Lecture 3 - Introduction to Linear Modeling

Sam Mason

2/4/2020

Variance & Covariance

Variance, σ^2, should be a familiar concept to us at this point in our statistical careers. Formally, variance is given by the equation:

$$\sigma^2 = \frac{\sum (x_i - \mu)^2}{N}$$

where large deviations from the population mean, μ, yield a consequently large variance. We can calculate the variance in R using the code:

```
sample <- rnorm(25)
sample.variance <- var(sample)
print(sample.variance)
```

```
## [1] 0.7235162
```

We can rewrite the formal definition of population variance this way

$$\sigma^2 = \frac{\sum (x_i - \mu)(x_i - \mu)}{N}$$

to highlight the product of two deviations from the mean. These deviations, are, of course, equivalent in direction (positive or negative) and magnitude, yiedling only positive values in the numerator. Here we are using only a single vector of data, but we can rewrite this formula once again to accomodate a second vector of data of length N.

$$Cov(\overrightarrow{x}, \overrightarrow{y}) = \frac{\sum (x_i - \mu_x)(y_i - \mu_y)}{N}$$

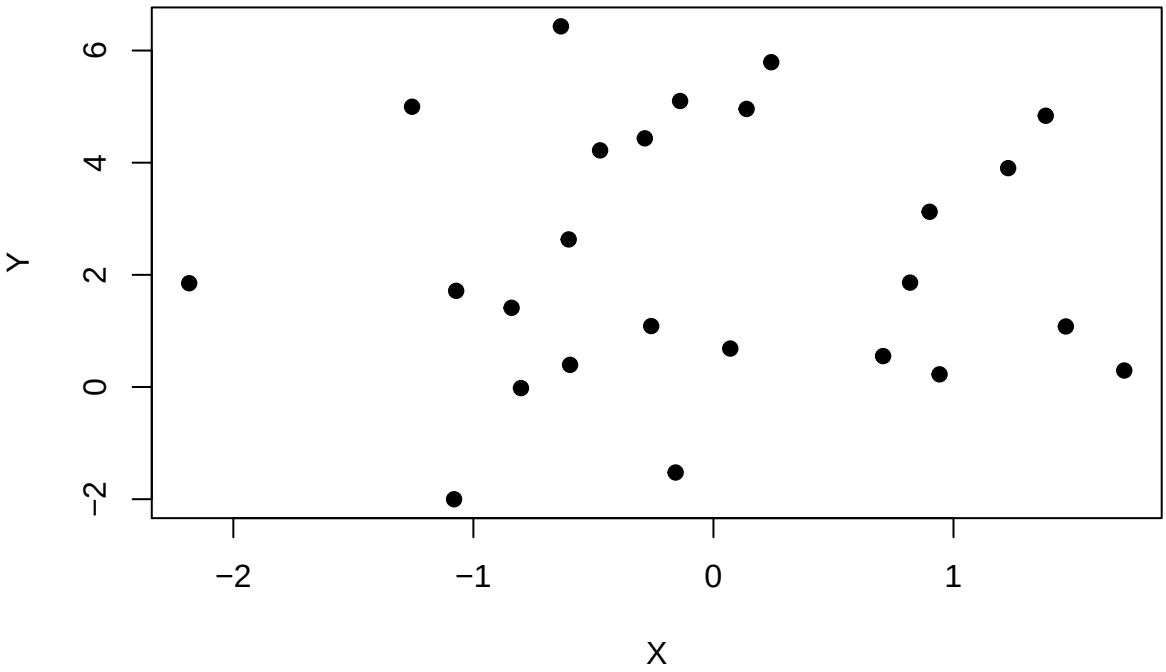
Here, we are not calculating the *variance* in a single set of data, but the **covariance** between two different sets of data. This equation, if it could talk, would ask the question, *in general, how does y deviate from its mean when x deviates from its mean?* For example, if negative deviations in x were *generally* accompanied by positive deviations in y, the covariance would be negative. If large deviations in x *generally* accompaied large deviation in y, the covariance would be large.

We can calculate the covariance between two variables of the same size in R using the code:

```
set.seed(5)
sample.x <- rnorm(25)
sample.y <- rnorm(25, 2, 2)
covariance <- cov(sample.x, sample.y)
print(covariance)
```

```
## [1] 0.0566686
```

Excellent! Roughly 0.057, umm, covariance units? Wait, what does this value tell us? Okay, so the direction makes sense. The covariance of these two values is positive, meaning that, *generally*, positive deviations in one correspond to positive deviations in the other and negative deviations in one correspond to negative deviations in the other. Take a second to visualize this trend as a scatterplot. Mine looks like this:



Let's pluck this covariance value from the mathematical ethos by dividing by the variance of $\overrightarrow{x}$. Examine the covariance formula again. You can imagine many vectors $\overrightarrow{x}$ and $\overrightarrow{y}$ that are characterized by the same covariance value. Here's one for example:

```
set.seed(5)
sample.x.prime <- rnorm(25, 10, 0.25)
sample.y.prime <- rnorm(25, 0.5, 8)
cov(sample.x.prime, sample.y.prime)
```

```
## [1] 0.0566686
```

However, by dividing by the variance of $\overrightarrow{x}$, we can differentiate between the datasets.

```
set.seed(5)
sample.x <- rnorm(25)
sample.y <- rnorm(25, 2, 2)
cov(sample.x, sample.y)/var(sample.x)
```

```
## [1] 0.05916972
```

```
sample.x.prime <- rnorm(25, 10, 0.25)
sample.y.prime <- rnorm(25, 0.5, 8)
cov(sample.x.prime, sample.y.prime)/var(sample.x.prime)
```

```
## [1] 5.045692
```

Alright, I think it's time for the big reveal! Let's quantify the effect that variable x has on variable y using a linear model with a normal error structure. In R, the code looks like this:

```
set.seed(5)
sample.x <- rnorm(25)
sample.y <- rnorm(25, 2, 2)
mod <- lm(sample.y ~ sample.x) # a linear model where y is a function of x
```

```r
summary(mod)
```

```
##
## Call:
## lm(formula = sample.y ~ sample.x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.2606 -1.8141 -0.5105  2.1275  4.1449
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.32358    0.47571   4.884 6.21e-05 ***
## sample.x     0.05917    0.49586   0.119    0.906
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.377 on 23 degrees of freedom
## Multiple R-squared:  0.0006187,  Adjusted R-squared:  -0.04283
## F-statistic: 0.01424 on 1 and 23 DF,  p-value: 0.9061
```

Go to the `Coefficients:` table in the model output and find the estimated slope below the estimated intercept. Does that value look familiar? HOLY SMOKES! It's the same number we get when we divide $Cov(\overrightarrow{x}, \overrightarrow{y})$ by σ_x^2.

In R, calculate the intercept of the line given by this linear model. **HINT:** The coordinate given by the mean of x and the mean of y falls on the line.

CHALLENGE: Calculate the residual standard error. **HINT:** You will need to look up the formula online.