Summarizing Numerical Associations

Overplotting, correlation and the least squares line

When we first discussed constructing summaries for numerical data, you may have noticed that we left out the case when we are working with two numerical variables. This is a *very* common scenario in statistics and data science– so much so that it deserves its own set of notes! In this lecture, we will discuss how we can make visualizations and calculate summary statistics involving two numerical variables. Then, we will introduce a third method of describing data: building a **model**.

Overplotting

First, we should spotlight an issue that can arise when visualizing numerical associations. This issue may have the potential to hide an association between them if it is not treated.

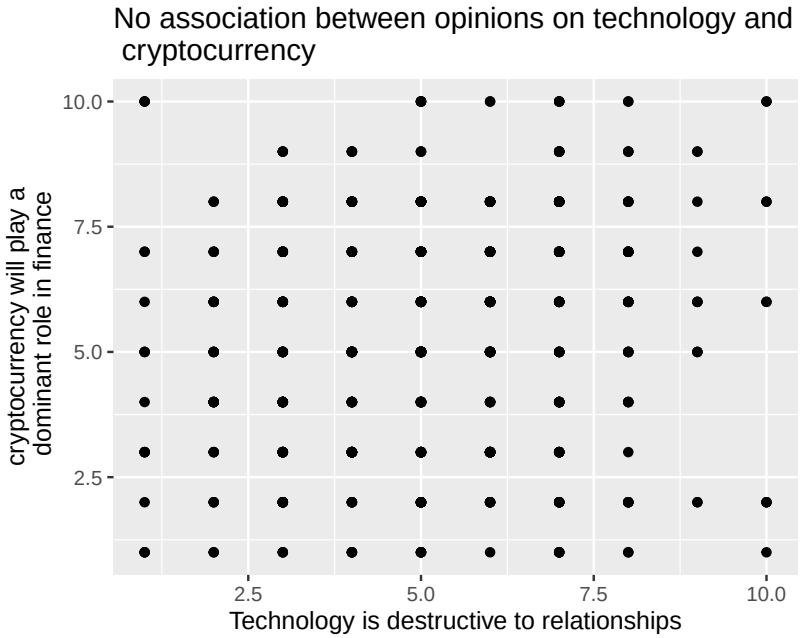
Let's examine the class survey dataset from earlier in this course. Stat 20 students filled out a survey that asked them their opinion on several topics including:

What is your opinion of the following statement: "Cryptocurrency will play a dominant role in the global financial system".

1 2 3 4 5 6 7 8 9 10

Strongly disagree ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ Strongly agree

The result was a data frame with 619 rows (one for every respondent) and 2 columns of discrete numerical data. A natural way to visualize this data is by creating a **scatter plot.**



The eye is immediately drawn to the eerie geometric regularity of this data. Isn't real data messier than this? What's going on?

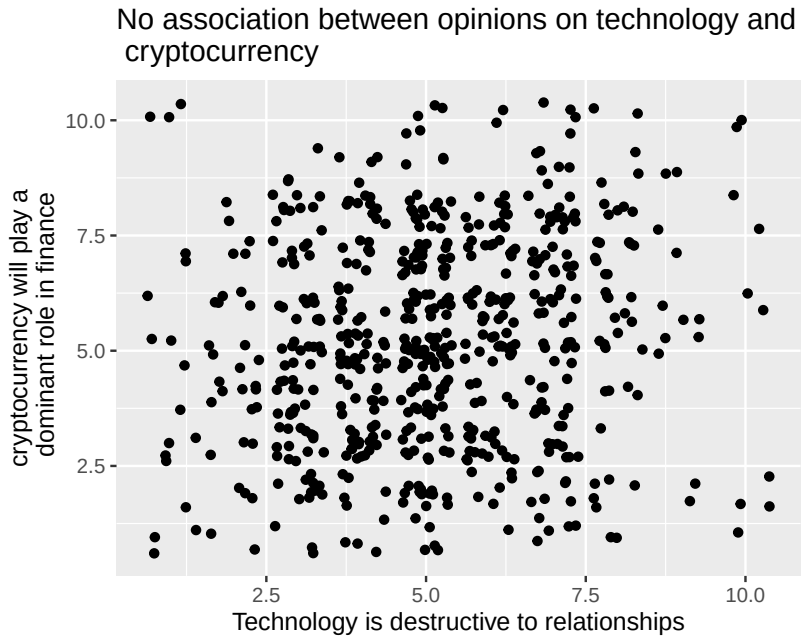
A hint is in the sample size. The number of observations in the data set was over 600 and yet the number of points shown here is just a bit under 100. Where did those other observations go?

It turns out they are in this plot, *they're just piled on top of one another!* Since there are only 10 possible values for each question, many students ended up selecting the same values for both, leading their points to be drawn on top of one another.

This phenomenon is called **overplotting** and it is very common in large data sets. There are several strategies for dealing with it, but here we cover two of them.

One approach to fixing the problem of points piled on top of one another is to unpile them by adding just a little bit of random noise to their x- and y-coordinate. This technique is called **jittering** and can be done in `ggplot2` by replacing `geom_point()` with **`geom_jitter()`**.

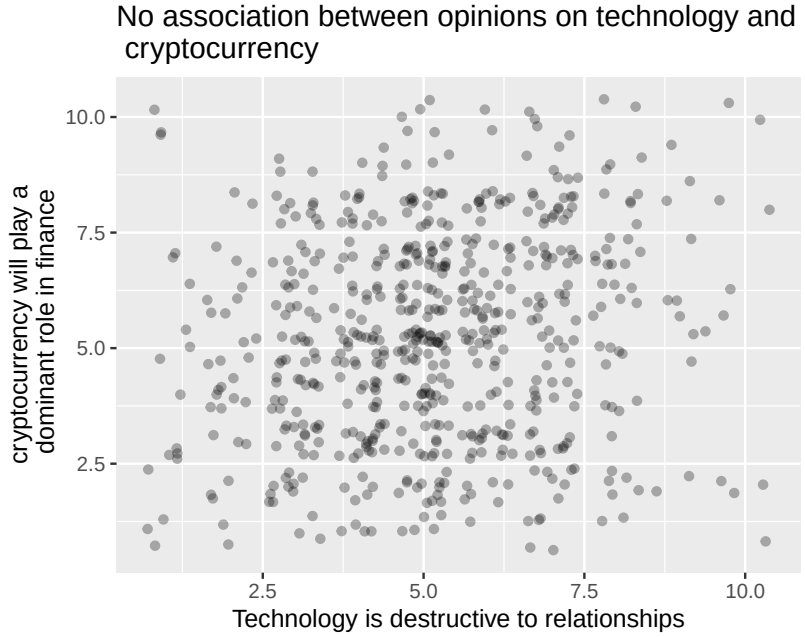
```
ggplot(class_survey, aes(x = tech,
                         y = crypto)) +
    geom_jitter() +
    labs(x = "Technology is destructive to relationships",
         y = "cryptocurrency will play a\n dominant role in finance",
         title = "No association between opinions on technology and \n cryptocurrency")
```



Ahh . . . there are those previously hidden students. Interestingly, the title on the first plot still holds true: even when we're looking at *all* of the students, there doesn't appear to be much of a pattern. That is certainly not the case in all overplotted data sets! Often overplotting will obscure a pattern that jumps out after the overplotting has been attended to.

The second technique is to make the points transparent by changing an aesthetic attribute (setting) called the **alpha value.** Let's combine transparency with jittering to understand the effect.

```
ggplot(class_survey, aes(x = tech,
                         y = crypto)) +
    geom_jitter(alpha = .3) +
    labs(x = "Technology is destructive to relationships",
         y = "cryptocurrency will play a\n dominant role in finance",
         title = "No association between opinions on technology and \n cryptocurrency")
```



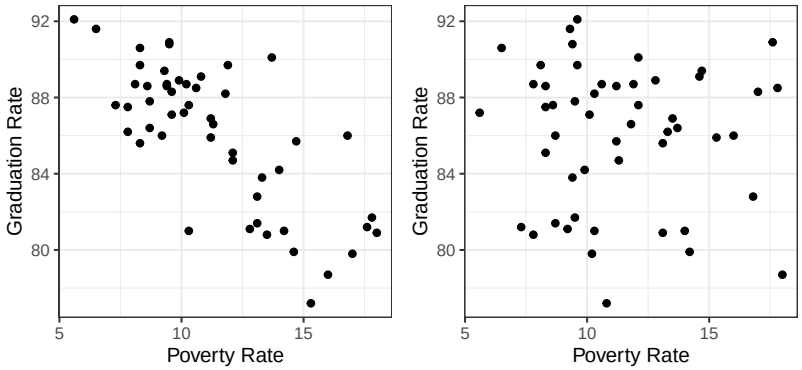
The `alpha` argument runs between 0 and 1, where 1 is fully opaque and 0 is fully see-through. Here, `alpha = .3`, which

changes all observations from black to gray. Where the points overlap, their alpha values add to create a dark blob.

There's still no sign of a strong association between these variables, but at least, by taking overplotting into consideration, we've made that determination after incorporating all of the data.

Associations and the correlation coefficient

Which of the following plots do you think depicts the relationship between the high school graduation rate and the poverty rate among the 50 US states?



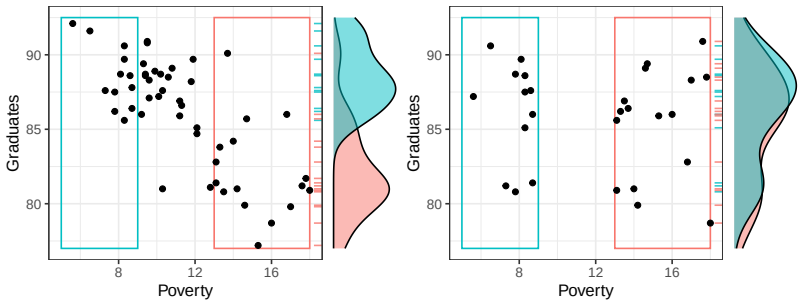
If you guessed the plot on the left, you are correct .

States with higher poverty rates tend to have lower graduation rates. This is a prime example of two variables that are *associated*. In a previous set of notes we defined association between two categorical variables, but lets replace that with a more general definition that can apply here.

Association There is an association between two variables if the conditional distribution of one varies as you move across values of the other.

You can detect associations in scatter plots by scanning from left to right along the x-axis and determining whether or not the conditional distribution of the y-variable is changing or not.

In the figure to the left below, when you look first to the states with low poverty rates (in the blue box), you find that the conditional distribution of the graduation rate (represented by the blue density curve along the right side of the scatter plot) is high: most of those states have graduation rates between 85% and 90%. When you scan to the right in that scatter plot, and condition on having a high poverty rate (the states in the red box), the conditional distribution shifts downwards. Those states have graduations rates in the low 80%s.



These density curves are conditional distributions because we've set a condition on the data we're visualizing. When focusing on the data that's in the blue box, for example, we've in effect set up a filter where Poverty < 9.

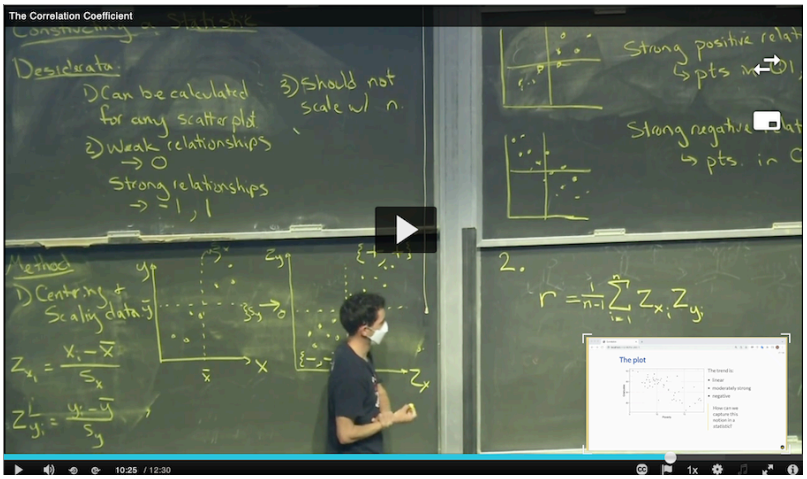
The plot on the right, by contrast, exhibits no association between poverty rate and graduation rate. When we compare the low poverty states with the high poverty states, their conditional distributions of Graduation rate are essentially the same.

So we can use the simple scatter plot to determine whether or not two numerical variables are associated, but sometimes a graphic isn't enough. In these notes we'll move from graphical summaries to numerical summaries and construct two different approaches to capturing these associations in numbers: the correlation coefficient and the simple linear model.

The Correlation Coefficient

Let's set out to engineer our first numerical summary in the same manner that we have previously, by laying out the properties that we'd like our summary to have.

Please watch the following 12 minute video.



Correlation coefficient, r The correlation coefficient, r, between two variables x and y is

$$r = \frac{1}{n-1}\sum_{i=1}^{n}\left(\frac{x_i - \bar{x}}{s_x}\right)\left(\frac{y_i - \bar{y}}{s_y}\right)$$

Several different statistics have been proposed for measuring association. This is the most common and is more specifically called the Pearson correlation.

Example: Poverty and Graduation rate

The data frame used to create the scatter plot above on the left looks like this.

```
# A tibble: 51 x 2
   Graduates Poverty
       <dbl>   <dbl>
 1      79.9    14.6
 2      90.6     8.3
 3      83.8    13.3
 4      80.9    18
 5      81.1    12.8
 6      88.7     9.4
 7      87.5     7.8
 8      88.7     8.1
 9      86      16.8
```

```
10      84.7    12.1
# i 41 more rows
```

Since it is a data frame, we can use the `summarize()` function to calculate our summary statistic.

```
poverty |>
  summarize(r = cor(Poverty, Graduates))
```

```
# A tibble: 1 x 1
      r
   <dbl>
1 -0.747
```

The value of -0.747 tells us that the linear association between these variables is negative and reasonably strong. This is our first example of a *bivariate* summary statistic: there are two variables that we put inside the `cor()` function to compute our statistic.

Let's repeat this calculation for the data frame that created the shapeless scatter plot with no association, `poverty_shuffled`.

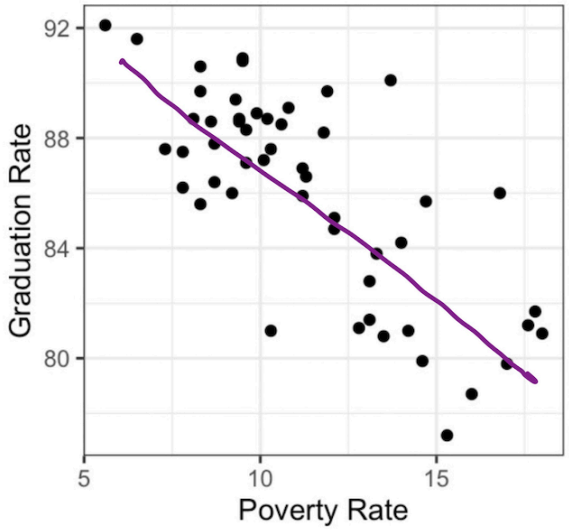
```
poverty_shuffled |>
  summarize(r = cor(Poverty, Graduates))
```

```
# A tibble: 1 x 1
        r
    <dbl>
1 -0.0546
```

As expected, that scatter plot yields a correlation coefficient very close to zero because the points are scattered across all four quadrants of the plot.

The Simple Linear Model

Another approach to summarizing the linear association is to just … draw a line.



This line serves both as a graphical summary and *also* as a numerical summary. After all, every line that you draw on a scatter plot is defined by two numbers: the slope and the y-intercept. This line is called a *simple linear model*.

Simple Linear Model An expression for a possible value of the y variable, $\hat{y}$, as a linear function of the x variable with slope b_1 and y-intercept b_0.

$$\hat{y} = b_0 + b_1 x$$

Therefore, a simple linear model captures the linear relationship of two variables in not one but *two* summary statistics, b_0 and b_1.

For the line above, we can do our best to eye-ball these. The line appears to rise -2 percentage points for every 2.5 that it runs, so I'd estimate the slope to be about $-2/2.5 = -0.8$. If I were to draw the line all the way to the left until it crossed the

y-axis at a poverty rate of 0, its y-intercept would be around 95. So I could express the line that is drawn above as:

$$\hat{y} = 95 - 0.8x$$

The Least Squares Line

If that felt a little shifty to you - drawing a line by hand and then eyeballing its slope and intercept - we can be more precise by using a more precisely-defined type of linear model: the least squares line. This is a method that we'll study in depth when we get to the unit on prediction, but for now, we'll use it because it makes calculation very easy. You can find the slope and intercept of the least squares line using statistics that we're already familiar with: $\bar{x}, \bar{y}, s_x, x_y$, and r.

Least Squares Slope

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

Least Squares Intercept

$$b_0 = \bar{y} - b_1\bar{x}$$

So how does this line look compared to the hand-drawn line? Let's calculate the slope and intercept. In R, we can do this with a function called **`lm()`.** To see the slope and intercept for our model, we can print out our model object.

```
m1 <- lm(formula = Graduates ~ Poverty, data = poverty)
m1
```

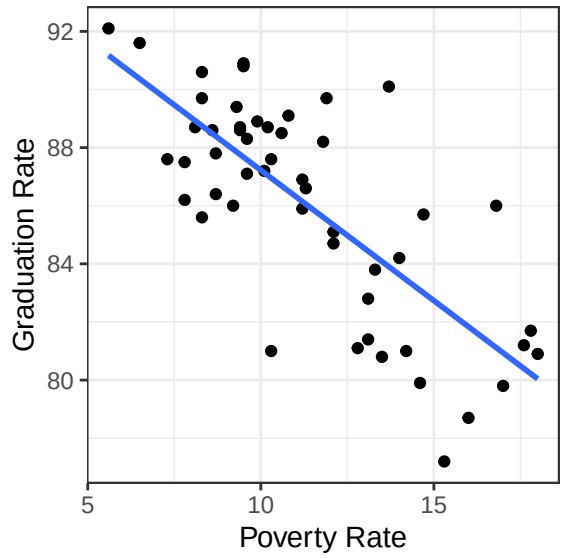
```
Call:
lm(formula = Graduates ~ Poverty, data = poverty)

Coefficients:
(Intercept)      Poverty
    96.2022      -0.8979
```

The syntax for `lm()` uses what's called "formula notation" in R. The first argument is a formula of the form `y ~ x` and can be read as, "Explain the y as a function of the x". In the second argument, you specify which data frame contains the variables used in the formula. If we want to use to save this slope and intercept for later use, we can save it into an object, just like a data frame or a vector can be saved.

We can then add our model to our scatter plot. This can be done with the **`geom_smooth()`** layer in `ggplot2` and the `method = "lm"` argument (you do not need to worry about the purpose of the `se` argument).

```
ggplot(poverty, aes(x = Poverty,
                    y = Graduates)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(x = "Poverty Rate",
       y = "Graduation Rate") +
  theme_bw()
```



That works remarkably well!

Interpreting the slope and intercept

So if the correlation coefficient measures the strength of the linear relationship between two variables, what exactly are the slope and intercept of a linear model between involving these two variables measuring?

The slope captures the expected change in the y variable associated with the x variable changing by 1 unit. In this example, states that are separated by 1 percentage point in their poverty rate tend to be separated by about -.89 in their graduation rate. This is distinct from what the correlation tells us because while r will stay the same regardless of the units in which the data is measured, b_1 is expressly designed to tell us how those units of measurement relate to one another.

What about the intercept? **It tells us the value that we'd expect the y to take when the x takes a value of zero.** *Sometimes that's an informative statistic, sometime it is not.* In this setting, do you really expect the graduation rate to be around 96% when their poverty rate is zero? What would it even look like for a state to have a poverty rate of zero? The abstraction of the linear model allows us to ponder such a world, but the reality of economics in the US is that we would never actually observe poverty rates of zero.

So what good is the intercept? Well, it's useful in helping us calculate a **residual.**

Residuals

One of the benefits of explaining the association between two variables with a line instead of just the correlation coefficient is that it allows us to calculate what we would *expect* an observation's y-value to be based on its x value, so that we can see how far our expectation is from reality. That gap between expectation and reality is called a *residual*.

Residual ($\hat{e}_i$) The difference between the observed value of a data point, y_i, and the value that we would expect ac-

cording to a linear model, $\hat{y}_i$.

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

Let's calculate the residual for California. Here is that row in the data set.

$\hat{y}_i$ is said "y hat sub i" and is also called the "fitted value".

```
poverty |>
  filter(State == "California") |>
  select(State, Graduates, Poverty)
```

```
# A tibble: 1 x 3
  State      Graduates Poverty
  <chr>           <dbl>   <dbl>
1 California       81.1    12.8
```

This shows us that for California, $y = 81.1$, so the next step is to find where the line passes through California's x-value, $x = 12.8$. There are several ways to do that calculation, including using R like a calculator and simply plugging that value into the equation for the line show above.

```
y_hat <- 96.2022 - 0.8979 * 12.8
y_hat
```

```
[1] 84.70908
```

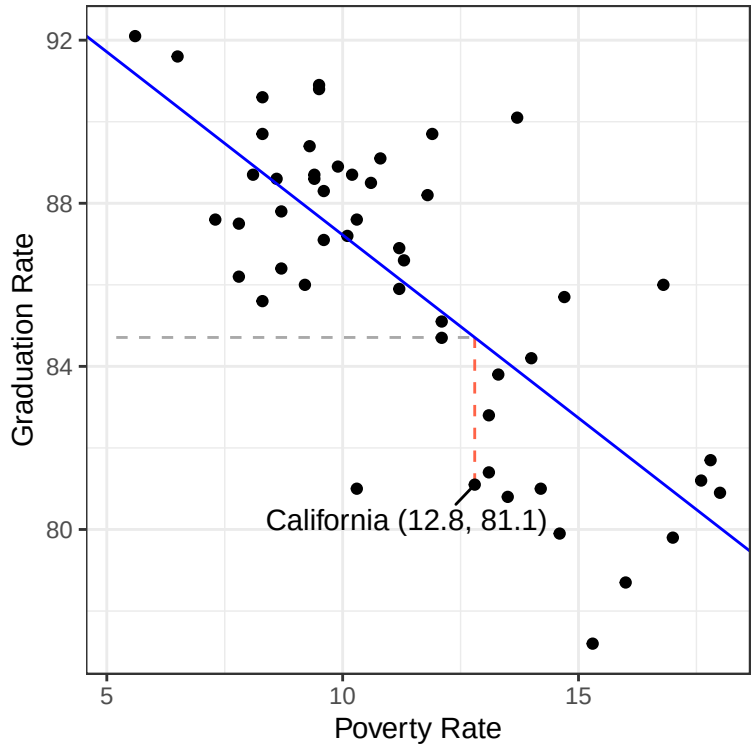
With that in hand, we can calculate California's residual.

```
81.1 - y_hat
```

```
[1] -3.60908
```

This residual tells us that California is actually a bit of an underachiever. Among states with a poverty rate around 12.8, we would expect their graduate rate to be around 84.7. California's rate, however, is 81.1, a decrease of 3.6.

The calculation of the residual can be seen in the plot below.



The horizontal dashed line represents $\hat{y} = 84.7$, the y-value of the least squares line when it passes through $x = 12.8$. The vertical red dashed line is the residual: the distance between the line and the observation in the y direction.

Residuals open up a new avenue for numerical statistics. While the slope and intercept are two statistics that tell us about the *overall* linear relationship between the two variables, each residual is a statistic that tells us whether an *individual* observation's y-value is higher or lower than we'd expect based on its x-value.

While using R as a calculator directly to obtain *one* residual is somewhat efficient, this changes when you would like to calculate a residual for each point in your data set. Imagine having to write n lines of code, one for each observation: y_1, y_2, and all the way to y_n! Luckily, when you save a linear model into an object, you store lots of useful information, including $\hat{y}_i$ and $\hat{e}_i$ for every observation y_i. These can be accessed via the `fitted()` and `residuals()` functions, respectively.

If you have n data points, you can calculate n residuals. This is described below.

`fitted()` and `residuals()` return vectors. To match the vectors up with the observations, we can *mutate* them as columns onto the original data frame. From here, we can isolate the residual for the state of California as before.

```
poverty |>
  mutate(y_hat = fitted(m1),
         e_hat = residuals(m1)) |>
  select(State, Graduates, Poverty, y_hat, e_hat) |>
  filter(State == "California")
```

```
# A tibble: 1 x 5
  State      Graduates Poverty y_hat e_hat
  <chr>          <dbl>   <dbl> <dbl> <dbl>
1 California      81.1    12.8  84.7 -3.61
```

Summary

In these notes we considered the question of how to capture the association between two variables with both visualizations and numerical summary statistics. The **correlation coefficient** is one of the most common statistics to use in this case: it captures the strength and direction of the linear trend. This statistic can be used, along with other simple summary statistics, to calculate the slope and intercept of the **least squares line**. The least squares line is an alternative approach to summarizing the linear relationship between two numerical variables. It has the advantage of providing an expectation for the y-value of every observation, which allows us to calculate residuals which are expressions of whether each observation is higher or lower than we'd expect.

We'll spend time practicing calculating these statistics - and looking at lots of scatter plots - in class.