Linear Regression I

Dav King

3/17/22

Coming up

- Lab due on Friday at 11:59 PM.

Main ideas

- Learn about what we use models for.
- Begin to learn the language of linear models in R.

Modeling

- We use models to...
  - understand relationships
  - assess differences
  - make predictions
- We will focus on **linear** models but there are many other types.

Packages

In addition to using the `tidyverse` and `viridis` packages, today we will be using data from the `fivethirtyeight` package with data on candy rankings from this 2017 article. The variables in this dataset are discussed in more detail here. We'll also be using the `tidymodels` package to run regressions.

Distribution of Variables

Let's start by looking at the variables in this dataset.

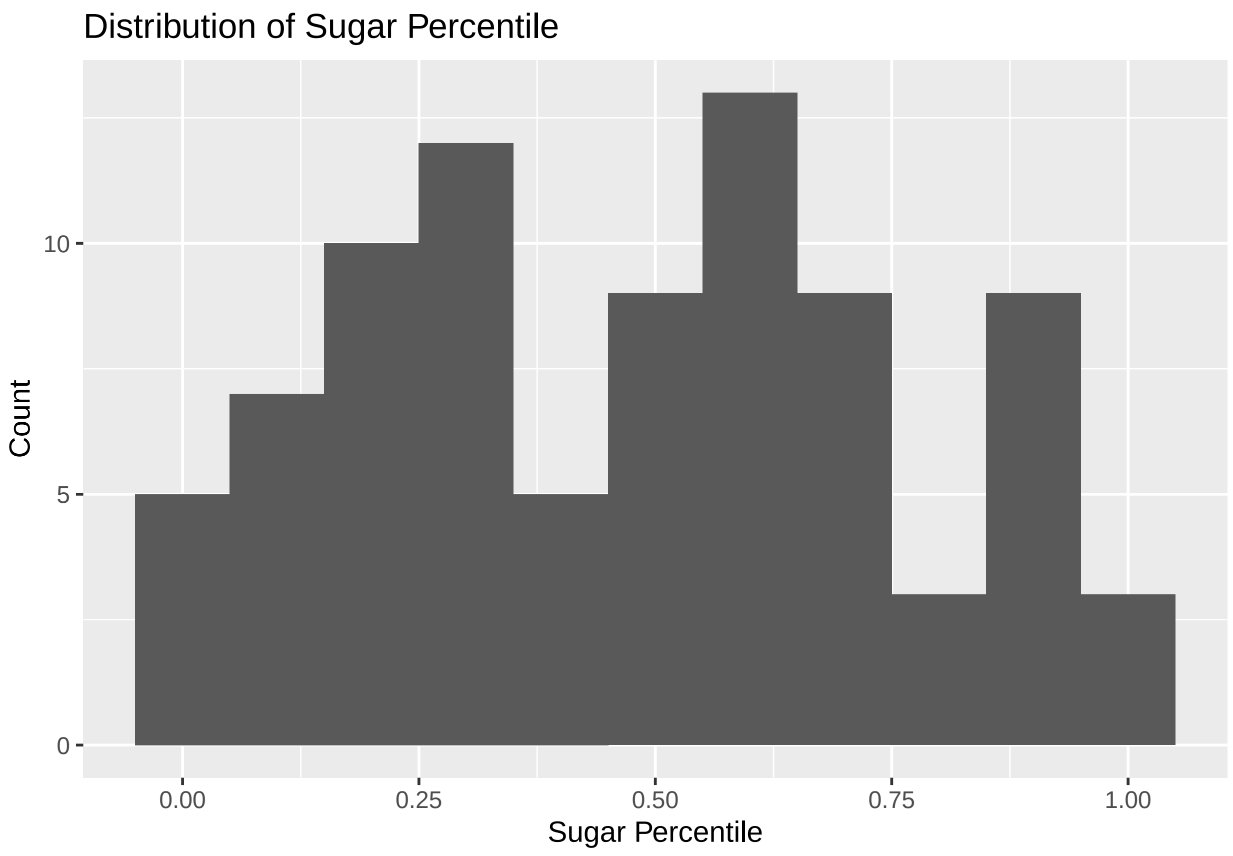
```
glimpse(candy_rankings)
```

```
## Rows: 85
## Columns: 13
## $ competitorname   <chr> "100 Grand", "3 Musketeers", "One dime", "One quarter~
## $ chocolate        <lgl> TRUE, TRUE, FALSE, FALSE, FALSE, TRUE, TRUE, FALSE, F~
## $ fruity           <lgl> FALSE, FALSE, FALSE, FALSE, TRUE, FALSE, FALSE, FALSE~
## $ caramel          <lgl> TRUE, FALSE, FALSE, FALSE, FALSE, FALSE, TRUE, FALSE,~
## $ peanutyalmondy   <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, TRUE, TRUE, TRUE, ~
## $ nougat           <lgl> FALSE, TRUE, FALSE, FALSE, FALSE, FALSE, TRUE, FALSE,~
## $ crispedricewafer <lgl> TRUE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE~
## $ hard             <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALS~
## $ bar              <lgl> TRUE, TRUE, FALSE, FALSE, FALSE, TRUE, TRUE, FALSE, F~
## $ pluribus         <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, TRUE~
## $ sugarpercent     <dbl> 0.732, 0.604, 0.011, 0.011, 0.906, 0.465, 0.604, 0.31~
## $ pricepercent     <dbl> 0.860, 0.511, 0.116, 0.511, 0.511, 0.767, 0.767, 0.51~
## $ winpercent       <dbl> 66.97173, 67.60294, 32.26109, 46.11650, 52.34146, 50.~
```

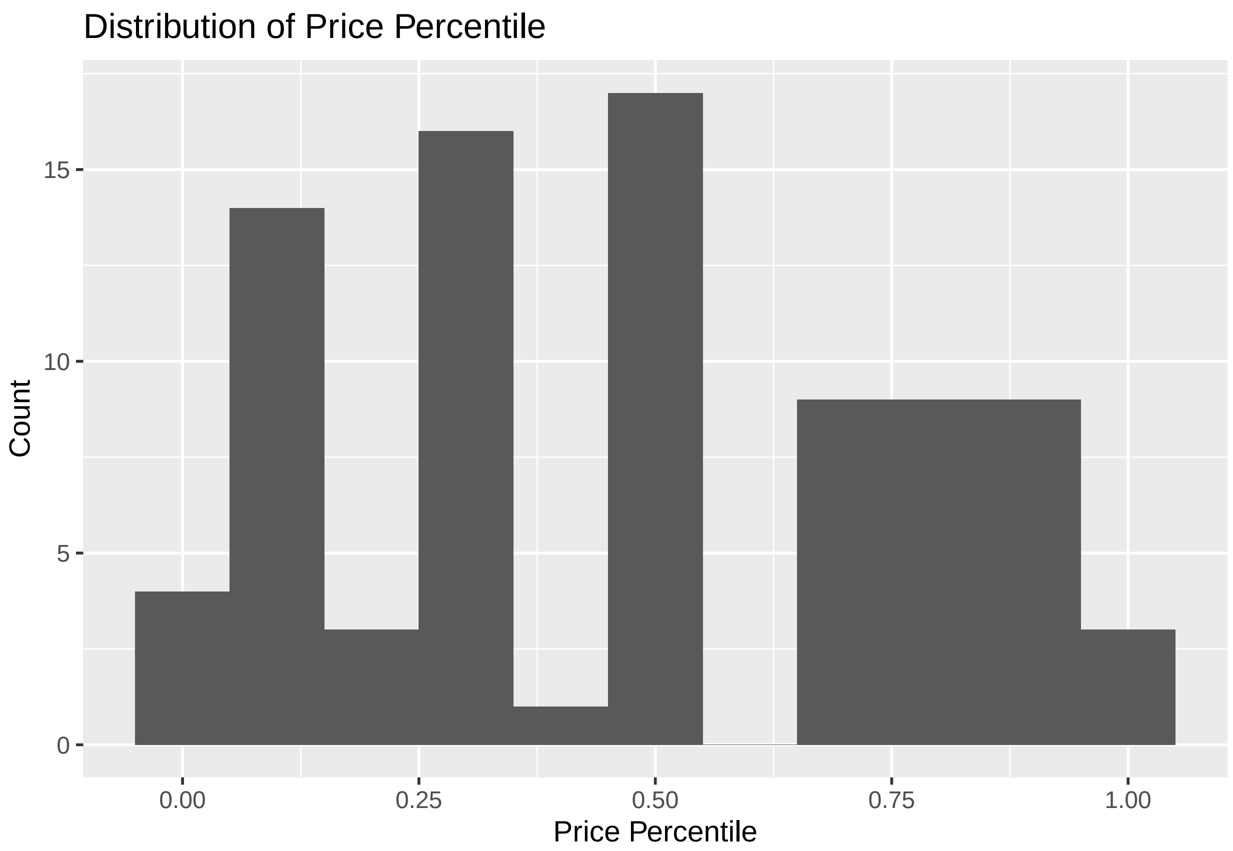
Three numerical variables of potential interest in this dataset are the sugar percentile, the unit price percentile, and the percentage of time that the candy bar won against its competitors. Let's look at the distribution of these variables. What units are these variables measured in?

The units are all percentiles, and we don't know what their base units were.

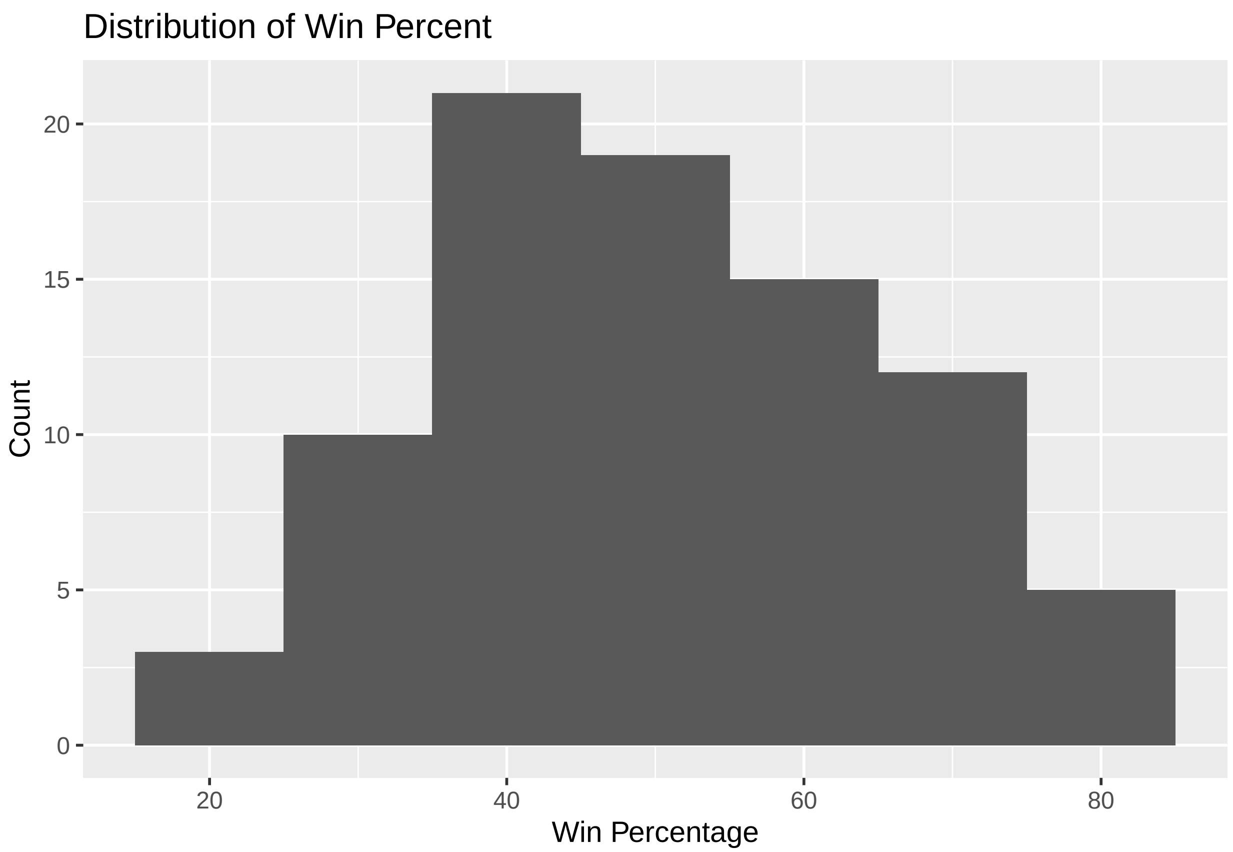
```
ggplot(data = candy_rankings, aes(x = sugarpercent)) +
  geom_histogram(binwidth = .1) +
  labs(title = "Distribution of Sugar Percentile", x = "Sugar Percentile", y =
         "Count")
```



```
ggplot(data = candy_rankings, aes(x = pricepercent)) +
  geom_histogram(binwidth = .1) +
  labs(title = "Distribution of Price Percentile", x = "Price Percentile", y =
         "Count")
```



```r
ggplot(data = candy_rankings, aes(x = winpercent)) +
  geom_histogram(binwidth = 10) +
  labs(title = "Distribution of Win Percent", x = "Win Percentage", y = "Count")
```



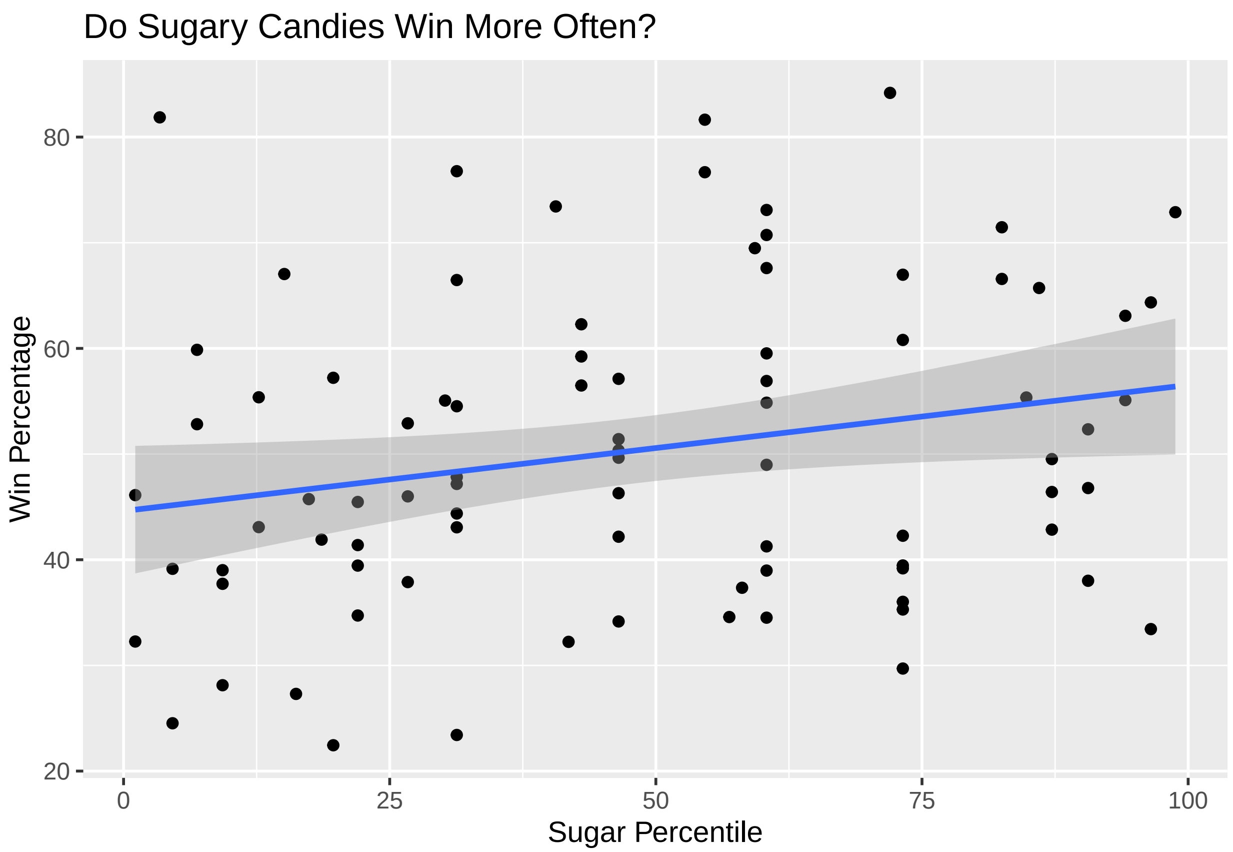
Let's say you wanted to see if sugary candies are more likely to win. First, we notice that the sugar and price percentiles are on a scale from 0 to 1, while the win percentage variable is on a scale from 0 to 100. Let's change the sugar and price variables to the same scale as win percentage by creating new variables on a 0 to 100 scale to be on the same scale as win percentage.

```
candy_rankings<- candy_rankings %>%
  mutate(sugarpercent100 = sugarpercent * 100)
```

```
candy_rankings<- candy_rankings %>%
  mutate(pricepercent100 = pricepercent * 100)
```

```
ggplot(data = candy_rankings, aes(x = sugarpercent100, y = winpercent)) +
  labs(title= "Do Sugary Candies Win More Often?", x = "Sugar Percentile", y =
       "Win Percentage") +
  geom_point() +
  geom_smooth(method = "lm")
```

```
## ‘geom_smooth()‘ using formula ’y ~ x’
```



Some Terminology

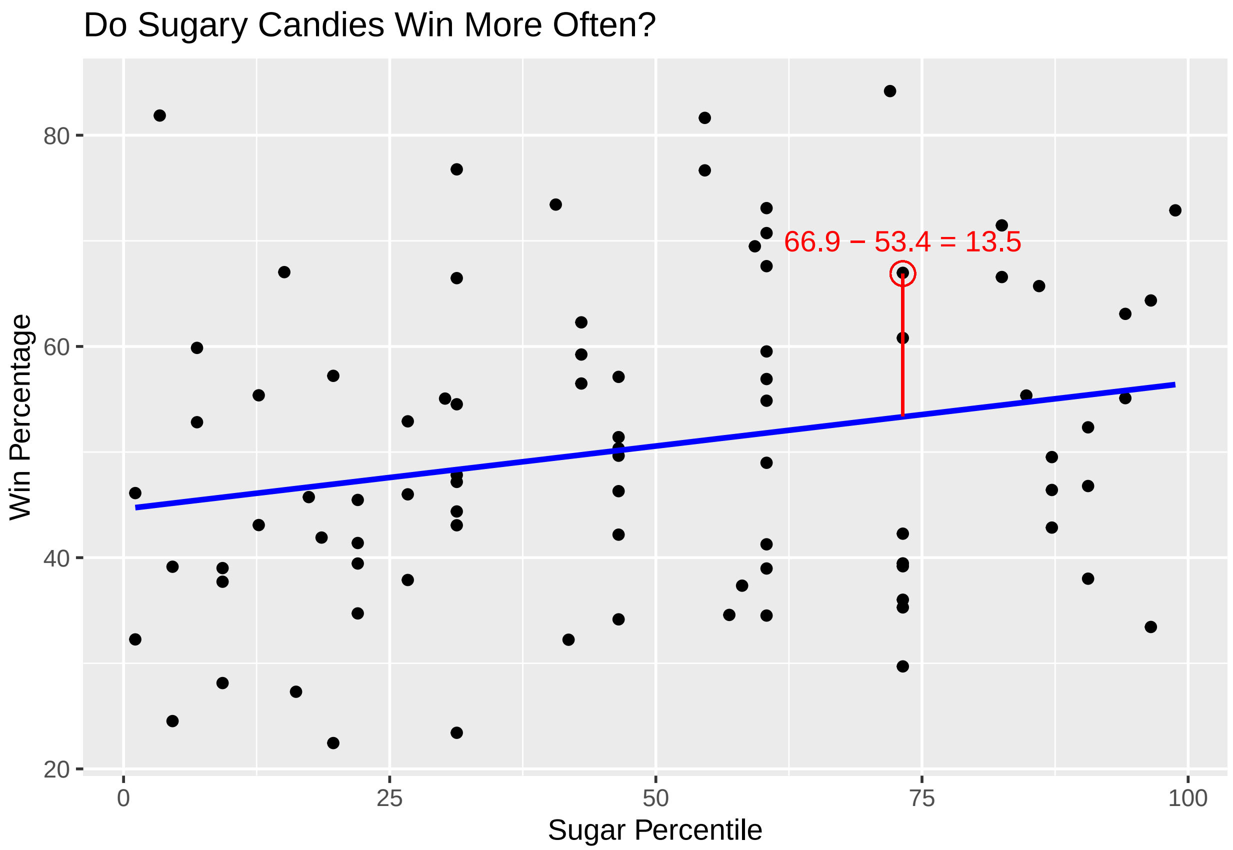
- Response Variable: Variable whose behavior or variation you are trying to understand, on the y-axis. Also called the dependent variable.
- Explanatory Variable: Other variables that you want to use to explain the variation in the response, on the x-axis. Also called independent variables, predictors, or features.
- Predicted value: Output of the model function
  - The model function gives the typical value of the response variable conditioning on the explanatory variables (what does this mean?)
- Residuals: Shows how far each case is from its predicted value
  - Residual = Observed value - Predicted value
  - Tells how far above/below the model function each case is

Question: What does a negative residual mean? Which candies on the plot have have negative residuals, those below or above the line?

Negative residuals are any where our actual values were less than our predicted values - in other words, anything below the line.

```
ggplot(data = candy_rankings, aes(x = sugarpercent100, y = winpercent)) +
  labs(title = "Do Sugary Candies Win More Often?", x = "Sugar Percentile", y =
         "Win Percentage") +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, col = "blue", lty = 1, lwd = 1) +
  geom_segment(aes(x = 73.2, y = 66.9, xend = 73.2, yend = 53.38), col = "red") +
  geom_point(aes(x = 73.2, y = 66.9), col = "red", shape = 1, size = 4) +
   annotate("text", x = 73.2, y = 70, label = "66.9 - 53.4 = 13.5", color =
              "red", size = 4)
```

```
## 'geom_smooth()' using formula 'y ~ x'
```

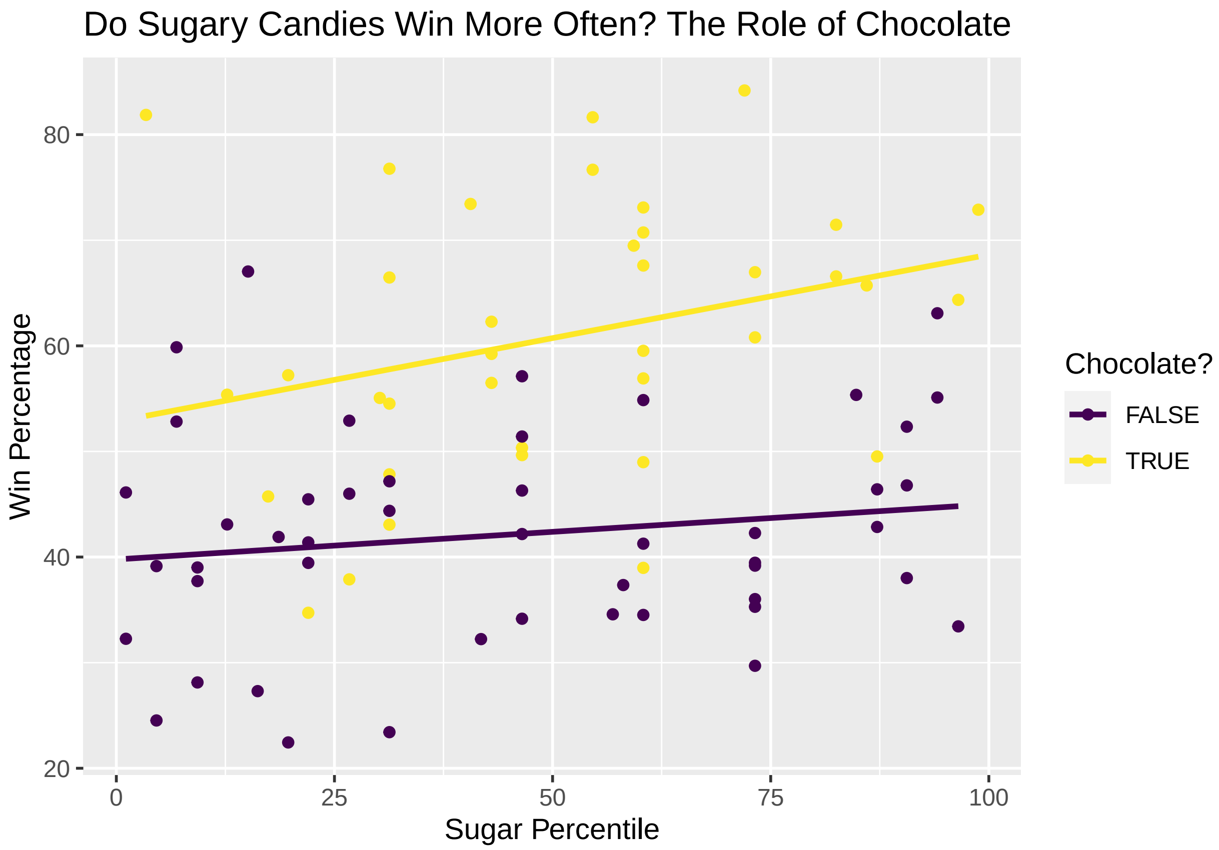


Multiple Explanatory Variables

How, if at all, does the relationship between sugar percentile and win percentage vary by whether or not the candy is chocolate?

```
ggplot(data = candy_rankings, aes(x = sugarpercent100, y = winpercent,
                                  color = factor(chocolate))) +
  scale_color_viridis(discrete = TRUE, option = "D", name = "Chocolate?") +
  labs(title = "Do Sugary Candies Win More Often? The Role of Chocolate", x =
         "Sugar Percentile", y = "Win Percentage") +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE)
```

```
## `geom_smooth()` using formula 'y ~ x'
```



We will talk about multiple explanatory variables in the following weeks.

Models - upsides and downsides

- Models can sometimes reveal patterns that are not evident in a graph of the data. This is a great advantage of modeling over simple visual inspection of data.
- There is a real risk, however, that a model is imposing structure that is not really there on the scatter of data, just as people imagine animal shapes in the stars. A skeptical approach is always warranted.

Variation in the Model

- This is just as important as the model, if not more!
- Statistics is the explanation of variation in the context of what remains unexplained.
- The scatter suggests that there might be other factors that account for large parts of candy win percentage variability, or perhaps just that randomness plays a big role.
- Adding more explanatory variables to a model can sometimes usefully reduce the size of the scatter around the model. (We'll talk more about this later.)

How do we use models?

1. Explanation: Characterize the relationship between y and x via slopes for numerical explanatory variables or differences for categorical explanatory variables.
2. Prediction: Plug in x, get the predicted y.

Intepreting Models, Some Helpful Commands

- **`tidy`**: Constructs a tidy data frame summarizing model's statistical findings
- **`glance`**: Constructs a concise one-row summary of the model (we'll use this Weds)
- **`augment`**: Adds columns (e.g. predictions, residuals) to the original data that was modeled

Returning to our earlier question, do sugary candies win more often?

```
sugarwins <- linear_reg() %>%
  set_engine("lm") %>%
  fit(winpercent ~ sugarpercent100, data = candy_rankings)
tidy(sugarwins)
```

```
## # A tibble: 2 x 5
##   term            estimate std.error statistic  p.value
##   <chr>              <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)       44.6      3.09       14.5  2.20e-24
## 2 sugarpercent100    0.119    0.0556      2.14 3.49e- 2
```

$$W\widehat{inPer}cent = 44.6 + 0.12\ SugarPercentile$$

- Slope: For each additional percentile of sugar, the win percentage is expected to be higher, on average, by 0.12 percentage points.
- **Question**: Why is it important to know what units the variable is measured in here? Why did we change the scale of percentile here earlier?

A linear regression line tells us how much one unit increase should increase our expected output - but that cannot be interpreted if we don't know what our expected output is measured in. We need units to explain anything.

- Intercept: Candies that are in the 0th percentile for sugar content are expected to win 44.6 percent of the time, on average.
  - Does this make sense?

  To an extent, yes - we know that sugar is not a great predictor of win percentage, and 44.6 is still reasonably close to random chance.

The linear model with a single predictor

- We're interested in the β_0 (population parameter for the intercept) and the β_1 (population parameter for the slope) in the following model:

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

- Unfortunately, we can't get these values
- So we use sample statistics to estimate them:

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

Least squares regression

The regression line minimizes the sum of squared residuals.

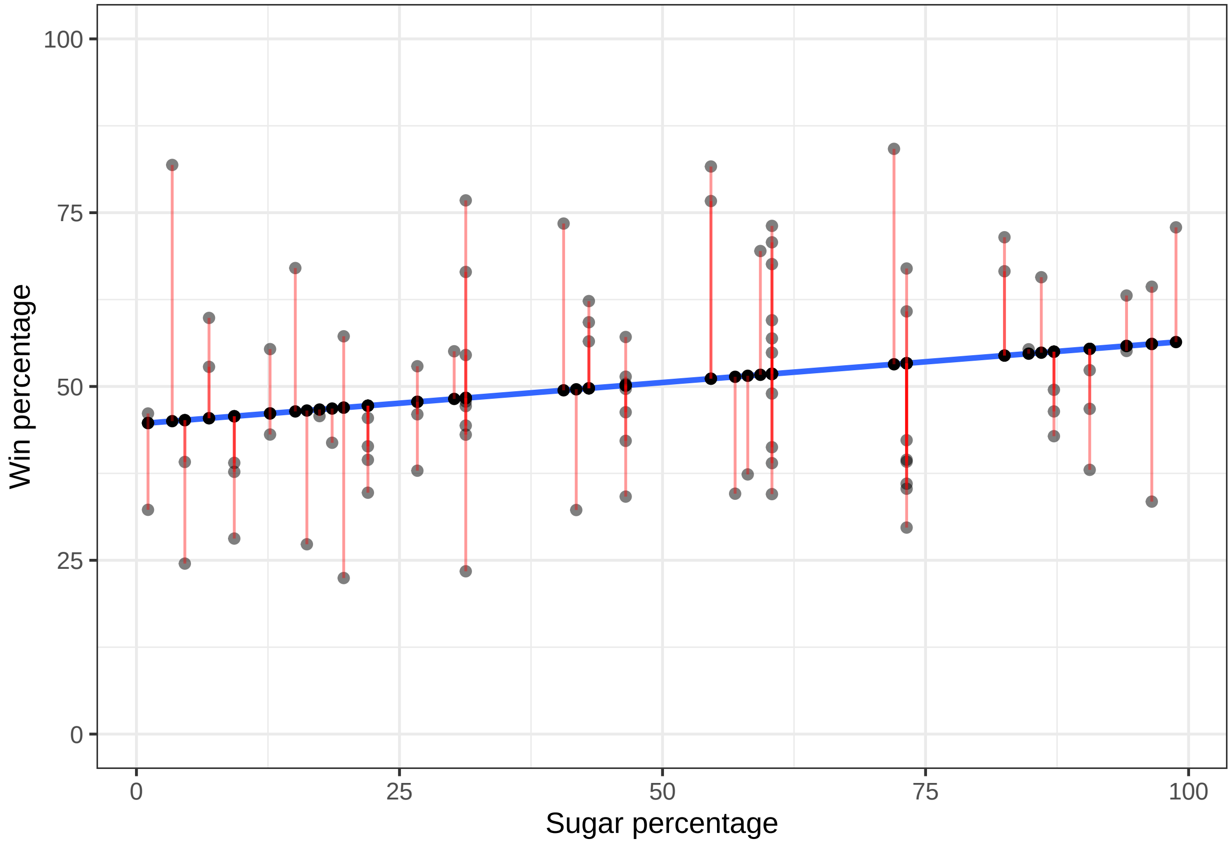
- **Residuals**: $e_i = y_i - \hat{y}_i$,
- The regression line minimizes $\sum_{i=1}^{n} e_i^2$.
- Equivalently, minimizing $\sum_{i=1}^{n} [y_i - (b_0 + b_1 \ x_i)]^2$
- **Question**: Why do we minimize the *squares* of the residuals?

Because we want to deal with negative and positive residuals in the same way, as well as pay special attention to removing large errors.

Visualizing Residuals

```
sugar_wins <- linear_reg() %>%
  set_engine("lm") %>%
  fit(winpercent ~ sugarpercent100, data = candy_rankings)

sugarwins %>%
  augment(candy_rankings) %>%
  ggplot(mapping = aes(x = sugarpercent100, y = winpercent)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE) +
  geom_point(mapping = aes(y = .pred)) +
  geom_segment(mapping = aes(xend = sugarpercent100, yend = .pred),
               alpha = 0.4, color="red") +
  ylim(0, 100) +
  labs(x = "Sugar percentage", y = "Win percentage") +
  theme_bw()
```



Check out the applet here to see this process in action.

Properties of the least squares regression line

- The estimate for the slope, b_1, has the same sign as the correlation between the two variables.
- The regression line goes through the center of mass point, the coordinates corresponding to average x and average y: $(\bar{x}, \bar{y})$
- The sum of the residuals is zero:

$$\sum_{i=1}^{n} e_i = 0$$

- The residuals and x values are uncorrelated.

Categorical Predictors

- Non-continuous predictors: Let's say that we want to see if candies with chocolate are more popular than those without chocolate. Our chocolate variable has two categories: chocolate and not chocolate. Predictors such as this are known as **dummy variables** and are typically coded as "1" (if true) and "0" (if false).

Let's run a model with the chocolate variable.

```
chocolatewins <- lm(winpercent ~ factor(chocolate), data = candy_rankings)
tidy(chocolatewins)
```

```
## # A tibble: 2 x 5
##   term                   estimate std.error statistic  p.value
##   <chr>                     <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)                42.1      1.65     25.6  4.05e-41
## 2 factor(chocolate)TRUE      18.8      2.50      7.52 5.86e-11
```

- **Slope:** Candies with chocolate are expected, on average, to have a winning percentage 18.8 percentage points higher than candies that don't have chocolate.
  - Compares baseline level (`chocolate = 0`) to other level (`chocolate = 1`).
- **Intercept:** Candies that don't have chocolate are expected, on average, to win 42.1 percent of the time.

What about categorical predictors with more than two levels.

Let's say that you are interested in which combination of chocolate and peanuts/almonds is the most popular. What would you do to create a variable to represent each of these options. Let's use `case_when` here.

```
candy_rankings<-candy_rankings %>%
   mutate(choc_peanutyalmondy =
            case_when(chocolate == 1 & peanutyalmondy == 0 ~ 'just chocolate',
                      chocolate == 1 & peanutyalmondy == 1 ~ 'both',
                      chocolate == 0 & peanutyalmondy == 1 ~
                        'justpeanutyalmondy',
                      chocolate == 0 & peanutyalmondy == 0 ~ "neither"))
```

Now, let's run the model.

```
bestcombowins <- lm(winpercent ~ factor(choc_peanutyalmondy), data = candy_rankings)
tidy(bestcombowins)
```

```
## # A tibble: 4 x 5
##   term                                       estimate std.error statistic  p.value
##   <chr>                                         <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)                                    68.5      3.16     21.7  1.69e-35
## 2 factor(choc_peanutyalmondy)just chocola~      -11.2      3.84     -2.92 4.51e- 3
## 3 factor(choc_peanutyalmondy)justpeanutya~      -33.6      8.35     -4.03 1.26e- 4
## 4 factor(choc_peanutyalmondy)neither            -26.0      3.54     -7.35 1.43e-10
```

When the categorical explanatory variable has many levels, the levels are encoded to dummy variables.

Each coefficient describes the expected difference between combinations of chocolate in that particular candy compared to the baseline level.

Question: Notice here that we have three variables in the model. Why don't we have four since there are four possibilities? Why don't we just use one variable here by adding the two categories together?

I believe this means that there are no instances of `both`. We can't just use one variable by adding two categories together because the different candies have different qualities and they are discrete categorical, not continuous.

Correlation does not imply causation!

Remember this when interpreting model coefficients.

Prediction with Models

Do Candies with more ingredients do better?

Does the number of ingredients in a candy bar make it do better? Let's create a variable measuring the number of ingredients and see how well it does.

```
candy_rankings <- candy_rankings %>%
  mutate(number_ingredients = chocolate + fruity + caramel + peanutyalmondy +
           nougat + crispedricewafer)
```

Do you think a candy bar with all of these things at the same time sounds good?

```
ingredientsmodel <- lm(winpercent ~ number_ingredients, data = candy_rankings)
tidy(ingredientsmodel)
```

```
## # A tibble: 2 x 5
##   term               estimate std.error statistic  p.value
##   <chr>                 <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)            35.5      2.48     14.3  3.97e-24
## 2 number_ingredients     10.7      1.55      6.93 8.22e-10
```

On average, what percentage of the time would you expect a candy bar with four ingredients to win?

$$Win\widehat{Percentage} = 35.5 + 10.7 \ Ingredients_{number}$$

```
35.5 + 10.7 * 4
```

```
## [1] 78.3
```

"On average, we expect candies with four ingredients to win 78.3 percent of the time."

Warning: We "expect" this to happen, but there will be some variability. (We'll learn about measuring the variability around the prediction later.)

On average, how often would you expect a candy with all seven ingredients to win? Is this realistic? Do any bars like this actually exist in the data?

You would expect it to win 110.4% of the time. This is not realistic and if you look at the data, there are no candies with more than 4 ingredients.

```
35.5 + 10.7 * 7
```

```
## [1] 110.4
```

```
candy_rankings %>%
  summarise(max(number_ingredients))
```

```
## # A tibble: 1 x 1
##   `max(number_ingredients)`
##                       <int>
## 1                         4
```

Would you eat a candy bar with all seven ingredients?

Watch out for extrapolation!

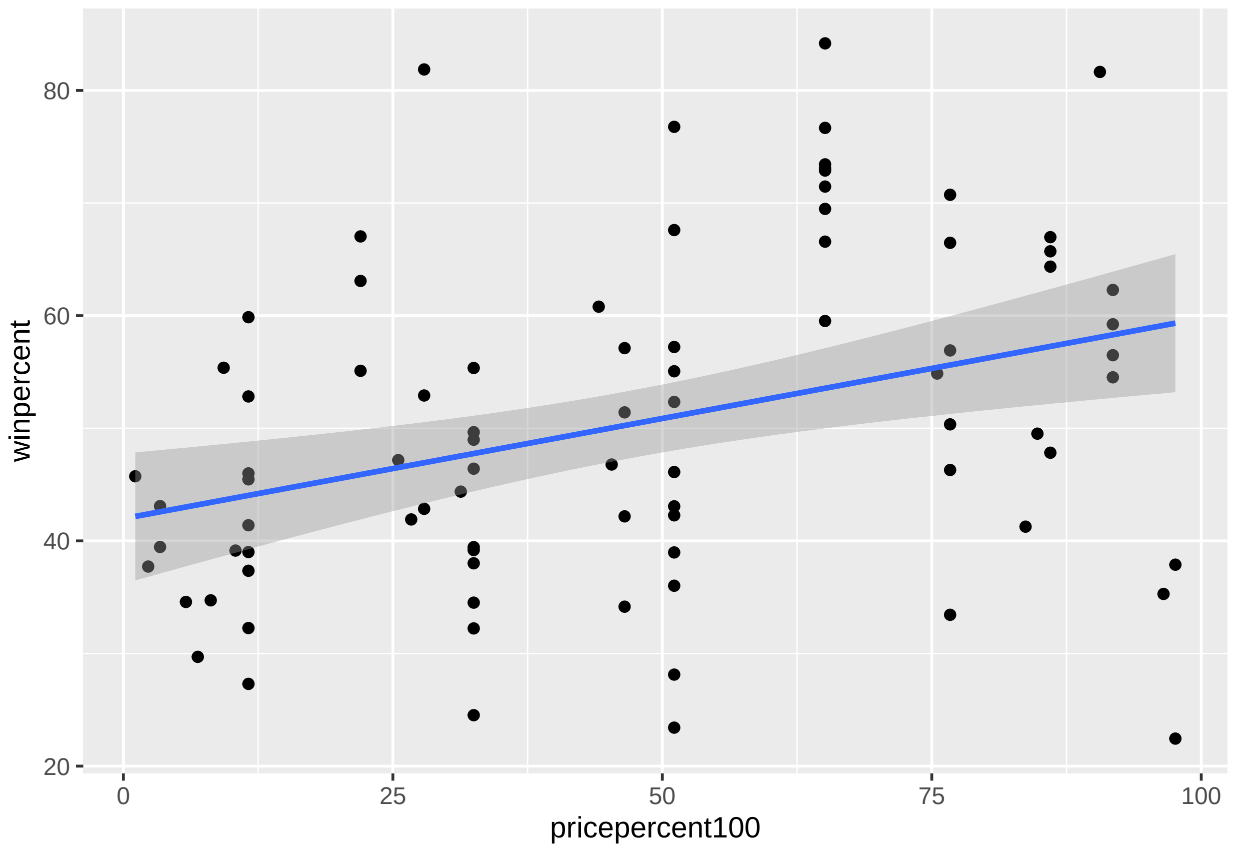
"When those blizzards hit the East Coast this winter, it proved to my satisfaction that global warming was a fraud. That snow was freezing cold. But in an alarming trend, temperatures this spring have risen. Consider this: On February 6th it was 10 degrees. Today it hit almost 80. At this rate, by August it will be 220 degrees. So clearly folks the climate debate rages on." -Stephen Colbert, April 6th, 2010 (Open Intro Stats, page 322)

Practice

Exercise 1: Do you get your money's worth when buying expensive candy? First, use `ggplot` to visualize the relationship between cost percentile and win percentage as we did for sugar percentile and win percentage.

```
ggplot(candy_rankings, aes(x = pricepercent100, y = winpercent)) +
  geom_point() +
  geom_smooth(method = "lm")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



```r
costmodel <- lm(winpercent ~ pricepercent100, data = candy_rankings)
tidy(costmodel)
```

```
## # A tibble: 2 x 5
##   term            estimate std.error statistic  p.value
##   <chr>              <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)       42.0      2.91       14.4  2.39e-24
## 2 pricepercent100    0.178    0.0530      3.35 1.21e- 3
```

Interpretation: For each percentile increase in price, we would expect an increase of 0.178 in our win percentage.

Slope: 0.178

Intercept: 42.0

Exercise 2: How popular are candies that have caramel? Run a linear model with caramel as the explanatory variable and win percentage as the response variable and interpret it.

```r
caramodel <- lm(winpercent ~ caramel, data = candy_rankings)
tidy(caramodel)
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)    48.9       1.72     28.5  1.22e-44
## 2 caramelTRUE     8.42      4.23      1.99 4.99e- 2
```

Interpretation: We would expect a candy containing caramel to have a win percentage 8.42 points higher than a candy not containing caramel.

Slope: 8.42

Intercept: 48.9

Exercise 3: Which candy is the sweetest: chocolate, caramel, candies with both, or candies with neither? Create a variable that represents this using `case_when` and then run and interpret a linear model with sugar content as the dependent variable.

Slopes:

Intercept:

Additional Resources

- https://r4ds.had.co.nz/model-basics.html

Tasks For Next Class

- Watch Prep Videos
- Take Prep Quiz by Tuesday March 22 at 11:59 PM.