

Likert analysis

Contents

Packages	1
Dataset	2
cleaning the date	2
.	3
EDA	3
Ohip by age and gender	4
OHIP	5
OHIP by gender	5
Regression analysis	7
OHIP by age	8
Analysis each question	9
Likert plot with sjPlot	11
Men and Women Likert	13
Data cleaning, data wrangling and Likert analysis	

Packages

```
pacman::p_load(janitor,  
               tidyverse,  
               gtsummary,  
               patchwork,  
               reporter,  
               sjPlot,  
               scales,  
               likert)
```

```
theme_set(theme_minimal())
```

Dataset

```
df <- read_csv("https://docs.google.com/spreadsheets/d/e/2PACX-1vS8G_1W0roX3Bj1FSI-aV7qcMc3bb4osbttxy-g")
```

```
## Rows: 132 Columns: 33
```

```
## -- Column specification -----
```

```
## Delimiter: ","
```

```
## chr  (15): Gender, Difficulty pronouncing words, Diminished sense of taste, ...
```

```
## dbl  (17): Count, Birth year, Difficulty pronouncing words Weighted, Diminis...
```

```
## date  (1): Datum
```

```
##
```

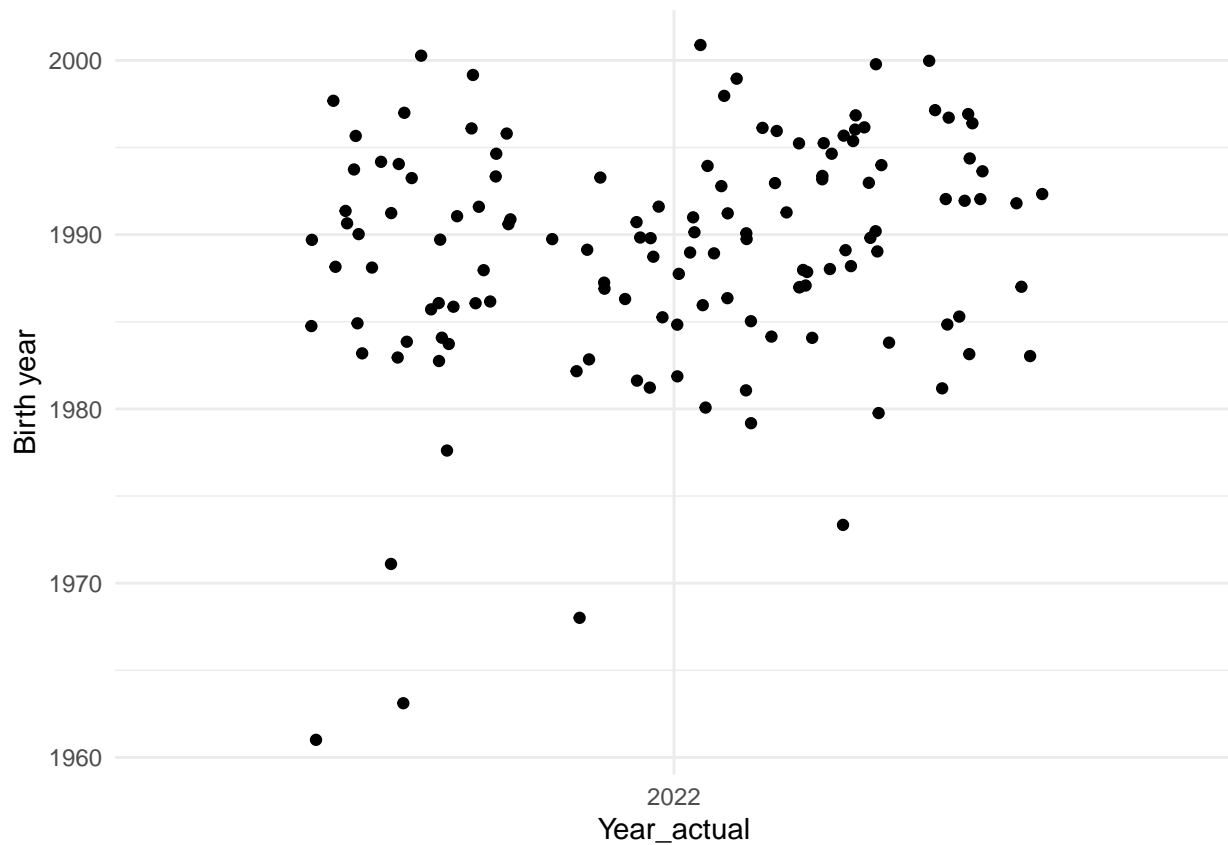
```
## i Use 'spec()' to retrieve the full column specification for this data.
```

```
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

Create a new column with the year and calculate the age

cleaning the date

```
df %>%  
  mutate(Year_actual = format(as.Date(Datum, format="%d/%m/%Y"), "%Y")) %>%  
  relocate(Year_actual, .after = Datum) %>%  
  ggplot(aes(x = Year_actual,  
             y = `Birth year`)) +  
  geom_jitter()
```



Clean the age

```
df <- df %>%
  mutate(Year_actual = format(as.Date(Datum, format="%d/%m/%Y"), "%Y")) %>%
  relocate(Year_actual, .after = Datum) %>%
  # Convert to int
  mutate(`Birth year` = as.integer(`Birth year`)) %>%
  mutate(Year_actual = as.integer(Year_actual)) %>%
  # calculate the age
  mutate(Age = Year_actual - `Birth year`) %>%
  relocate(Age, .after = `Birth year`) %>%
  select(-c(Year_actual, `Birth year`, Datum))
```

EDA

How many patients?

```
df %>%
  janitor::tabyl(Gender) %>%
  adorn_pct_formatting() %>%
  adorn_totals("row")
```

```
##   Gender    n percent
```

```
##      Man   60   45.5%
##      Woman  72   54.5%
##      Total 132      -
```

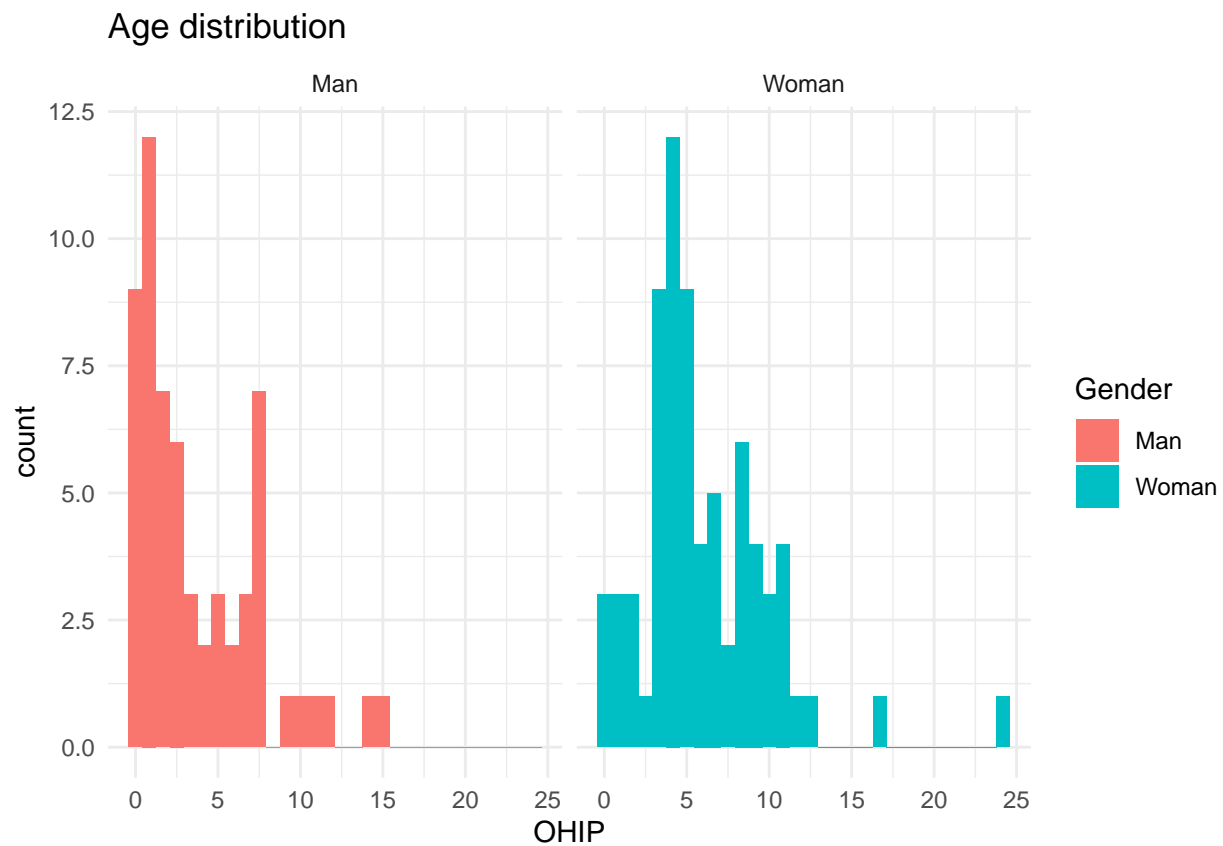
```
summary(df$Age)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      21.00  28.00   32.00   32.97  37.00   61.00
```

Ohip by age and gender

```
df %>%
  ggplot(aes(x = OHIP,
             fill = Gender)) +
  geom_histogram() +
  facet_grid(. ~ Gender) +
  labs(title = "Age distribution")
```

```
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
```



Table

```
df %>%
  select(Age, Gender, OHIP) %>%
  gtsummary::tbl_summary(by = Gender) %>%
  gtsummary::add_p()
```

```
## Table printed with 'knitr::kable()', not {gt}. Learn why at
## http://www.danielsjoberg.com/gtsummary/articles/rmarkdown.html
## To suppress this message, include 'message = FALSE' in code chunk header.
```

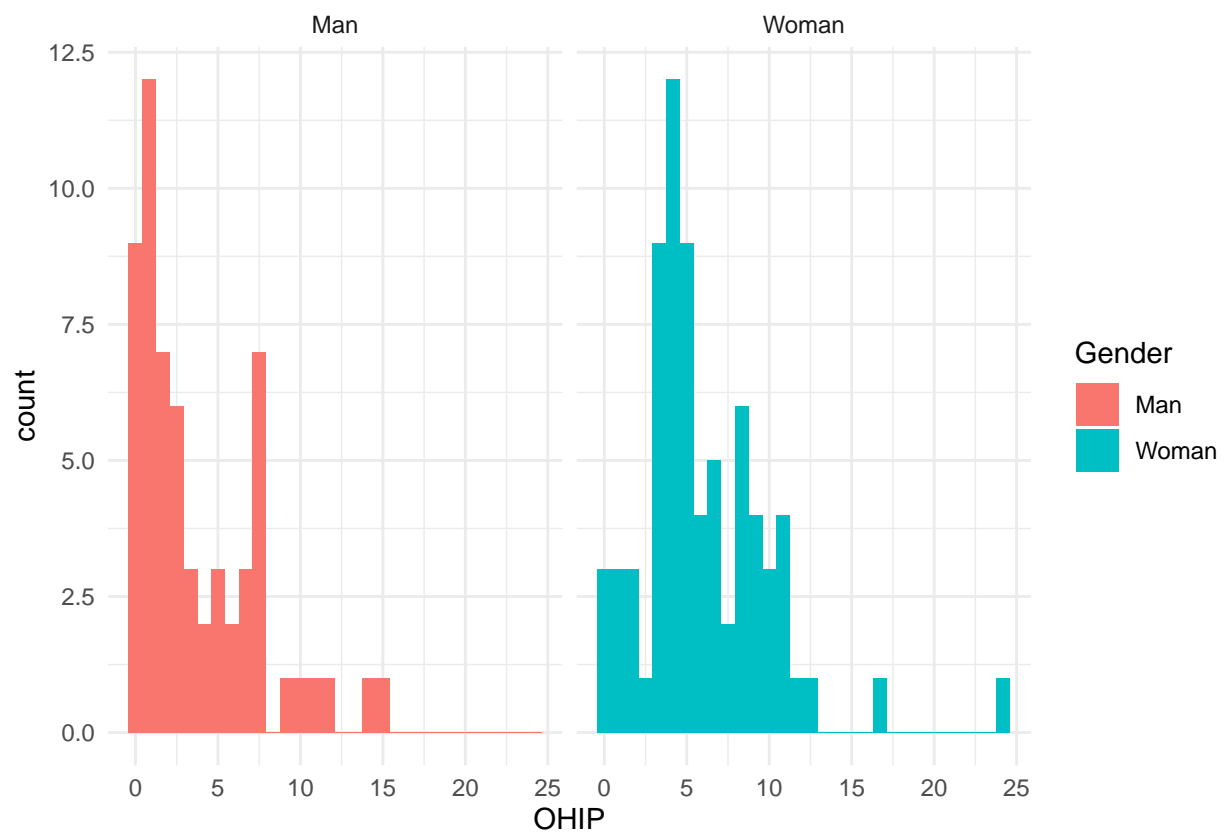
Characteristic	Man, N = 60	Woman, N = 72	p-value
Age	35 (31, 38)	30 (27, 34)	<0.001
OHIP	2.2 (0.9, 6.6)	5.0 (3.6, 8.3)	<0.001

OHIP

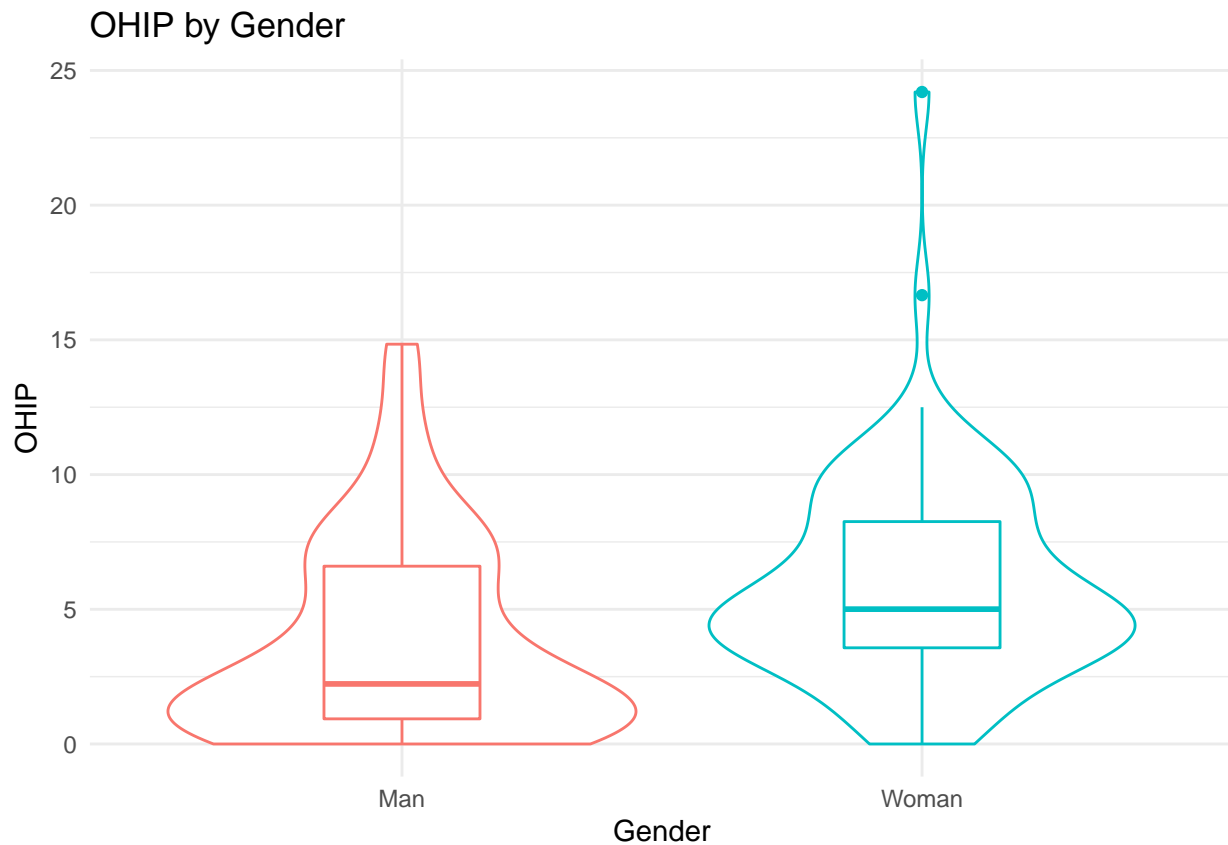
OHIP by gender

```
df %>%
  ggplot(aes(x = OHIP,
             fill = Gender)) +
  geom_histogram() +
  facet_wrap(Gender ~ .)
```

```
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
```



```
df %>%
  ggplot(aes(y = OHIP,
             x = Gender,
             color = Gender)) +
  geom_violin() +
  geom_boxplot(width = .3) +
  theme(legend.position = "none") +
  labs(title = "OHIP by Gender")
```



Regression analysis

```
df %>%
  with(glm(OHIP ~ Age + Gender)) %>%
  gtsummary::tbl_regression()
```

```
## Table printed with 'knitr::kable()', not {gt}. Learn why at
## http://www.danielsjoberg.com/gtsummary/articles/rmarkdown.html
## To suppress this message, include 'message = FALSE' in code chunk header.
```

Characteristic	Beta	95% CI	p-value
Age	-0.08	-0.18, 0.02	0.14
Gender			
Man			
Woman	2.0	0.64, 3.3	0.005

```
df %>%
  with(glm(OHIP ~ Age + Gender)) %>%
  report::report()
```

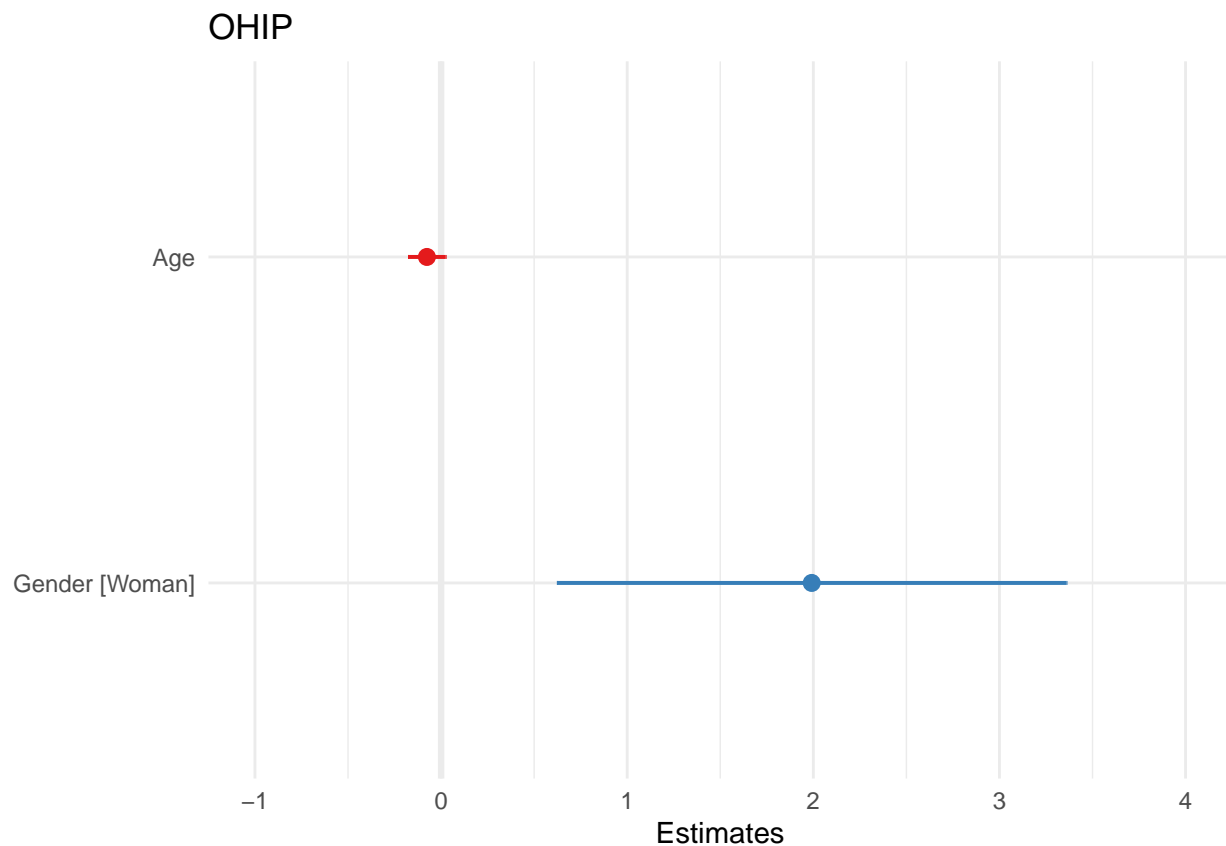
```
## Warning: 'effectsize::interpret_d' is deprecated.
```

```
## Use 'interpret_cohens_d' instead.
## See help("Deprecated")

## Warning: 'effectsize::interpret_d' is deprecated.
## Use 'interpret_cohens_d' instead.
## See help("Deprecated")

## We fitted a linear model (estimated using ML) to predict OHIP with Age and Gender (formula: OHIP ~ Age + Gender)
##
## - The effect of Age is statistically non-significant and negative (beta = -0.08, 95% CI [-0.18, 0.02])
## - The effect of Gender [Woman] is statistically significant and positive (beta = 1.99, 95% CI [0.6, 3.4])
##
## Standardized parameters were obtained by fitting the model on a standardized version of the dataset.
```

```
df %>%
  with(glm(OHIP ~ Age + Gender)) %>%
  sjPlot::plot_model()
```

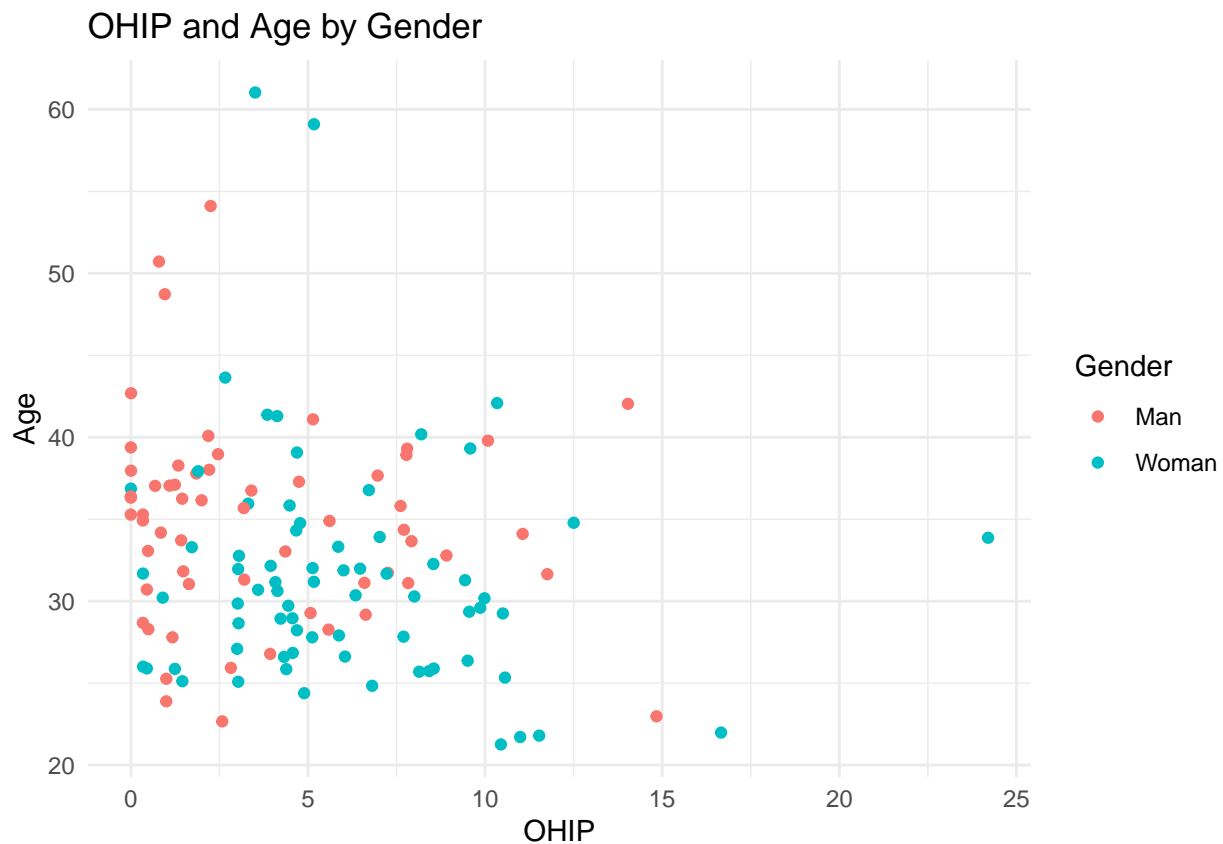


OHIP by age

```
df %>%
  ggplot(aes(x = OHIP,
             y = Age,
             color = Gender)) +
```



```
geom_jitter() +
labs(title = "OHIP and Age by Gender")
```



Analysis each question

```
df_long_values <- df %>%
  select(-c(`Difficulty pronouncing words`:`Totally unable to function`)) %>%
  pivot_longer(`Difficulty pronouncing words Weighted`:`Totally unable to function Weighted`,
               names_to = "OHIP_item",
               values_to = "OHIP_value_weighted")
```

```
df_long_likert <- df %>%
  select(Gender, c(`Difficulty pronouncing words`:`Totally unable to function`)) %>%
  pivot_longer(`Difficulty pronouncing words`:`Totally unable to function`,
               names_to = "likert_item",
               values_to = "likert_value")
```

Data summary

```
df_long_likert_summary <- df_long_likert %>%
  group_by(likert_item, likert_value) %>%
  count(name = "n_answers") %>%
  group_by(likert_item) %>%
```

```
mutate(percent_answers = n_answers / sum(n_answers)) %>%
ungroup() %>%
mutate(percent_answers_label = percent(percent_answers, accuracy = 1))
```

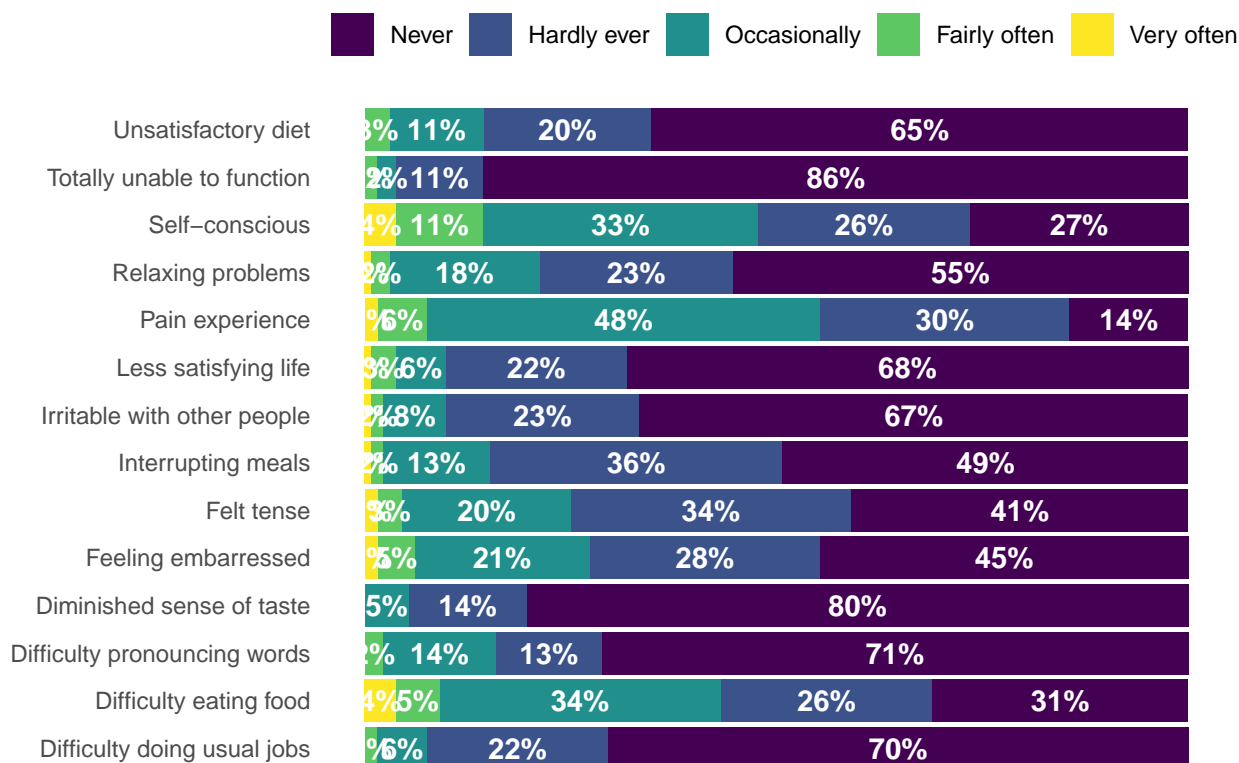
```
head(df_long_likert_summary)
```

```
## # A tibble: 6 x 5
##   likert_item      likert_value n_answers percent_answers percent_answers_l~
##   <chr>          <chr>         <int>         <dbl> <chr>
## 1 Difficulty doing us~ Fairly often         2      0.0152 2%
## 2 Difficulty doing us~ Hardly ever        29      0.220 22%
## 3 Difficulty doing us~ Never             93      0.705 70%
## 4 Difficulty doing us~ Occasionally        8      0.0606 6%
## 5 Difficulty eating f~ Fairly often        7      0.0530 5%
## 6 Difficulty eating f~ Hardly ever        34      0.258 26%
```

```
df_long_likert_summary <- df_long_likert_summary %>%
  mutate(likert_value = fct_relevel(likert_value,
                                    "Never",
                                    "Hardly ever",
                                    "Occasionally",
                                    "Fairly often",
                                    "Very often"))
```

```
df_long_likert_summary %>%
  ggplot(aes(x = likert_item,
             y = percent_answers,
             fill = likert_value)) +
  geom_col() +
  geom_text(aes(label = percent_answers_label),
            position = position_stack(vjust = 0.5),
            color = "white",
            fontface = "bold") +
  coord_flip() +
  scale_x_discrete() +
  scale_fill_viridis_d() +
  labs(title = "Title",
       x = NULL,
       fill = NULL) +
  theme_minimal() +
  theme(axis.text.x = element_blank(),
        axis.title.x = element_blank(),
        panel.grid = element_blank(),
        legend.position = "top")
```

Title



Likert plot with sjPlot

```
df_for_likert <- df %>%
  # select some columns
  select(Age:`Totally unable to function`) %>%

  # create an ID to not to mess with the pivot_wider afterwards
  mutate(id = row_number()) %>%
  relocate(id, .before = Age) %>%

  # reshape for easy wrangling
  pivot_longer(cols = -c(id, Age, Gender)) %>%

  # reorder the levels
  mutate(value = fct_relevel(value,
                              "Never",
                              "Hardly ever",
                              "Occasionally",
                              "Fairly often",
                              "Very often")) %>%

  # convert to a numeric variable
  # mutate(value_num = case_when(
  #   value == "Never" ~ "1",
  #   value == "Hardly ever" ~ "2",
  #   value == "Occasionally" ~ "3",
```

```

# value == "Fairly often" ~ "4",
# TRUE ~ "5"
# )) %>%
# mutate(value_num = as.integer(value_num)) %>%

# select only these columns

# collapse categories
mutate(value = case_when(
  value == "Fairly often" ~ "Very often",
  TRUE ~ as.character(as.character(value))
)) %>%

```

```

# select(id, name, value_num) %>%
select(id, name, value) %>%
# reshape for the graph

pivot_wider(names_from = name,
             values_from = value)

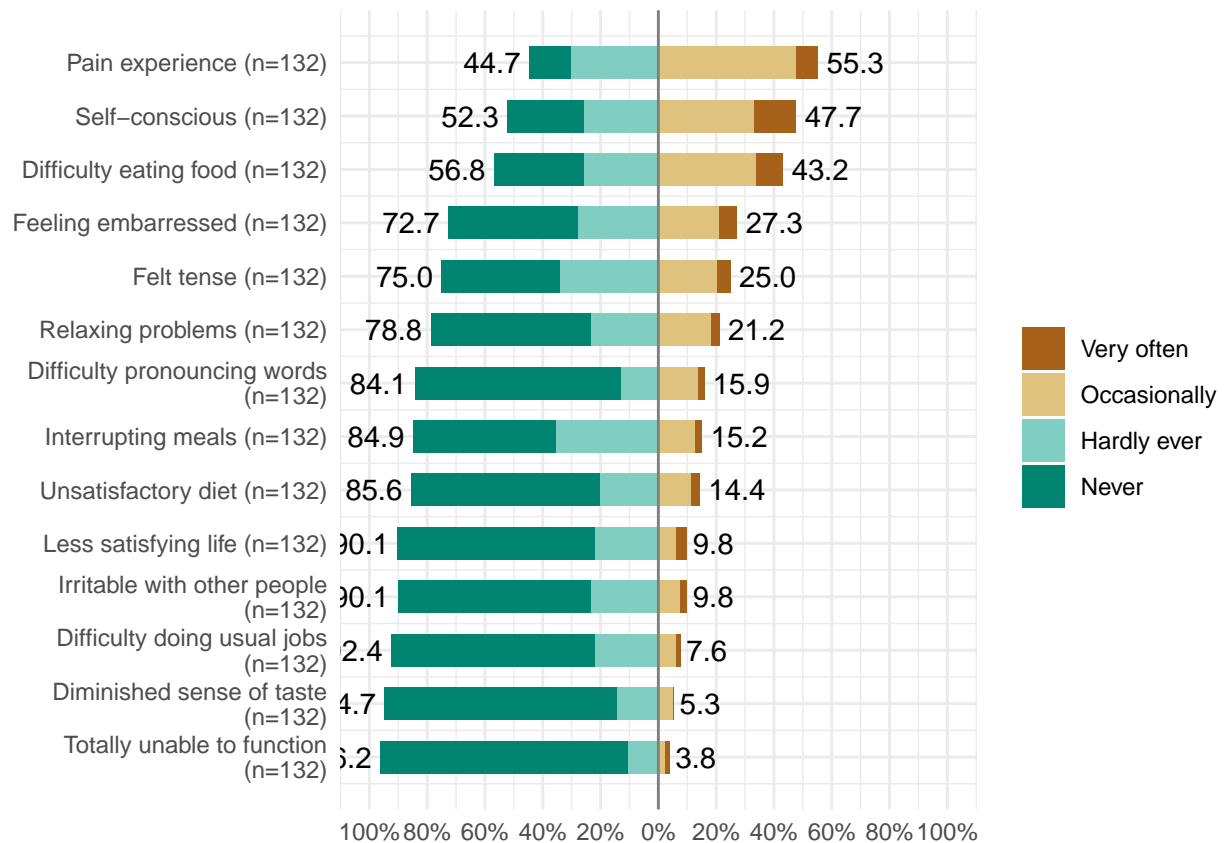
```

```

df_for_likert %>%
  select(-id) %>%
  dplyr::mutate_all(., ~ ordered(., levels = c("Very often", "Occasionally", "Hardly ever", "Never"))) %>%
  # janitor::tabyl(`Pain experience`)

plot_likert(
  # grid.range = c(1.2, 1.4),
  # expand.grid = FALSE,
  sort.frq = "pos.desc",
  values = "sum.outside"
  # show.prc.sign = TRUE
)

```



```
ggsave("likert_all.pdf", width = 23, height = 15, units = "cm")
```

Men and Women Likert

Men

```
df_for_likert_man <- df %>%
  filter(Gender == "Man") %>%
  # select some columns
  select(Age:`Totally unable to function`) %>%

  # create an ID to not to mess with the pivot_wider afterwards
  mutate(id = row_number()) %>%
  relocate(id, .before = Age) %>%

  # reshape for easy wrangling
  pivot_longer(cols = -c(id, Age, Gender)) %>%

  # reorder the levels
  mutate(value = fct_relevel(value,
                             "Never",
                             "Hardly ever",
                             "Occasionally",
                             "Fairly often",
```

```

                                "Very often")) %>%
# convert to a numeric variable
# mutate(value_num = case_when(
#   value == "Never" ~ "1",
#   value == "Hardly ever" ~ "2",
#   value == "Occasionally" ~ "3",
#   value == "Fairly often" ~ "4",
#   TRUE ~ "5"
# )) %>%
# mutate(value_num = as.integer(value_num)) %>%

# select only these columns

# collapse categories
mutate(value = case_when(
  value == "Fairly often" ~ "Very often",
  TRUE ~ as.character(as.character(value))
)) %>%

# select(id, name, value_num) %>%
select(id, name, value) %>%
# reshape for the graph

pivot_wider(names_from = name,
             values_from = value)

```

```

men_plot <- df_for_likert_man %>%
  select(-id) %>%
  dplyr::mutate_all(., ~ ordered(., levels = c("Very often", "Occasionally", "Hardly ever", "Never"))) %>%
  # janitor::tabyl(`Pain experience`)

plot_likert(
  # grid.range = c(1.2, 1.4),
  # expand.grid = FALSE,
  sort.frq = "pos.desc",
  values = "sum.outside"
  # show.prc.sign = TRUE
) +
labs(title = "Men")

```

```

ggsave("likert_man.pdf", width = 23, height = 15, units = "cm")

```

Women Likert

```

df_for_likert_women <- df %>%
  filter(Gender == "Woman") %>%

```

```

# select some columns
select(Age:`Totally unable to function`) %>%

# create an ID to not to mess with the pivot_wider afterwards
mutate(id = row_number()) %>%
relocate(id, .before = Age) %>%

# reshape for easy wrangling
pivot_longer(cols = -c(id, Age, Gender)) %>%

# reorder the levels
mutate(value = fct_relevel(value,
                           "Never",
                           "Hardly ever",
                           "Occasionally",
                           "Fairly often",
                           "Very often")) %>%

# convert to a numeric variable
# mutate(value_num = case_when(
#   value == "Never" ~ "1",
#   value == "Hardly ever" ~ "2",
#   value == "Occasionally" ~ "3",
#   value == "Fairly often" ~ "4",
#   TRUE ~ "5"
# )) %>%
# mutate(value_num = as.integer(value_num)) %>%

# select only these columns

# collapse categories
mutate(value = case_when(
  value == "Fairly often" ~ "Very often",
  TRUE ~ as.character(as.character(value))
)) %>%

# select(id, name, value_num) %>%
select(id, name, value) %>%
# reshape for the graph

pivot_wider(names_from = name,
            values_from = value)

women_plot <- df_for_likert_women %>%
  select(-id) %>%
  dplyr::mutate_all(., ~ ordered(., levels = c("Very often", "Occasionally", "Hardly ever", "Never")))
# janitor::tabyl(`Pain experience`)

plot_likert(

```

```
# grid.range = c(1.2, 1.4),  
# expand.grid = FALSE,  
sort.frq = "pos.desc",  
values = "sum.outside"  
#show.prc.sign = TRUE  
) +  
labs(title = "Women")
```

```
ggsave("likert_woman.pdf", width = 23, height = 15, units = "cm")
```