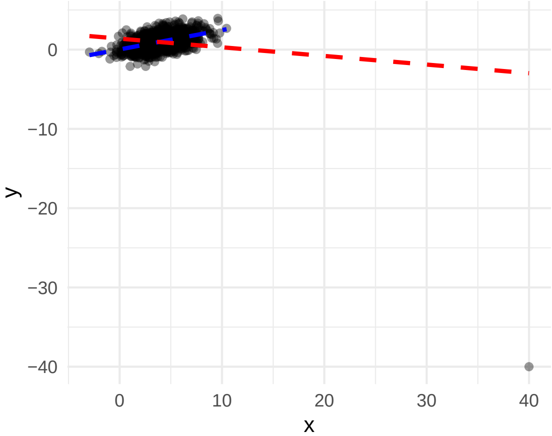
- MSE is often used to evaluate the predictive performance of statistical models with continuous outcomes.

OLS assumptions

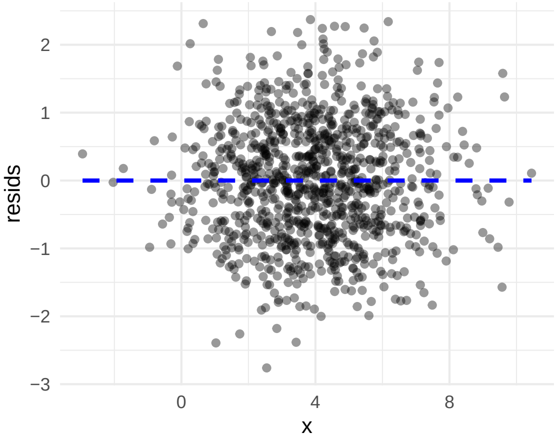
- x and y are independently and identically distributed (IID).
  - The sample x must contain some variability. Specifically, $var(x) > 0$.
  - Large outliers are unlikely.
- The conditional distribution of u given x has a mean of zero.
  - Errors are independent $E[u_i|x_i] = E[u_i] = 0$.
  - Errors have constant variance $var(u_i) = \sigma^2$.
  - Errors are uncorrelated.

Violating the large outlier assumption

Observe how a large outlier can pull down the entire regression line.



$E[u_i|x_i] = 0$



Homoskedasticity and heteroskedasticity

- The $E[u|x] = E[u] = 0$ implies **homoskedasticity**
  - The variance of u_i is equal for all values of x_i, $var(u_i) = \sigma^2$.
- **Heteroskedasticity** exists when this assumption is violated.
  - It can result in inefficient point estimates and biased standard errors.

The Gauss-Markov Theorem

- If these assumptions hold and the errors are homoskedastic, the OLS estimator $\hat{\beta}_1$ is **BLUE**: the **Best Linear conditionally Unbiased Estimator**.
- **Best** implies that $\hat{\beta}_1$ is the best of all possible linear conditionally unbiased estimators.
  - $\hat{\beta}_1$ produces the smallest mean squared error of all possible estimators $\tilde{\beta}_1$.
- **Linear** requires the dependent variable y to be a linear function of the parameters in the model.
  - This does *not* require the relationship between x and y to be linear. e.g. $y = 1 + 2x^2$ is linear in parameters.
- **conditionally Unbiased** implies $E[\hat{\beta}_1] = \beta_1$.
  - The expectation of the estimated coefficient $\hat{\beta}_1$ is equal to the population parameter β_1 after conditioning on x.

Summary

- OLS regression is used when we assume y can be modeled as a linear combination of parameters.
- We assume a population model, $y = \beta_0 + \beta_1 x + u$.
- We use a sample of data to estimate the relationship between y and x in the population.
- The equation $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i + \hat{u}_i$ minimizes the sum of the squared residuals.
- If the sample is IID and the errors are unrelated to x, we can assume that $\hat{\beta}_1$ is the best estimator of β_1.

Estimating β_0 and β_1 using `lm()`

```r
model <- lm(y ~ x, data = df)
```

Estimating β_0 and β_1 using `lm()`

```
summary(model)
```

```
##
## Call:
## lm(formula = y ~ x, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.76000 -0.65289 -0.02834  0.62889  2.37092
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.02283    0.06288   0.363    0.717
## x            0.24402    0.01424  17.134   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8754 on 998 degrees of freedom
## Multiple R-squared:  0.2273, Adjusted R-squared:  0.2265
```

Estimating β_0 and β_1 using stan_glm()

We can also run the same model using Bayesian estimation.

```
model2 <- stan_glm(y ~ x, data = df)
```

```
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 7.4e-05 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition wou
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
```

Comparing `lm` and `stan_glm`

Let's compare the coefficients across the two models. We can see that they are very close. We will discuss the differences in these approaches more next week.

```
print(model$coefficients) # lm
```

```
## (Intercept)           x
##  0.02282593  0.24401834
```

```
print(model2$coefficients) # stan_glm
```

```
## (Intercept)           x
##  0.02240602  0.24432927
```

Comparing `lm` and `stan_glm`

We can also compare the standard deviations of the residuals, σ. The results are almost identical.

```
sigma(model)
```

```
## [1] 0.8754068
```

```
sigma(model2)
```

```
## [1] 0.8756781
```

Next week

- Introduction to Bayesian statistics