RESPONSI_DS-(A)

_<123180145>

Intro

0. Cuci tangan dengan sabun hingga benar-benar bersih dengan durasi mencuci tangan kurang lebih 20 dtk
1. Kerjakan Soal-soal yang ada! Jangan lupa author dinamai!
2. Responsi terdiri dari 2 bagian yaitu bagian pertama dan bagian kedua
3. Jawab dengan membuat chunk dibawah soal!
4. Durasi pengerjaan sesuai kesepakatan yaitu 2 jam mulai pukul 20.00 pagi hingga 22.00 malam tanggal 22 Januari 2021
5. No toleransi pengumpulan telat. Telat tiap 3 menit akan ada pengurangan nilai 5 point dengan maksimal pengurangan 25 point. Telat lebih dari 15 menit atau melebihi pukul 22.15 dianggap **GUGUR**.
6. Misal soal rancu bisa menghubungi asisten terkait.
7. Pengumpulan hanya dalam bentuk **WORD Document atau PDF**. Jika pengumpulan dalam bentuk **Rmd** akan dianggap tidak mengumpulkan jaawaban. Pastikan jawaban dapat dijalankan dengan baik.
8. Tenang, untuk responsi kali ini nilai akan diobral, nilai maksimal adalah 350 dari 100. Jadi, kemungkinan dapat nilai bagus besar kok.
9. Isi juga review/feedback/kritik/saran/masukan yang sudah disediakan di bagian paling bawah soal. ***WAJIB***
10. Selamat memutus rantai gabut :) jangan lupa jaga kesehatan

Persiapan

Load library apa saja yang kira-kira digunakan! Lalu load dataset 'googleplay.csv' dan 'googleplay_user_review.csv'!

```
library(here)
```

```
## here() starts at D:/R Studio File/RMarkdown/praktikum2020
```

```
library(vroom)
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
```

```r
library(tidymodels)
```

```
## -- Attaching packages -------------------------------------- tidymodels 0.1.2 --

## v broom     0.7.2      v recipes   0.1.15
## v dials     0.0.9      v rsample   0.0.8
## v ggplot2   3.3.2      v tibble    3.0.4
## v infer     0.5.4      v tidyr     1.1.2
## v modeldata 0.1.0      v tune      0.1.2
## v parsnip   0.1.5      v workflows 0.2.1
## v purrr     0.3.4      v yardstick 0.0.7

## -- Conflicts ----------------------------------------- tidymodels_conflicts() --
## x purrr::discard()  masks scales::discard()
## x dplyr::filter()   masks stats::filter()
## x dplyr::lag()      masks stats::lag()
## x yardstick::spec() masks vroom::spec()
## x recipes::step()   masks stats::step()
```

```r
library(tidytext)
library(ggplot2)

googleplaystore <- vroom(here::here("Responsi", "googleplaystore.csv"))
```

```
## Rows: 8,196
## Columns: 13
## Delimiter: ","
## chr [11]: App, Category, Size, Installs, Type, Price, Content Rating, Genres, Last Updated...
## dbl [ 2]: Rating, Reviews
##
## Use `spec()` to retrieve the guessed column specification
## Pass a specification to the `col_types` argument to quiet this message
```

```r
user_reviews <- vroom(here::here("Responsi", "googleplaystore_user_reviews.csv"))
```

```
## Rows: 64,295
## Columns: 5
## Delimiter: ","
## chr [3]: App, Translated_Review, Sentiment
## dbl [2]: Sentiment_Polarity, Sentiment_Subjectivity
##
## Use `spec()` to retrieve the guessed column specification
## Pass a specification to the `col_types` argument to quiet this message
```

Bagian Pertama

1. Tampilkan TOP 10 Aplikasi berdasarkan peringkat PENILAIAN/RATING yang diberikan user! **point 10**

```
googleplaystore %>% arrange(desc(Rating)) %>% head(n=10)
```

```
## # A tibble: 10 x 13
##    App   Category Rating Reviews Size  Installs Type  Price `Content Rating`
##    <chr> <chr>     <dbl>   <dbl> <chr> <chr>    <chr> <chr> <chr>
##  1 Hoji~ COMICS        5      15 37M   1,000+   Free  0     Everyone
##  2 Amer~ DATING        5       5 4.4M  1,000+   Free  0     Mature 17+
##  3 Awak~ DATING        5       2 70M   100+     Free  0     Mature 17+
##  4 Spin~ DATING        5       5 9.3M  500+     Free  0     Teen
##  5 Girl~ DATING        5       6 5.0M  100+     Free  0     Mature 17+
##  6 Onli~ DATING        5       5 5.0M  100+     Free  0     Mature 17+
##  7 Spee~ DATING        5       3 25M   100+     Free  0     Mature 17+
##  8 SUMM~ EVENTS        5       4 61M   500+     Free  0     Everyone
##  9 Pros~ EVENTS        5      16 2.3M  100+     Free  0     Everyone
## 10 Mind~ EVENTS        5       1 21M   100+     Free  0     Everyone
## # ... with 4 more variables: Genres <chr>, `Last Updated` <chr>, `Current
## #   Ver` <chr>, `Android Ver` <chr>
```

2. Tampilkan rata-rata RATING yang dihitung menggunakan fungsi buatan untuk setiap kategori aplikasi! **point 15**

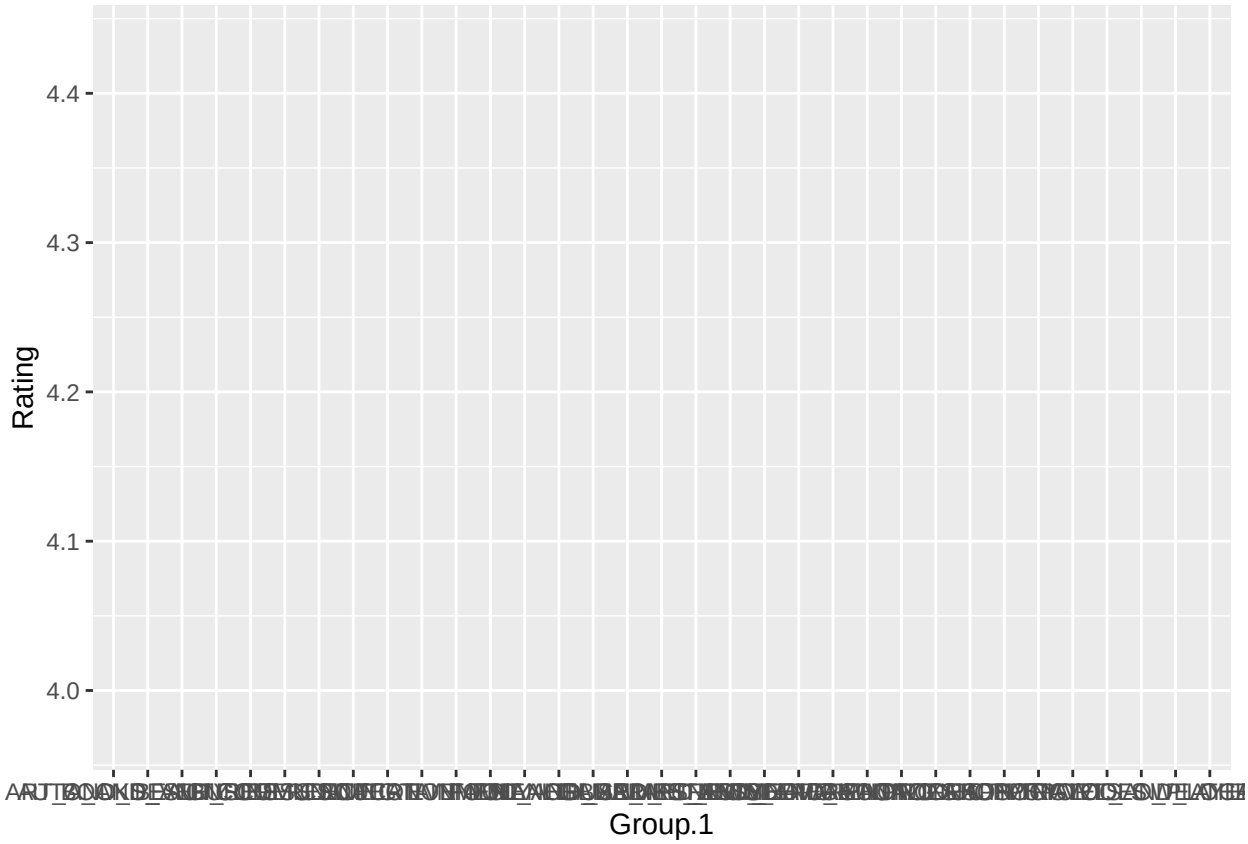
```
Rata2 <- aggregate(googleplaystore[, 3], list(googleplaystore$Category), mean)
Rata2
```

```
##               Group.1   Rating
## 1       ART_AND_DESIGN 4.357377
## 2    AUTO_AND_VEHICLES 4.190411
## 3               BEAUTY 4.278571
## 4  BOOKS_AND_REFERENCE 4.344970
## 5             BUSINESS 4.098479
## 6               COMICS 4.181481
## 7        COMMUNICATION 4.121484
## 8               DATING 3.970149
## 9            EDUCATION 4.364407
## 10       ENTERTAINMENT 4.135294
## 11              EVENTS 4.435556
## 12              FAMILY 4.179664
## 13             FINANCE 4.115563
## 14      FOOD_AND_DRINK 4.172340
## 15                GAME 4.247368
## 16  HEALTH_AND_FITNESS 4.243033
## 17      HOUSE_AND_HOME 4.150000
## 18  LIBRARIES_AND_DEMO 4.178125
## 19           LIFESTYLE 4.093355
## 20 MAPS_AND_NAVIGATION 4.036441
## 21             MEDICAL 4.166552
## 22  NEWS_AND_MAGAZINES 4.121569
## 23           PARENTING 4.300000
## 24     PERSONALIZATION 4.332215
## 25         PHOTOGRAPHY 4.157414
## 26        PRODUCTIVITY 4.183389
## 27            SHOPPING 4.230000
```

```
## 28              SOCIAL 4.247291
## 29              SPORTS 4.216154
## 30               TOOLS 4.039554
## 31    TRAVEL_AND_LOCAL 4.069519
## 32       VIDEO_PLAYERS 4.044595
## 33             WEATHER 4.243056
```

3. Berdasarkan soal nomor 2, buat plot untuk memvisualisasikan hasilnya! (Bentuk plot bebas) **point 15**

```
Rata2 %>% ggplot(aes(Group.1, Rating))
```



Info untuk 2 soal 4-5:Terdapat dua dataset yang digunakan. Satu dataset untuk info aplikasi dan satu dataset lagi untuk kumpulan reviewnya.

4. Dari kedua dataset tersebut, buat satu variable data baru yang isinya NAMA APLIKASI, RATING, dan JUMLAH REVIEW Positif dan/atau Negatif dan/atau Neutral (boleh semua, boleh pilih salah satu)lalu tampilkan isi data tabel tersebut! **point 20**

```
new_data <- inner_join(googleplaystore, user_reviews, by = "App") %>%
  select(App, Rating, Sentiment) %>% count(Sentiment)
new_data
```

```
## # A tibble: 4 x 2
```

```
##   Sentiment     n
##   <chr>     <int>
## 1 nan       24155
## 2 Negative   7618
## 3 Neutral    4584
## 4 Positive  21739
```

5. Dalam dunia data scientist, sebelum melakukan pemodelan ada baiknya data dilakukan preprocessing terlebih dahulu. Dengan dataset review yang sudah dimasukkan oleh user, lakukan sebuah preprocessing data SEDERHANA yang menurut kalian dapat dilakukan untuk dataset tersebut agar dataset bisa siap untuk dimodelkan (simpan hasil preprocessing dalam variabel baru)!

Clue : Clean, Tidy, no redundacy, no dupe, no null , lan liya-liyane **point 40**

```
googleplaystore %>%
  inner_join(user_reviews, by = "App") %>%
  filter(Translated_Review != "nan") %>%
  unnest_tokens(word, Translated_Review) %>%
  anti_join(stop_words)
```

```
## Joining, by = "word"
```

```
## # A tibble: 362,048 x 17
##    App   Category Rating Reviews Size  Installs Type  Price `Content Rating`
##    <chr> <chr>     <dbl>   <dbl> <chr> <chr>    <chr> <chr> <chr>
##  1 Colo~ ART_AND~    3.9     967 14M   500,000+ Free  0     Everyone
##  2 Colo~ ART_AND~    3.9     967 14M   500,000+ Free  0     Everyone
##  3 Colo~ ART_AND~    3.9     967 14M   500,000+ Free  0     Everyone
##  4 Colo~ ART_AND~    3.9     967 14M   500,000+ Free  0     Everyone
##  5 Colo~ ART_AND~    3.9     967 14M   500,000+ Free  0     Everyone
##  6 Colo~ ART_AND~    3.9     967 14M   500,000+ Free  0     Everyone
##  7 Colo~ ART_AND~    3.9     967 14M   500,000+ Free  0     Everyone
##  8 Colo~ ART_AND~    3.9     967 14M   500,000+ Free  0     Everyone
##  9 Colo~ ART_AND~    3.9     967 14M   500,000+ Free  0     Everyone
## 10 Colo~ ART_AND~    3.9     967 14M   500,000+ Free  0     Everyone
## # ... with 362,038 more rows, and 8 more variables: Genres <chr>, `Last
## #   Updated` <chr>, `Current Ver` <chr>, `Android Ver` <chr>, Sentiment <chr>,
## #   Sentiment_Polarity <dbl>, Sentiment_Subjectivity <dbl>, word <chr>
```

Bagian Kedua

Referensi mengerjakan: https://www.tidytextmining.com/

1. Import library tidymodels, vroom, here, tidytext dan dua dataset ke dalam objek R **nilai 10**

```
library(tidymodels)
library(vroom)
library(here)
library(tidytext)

user_reviews <- vroom(here("Responsi", "googleplaystore_user_reviews.csv"))
```

```
## Rows: 64,295
## Columns: 5
## Delimiter: ","
## chr [3]: App, Translated_Review, Sentiment
## dbl [2]: Sentiment_Polarity, Sentiment_Subjectivity
##
## Use `spec()` to retrieve the guessed column specification
## Pass a specification to the `col_types` argument to quiet this message
```

```
googleplaystore <- vroom(here("Responsi", "googleplaystore.csv"))
```

```
## Rows: 8,196
## Columns: 13
## Delimiter: ","
## chr [11]: App, Category, Size, Installs, Type, Price, Content Rating, Genres, Last Updated...
## dbl [ 2]: Rating, Reviews
##
## Use `spec()` to retrieve the guessed column specification
## Pass a specification to the `col_types` argument to quiet this message
```

2. Joining dua dataset menggunakan inner join **nilai 10**

```
joining_data <- googleplaystore %>%
  inner_join(user_reviews, by = "App")
joining_data
```

```
## # A tibble: 58,096 x 17
##    App   Category Rating Reviews Size  Installs Type  Price `Content Rating`
##    <chr> <chr>     <dbl>   <dbl> <chr> <chr>    <chr> <chr> <chr>
##  1 Colo~ ART_AND~    3.9     967 14M   500,000+ Free  0     Everyone
##  2 Colo~ ART_AND~    3.9     967 14M   500,000+ Free  0     Everyone
##  3 Colo~ ART_AND~    3.9     967 14M   500,000+ Free  0     Everyone
##  4 Colo~ ART_AND~    3.9     967 14M   500,000+ Free  0     Everyone
##  5 Colo~ ART_AND~    3.9     967 14M   500,000+ Free  0     Everyone
##  6 Colo~ ART_AND~    3.9     967 14M   500,000+ Free  0     Everyone
##  7 Colo~ ART_AND~    3.9     967 14M   500,000+ Free  0     Everyone
##  8 Colo~ ART_AND~    3.9     967 14M   500,000+ Free  0     Everyone
##  9 Colo~ ART_AND~    3.9     967 14M   500,000+ Free  0     Everyone
## 10 Colo~ ART_AND~    3.9     967 14M   500,000+ Free  0     Everyone
## # ... with 58,086 more rows, and 8 more variables: Genres <chr>, `Last
## #   Updated` <chr>, `Current Ver` <chr>, `Android Ver` <chr>,
## #   Translated_Review <chr>, Sentiment <chr>, Sentiment_Polarity <dbl>,
## #   Sentiment_Subjectivity <dbl>
```

3. Tahap pre-processing data. Ketika ingin melakukan analisis sentimen beberapa hal harus dilakukan sebelum data dapat digunakan. Bersihkan dan rapikan data dengan membuang data yang "nan" di bagian Translated_review. Setelah itu, data juga harus dibersihkan dari kata-kata yang mengandung stop_word (seperti: a, a's, after, dll). Data yang siap diolah juga harus ditokenisasi yaitu proses membagi teks dari paragraf atau kalimat ke kata. Hasil dari tokenisasi adalah tiap baris data hanya mengandung 1 kata. **nilai 15**

```r
tidy_user_reviews <- joining_data %>%
  filter(Translated_Review != "nan") %>%
  unnest_tokens(word, Translated_Review) %>%
  anti_join(stop_words)
```

```
## Joining, by = "word"
```

```r
tidy_user_reviews
```

```
## # A tibble: 362,048 x 17
##    App   Category Rating Reviews Size  Installs Type  Price `Content Rating`
##    <chr> <chr>     <dbl>   <dbl> <chr> <chr>    <chr> <chr> <chr>
##  1 Colo~ ART_AND~    3.9     967 14M   500,000+ Free  0     Everyone
##  2 Colo~ ART_AND~    3.9     967 14M   500,000+ Free  0     Everyone
##  3 Colo~ ART_AND~    3.9     967 14M   500,000+ Free  0     Everyone
##  4 Colo~ ART_AND~    3.9     967 14M   500,000+ Free  0     Everyone
##  5 Colo~ ART_AND~    3.9     967 14M   500,000+ Free  0     Everyone
##  6 Colo~ ART_AND~    3.9     967 14M   500,000+ Free  0     Everyone
##  7 Colo~ ART_AND~    3.9     967 14M   500,000+ Free  0     Everyone
##  8 Colo~ ART_AND~    3.9     967 14M   500,000+ Free  0     Everyone
##  9 Colo~ ART_AND~    3.9     967 14M   500,000+ Free  0     Everyone
## 10 Colo~ ART_AND~    3.9     967 14M   500,000+ Free  0     Everyone
## # ... with 362,038 more rows, and 8 more variables: Genres <chr>, `Last
## #   Updated` <chr>, `Current Ver` <chr>, `Android Ver` <chr>, Sentiment <chr>,
## #   Sentiment_Polarity <dbl>, Sentiment_Subjectivity <dbl>, word <chr>
```

4. Sentimen analisis dapat menggunakan beberapa jenis metode berdasarkan sentiment lexicon. Ada beberapa sentiment lexicon seperti bing, afinn, dan nrc. Gunakan sentiment lexicon nrc untuk mendapatkan jumlah kata untuk 10 kategori nrc (positive, negative, fear, surprise, dll). ***nilai 15***

```r
nrc_n <- tidy_user_reviews %>%
  inner_join(get_sentiments("nrc")) %>%
  count(word, sort = TRUE)
```

```
## Registered S3 methods overwritten by 'readr':
##   method           from
##   format.col_spec  vroom
##   print.col_spec   vroom
##   print.collector  vroom
##   print.date_names vroom
##   print.locale     vroom
##   str.col_spec     vroom
```

```
## Joining, by = "word"
```

```r
nrc_n
```

```
## # A tibble: 2,673 x 2
##    word       n
##    <chr>  <int>
```

```
##  1 money   8766
##  2 love    8646
##  3 fun     5187
##  4 pay     4956
##  5 bad     4450
##  6 time    4237
##  7 hate    2525
##  8 level   2396
##  9 pretty  2320
## 10 star    2292
## # ... with 2,663 more rows
```

5. Kita dapat mengetahui banyaknya kata tiap kategori nrc untuk tiap aplikasi. Cobalah untuk mencari banyak kata tiap kategori nrc yang dikelompokkan berdasarkan nama aplikasi. **nilai 15**

```
user_reviews_nrc <- tidy_user_reviews %>%
  inner_join(get_sentiments("nrc")) %>%
  group_by(App) %>%
  count(sentiment, sort = TRUE) %>%
  spread(sentiment, n, fill = 0) %>%
  ungroup()
```

```
## Joining, by = "word"
```

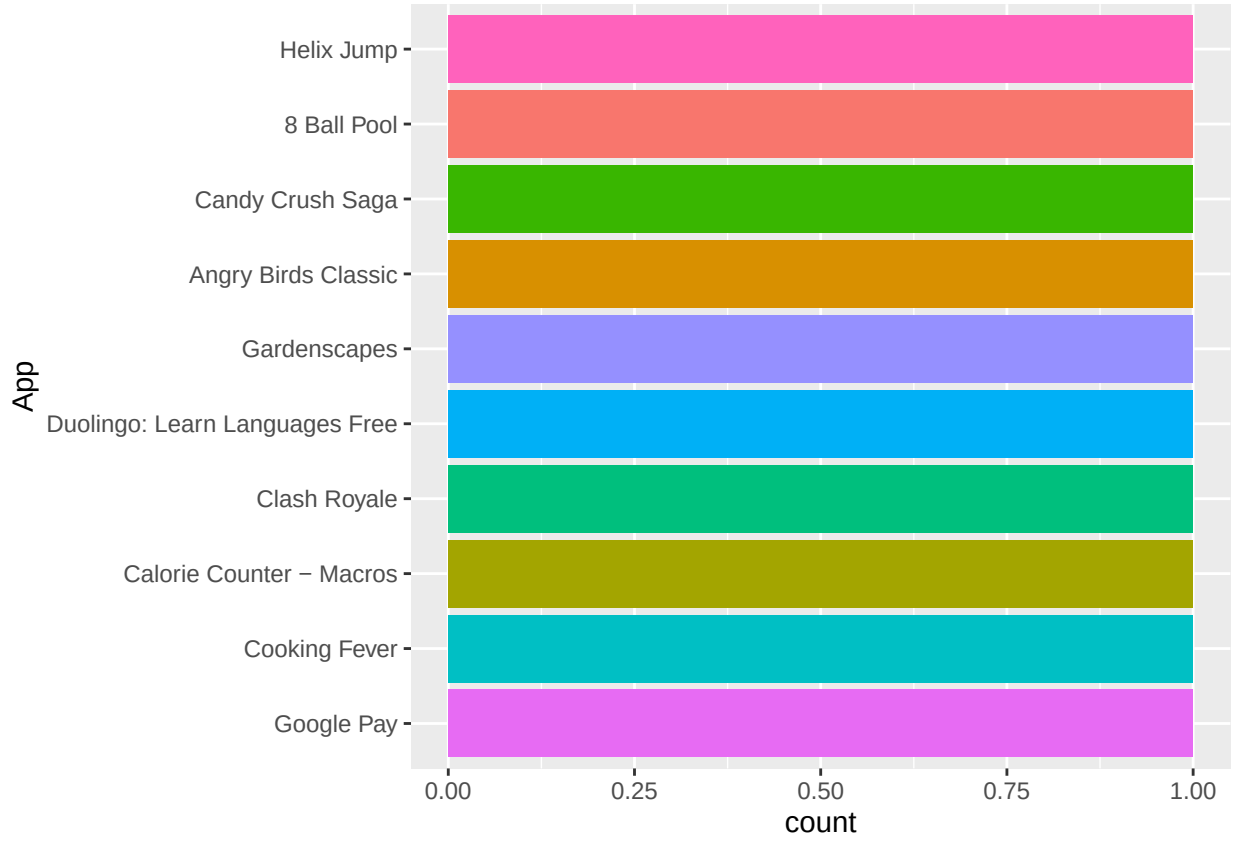
```
user_reviews_nrc
```

```
## # A tibble: 758 x 11
##    App   anger anticipation disgust  fear   joy negative positive sadness
##    <chr> <dbl>        <dbl>   <dbl> <dbl> <dbl>    <dbl>    <dbl>   <dbl>
##  1 10 B~     8           38      10    12    96       26      176      16
##  2 11st      8           20       2    12    13       18       27      10
##  3 1800~     8           40       4     6    34       16       76       6
##  4 21-D~    20           82      16    20    68       46      149      26
##  5 2Dat~     6           31       7     3    31       11       45       4
##  6 2GIS~     5           12       5     6     8       12       24       6
##  7 2ndL~     3           19       3     3    14       31       19      19
##  8 2Red~     3            7       3     2    13        5       22       3
##  9 30 D~    11           16       6    12    25       19       39      15
## 10 365S~     1            2       0     1     3        2        3       1
## # ... with 748 more rows, and 2 more variables: surprise <dbl>, trust <dbl>
```

6. Setelah mendapatkan jumlah kata tiap kategori tiap aplikasi, kita dapat mngetahui aplikasi mana yang memiliki kata dengan kategori 'surprise' terbanyak untuk tiap aplikasi. Kita akan memvisualisasikan dengan grafik batang 10 aplikasi dengan jumlah kata kategori 'surprise' terbanyak. ***nilai 20***

```
user_reviews_nrc %>%
  arrange(desc(user_reviews_nrc)) %>%
  top_n(10) %>%
  ggplot(aes(reorder(App, surprise), fill = App)) +
  geom_bar(show.legend = FALSE) +
  coord_flip() +
  labs(
    x = "App"
  )
```

```
## Selecting by trust
```



7. Selain menggunakan sentiment lexicon 'nrc', sentimen analisis juga dapat menggunakan sentiment lexicon 'bing'. Bing hanya akan memberikan label untuk tiap kata positif atau negatif saja. Carilah kata positif yang paling umum dan kata negatif yang paling sering digunakan saat memberikan review pada aplikasi! ***nilai 15***

```
bing_word_counts <- tidy_user_reviews %>%
  inner_join(get_sentiments("bing") ) %>%
  count(word, sentiment, sort = TRUE)
```

```
## Joining, by = "word"
```

```
bing_word_counts
```

```
## # A tibble: 2,286 x 3
##    word    sentiment     n
##    <chr>   <chr>     <int>
##  1 love    positive   4323
##  2 easy    positive   1988
##  3 fun     positive   1729
##  4 free    positive   1574
##  5 nice    positive   1531
##  6 awesome positive    941
```

```
##  7 bad     negative    890
##  8 amazing positive    820
##  9 hard    negative    707
## 10 issue   negative    701
## # ... with 2,276 more rows
```

8. Pembacaan data akan lebih mudah jika ditampilkan dalam bentuk grafik. Tampilkan grafik 10 kata positif dan negatif terbanyak! ***nilai 20***

```
# bing_word_counts %>%
#   ___(___) %>%
#   ___(10) %>%
#   ungroup() %>%
#   mutate(word = reorder(___, n)) %>%
#   ___(aes(___, n, fill = sentiment)) +
#   geom___(show.legend = FALSE) +
#   facet_wrap(~sentiment, scales = "free_y") +
#   labs(y = "Contribution to sentiment",
#        x = NULL) +
#   ___flip()
```

9. Penganalisis data membutuhkan jumlah kata tiap kategori yang belum digabung dengan sentiment lexicon untuk menghitung rasio positif, ratio negatif dan net sentiment. Bantulah penganalisis tersebut untuk mendapatkan jumlah kata tiap kategori dari data yang sudah dirapikan! ***nilai 15***

10. Selanjutnya penganalisis data ingin mendapatkan jumlah kata positif, jumlah kata negatif, rasio positif (jumlah kata positif/jumlah keseluruhan kata), rasio negatif (jumlah kata negatif/jumlah keseluruhan kata), dan net sentiment (jumlah kata positif - jumlah kata negatif) dengan menggunakan sentiment lexicon bing untuk tiap kategorinya. Tabel yang diinginkan oleh analisis adalah seperti berikut ***nilai 40***

Category	poisitive	negative	words	positive_ratio	negative_ratio	net_sentiment

11. Jangan lupa untuk menampilkan dalam bentuk grafik antara net_sentiment dengan kategori! ***nilai 25***

12. Penganalisis data ternyata juga ingin membandingkan antara positive_ratio tiap kategori aplikasi yang menggunakan sentiment lexicon bing dengan positive_ratio yang dihitung dari kolom Sentiment yang sudah tersedia dari awal. Apakah hasilnya sama atau ada perbedaan? Tampilkan juga grafik positive_ratio keduanya. ***nilai 50***

Kritik/saran/masukan/feedback/review/uneg-uneg:

========== SELESAI ==========