An advanced example of a PMF visualization

```r
library(tidyverse)
county_complete <- read_rds(
  path = url("http://spring18.cds101.com/files/datasets/county_complete.rds"))
nebraska_iowa <- county_complete %>%
  filter(state == "Iowa" | state == "Nebraska")
```

The following example assumes you have already read the standard reading on *Probability mass functions*.

Histograms and PMFs are useful while you are exploring data and trying to identify patterns and relationships. Once you have an idea what is going on, a good next step is to design a visualization that makes the patterns you have identified as clear as possible.

In our prior comparison of the average work travel times for Nebraska and Iowa, we saw that there was a large overlap in travel times between 11 and 29 minutes, but that the overlap wasn't exact. So it makes sense to zoom in on that part of the graph, and to transform the data to emphasize differences. To do this, we need the values of the PMF in each of the bins. The `ggplot_build()` function allows us to extract these numerical values, although it takes a couple of steps to do. First, we need to create a histogram of the travel times in Iowa and Nebraska and assign it to a variable:

```r
nebraska_iowa_histogram <- nebraska_iowa %>%
  ggplot() +
  geom_histogram(
    mapping = aes(x = mean_work_travel, fill = state), binwidth = 1)
```

Next, we use `ggplot_build()` to convert the figure into a `list()` of information about the plot:

```r
nebraska_iowa_figure_list <- ggplot_build(nebraska_iowa_histogram)
```

A `list()` is a data type that we haven't used in the course yet. It's a convenient alternative to the `tibble` when you need to store uneven or very different kinds of information. Like the `tibble`, you can label the entries in a list. Our list `nebraska_iowa_figure_list` has several labels containing metadata about the plot:

```r
names(nebraska_iowa_figure_list)
```

```
## [1] "data"   "layout" "plot"
```

The one that we want to use is named `data`. To get the information inside of the `data` label, we use the `pluck()` function from `tidyverse`.

```r
nebraska_iowa_figure_df <- nebraska_iowa_figure_list %>%
  pluck("data", 1) %>%
  as_tibble()
```

The 1 inside of `pluck()` is necessary to get the data table stored inside of `data` (without it, we just get a `list()` data type back, which isn't helpful). We've also converted the data table into a `tibble` for convenience.

There are 17 columns in `nebraska_iowa_figure_df`. Let's use `glimpse()` to get a list of the variable names and previews of the first few entries:

```r
nebraska_iowa_figure_df %>%
  glimpse()
```

```
## Observations: 38
## Variables: 17
## $ fill     <chr> "#00BFC4", "#F8766D", "#00BFC4", "#F8766D", "#00BFC4"...
## $ y        <dbl> 1, 1, 2, 3, 5, 7, 10, 14, 12, 19, 8, 17, 8, 19, 11, 2...
## $ count    <dbl> 1, 0, 2, 1, 5, 2, 10, 4, 12, 7, 8, 9, 8, 11, 11, 17, ...
## $ x        <dbl> 11, 11, 12, 12, 13, 13, 14, 14, 15, 15, 16, 16, 17, 1...
## $ xmin     <dbl> 10.5, 10.5, 11.5, 11.5, 12.5, 12.5, 13.5, 13.5, 14.5,...
## $ xmax     <dbl> 11.5, 11.5, 12.5, 12.5, 13.5, 13.5, 14.5, 14.5, 15.5,...
## $ density  <dbl> 0.01075269, 0.00000000, 0.02150538, 0.01010101, 0.053...
## $ ncount   <dbl> 0.08333333, 0.00000000, 0.16666667, 0.05882353, 0.416...
## $ ndensity <dbl> 7.750000, 0.000000, 15.500000, 5.823529, 38.750000, 1...
## $ PANEL    <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...
## $ group    <int> 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1,...
## $ ymin     <dbl> 0, 1, 0, 2, 0, 5, 0, 10, 0, 12, 0, 8, 0, 8, 0, 11, 0,...
## $ ymax     <dbl> 1, 1, 2, 3, 5, 7, 10, 14, 12, 19, 8, 17, 8, 19, 11, 2...
## $ colour   <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ size     <dbl> 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5...
## $ linetype <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...
## $ alpha    <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
```

For our purposes, we want the `x` (values along horizontal axis), `density` (same as PMF), and `group` (created by `fill = state`) columns. We extract those using `select()` and then use `rename()` and `recode()` to give better names to the columns and categorical labels:

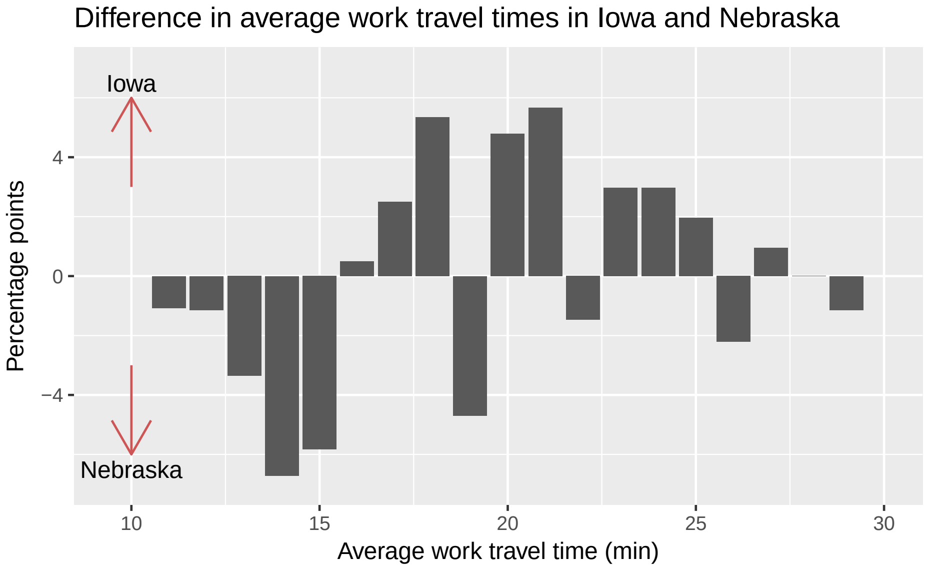
```r
nebraska_iowa_pmf <- nebraska_iowa_figure_df %>%
  select(x, density, group) %>%
  rename(mean_travel_time = x, pmf = density, state = group) %>%
  mutate(state = recode(state, `1` = "Iowa", `2` = "Nebraska"))
```

With all of that work done, we can now calculate the difference in the Iowa and Nebraska PMFs. To do that, we need to `spread()` the `state` column into separate `Nebraska` and `Iowa` columns, then use `mutate()` to subtract the `Nebraska` PMF from the `Iowa` PMF:

```r
nebraska_iowa_percent_difference <- nebraska_iowa_pmf %>%
  spread(key = state, value = pmf) %>%
  mutate(percent_difference = 100 * (Iowa - Nebraska)) %>%
  select(-Iowa, -Nebraska)
```

We remove the `Iowa` and `Nebraska` columns afterward, as we no longer need them after taking the difference. Now we can create a bar chart of the differences between Nebraska and Iowa travel times, which was the goal of this procedure:

```r
nebraska_iowa_percent_difference %>%
  ggplot() +
  geom_col(mapping = aes(x = mean_travel_time, y = percent_difference)) +
  coord_cartesian(xlim = c(9.5, 30), ylim = c(-7, 7))
```



The arrows indicate that a taller bar in the $y > 0$ region means the travel time is greater in Iowa, while a taller bar in the $y < 0$ region means the travel time is greater in Nebraska. This figure makes the pattern clearer: longer work travel times are more common in Iowa than in Nebraska. For now we should hold this conclusion only tentatively. We used the same dataset to identify an apparent difference and then chose a visualization that makes the difference apparent. We can't be sure this effect is real; it might be due to random variation. When we learn about statistical inference later on, we'll have the tools necessary to better answer that question.

Credits

This work, *An advanced example of a PMF visualization*, is a derivative of Allen B. Downey, "Chapter 3 Probability mass functions" in *Think Stats: Exploratory Data Analysis*, 2nd ed. (O'Reilly Media, Sebastopol, CA, 2014), used under CC BY-NC-SA 4.0. *An Advanced Example of a PMF Visualization* is licensed under CC BY-NC-SA 4.0 by James Glasbrenner.