The ECDF

The ECDF is a function. In fact, "ECDF"" stands for "empirical cumulative distribution function." Just like other mathematical functions, the ECDF takes an input and produces an output. In mathematical terms, it happens to be a step function, which is usually flat but occasionally features discontinuous jumps. When plotted, these occasional jumps cause the ECDF to look A series of steps, where the steps are unevenly placed both vertically and horizontally.

Here's the formal definition:

ECDF: Suppose a list of numbers. The EDCF (of this list) is a function that takes an input and returns as an output the percentage (or proportion) of entries in the list that are less than or equal to the input.

For example, suppose the list -3, 5, 7, 7, and 9. What value would the ECDF of this list return if given 4 as an input?

To answer this, we would simply calculate the percentage (or proportion) of entries that are less than or equal to 4. In this case, we have $\frac{\text{number of items less than or equal to 4}}{\text{total number of items in the list}} \times 100\% = \frac{1}{5} \times 100\% = 0.2 \times 100\% = 20\%$.[1]

Similarly, what value would the ECDF of this same list return if given 5 as an input? In this case, we would simply calculate the percentage of entries that are less than or equal to 5, giving us $\frac{\text{number of items less than or equal to 5}}{\text{total number of items in the list}} \times 100\% = \frac{2}{5} \times 100\% = 0.4 \times 100\% = 40\%$.

Review Exercises

1. Suppose the list -4, -1, 0, 0, 0, 1, 2, 2, 2, 7. What is the ECDF of this list if -5 is given as an input? What about -4? What about 0? What about 7? What about 125?
2. Suppose I reordered the list to 0, -4, 1, -1, 7, 0, 2, 0, 2, 2. Would this change the answer to the previous question?

Plotting the ECDF by Hand

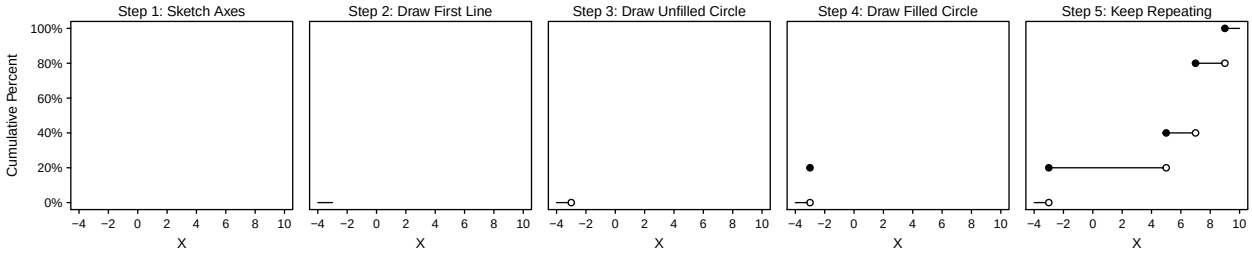
When plotting the ECDF by hand, it is helpful to know the jump height. If we are interested in percentages, then the jump height equals $\frac{1}{\text{total number of items in the list}} \times 100\%$. For the list above, we have a jump height of $\frac{1}{5} \times 100\% = 0.2 \times 100\% = 20\%$.[2]

1. Start by sketching the axes.
    a. The horizontal axis should range from slightly below the smallest value in the list to slightly above the largest value in the list. For the list above, we might choose -4 (slightly below -3) to 10 (slightly above 9).
    b. The vertical axis should range from 0% to 100%.[3]
2. Starting at the far left edge of the horizontal axis and at 0% on the vertical axis, draw a horizontal line to the right until you encounter an entry in the list. The first value(s) you encounter will be the smallest value(s) in the list.
3. Draw a small, unfilled circle on the right endpoint of this line, indicating that the line does not include this point.
4. Draw a small, filled circle one jump height above this filled circle. This filled circle indicates that the next line we'll draw (see the step below) does include this point. Note: if you happen to encounter two points rather than one, go up two jumps. If you happen to encounter three points, then go up three jumps, and so on.
5. Starting at the solid circle, start drawing a horizontal line to the right and repeat steps 2-5 until you reach the end of the horizontal axis.
6. As a check, you should have reached 100% when you get to the largest value(s) in the list.

[1] If we wanted the proportion, rather than the percentage, we would simply drop the 100% from out calculations, giving us 0.2.

[2] If we want the ECDF to return a proportion, then we just drop the 100% from the jump height calculation.

[3] Or 0 to 1 if we prefer the proportion.



1. Suppose the list 1, 2, 2, 4. Plot the ECDF by hand.
2. Suppose the list -4, -1, 0, 0, 0, 1, 2, 2, 2, 7. Plot the ECDF by hand.

Plotting an ECDF with `ggplot()`

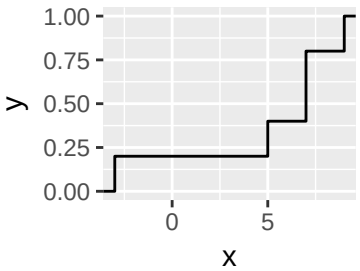
I've been emphasizing all semester that each ggplot we draw has three crucial components: a data frame, aesthetics, and a geometry. That's *almost* right. It turns out that each geometry is associated with a particular *statistic* and vice versa. For example, `geom_histogram()` is associated with the bin statistic (i.e., binning, or counting the observations in each bin), `geom_density()` is associated with the density statistic, and `geom_line()` is associated with the identity statistic (i.e., doing nothing to the data). Sometimes, it is easier to specify a statistic rather than a geometry. This is purely a matter of convenience.

In the case of producing an ECDF plot, it is easier to specify `stat_ecdf()`, which has `geom_step()` as its default geometry. To the best of my knowledge, this is the only time this semester we'll prefer to specify a statistic rather than a geometry.[4] Notice that `ggplot()` calculates a cumulative proportion rather than a cumulative percent.

```
# load packages
library(ggplot2)

# create data frame
x <- c(-3, 5, 7, 7, 9) # create numeric vector
df <- data.frame(x)  # place the vector inside a data frame

# draw ecdf plot
ggplot(df, aes(x = x)) +
  stat_ecdf() # or use geom_step(stat = "ecdf")
```



Review Exercises

1. Load the nominate data set and plot the ECDF for Republicans in the 114th Congress. Hint: You'll want to use the `subset()` function. Use the plot to estimate the percent of Republicans in the 114th Congress that are more conservative than Paul Ryan, who has an ideology score of 0.56. Repeat for Kevin McCarthy, who has an ideology score of 0.46.

[4] Alternatively, we could use `geom_step()` (a step function geometry) and specify that we would like to use the ECDF statistic (rather than the default identity statistic) by supplying `stat = "ecdf"` as an argument to `geom_step`. I think you'll agree it's easier to just use `stat_ecdf()`.

2. Plot the ECDF for Democrats in the 114th Congress. Use the plot to estimate the percent of Democrats in the 114th Congress that are more liberal than Nancy Pelosi, who has an ideology score of -0.49. Repeat for Steny Hoyer, who has an ideology score of -0.38.

Creating an ECDF in R

We can also create an ECDF function in R using the `ecdf()` function. We supply this function with a numeric vector and it outputs a new R function. Let me emphasize that again: ***`ecdf()` does not return a scalar, vector, or data frame–t returns a new function.*** This new function takes a scalar (or vector) as an argument and returns the value of the ECDF function at that points (or those points).

```
x <- c(-3, 5, 7, 7, 9) # create numeric vector
ecdf0 <- ecdf(x)  # create ecdf function

# using ecdf0, the new function we created
ecdf0(4)  # evalate the ecdf at 4
```

```
## [1] 0.2
```

```
ecdf0(5)  # evalate the ecdf at 5
```

```
## [1] 0.4
```

1. Load the nominate data set and use the `ecdf()` function to create an ECDF function for Republicans in the 114th Congress. Use the function you created to calculate the percent of Republicans in the 114th Congress that are more conservative than Paul Ryan, who has an ideology score of 0.56. Repeat for Kevin McCarthy, who has an ideology score of 0.46. Hint: Remember that the ECDF returns the values *less than* (i.e., more liberal than) or equal to the input. For these questions, you can consider *less than* and *less than or equal to* equivalent, so don't stress about the equality bit.
2. Use the `ecdf()` function to create an ECDF function for Democrats in the 114th Congress. Use the function you created to estimate the percent of Democrats that are more liberal than Nancy Pelosi, who has an ideology score of -0.49. Repeat for Steny Hoyer, who has an ideology score of -0.38.