

# Working a qualtrics project

John Little

## Load library packages

The key with this sample data is that the SPSS data is labeled, and the CSV data is not. The CSV data is easier to work with in R, but the labeled data is easier to work with in SPSS. The goal is to get the best of both worlds. The labeled data is imported from SPSS, and the CSV data is imported from the CSV file. The labeled data is then joined with the CSV data, and the result is a data frame that has both the labeled data and the CSV data. The labeled data is easier to use in visualizations, and the CSV data is easier to use in analysis.

```
library(tidyverse)
```

Warning: package 'readr' was built under R version 4.3.2

Warning: package 'dplyr' was built under R version 4.3.2

Warning: package 'stringr' was built under R version 4.3.2

```
-- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
v dplyr      1.1.4      v readr      2.1.5
v forcats    1.0.0      v stringr    1.5.1
v ggplot2    3.4.4      v tibble     3.2.1
v lubridate  1.9.3      v tidyr      1.3.0
v purrr      1.0.2
-- Conflicts ----- tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()     masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become
```

```
library(janitor)
```

Attaching package: 'janitor'

The following objects are masked from 'package:stats':

chisq.test, fisher.test

```
library(haven)
```

Warning: package 'haven' was built under R version 4.3.2

```
library(gt)
```

## Import data

```
my_spss_df <- read_sav("data/student-satisfaction_SPSS.sav")
my_CSV_df <- read_csv("data/student-satisfaction_CSV_use-choice-text.csv",
                      n_max = 1)
```

Rows: 1 Columns: 32

-- Column specification -----

Delimiter: ","

chr (32): StartDate, EndDate, Status, IPAddress, Progress, Duration (in seco...

i Use `spec()` to retrieve the full column specification for this data.

i Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

## Wrangle

### My text example

Clean question text. This table has the original question text, as used in Qualtrics. It can be used to merge into the data frame containing the survey response data. Scroll right to see the question text as it was originally exported in line 1 of the CSV file from Qualtrics. And here's the thing: only the CSV file has the original text. The SPSS file has the labeled data, but not

the original question text. The CSV file has the question original text, but not the labeled data. The goal is to get the best of both worlds. The labeled data is imported from SPSS, and the CSV data is imported from the CSV file. The labeled data is then joined with the CSV data, and the result is a data frame that has both the labeled data and the CSV data. The labeled data is easier to use in visualizations, and the CSV data is easier to use in analysis.

```
my_CSV_df <- my_CSV_df |>
  janitor::clean_names() |>
  pivot_longer(cols = everything(), names_to = "question", values_to = "question_text")

my_CSV_df
```

```
# A tibble: 32 x 2
  question      question_text
  <chr>         <chr>
1 start_date    Start Date
2 end_date      End Date
3 status        Response Type
4 ip_address    IP Address
5 progress      Progress
6 duration_in_seconds Duration (in seconds)
7 finished      Finished
8 recorded_date Recorded Date
9 response_id   Response ID
10 recipient_last_name Recipient Last Name
# i 22 more rows
```

This table contains both the numeric response and the labeled categorical factors. The labeled factors are easier to use in visualizations. The numeric response are sometimes easier to use in analysis.

```
my_labeled_spss_df <- my_spss_df |>
  clean_names() |>
  select(start_date, response_id, starts_with("q")) |>
  # mutate(across(is.labelled, ~ as_factor(.x), .names = "{.col}_label")) |>
  mutate(across(is.labelled, ~ as_factor(.x))) |>
  # pivot_longer(cols = ends_with("_label"), names_to = "question", values_to = "response")
  pivot_longer(cols = starts_with("q"), names_to = "question", values_to = "response") |>
  select(start_date, response_id, question, response)
```

Warning: There was 1 warning in `mutate()`.

```
i In argument: `across(is.labelled, ~as_factor(.x))`.
Caused by warning:
! Use of bare predicate functions was deprecated in tidysselect 1.1.0.
i Please use wrap predicates in `where()` instead.
# Was:
data %>% select(is.labelled)

# Now:
data %>% select(where(is.labelled))
```

```
my_labeled_spss_df
```

```
# A tibble: 375 x 4
  start_date      response_id      question response
  <dtm>          <chr>          <chr>      <fct>
1 2023-01-21 08:50:45 R_e3ejzhVVTZuvmK0 q1      Slightly unhelpful
2 2023-01-21 08:50:45 R_e3ejzhVVTZuvmK0 q2      Slightly easy
3 2023-01-21 08:50:45 R_e3ejzhVVTZuvmK0 q3      Not useful at all
4 2023-01-21 08:50:45 R_e3ejzhVVTZuvmK0 q4      Slightly helpful
5 2023-01-21 08:50:45 R_e3ejzhVVTZuvmK0 q5      Extremely well
6 2023-01-21 08:50:45 R_e3ejzhVVTZuvmK0 q6      Moderately unsafe
7 2023-01-21 08:50:45 R_e3ejzhVVTZuvmK0 q7      Extremely good
8 2023-01-21 08:50:45 R_e3ejzhVVTZuvmK0 q8      Very Satisfied
9 2023-01-21 08:50:45 R_e3ejzhVVTZuvmK0 q9      Moderately well
10 2023-01-21 08:50:45 R_e3ejzhVVTZuvmK0 q10     Neither easy nor difficult
# i 365 more rows
```

```
my_numeric_spss_df <- my_spss_df |>
  clean_names() |>
  select(start_date, response_id, starts_with("q")) |>
  mutate(across(is.labelled, ~ as.numeric(.x))) |>
  pivot_longer(cols = starts_with("q"), names_to = "question", values_to = "response")
my_numeric_spss_df
```

```
# A tibble: 375 x 4
  start_date      response_id      question response
  <dtm>          <chr>          <chr>      <dbl>
1 2023-01-21 08:50:45 R_e3ejzhVVTZuvmK0 q1          5
2 2023-01-21 08:50:45 R_e3ejzhVVTZuvmK0 q2          3
3 2023-01-21 08:50:45 R_e3ejzhVVTZuvmK0 q3          5
4 2023-01-21 08:50:45 R_e3ejzhVVTZuvmK0 q4          3
```

```

5 2023-01-21 08:50:45 R_e3ejzhVVTZuvmK0 q5 1
6 2023-01-21 08:50:45 R_e3ejzhVVTZuvmK0 q6 6
7 2023-01-21 08:50:45 R_e3ejzhVVTZuvmK0 q7 1
8 2023-01-21 08:50:45 R_e3ejzhVVTZuvmK0 q8 7
9 2023-01-21 08:50:45 R_e3ejzhVVTZuvmK0 q9 3
10 2023-01-21 08:50:45 R_e3ejzhVVTZuvmK0 q10 4
# i 365 more rows

```

join question with question text

```

joined_df <- my_labeled_spss_df |>
  left_join(my_CSV_df, by = "question")

joined_df

```

```

# A tibble: 375 x 5
  start_date      response_id question response      question_text
  <dtm>          <chr>      <chr>    <fct>      <chr>
1 2023-01-21 08:50:45 R_e3ejzhVVTZuvmK0 q1      Slightly unhelp~ "Thank you f~
2 2023-01-21 08:50:45 R_e3ejzhVVTZuvmK0 q2      Slightly easy   "How easy or~
3 2023-01-21 08:50:45 R_e3ejzhVVTZuvmK0 q3      Not useful at a~ "How useful ~
4 2023-01-21 08:50:45 R_e3ejzhVVTZuvmK0 q4      Slightly helpful "How helpful~
5 2023-01-21 08:50:45 R_e3ejzhVVTZuvmK0 q5      Extremely well  "How well ma~
6 2023-01-21 08:50:45 R_e3ejzhVVTZuvmK0 q6      Moderately unsa~ "How safe or~
7 2023-01-21 08:50:45 R_e3ejzhVVTZuvmK0 q7      Extremely good  "How good or~
8 2023-01-21 08:50:45 R_e3ejzhVVTZuvmK0 q8      Very Satisfied  "How satisfi~
9 2023-01-21 08:50:45 R_e3ejzhVVTZuvmK0 q9      Moderately well "Overall, ho~
10 2023-01-21 08:50:45 R_e3ejzhVVTZuvmK0 q10     Neither easy no~ "How easy or~
# i 365 more rows

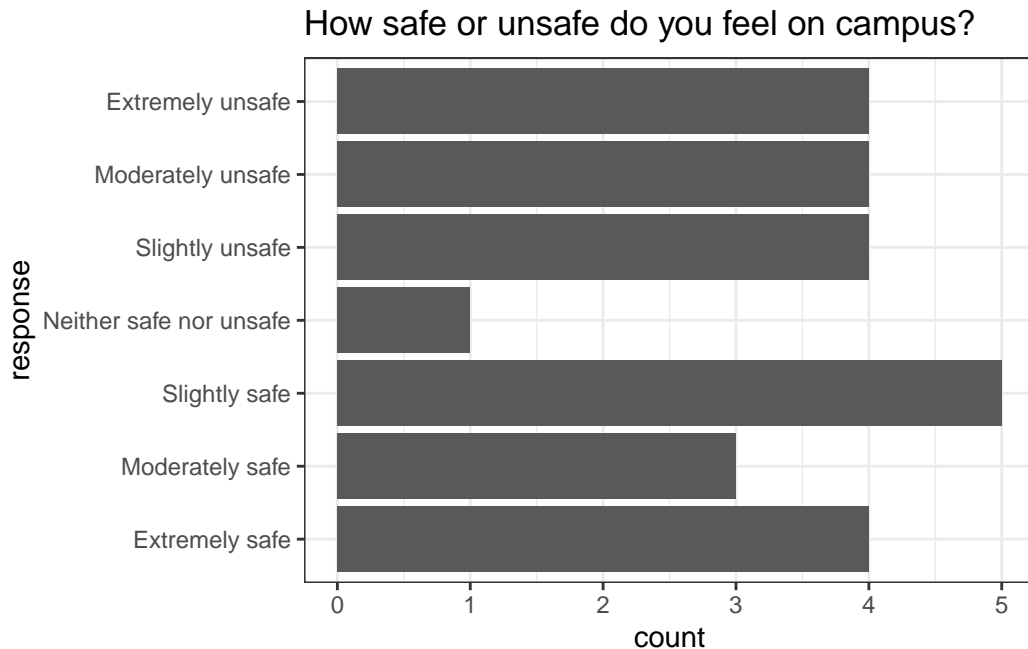
```

**viz**

```

joined_df |>
  filter(question == "q6") |>
  ggplot(aes(x = response)) +
  geom_bar() +
  coord_flip() +
  labs(title = joined_df |> filter(question == "q6") |> distinct(question_text) |> pull(question_text))
  theme_bw()

```



Iterate for all the questions

```
make_my_plots_df <- joined_df |>
  nest(data = -c(question, question_text)) |>
  mutate(my_barplot = map2(data, question_text, \(my_x, my_title) ggplot(my_x, aes(x = response)) +
    geom_bar() +
    coord_flip() +
    labs(title = my_title) +
    theme_bw())
make_my_plots_df
```

# A tibble: 15 x 4

	question <chr>	question_text <chr>	data <list>	my_barplot <list>
1	q1	"Thank you for participating in this stude~	<tibble>	<gg>
2	q2	"How easy or difficult is it to obtain the~	<tibble>	<gg>
3	q3	"How useful are the services provided at t~	<tibble>	<gg>
4	q4	"How helpful or unhelpful is the staff at ~	<tibble>	<gg>
5	q5	"How well maintained are the facilities at~	<tibble>	<gg>
6	q6	"How safe or unsafe do you feel on campus?"	<tibble>	<gg>
7	q7	"How good or bad is the quality of the foo~	<tibble>	<gg>
8	q8	"How satisfied or dissatisfied were you wi~	<tibble>	<gg>
9	q9	"Overall, how well do the professors at th~	<tibble>	<gg>

```

10 q10          "How easy or difficult is it to register f~ <tibble> <gg>
11 q12          "How reasonable or unreasonable is the cos~ <tibble> <gg>
12 q13          "Overall, how satisfied or dissatisfied ar~ <tibble> <gg>
13 q14          "How likely are you to attend this univers~ <tibble> <gg>
14 q15_nps_group "How likely are you to recommend this univ~ <tibble> <gg>
15 q15          "How likely are you to recommend this univ~ <tibble> <gg>

```

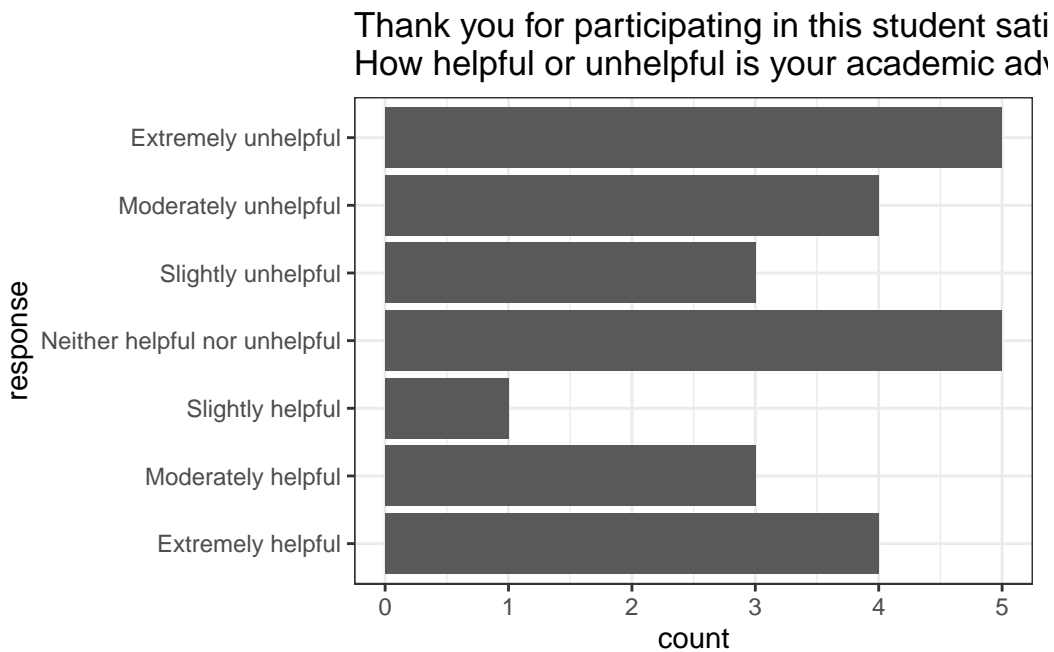
See the plots made in the previous example. Notice, although it is hard to see in the last plot, each plot has the labeled responses. Those labels are in order of the likkert scale because the data were downloaded as spss data and then wrangled as labelled categorical factors in R.

```

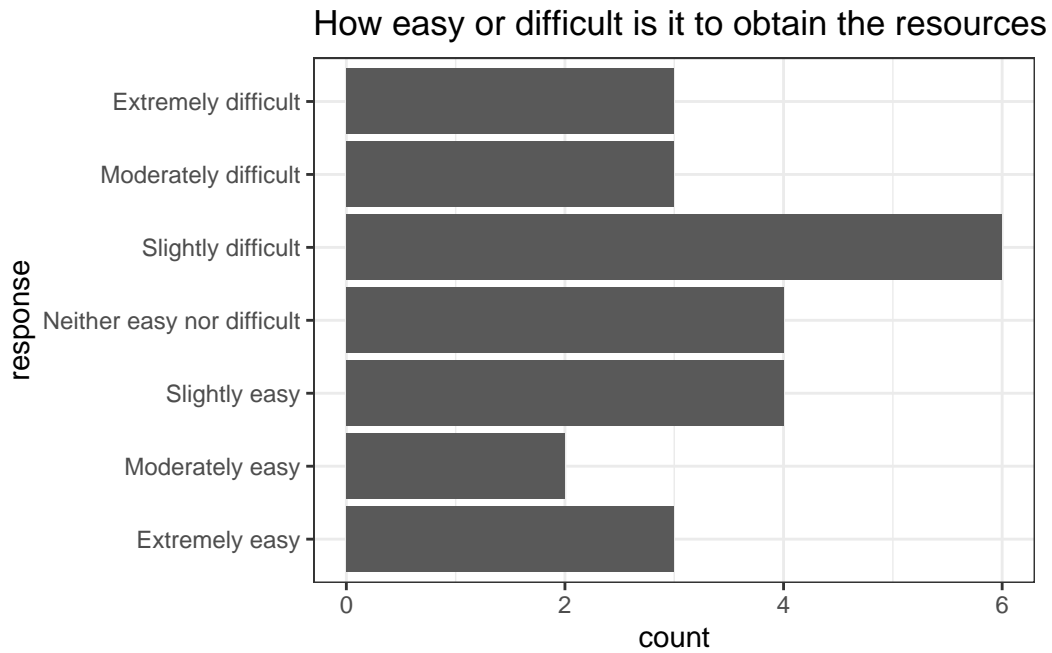
make_my_plots_df |>
  pull(my_barplot)

```

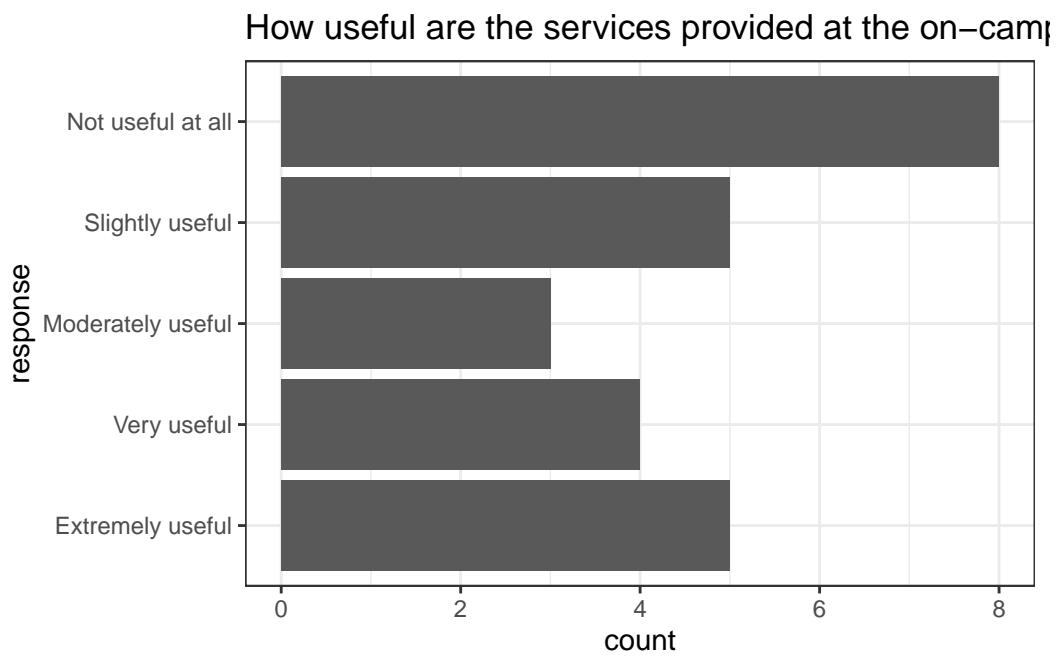
```
[[1]]
```



```
[[2]]
```

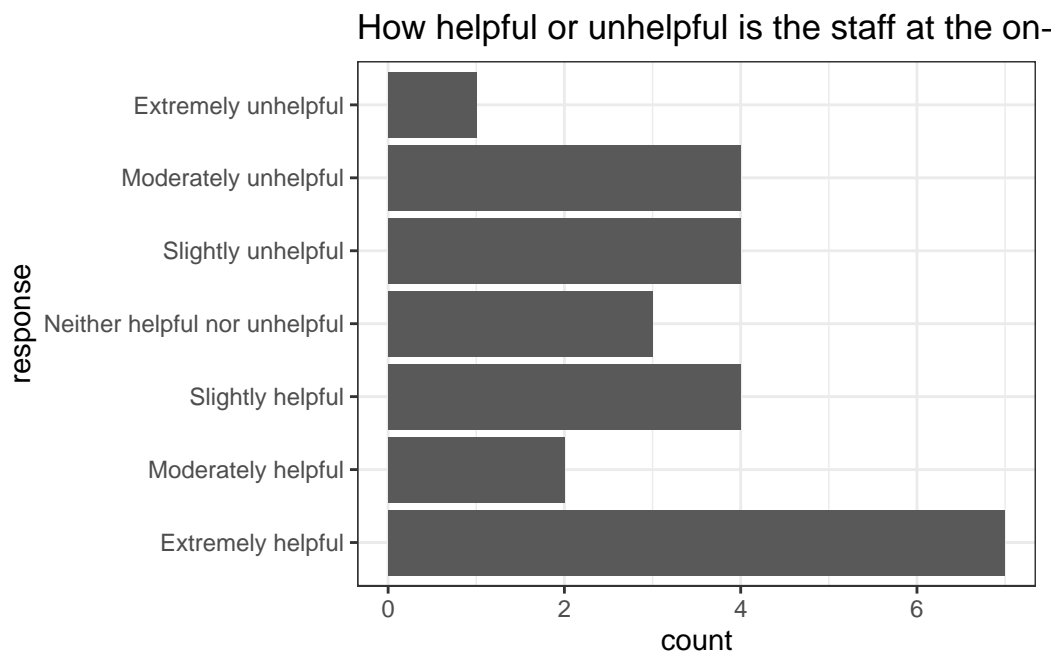


[[3]]

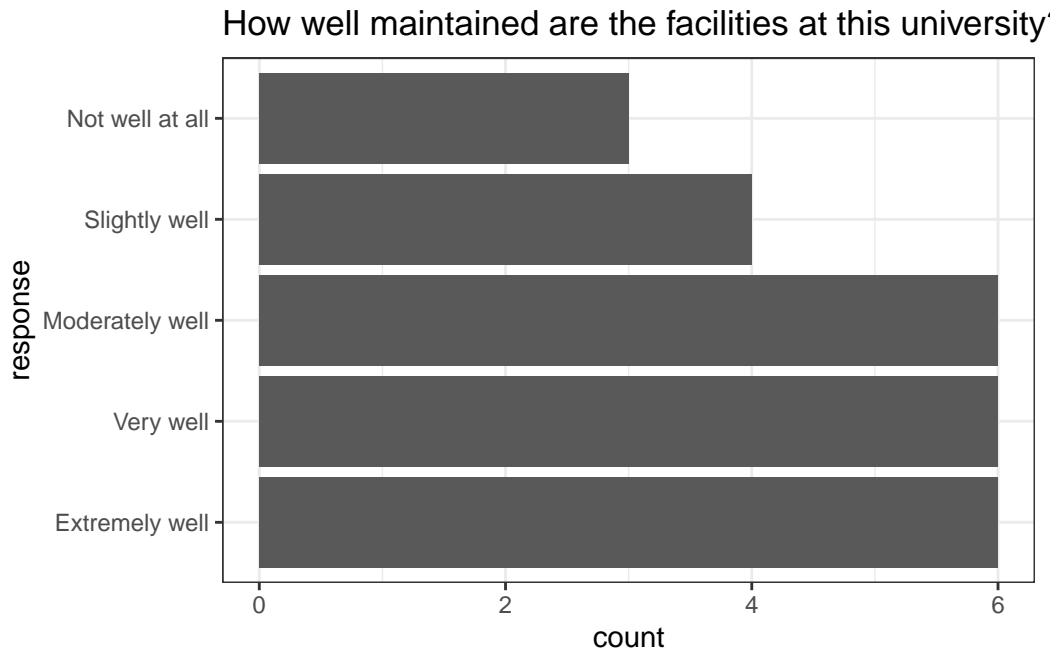




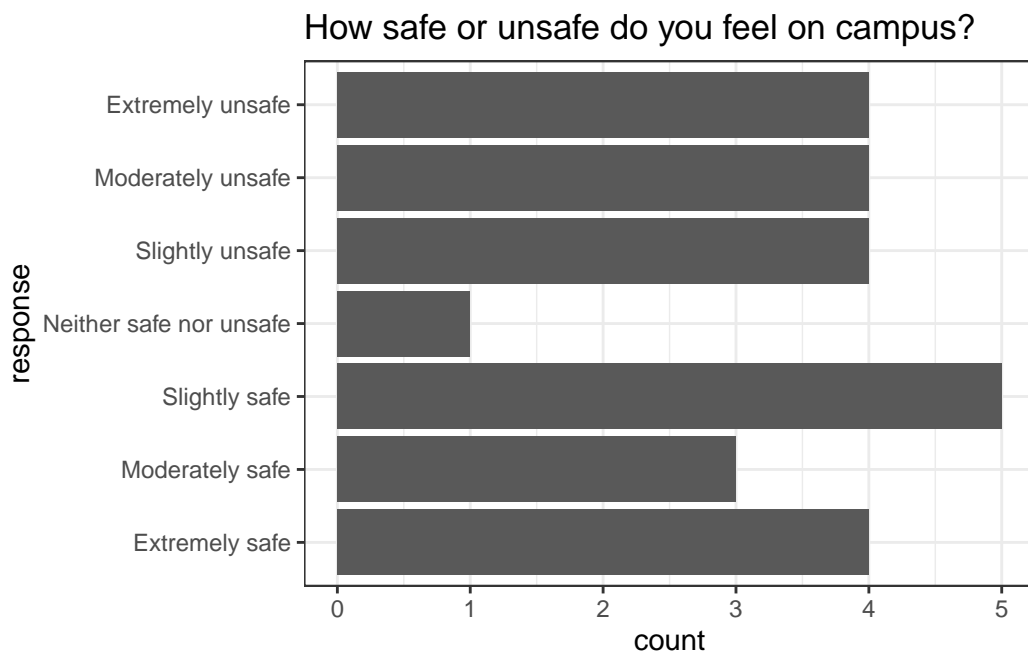
```
[[4]]
```



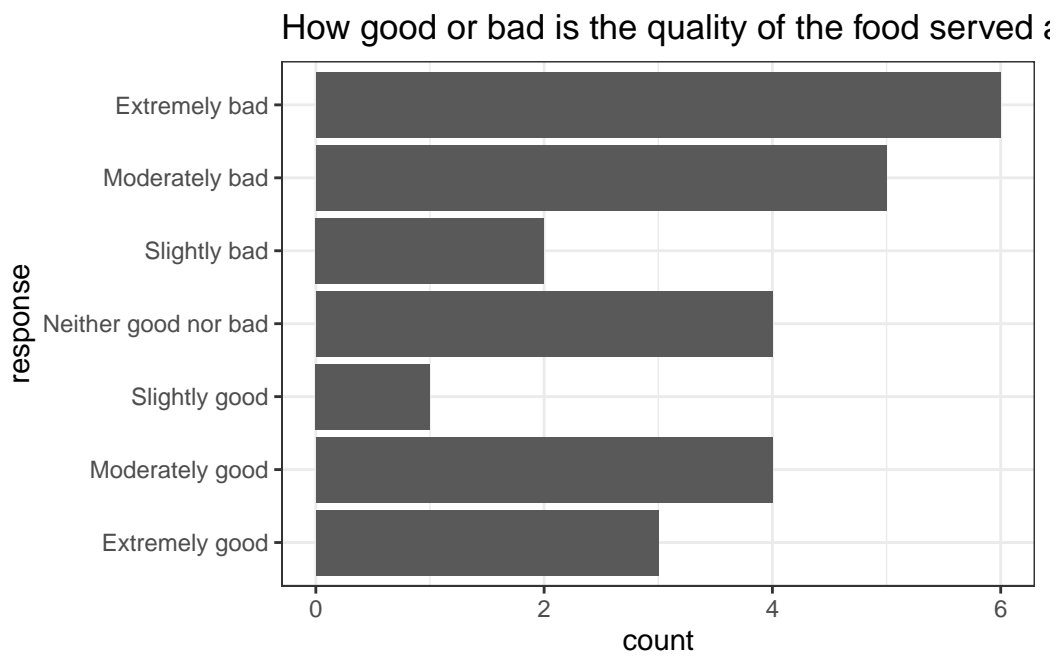
```
[[5]]
```



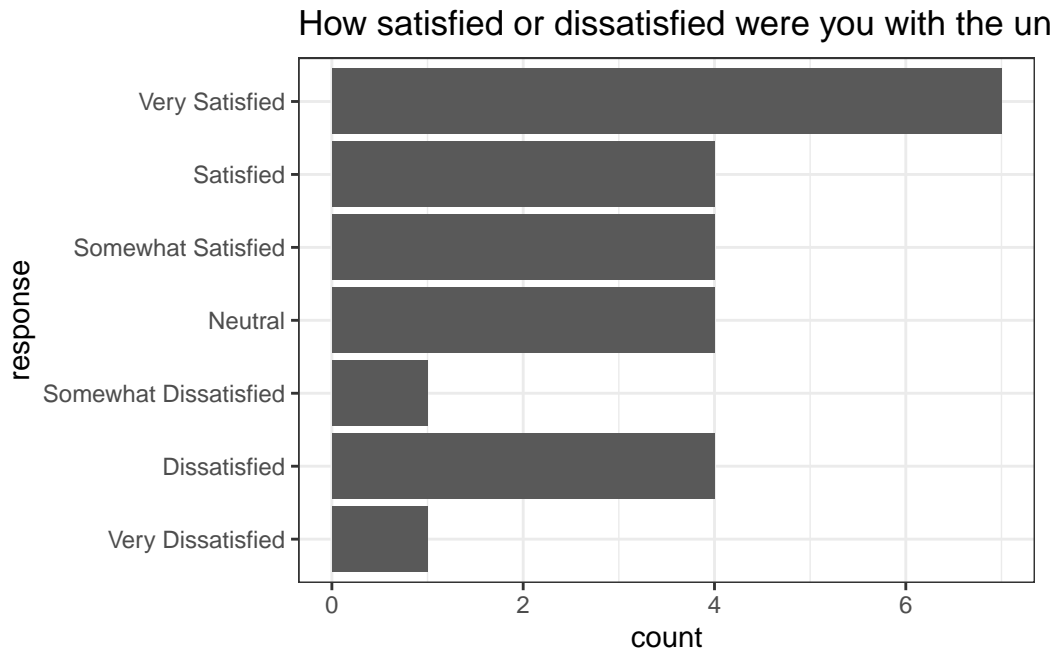
[[6]]



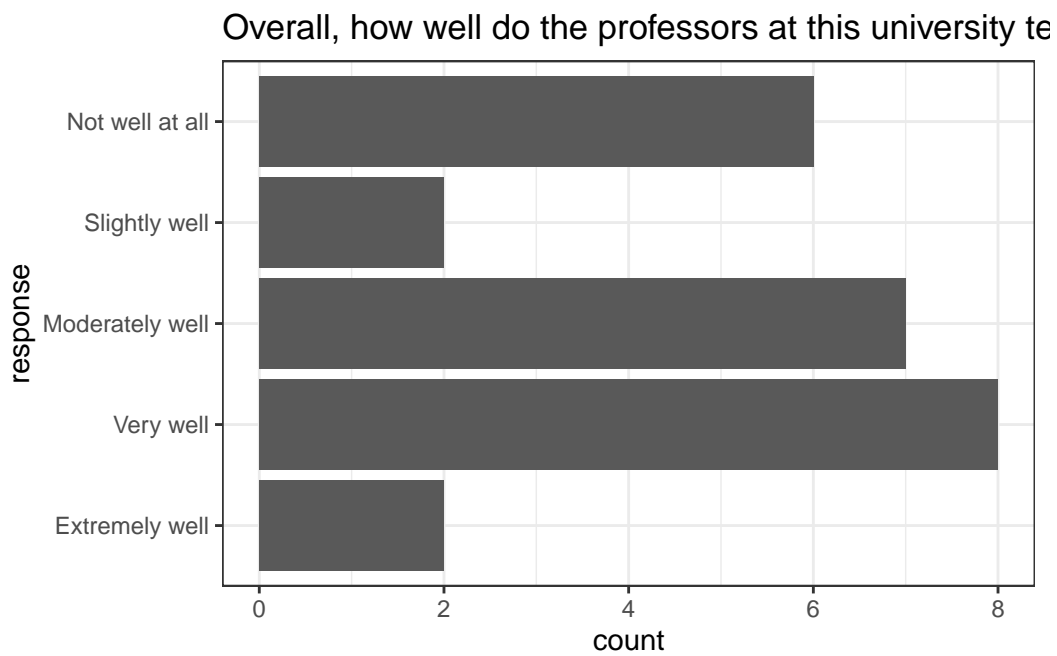
[[7]]



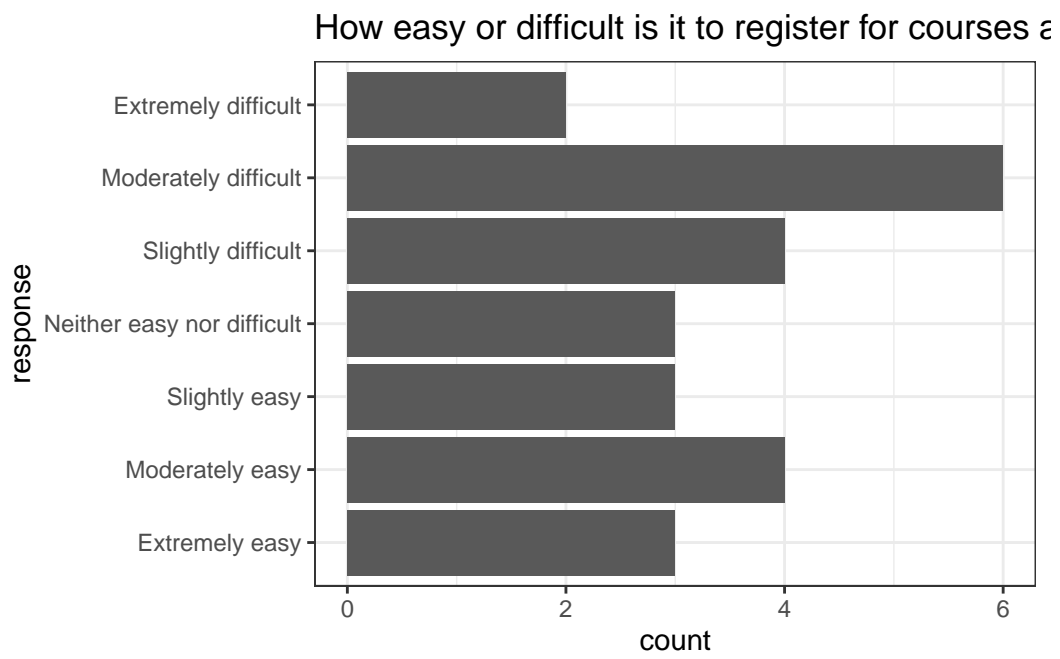
[[8]]



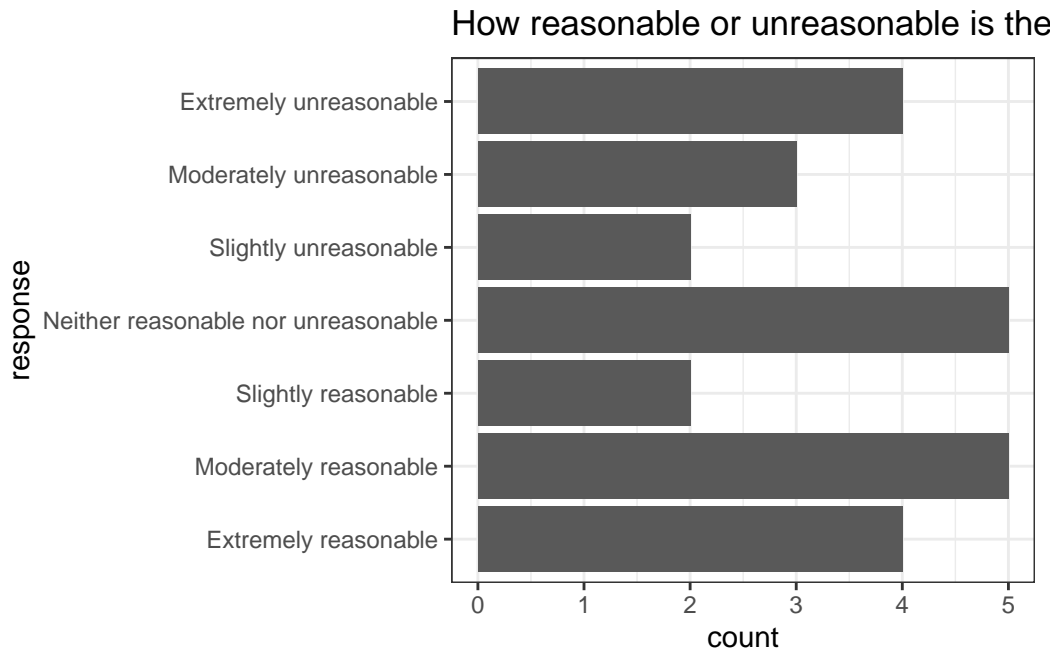
[[9]]



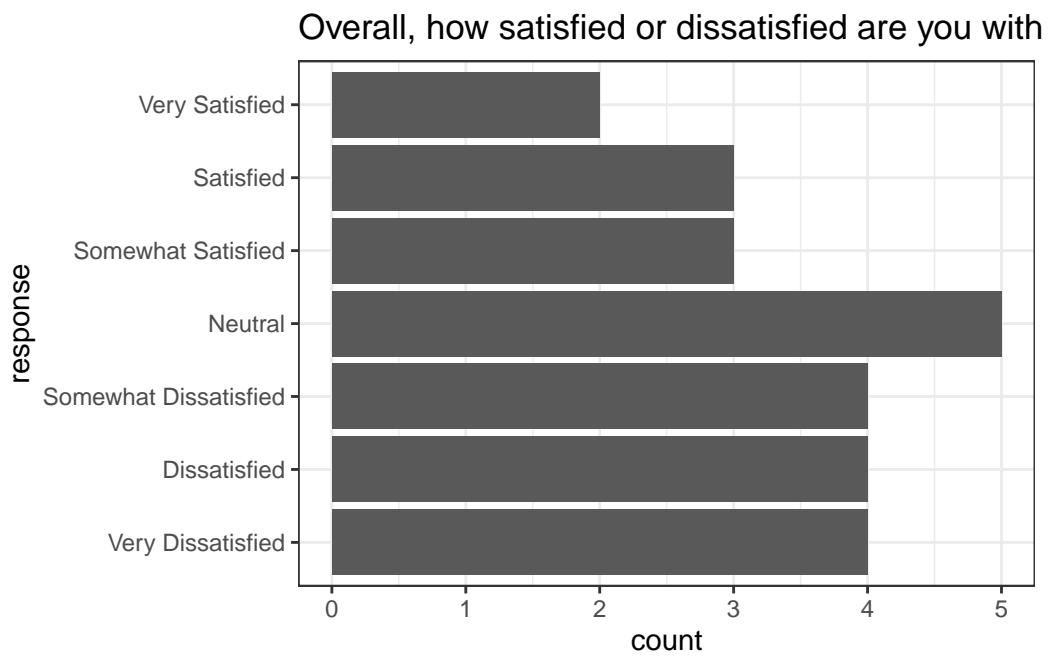
```
[[10]]
```



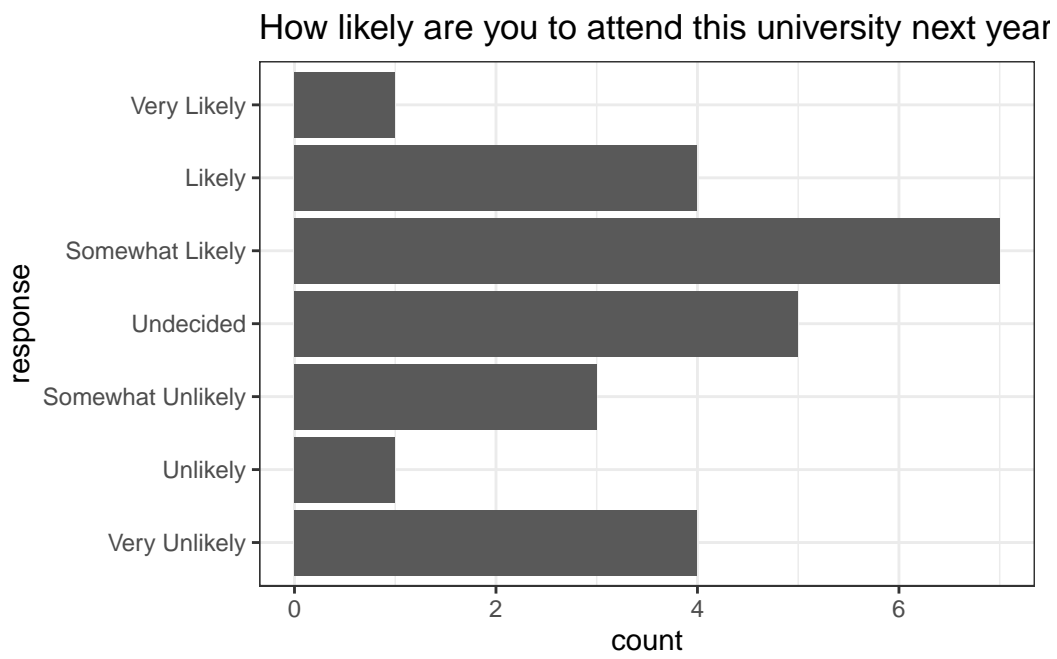
```
[[11]]
```



[[12]]

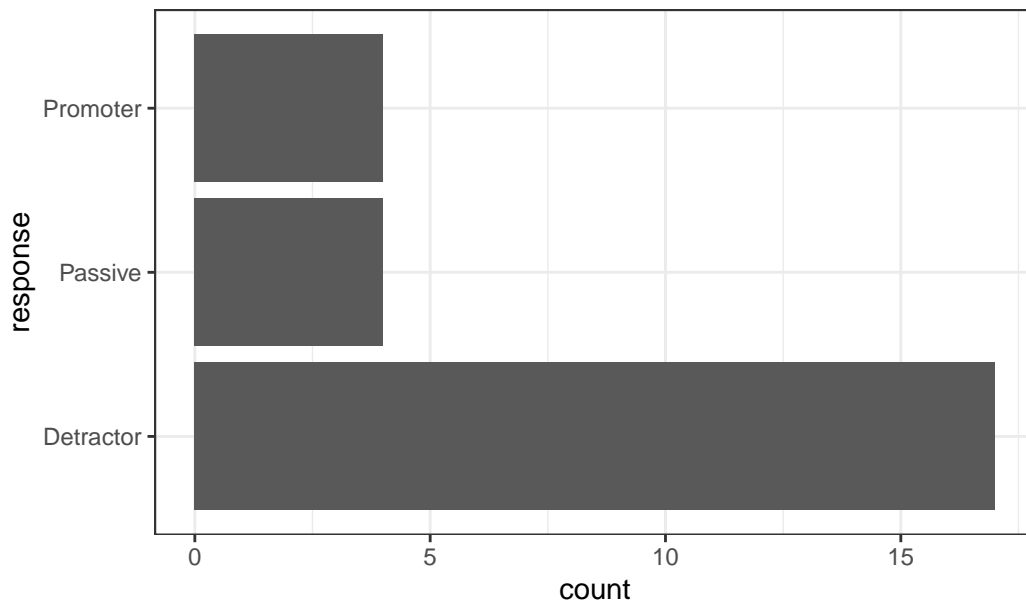


```
[[13]]
```



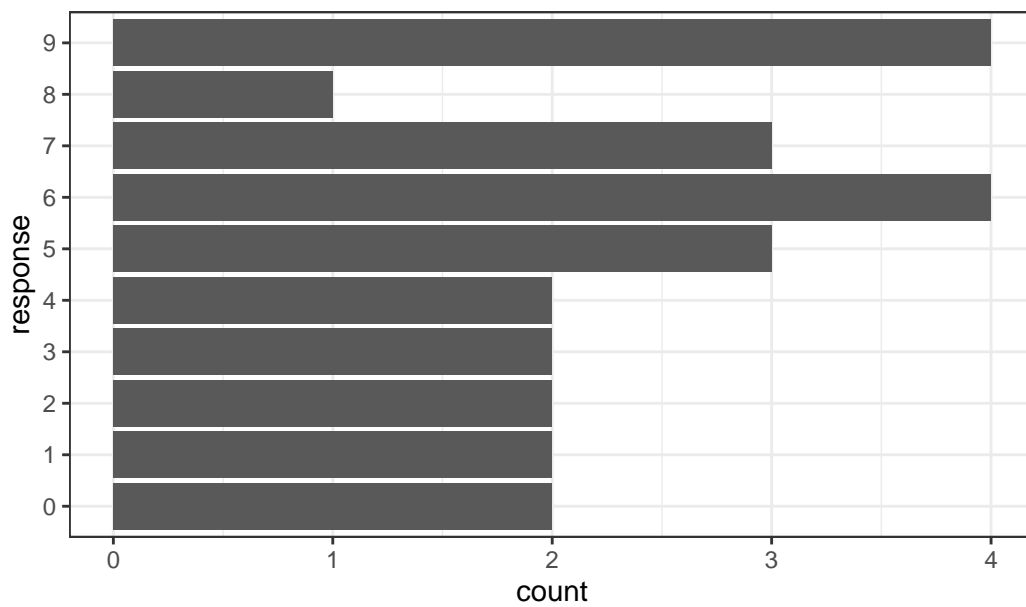
```
[[14]]
```

How likely are you to recommend this university to friends or colleagues?



[[15]]

How likely are you to recommend this university to friends or colleagues?





## Figure out percentages

```
joined_df |>
  filter(question == "q6") |>
  count(response)
```

```
# A tibble: 7 x 2
  response      n
  <fct>      <int>
1 Extremely safe      4
2 Moderately safe      3
3 Slightly safe       5
4 Neither safe nor unsafe 1
5 Slightly unsafe      4
6 Moderately unsafe      4
7 Extremely unsafe      4
```

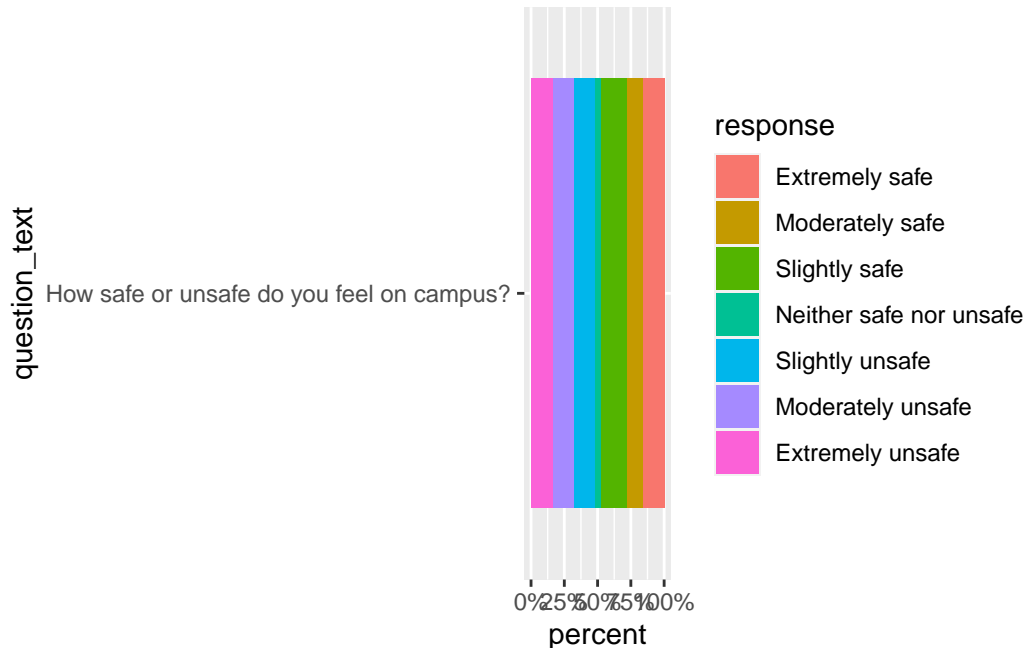
Determine percentage of response

```
joined_df |>
  filter(question == "q6") |>
  count(question_text, response) |>
  mutate(percent = n / sum(n))
```

```
# A tibble: 7 x 4
  question_text      response      n percent
  <chr>            <fct>    <int>   <dbl>
1 How safe or unsafe do you feel on campus? Extremely safe      4    0.16
2 How safe or unsafe do you feel on campus? Moderately safe      3    0.12
3 How safe or unsafe do you feel on campus? Slightly safe       5    0.2
4 How safe or unsafe do you feel on campus? Neither safe nor unsa~ 1    0.04
5 How safe or unsafe do you feel on campus? Slightly unsafe      4    0.16
6 How safe or unsafe do you feel on campus? Moderately unsafe      4    0.16
7 How safe or unsafe do you feel on campus? Extremely unsafe      4    0.16
```

visualize percentage

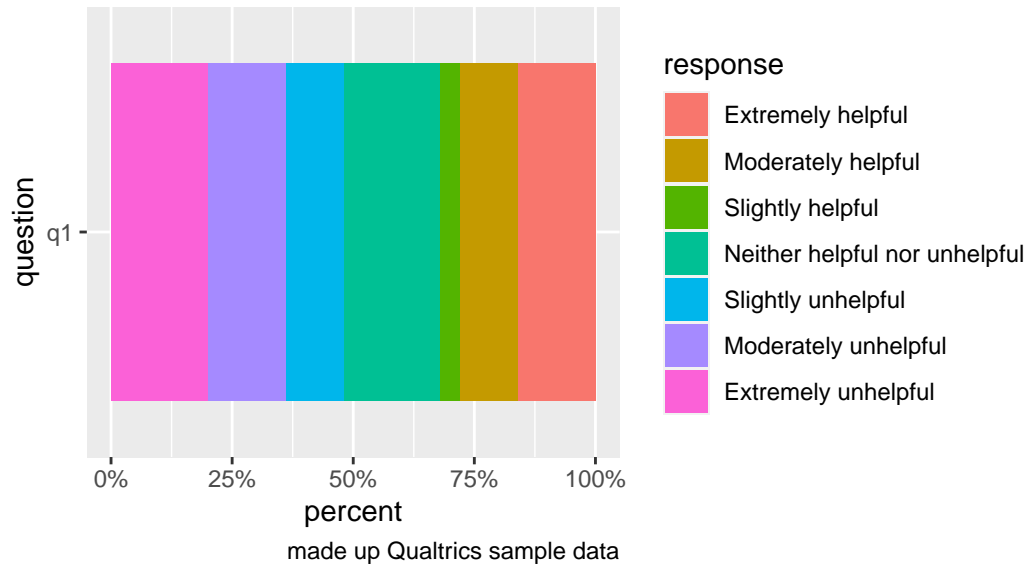
```
joined_df |>
  filter(question == "q6") |>
  count(question_text, response) |>
  mutate(percent = n / sum(n)) |>
  ggplot(aes(x = question_text, y = percent, fill = response)) +
  geom_col() +
  coord_flip() +
  scale_y_continuous(labels = scales::percent_format())
```



```
joined_df |>
  nest(data = -c(question_text)) |>
  mutate(my_percentages = map2(data, question_text, \(mydata, mytitle)
    mydata |> count(question, response) |>
    mutate(percent = n / sum(n)) |>
    ggplot(aes(x = question, y = percent, fill = response)) +
    geom_col() +
    coord_flip() +
    scale_y_continuous(labels = scales::percent_format()) +
    labs(title = mytitle,
          caption = "made up Qualtrics sample data")) |>
  pull(my_percentages)
```

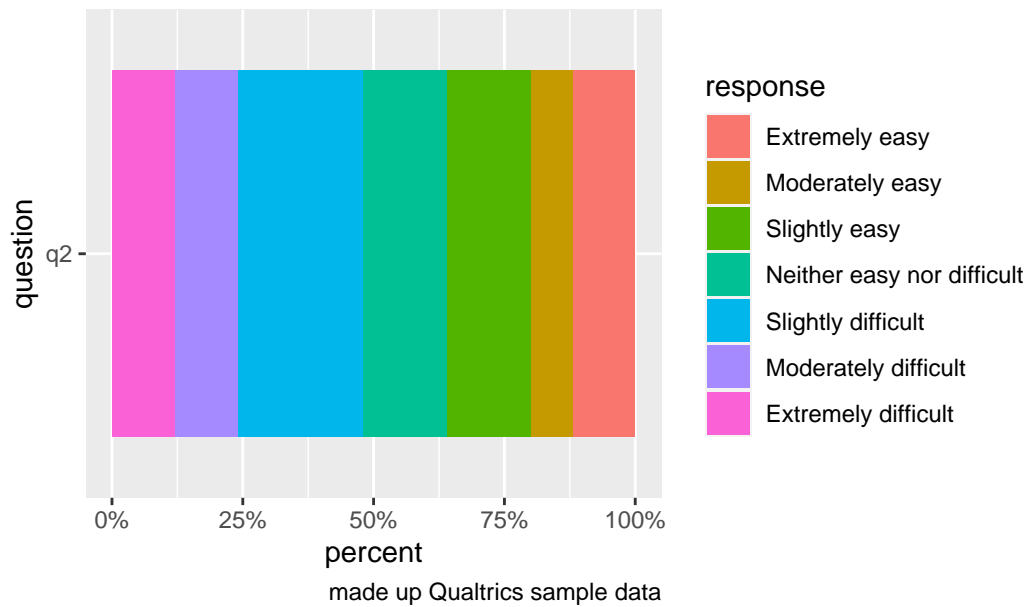
[[1]]

Thank you for participating in this student satisfaction survey. Yr  
How helpful or unhelpful is your academic advisor?



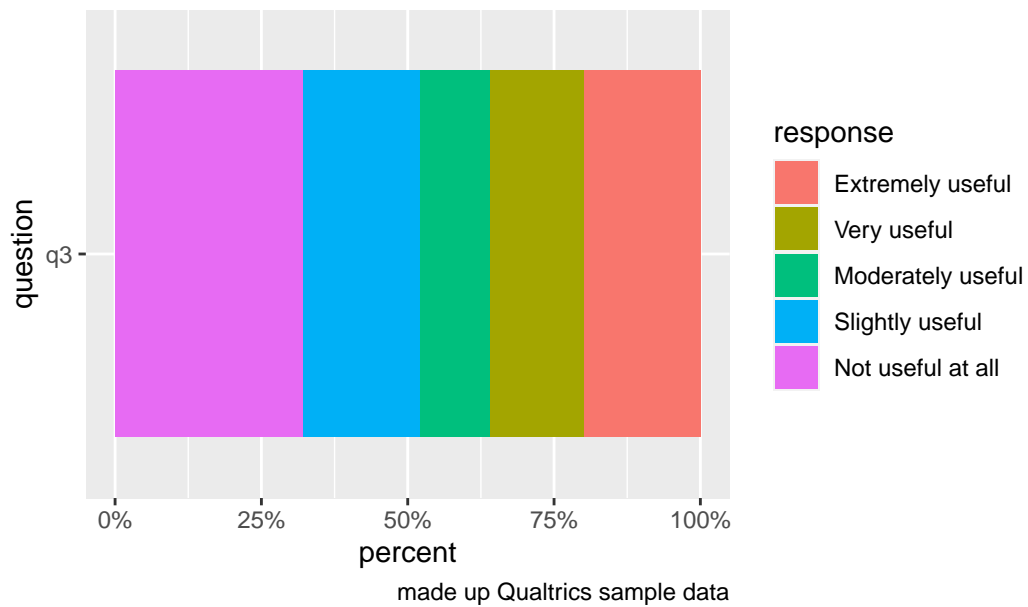
[[2]]

How easy or difficult is it to obtain the resources that you need f



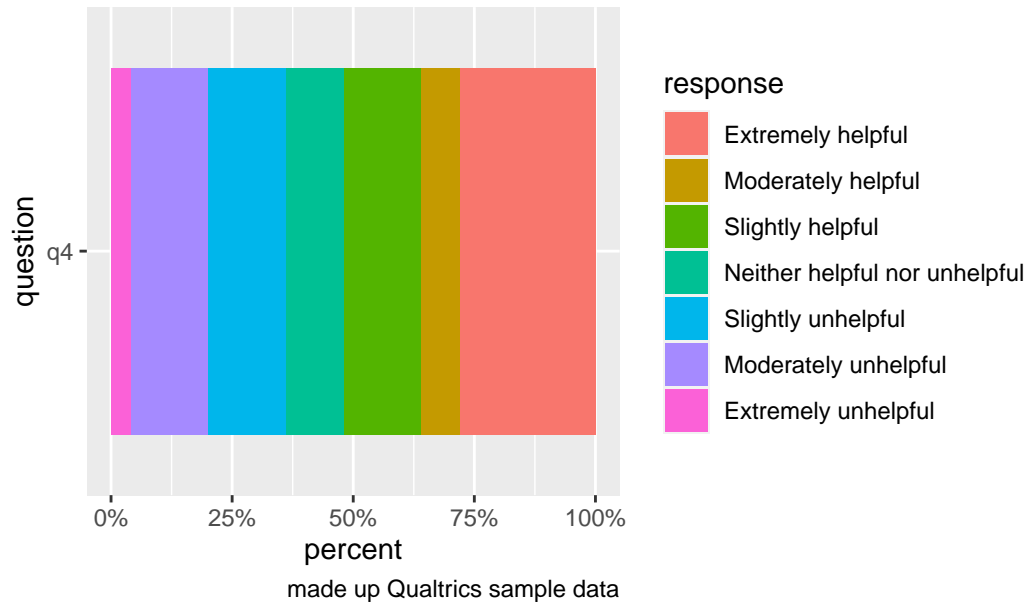
[[3]]

How useful are the services provided at the on-campus career



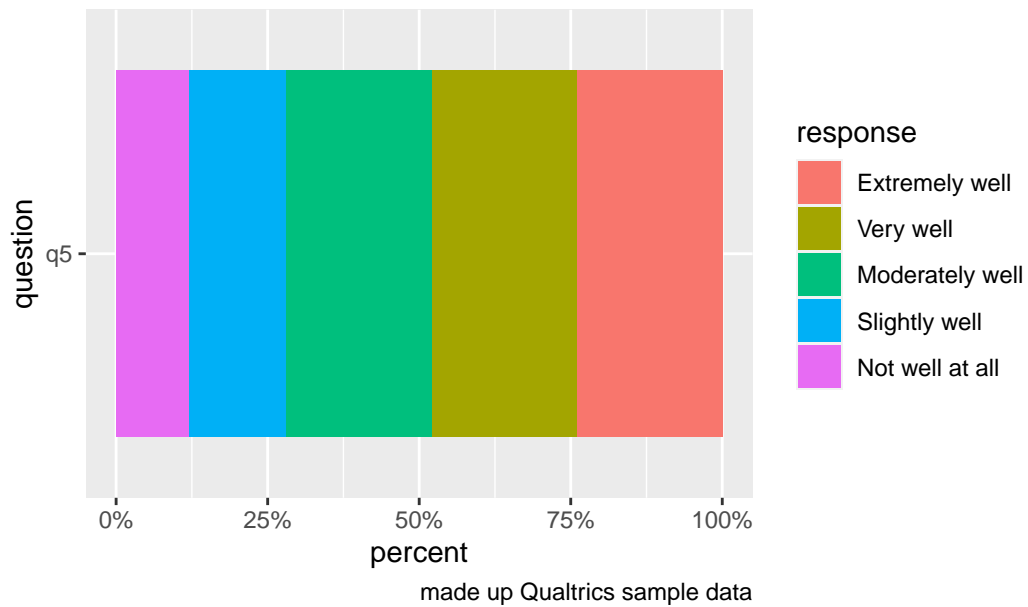
[[4]]

How helpful or unhelpful is the staff at the on-campus health ce



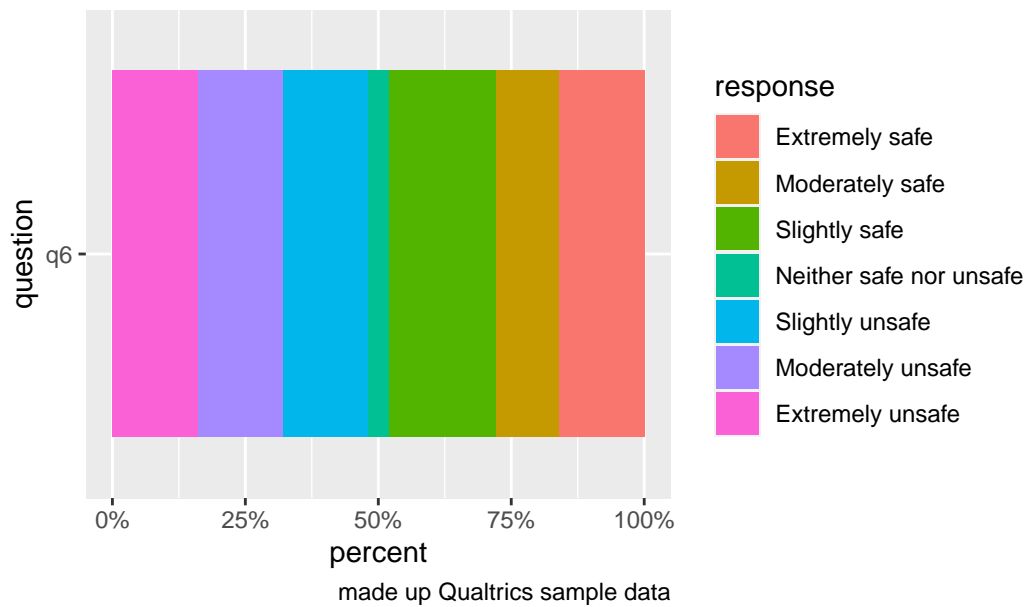
[[5]]

How well maintained are the facilities at this university?



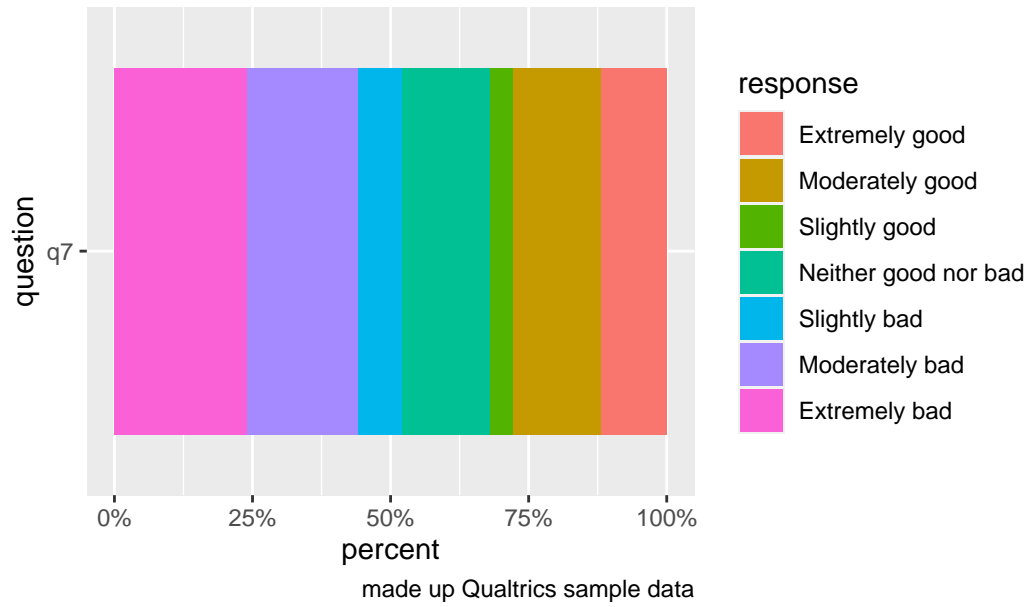
[[6]]

How safe or unsafe do you feel on campus?



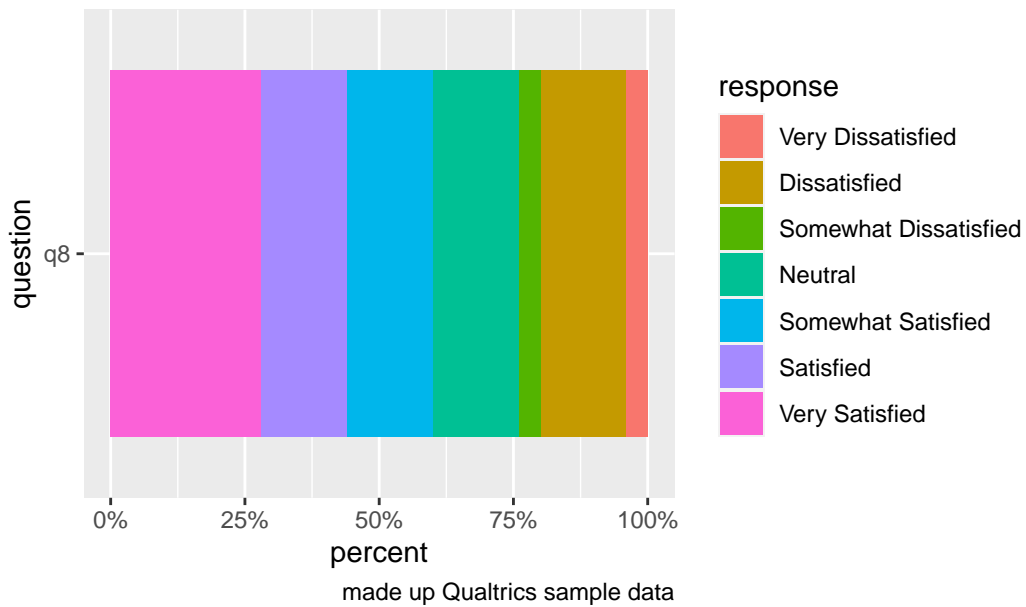
[[7]]

How good or bad is the quality of the food served at this univers



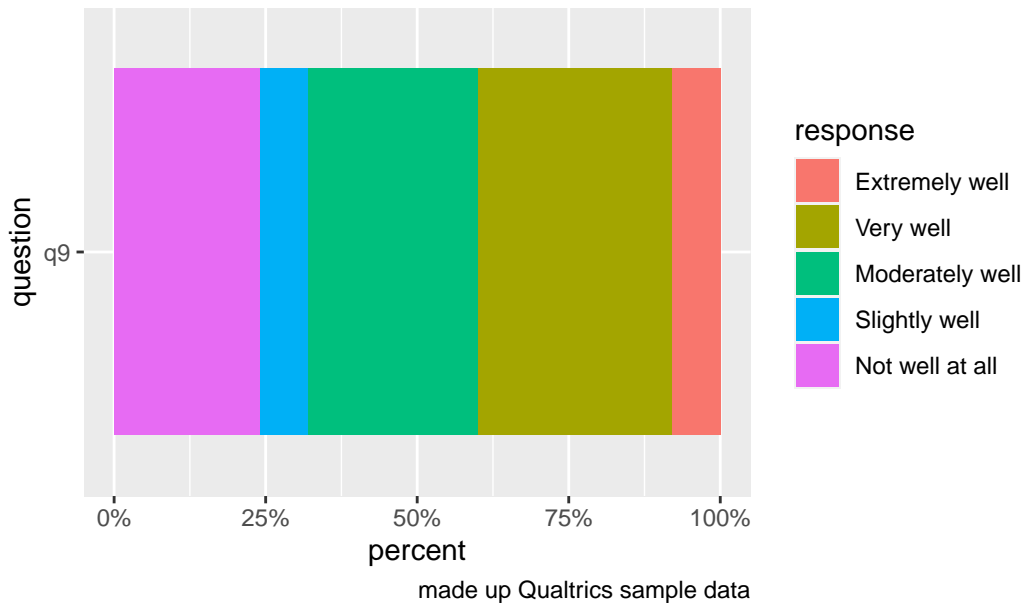
[[8]]

How satisfied or dissatisfied were you with the university sponsi



[[9]]

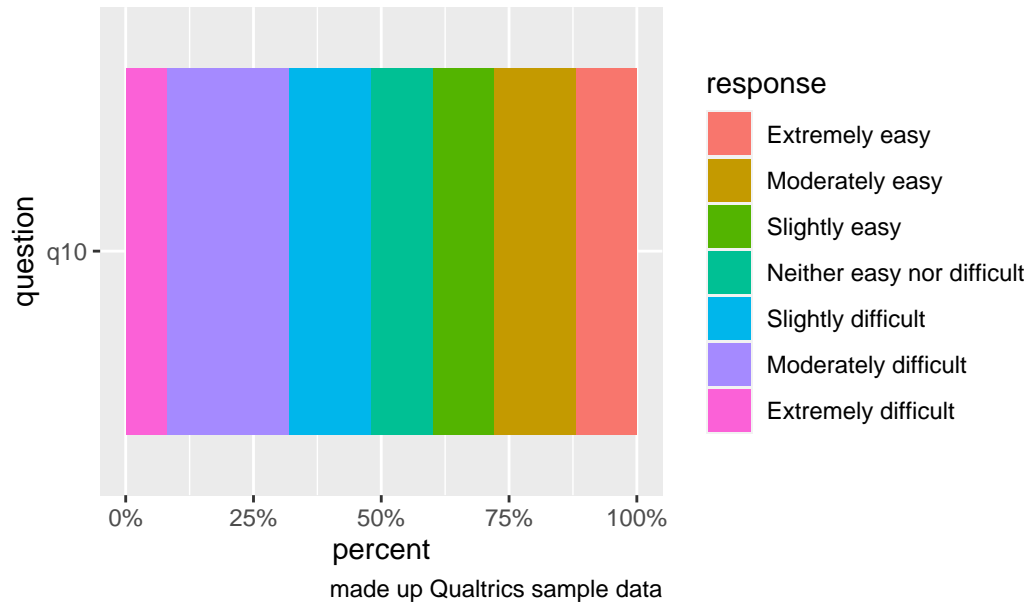
Overall, how well do the professors at this university teach?





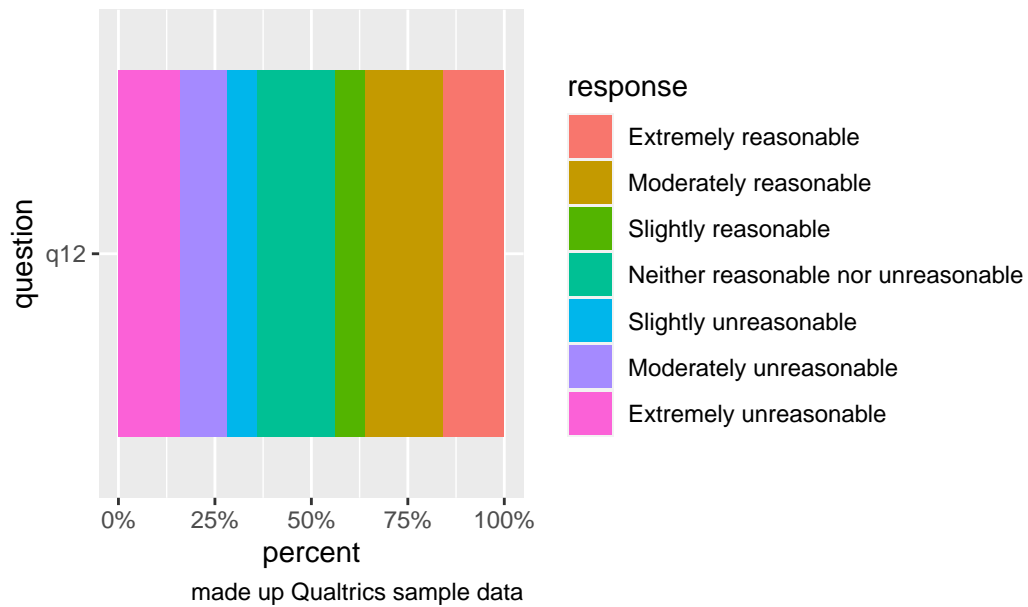
```
[[10]]
```

How easy or difficult is it to register for courses at this universit



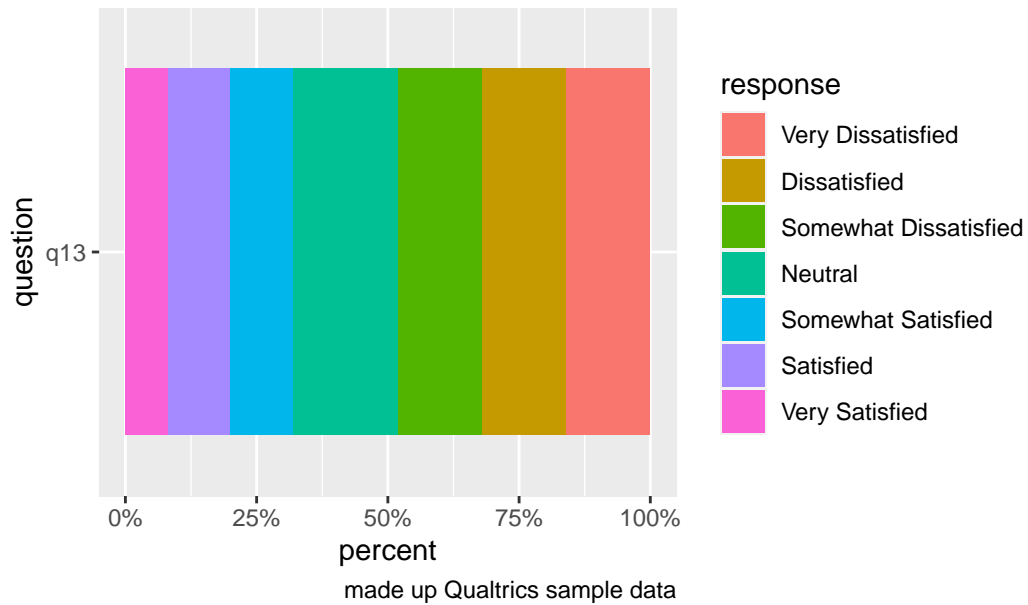
```
[[11]]
```

### How reasonable or unreasonable is the cost of courses/tuition

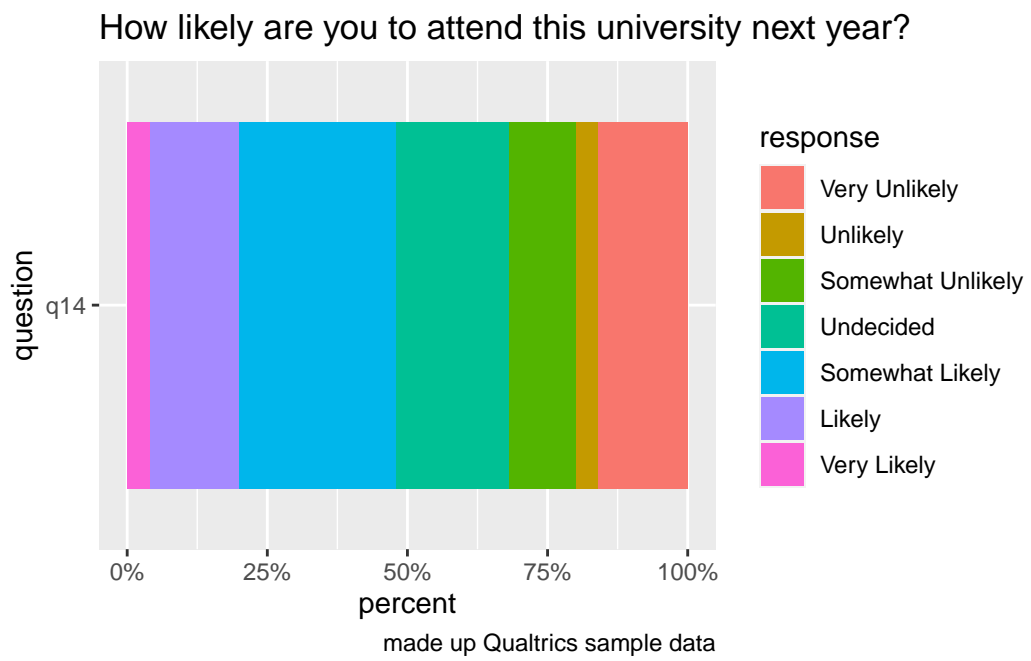


[[12]]

### Overall, how satisfied or dissatisfied are you with your experier

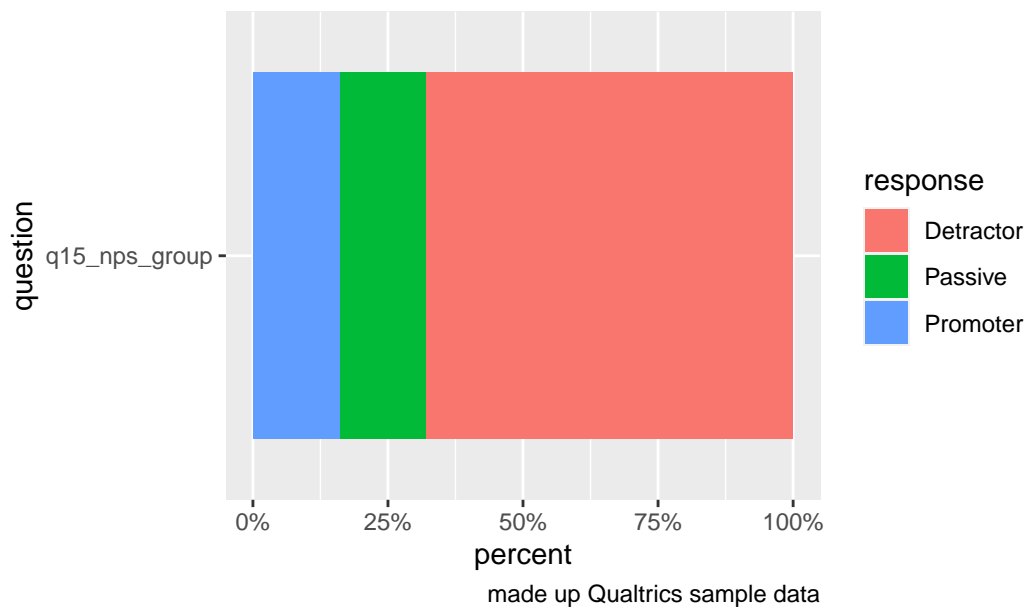


```
[[13]]
```



```
[[14]]
```

How likely are you to recommend this university to frie



[[15]]

How likely are you to recommend this university to friends or c

