Course evaluation

Hannah Meyer

05/02/2020

Contents

Read required libraries

```
library(tidyverse)
#> -- Attaching packages ------------------------------------------------------------------------------
#> v ggplot2 3.2.1     v purrr   0.3.2
#> v tibble  2.1.3     v dplyr   0.8.3
#> v tidyr   1.0.0     v stringr 1.4.0
#> v readr   1.3.1     v forcats 0.4.0
#> -- Conflicts ---------------------------------------------------------------------------------- tidy
#> x dplyr::filter() masks stats::filter()
#> x dplyr::lag()    masks stats::lag()
```

Read data

Survey data from 28 out of 35 participants of the Workshop:

```
survey <- read_csv("2020_Evaluations_DataVisualR_HMeyer.csv", skip = 2,
                   col_names = FALSE)
#> Parsed with column specification:
#> cols(
#>   .default = col_character(),
#>   X3 = col_double(),
#>   X8 = col_double(),
#>   X9 = col_double(),
#>   X10 = col_double(),
#>   X13 = col_double(),
#>   X15 = col_double(),
#>   X16 = col_double(),
#>   X24 = col_logical(),
#>   X25 = col_double(),
#>   X26 = col_double(),
#>   X27 = col_double()
#> )
#> See spec(...) for full column specifications.
```

Format into long format, change type and add short version of questions:

```
tb <- survey[1:5,] %>%
    column_to_rownames("X1") %>%
    t %>%
    as_tibble %>%
    select(-starts_with("Q1")) %>%
    mutate_each(funs(as.integer)) %>%
    pivot_longer(col=starts_with("Q"), names_to = "question_long",
                 values_to="score") %>%
    mutate(question=gsub("Q(\\d).*", "\\1", question_long)) %>%
    mutate(question_long=gsub("Q\\d\\.  (.*)", "\\1", question_long)) %>%
    drop_na
#> Warning: funs() is soft deprecated as of dplyr 0.8.0
#> Please use a list of either functions or lambdas:
#>
#>   # Simple named list:
#>   list(mean = mean, median = median)
#>
#>   # Auto named with `tibble::lst()`:
#>   tibble::lst(mean, median)
#>
#>   # Using lambdas
#>   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
#> This warning is displayed once per session.
#> Warning: NAs introduced by coercion
```

Prepare for plotting per data type

Set separate levels where appropriate, verbalise the selection levels:

```
q3 <- tb %>%
    filter(question == 3) %>%
    mutate(factorscores=factor(score, levels=1:3,
                               labels=c("too easy", "appropriate", "too hard")))
q2q4 <- tb %>%
    filter(question %in% c(2,4)) %>%
    mutate(factorscores=factor(score, levels=1:5,
                               labels=c("very poor", "poor", "average",
                                        "good", "very good")))

q5 <- tb %>%
    filter(question == 5) %>%
    mutate(factorscores=factor(score, levels=1:3,
                               labels=c("very much", "a little", "not at all")))
```

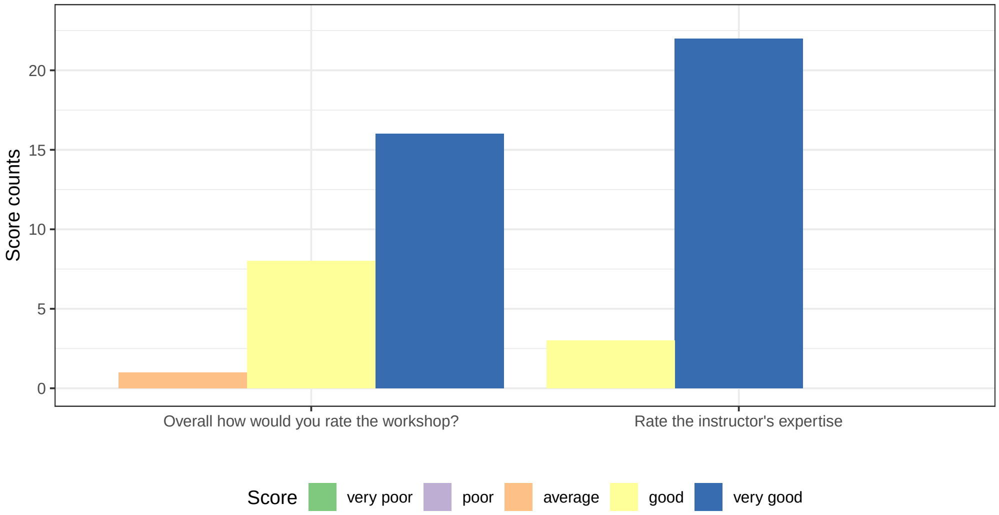
Visualise results

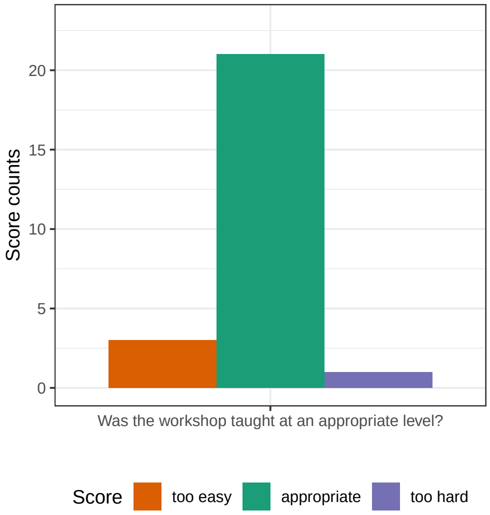
Show number of survey responses per score and question; summarise similar questions (Q2 and Q4) into one plot:

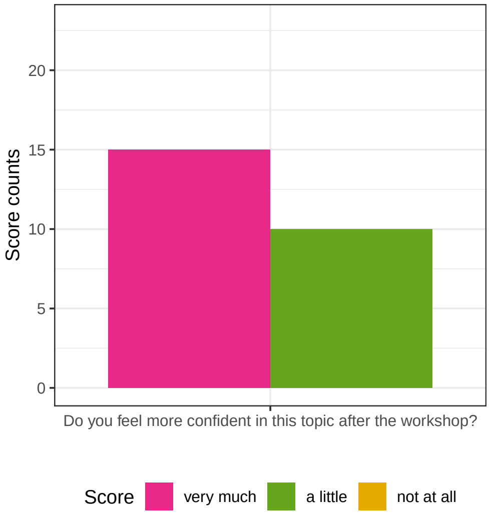
```r
q2q4_plot <- ggplot(data=q2q4,
                        aes(x=question_long, fill=factorscores))
q2q4_plot <- q2q4_plot + geom_bar(position=position_dodge(preserve = "single")) +
    scale_fill_brewer(type="qual", drop=FALSE) +
    coord_cartesian(ylim=c(0,23)) +
    labs(x="",
         y="Score counts",
         fill="Score") +
    theme_bw() +
    theme(legend.position = "bottom")

q3_plot <- ggplot(data=q3,
                        aes(x=question_long, fill=factorscores))
q3_plot <- q3_plot + geom_bar(position=position_dodge(preserve = "single")) +
    scale_fill_manual(values=c( "#d95f02", "#1b9e77","#7570b3"), drop=FALSE) +
    coord_cartesian(ylim=c(0,23)) +
    labs(x="",
         y="Score counts",
         fill="Score") +
    theme_bw() +
    theme(legend.position = "bottom")


q5_plot <- ggplot(data=q5,
                        aes(x=question_long, fill=factorscores))
q5_plot <- q5_plot + geom_bar(position=position_dodge(preserve = "single")) +
    scale_fill_manual(values=c("#e7298a","#66a61e", "#e6ab02"), drop=FALSE) +
    scale_x_discrete(labels="Do you feel more confident in this topic after the workshop?") +
    coord_cartesian(ylim=c(0,23)) +
    labs(x="",
         y="Score counts",
         fill="Score") +
    theme_bw() +
    theme(legend.position = "bottom")
```

Combine plots in plot grid:

```r
cowplot::plot_grid(q2q4_plot,
                   cowplot::plot_grid(q3_plot, q5_plot, nrow=1),
                   nrow=2,
                   align="v",
                   axis="r")
```


Score counts
0
5
10
15
20
Overall how would you rate the workshop?
Rate the instructor's expertise
Score
very poor
poor
average
good
very good


Score counts
0
5
10
15
20
Was the workshop taught at an appropriate level?
Score
too easy
appropriate
too hard


Score counts
0
5
10
15
20
Do you feel more confident in this topic after the workshop?
Score
very much
a little
not at all