plot_stackedbar_likert

Stackoverflow

2022-09-07

How to create stacked bar charts in ggplot 2 from Likert data

Credit: Stackoverflow Souce

Data

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
library(ggplot2)
library(scales)

# make the dataset
set.seed(8675309)
questions <- data.frame(Person = 1:20)

for (i in 1:20) {
  questions[[paste0('Q',i)]] <- sample(1:5, 20, replace=TRUE)
}
```

Creates a simple dataframe with 20 rows and 21 columns (20 questions, and a subject ID column)

Preparing Data

When preparing to generate a plot, you will almost always have to prep the data in some way. There are only two things I want to do here first before we plot. The first step is to make our data into a format which is referred to as Tidy Data. In the format we have it in now... it's okay to plot in Excel, but if we want to have a quality way of organizing and summarizing this data, we want to organize it to be in a "longer" table format. What we need is to organize in a way that has columns organized as:

Person | Question_num | Answer

```
questions <- questions %>% gather(key='Question_num', value='Answer', -Person)
```

The final thing I want to do here is to convert our column questions$Answer into a categorical variable, not a continuous number. V
into a factor. This also allows us to do two things at the same time that are quite useful here:

```
Setting the levels - this indicates which order you want the levels of the factor.
Setting the labels - this allows you to remap 1 to be "Approve" and 2 to be "Slightly Approve" and so o
```

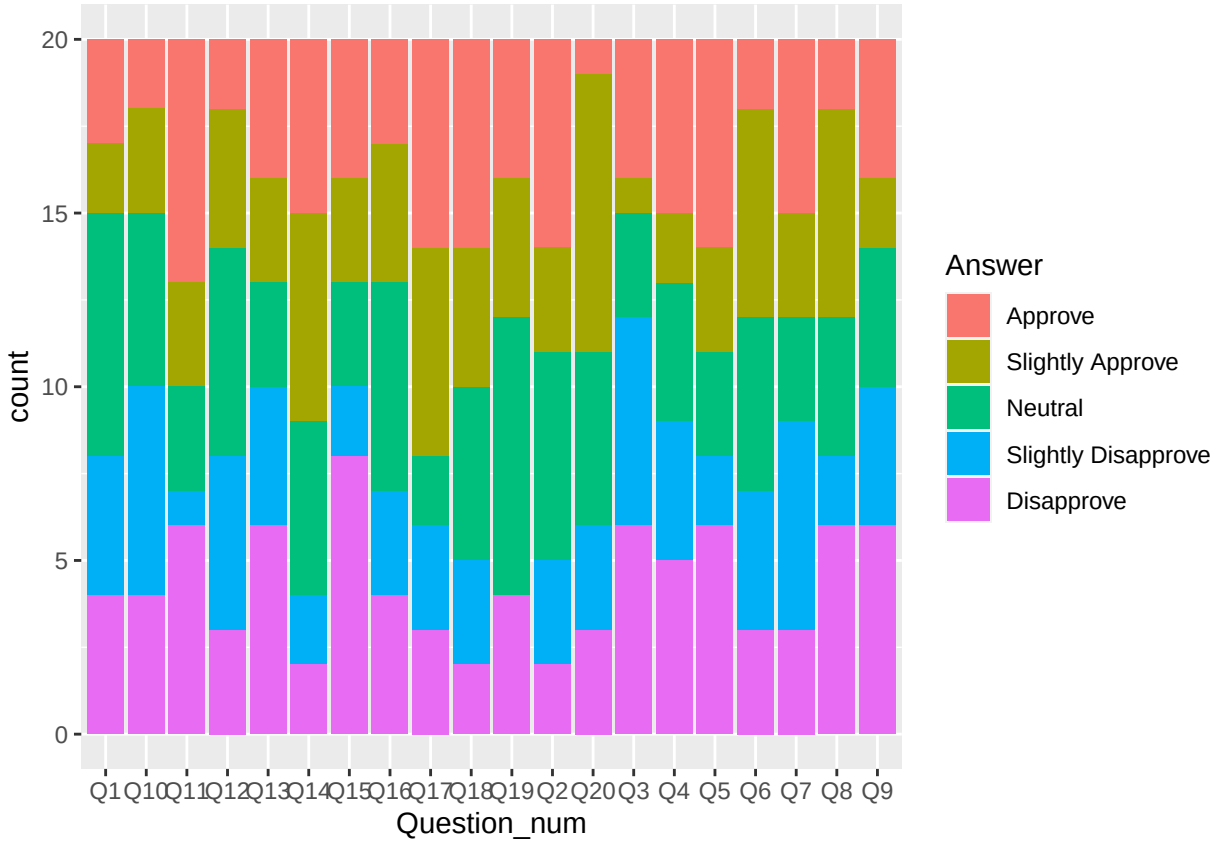
You can then check the data after and see that questions$Answer column is now composed of our labels() values, not numbers.

```
questions$Answer <- factor(questions$Answer,
    levels=1:5,
    labels=c('Approve','Slightly Approve','Neutral','Slightly Disapprove','Disapprove'))
```

Make the plot

We can then make the plot using the ggplot2 package. GGplot draws your data onto the plot area using geoms. In this case, we can use geom_bar() which will draw a barplot (totaling up the number/count of each item), and requires an x aesthetic only. If we set the fill color of each bar to be equal to the Answer column, then it will color-code the bars to be associated with the number of each answer for each question. By default, the bars are stacked on top of one another in the order that we set previously for the levels argument of the questions$Answer column.

```
ggplot(questions, aes(x=Question_num)) +
  geom_bar(aes(fill=Answer))
```



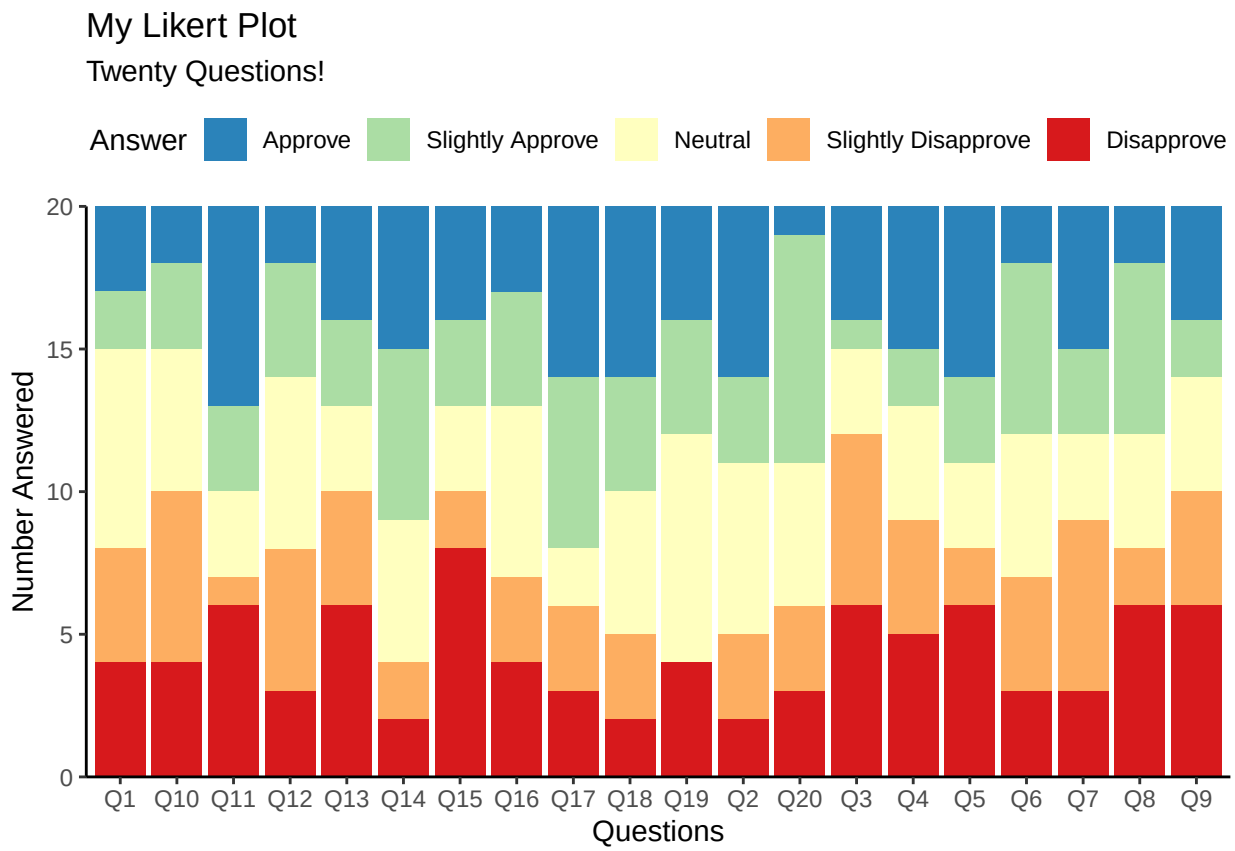
Modifying plot

There's a lot of things that are right with this plot and the general layout looks good. All that's left is to change the appearance in a few ways. We can do that by extending our plot code to change those aspects of the plot. Namely, I want to do the following:

```
Add a title and change some axis labels
Change the color scheme to one of the Brewer scales
Remove the whitespace in the y axis
Simplify the theme and move the legend to a different location
```

The full plot code now looks like this shown below. You should be able to identify which parts of the code are doing each thing referenced above.

```
ggplot(questions, aes(x=Question_num)) +
  geom_bar(aes(fill=Answer)) +
  scale_fill_brewer(palette='Spectral', direction=-1) +
  scale_y_continuous(expand=expansion(0)) +
  labs(
    title='My Likert Plot', subtitle='Twenty Questions!',
    x='Questions', y='Number Answered'
  ) +
  theme_classic() +
  theme(legend.position='top')
```



Displaying percentages

This is also fairly straightforward to do and often what one wants to do with a likert plot... so let's do it! We'll convert the counts into percentages in two stages. First, we'll get the axis and the bars setup to do that. Second, we'll overlay text on top of each bar to display the % answering that way for each question.

First, let's set the bars and y axis to be percentages, not counts. Our line to draw the bar geom was geom_bar(aes(fill=Answer)). There's a hidden default value for the position = "stack" inside that function as well (which we don't have to specify). The position argument deals with how ggplot should handle the situation when more than one bar needs to be drawn at that particular x value. In this case, it determines what to do with the 5 bars that correspond to each value of questions$Answer corresponding to each question.

"Stack", as you might assume, just stacks them on top of each other. Since we have 20 people answering each question, all of our bars are the same total height (20) for every question. What if you had only 19 people answering question #3? Well, that total bar height would be shorter than the rest.

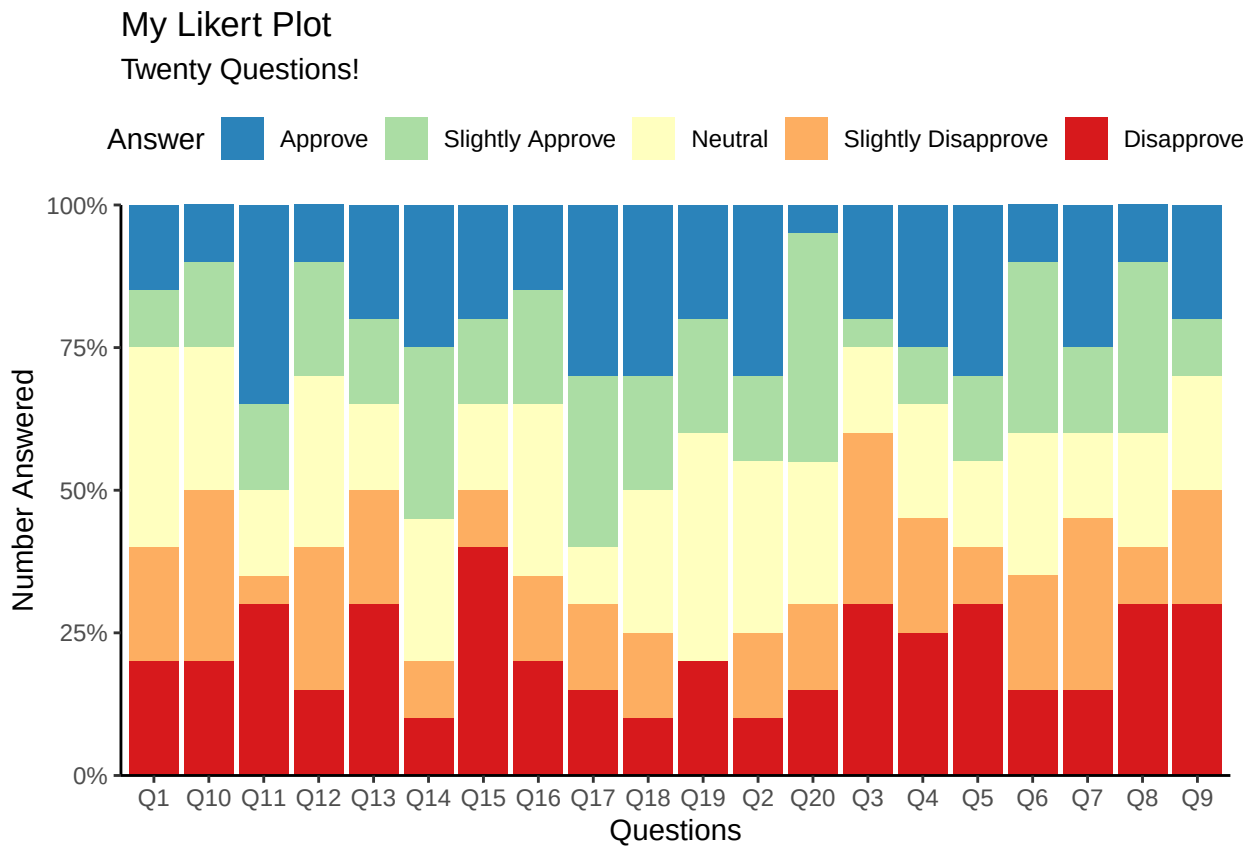
Normally, likert plots all show the bars the same height, because they are stacked according to the proportion of the whole they occupy for the total. In this case, we want each stack of bars to total up to 1. That means that 10 people answering one way should be mapped to a bar height of 0.5 (50%).

This is where the other position values come into play. We want to use position = "fill" to reference that we want the bars that need to be drawn at the same x axis position to be stacked... but not according to their value, but according to the proportion of the total value for that x axis position.

Finally, we want to fix our scale. If we just use position="fill" our y axis scale would have values of "0, 0.25, 0.50, 0.75, and 1.0" or something like that. We want that to look like "0%, 25%, 50%, 75%, 100%". You can do that within the scale_y_continuous() function and specify the labels argument. In this case, the scales

package has a convenient percent_format() function for just this purpose. Putting this together, you get the following:

```
ggplot(questions, aes(x=Question_num)) +
  geom_bar(aes(fill=Answer), position="fill") +
  scale_fill_brewer(palette='Spectral', direction=-1) +
  scale_y_continuous(expand=expansion(0), labels=scales::percent_format()) +
  labs(
    title='My Likert Plot', subtitle='Twenty Questions!',
    x='Questions', y='Number Answered'
  ) +
  theme_classic() +
  theme(legend.position='top')
```



Getting text on top

To put the text on top as percentages, that's unfortunately not quite as simple. For this, we need to summarize the data, and in this case the most simple way to do that would be to summarize before hand in a separate dataset, then use that to label the text using a text geom mapped to our summary data frame.

The summary data frame is created by specifying how we want to group our data together, then assigning n(), or the count of each answer, as the freq column value.

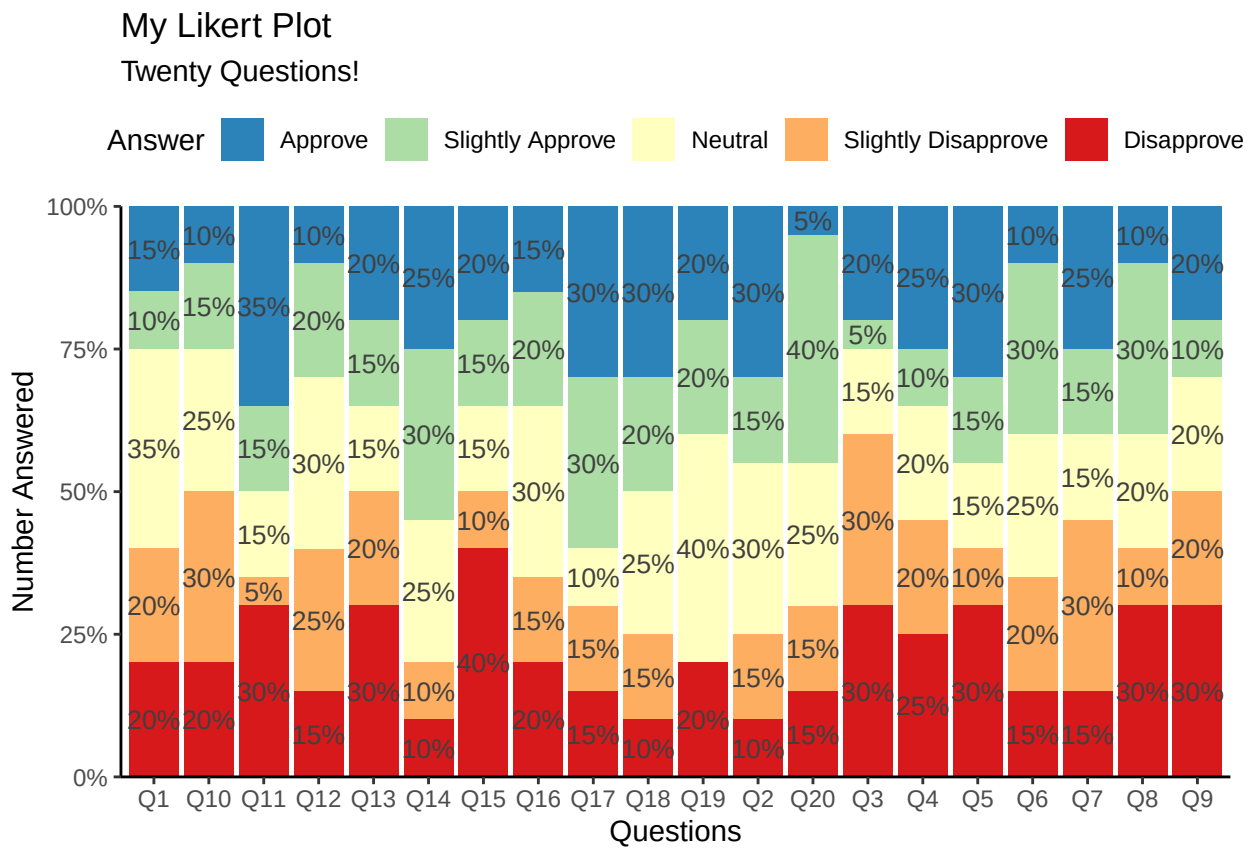
```
questions_summary <- questions %>%
  group_by(Question_num, Answer) %>%
  summarize(freq = n()) %>% ungroup()
```

```
## `summarise()` has grouped output by 'Question_num'. You can override using the
## `.groups` argument.
```

We then use that to map to a new geom: geom_text. The y value needs to be represented as a proportion again. Just like for geom_bar and the reasons above, we have to use the "fill" position. I also want to make sure the position is set to the "middle" vertically for each bar, so we have to specify a bit further by using position_fill(vjust=0.5) instead of just "fill".

You'll notice a final critical piece is that we're using a group aesthetic. This is very important. For the text geom, ggplot needs to know how the data is to be grouped. In the case of the bar geom, it was "obvious" (so-to-speak) that since the bars are colored differently, each color of bar was the separation. For text, this always needs to be specified (how to split the values) and we do this through the group aesthetic.

```
ggplot(questions, aes(x=Question_num)) +
  geom_bar(aes(fill=Answer), position="fill") +
  geom_text(
    data=questions_summary,
    aes(y=freq, label=percent(freq/20,1), group=Answer),
    position=position_fill(vjust=0.5),
    color='gray25', size=3.5
  ) +
  scale_fill_brewer(palette='Spectral', direction=-1) +
  scale_y_continuous(expand=expansion(0), labels=scales::percent_format()) +
  labs(
    title='My Likert Plot', subtitle='Twenty Questions!',
    x='Questions', y='Number Answered'
  ) +
  theme_classic() +
  theme(legend.position='top')
```


My Likert Plot
Twenty Questions!
Answer
Approve
Slightly Approve
Neutral
Slightly Disapprove
Disapprove
Number Answered
100%
75%
50%
25%
0%
Q1 Q10 Q11 Q12 Q13 Q14 Q15 Q16 Q17 Q18 Q19 Q2 Q20 Q3 Q4 Q5 Q6 Q7 Q8 Q9
Questions