Analysis

In this document, we do the analysis presented in the paper.

Currently, the analysis uses fake data.

Setup

```r
library(testthat)
```

Reading the data

```r
ratings <- readr::read_csv("ratings.csv", show_col_types = FALSE)
n_ratings <- nrow(ratings)
```

There are 1000 ratings.

Analysis

Connecting the ratings to the formations:

```r
songs <- dplyr::select(heyahmama::get_songs(), cd_title, song_title)
n_songs <- nrow(songs)
```

There are 270 songs.

```r
cds <- dplyr::select(heyahmama::get_cds(), cd_title, formation)
n_cds <-nrow(cds)
```

There are 22 CDs.

```r
songs_per_formation <- dplyr::select(merge(songs, cds), song_title, formation)

# Not yet
# testthat::expect_equal(n_songs, nrow(songs_per_formation))
if (n_songs != nrow(songs_per_formation)) {
  warning("Not all songs are found to be on a CD")
}

knitr::kable(head(songs_per_formation))
```

song_title	formation
10.000 luchtballonnen	3
Kusjessoldaten	3
Als het binnen regent	3
Jodelee	3
Kus van de juf	3
Jij bent de bom!	3

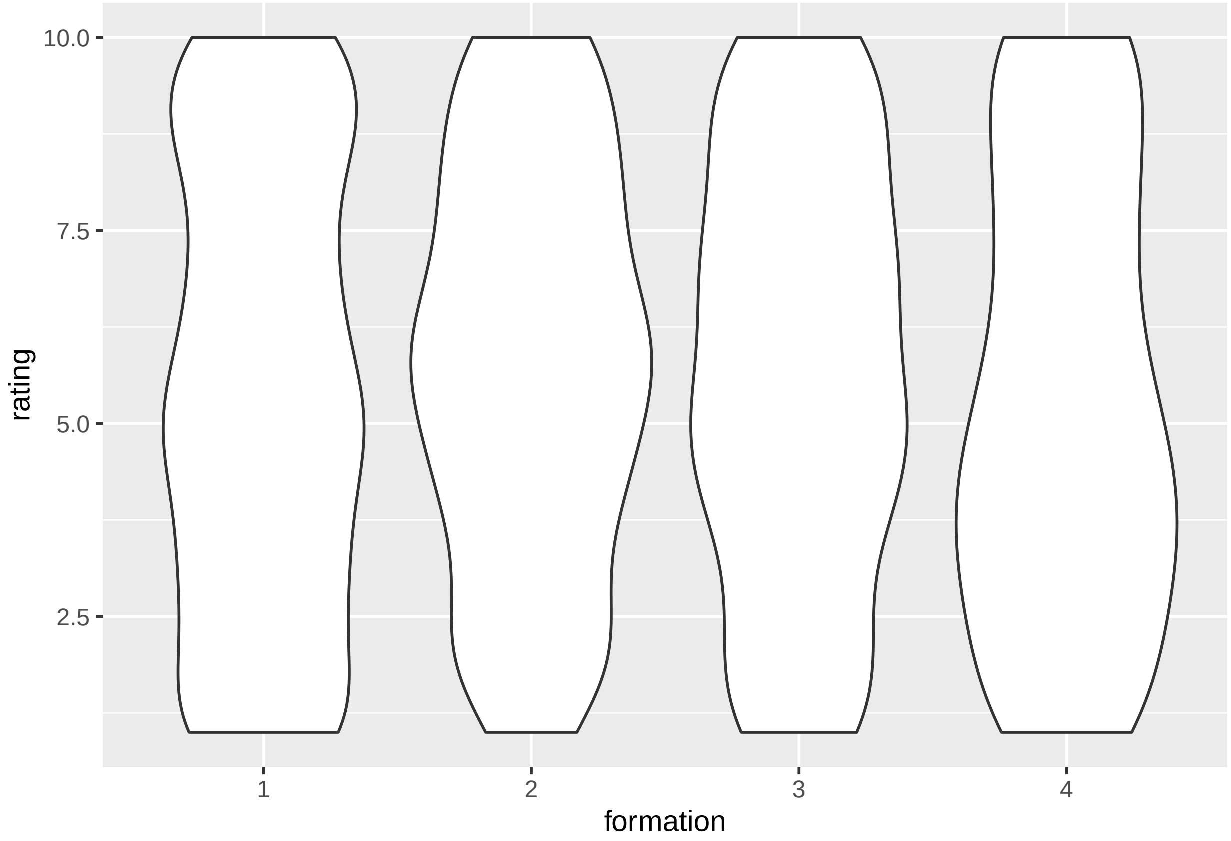
Add the formations to the ratings:

```
ratings_per_formation <- dplyr::select(merge(ratings, songs_per_formation), formation, rating)
ratings_per_formation$formation <- as.factor(ratings_per_formation$formation)
knitr::kable(head(ratings_per_formation))
```

formation	rating
1	4
1	4
1	6
3	8
3	5
3	10

Plot:

```
ggplot2::ggplot(
  ratings_per_formation,
  ggplot2::aes(x = formation, y = rating)
) + ggplot2::geom_violin()
```



Order formations by ratings:

```
average_rating_per_formation <-
  ratings_per_formation |> dplyr::group_by(formation) |> dplyr::summarise(average_rating = mean(rating)

ordered_average_rating_per_formation <-  average_rating_per_formation |> dplyr::arrange(dplyr::desc(ave
knitr::kable(ordered_average_rating_per_formation)
```

formation	average_rating
2	5.703911
3	5.613821
1	5.437956
4	5.274390

Statistics

Do the formations have different ratings?

```
ratings_1 <- ratings_per_formation[ratings_per_formation$formation == 1, ]$rating
ratings_2 <- ratings_per_formation[ratings_per_formation$formation == 2, ]$rating
ratings_3 <- ratings_per_formation[ratings_per_formation$formation == 3, ]$rating
ratings_4 <- ratings_per_formation[ratings_per_formation$formation == 4, ]$rating
p_12 <- ks.test(ratings_1, ratings_2, alternative = "two.sided")$p.value
#> Warning in ks.test.default(ratings_1, ratings_2, alternative = "two.sided"):
#> p-value will be approximate in the presence of ties
p_13 <- ks.test(ratings_1, ratings_3, alternative = "two.sided")$p.value
#> Warning in ks.test.default(ratings_1, ratings_3, alternative = "two.sided"):
#> p-value will be approximate in the presence of ties
p_14 <- ks.test(ratings_1, ratings_4, alternative = "two.sided")$p.value
#> Warning in ks.test.default(ratings_1, ratings_4, alternative = "two.sided"):
#> p-value will be approximate in the presence of ties
p_23 <- ks.test(ratings_2, ratings_3, alternative = "two.sided")$p.value
#> Warning in ks.test.default(ratings_2, ratings_3, alternative = "two.sided"):
#> p-value will be approximate in the presence of ties
p_24 <- ks.test(ratings_2, ratings_4, alternative = "two.sided")$p.value
#> Warning in ks.test.default(ratings_2, ratings_4, alternative = "two.sided"):
#> p-value will be approximate in the presence of ties
p_34 <- ks.test(ratings_3, ratings_4, alternative = "two.sided")$p.value
#> Warning in ks.test.default(ratings_3, ratings_4, alternative = "two.sided"):
#> p-value will be approximate in the presence of ties
p_values_table <- tibble::tribble(
  ~comparison, ~p_value,
  "12", p_12,
  "13", p_13,
  "14", p_14,
  "23", p_23,
  "24", p_24,
  "34", p_34
)
alpha <- 0.05
p_values_table$is_the_same <- p_values_table$p_value > alpha
knitr::kable(p_values_table)
```

comparison	p_value	is_the_same
12	0.4511496	TRUE
13	0.6318555	TRUE
14	0.8206113	TRUE

comparison	p_value	is_the_same
23	0.9954695	TRUE
24	0.1556527	TRUE
34	0.2831867	TRUE