Sistema de recomendación de aplicaciones médicas

Aida María González

# Análisis de aplicaciones

## Cargar datos

La API de R recibirá los datos de las aplicaciones y los cargará para poder ser utilizados. Estos datos procederán de la base de datos de App Store y Google Play en la categoría de Health and Fitness.

## Tidy data frame

Después de cargar las aplicaciones, se creará una estructura de datos ordenada (**tidy data frame**) para poder ser manipulado en la recomendación. Constará del id y descripción de cada aplicación.

## Limpieza de descripción

Una vez que tengamos el **tidy data frame** creado, se hará la limpieza de la descripción.

Si se estudian las descripciones de las aplicaciones, se puede ver que todavía quedan algunas que tienen la descripción en otro idioma diferente al inglés o filas nulas. Se eliminan dichas aplicaciones y filas. Estas son las aplicaciones que vamos a eliminar:

Se eliminan números y signos de puntuación. Así es como quedarían las descripciones de las aplicaciones hasta el momento:

## Formato tidy text

Para hacer el análisis, se necesitará separar las palabras de las descripciones de cada aplicación. Se crearán dos formatos **tidy text** para una palabra y para pares de palabras. En este proceso limpiaremos las *palabras vacías* como artículos, pronombres, verbos, etc. y a cada una de las palabras restantes aplicaremos un proceso llamado *stemming*. El *stemming* es un método para reducir una palabra a su raiz. Así mismo, también limpiará cualquier resto de caracteres que no sean palabras.

### Aplicaciones no útiles

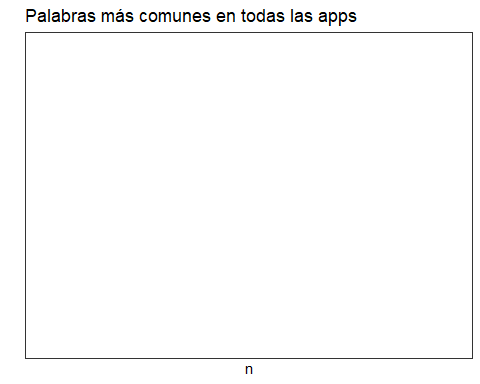
Se eliminarán las aplicaciones que no sean de utilidad. Primero se eliminarán las aplicaciones que contengan los siguientes pares de palabras: *weight loss*, *free trial*, *purchase subscription*, *weight lifting*, *trial period*, *lose weight*, *confirm purchase*, *fat burn*, *belly fat*, *burn workout*, *apple watch*, *heavy weight*, *slim down*, *burn calories*, *high intensity*, *body sculptor* y *for kids*.

Después se eliminarán las que contengan las siguientes palabras clave: *ovulation*, *fertil*, *wearable*, *baby*, *pregnancy*, *hypnosis*, *longevity*, *abs*, *heavy*, *watches*, *trial*, *membership*, *premium*, *subscription*, *mediation*, *purchase*, *yoga*, *pilates*, *zen*, *relax*, *mind*, *eat*, *food*, *sleep*, *tarot*, *slim*, *device*, *reiky*, *weight*, *pet*, *nutrition*, *calories*, *medication* y *kids*.

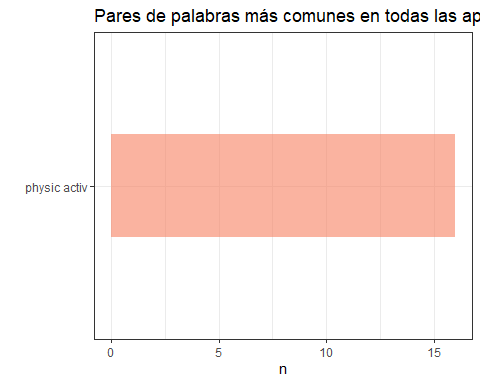
Situación de las aplicaciones final:

### Frecuencia de palabras

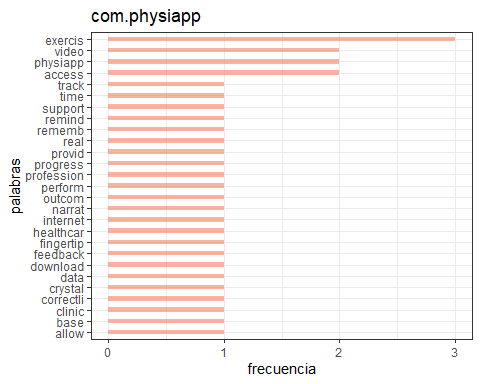
Podemos mirar las palabras más comunes de todas las applicaciones:

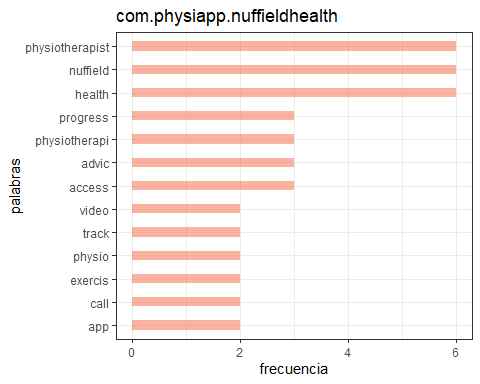


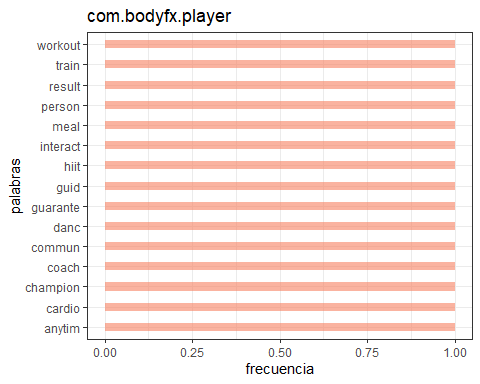
Los pares de palabras más comunes:

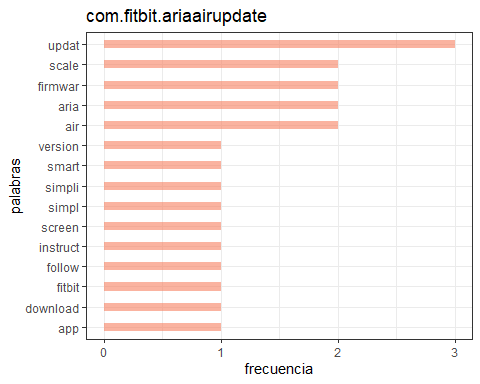


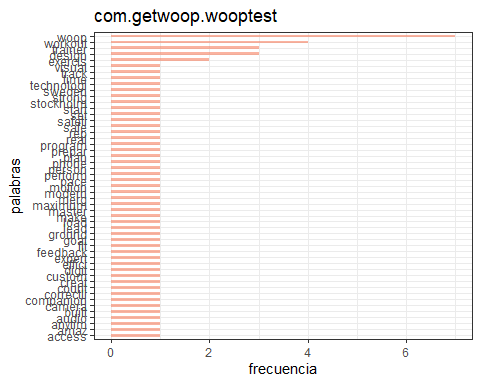
Y las palabras más comunes para cada aplicación.

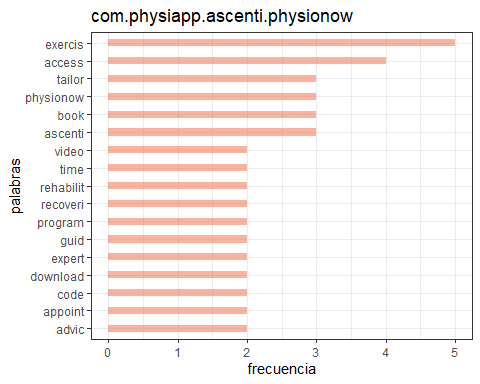


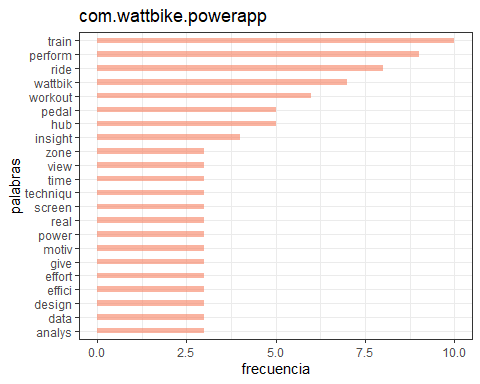


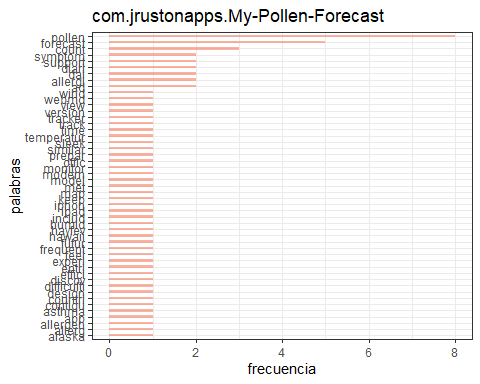


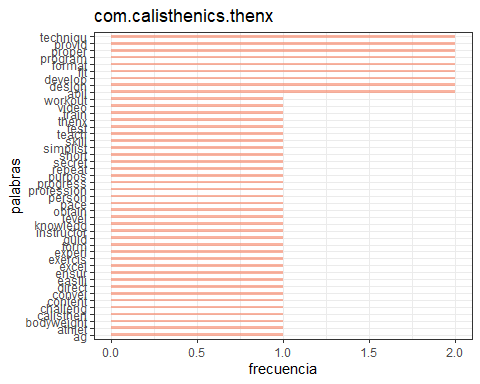


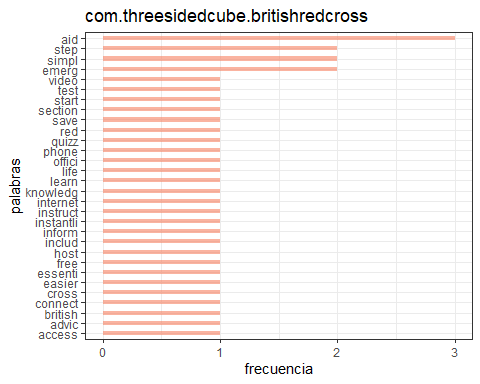


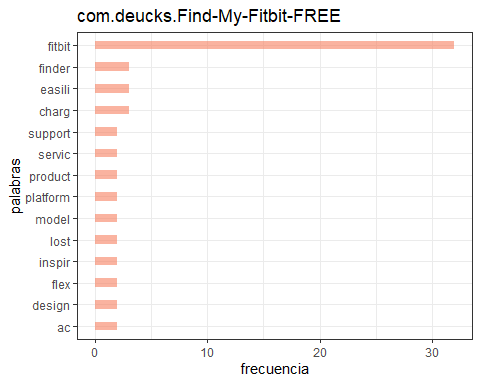


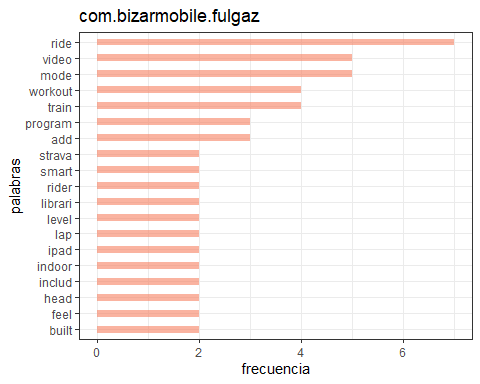


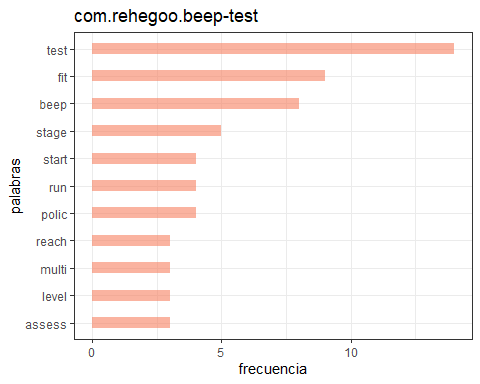


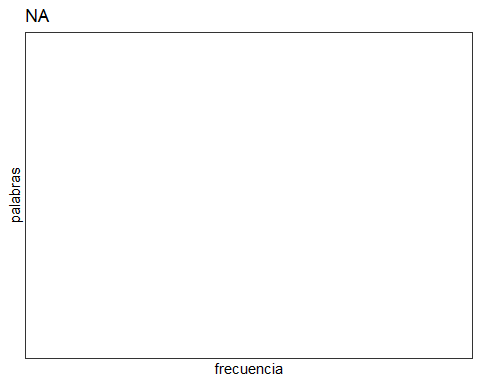


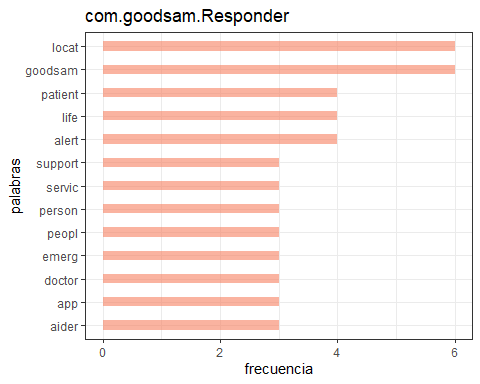


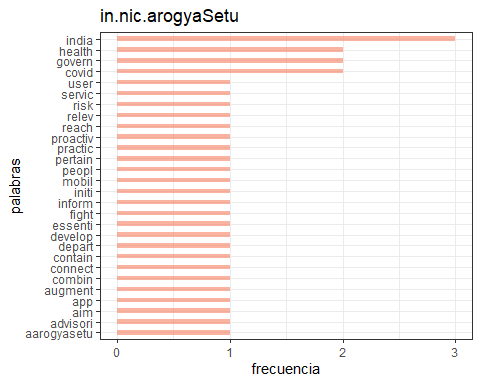


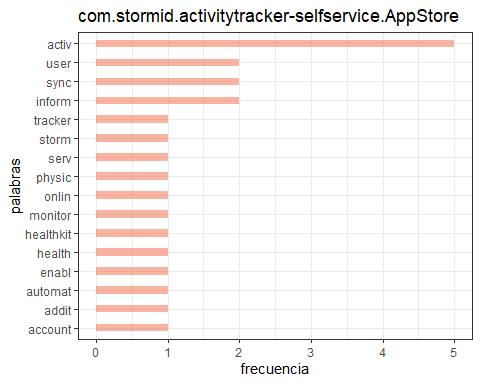


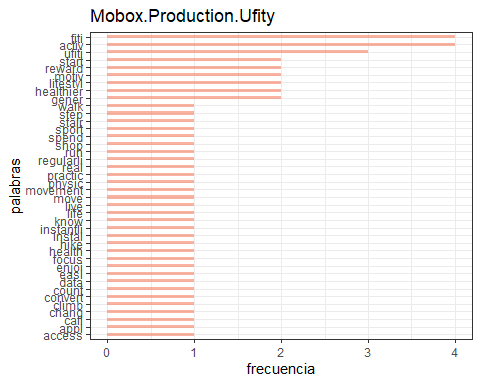


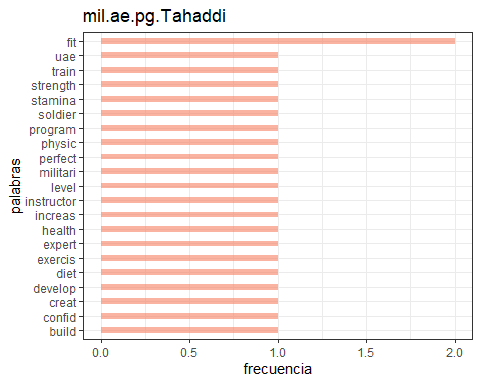


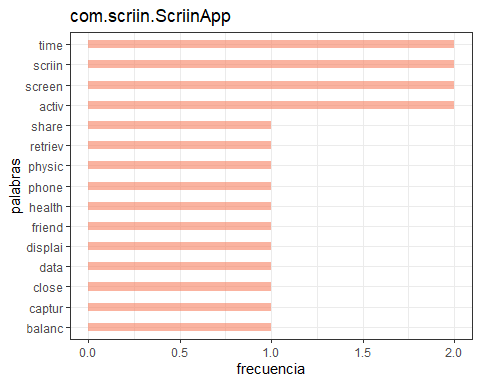


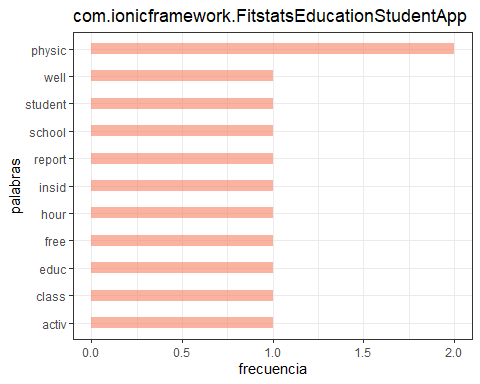


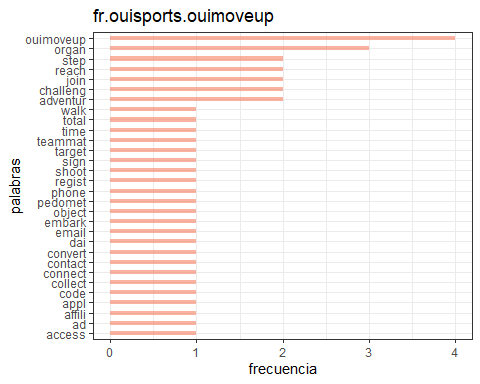


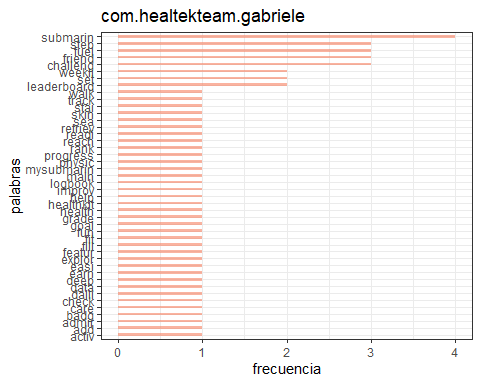


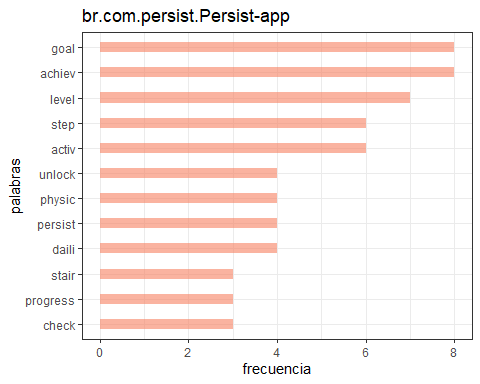


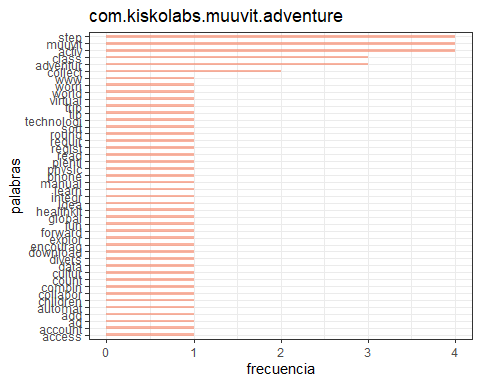


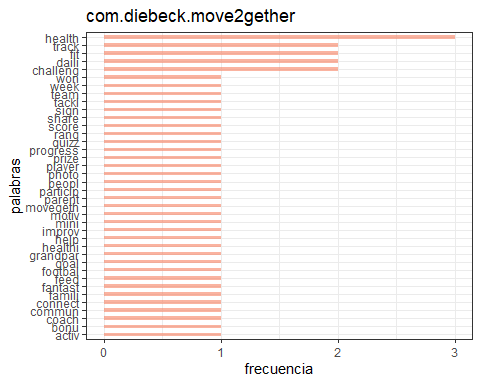


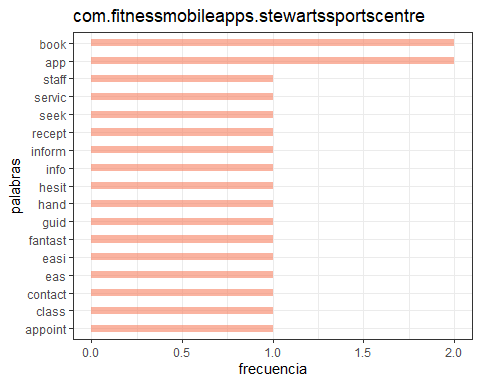


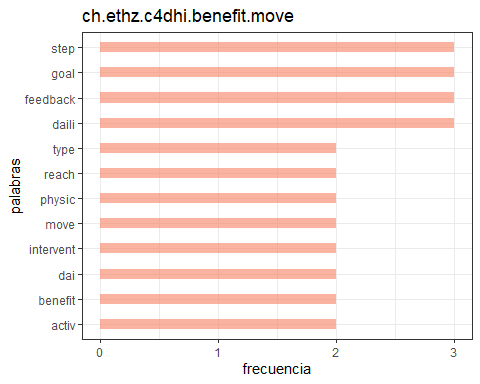


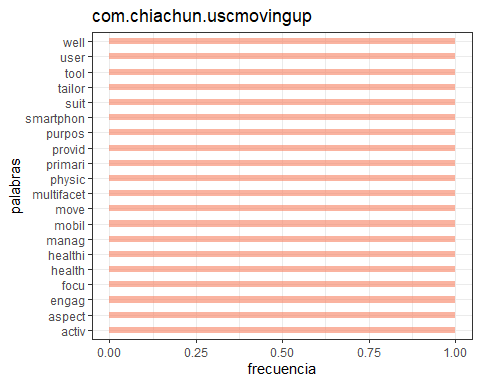


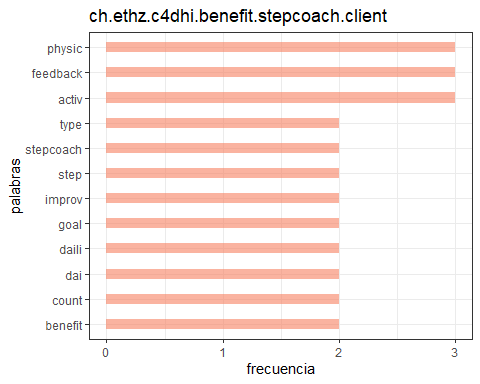


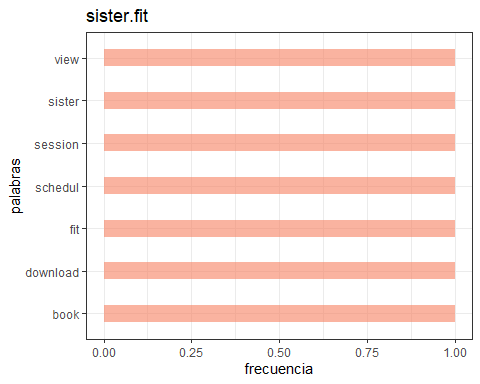


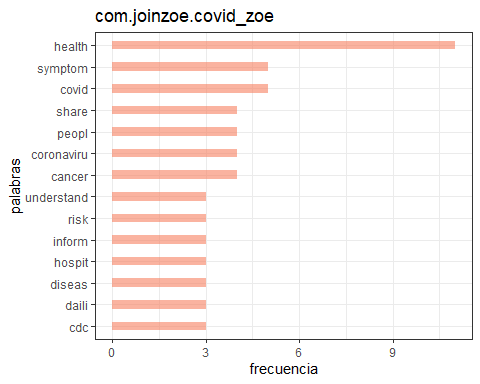


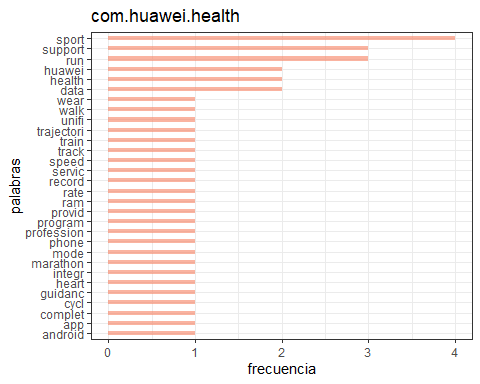


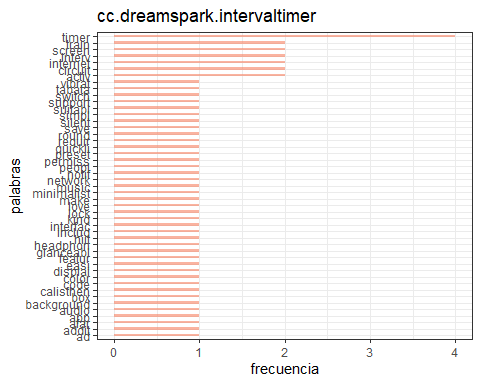


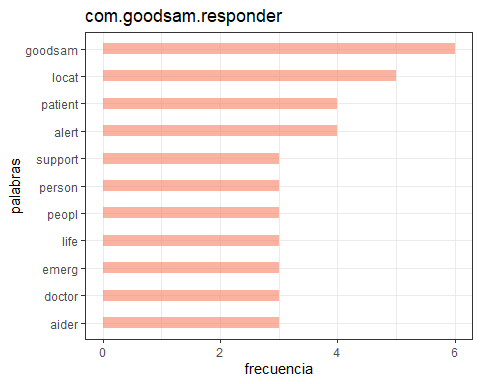


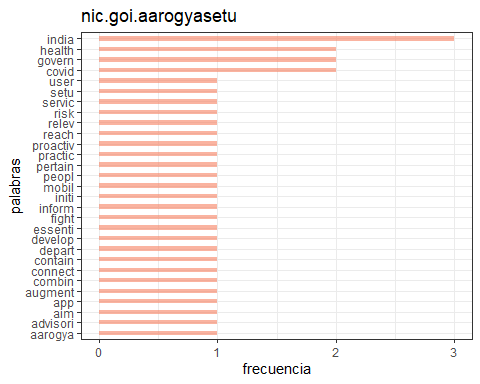


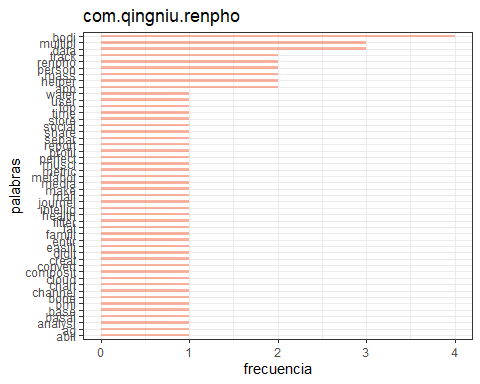


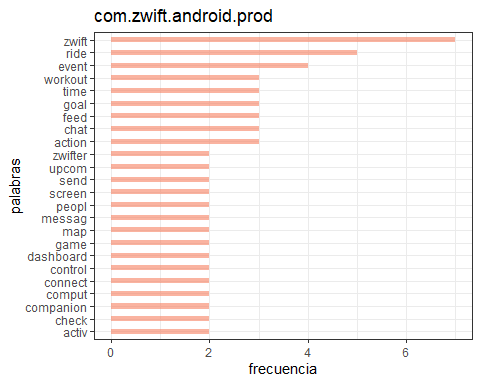


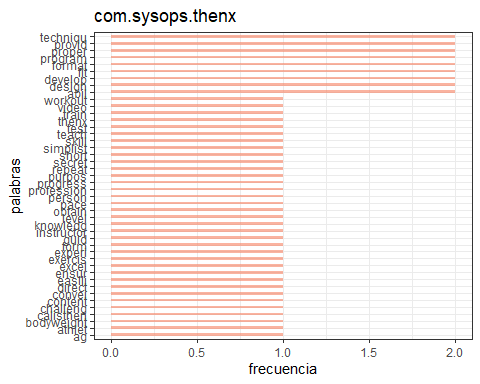


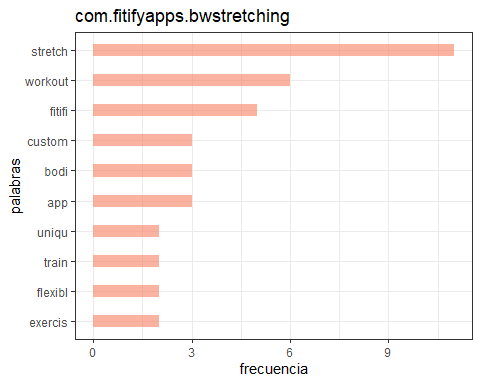










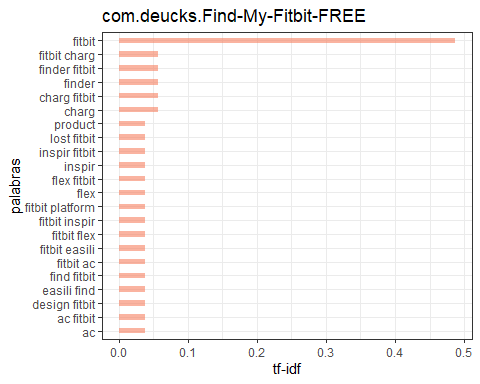


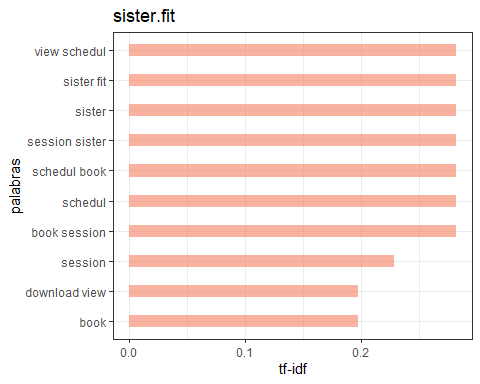
## Análisis de frecuencia de palabras y documentos: tf-idf

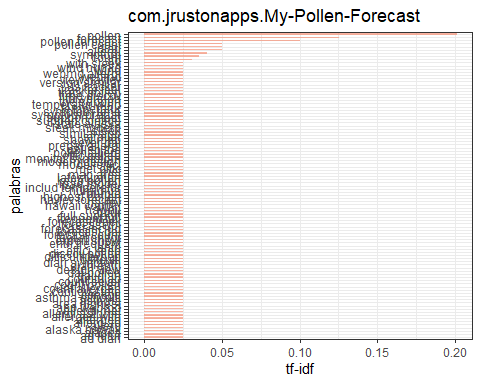
En la minería de datos, es muy común calcular la medida para conocer de qué trata un documento. Para ello, existe un algoritmo formado por dos ideas: **TF** y **IDF**. En primer lugar, se encuentra **Term Frequency** (TF), que mide la frecuencia de una palabra en un documento. En segundo lugar, se encuentra **Inverse Document Frequency** (IDF), donde el peso de una palabra aumenta a medida que su uso sea poco frecuente. Si juntamos estas dos medidas, tenemos como resultado **TF-IDF**, un algoritmo que da como resultado la importancia una palabra para cada documento.

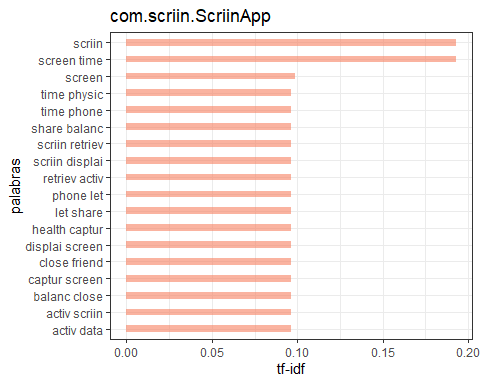
Calculamos ahora dichos valores para las frecuencias de una palabra y las de pares de palabras por separado; y además una palabra y pares de palabras juntos.

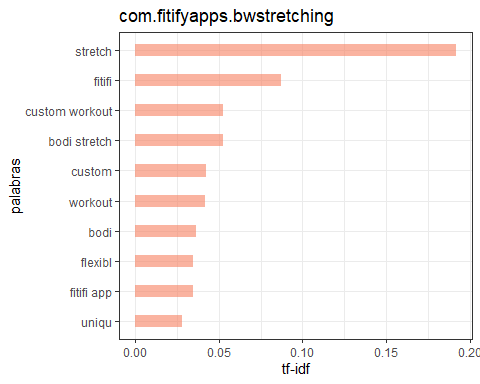
Con los valores de **TF-IDF** nos será más fácil valorar y analizar cada documento. Vamos a mostrar las gráficas para cada aplicación con las medidas de **TF-IDF** de una palabra y pares de palabras juntos más significativas.

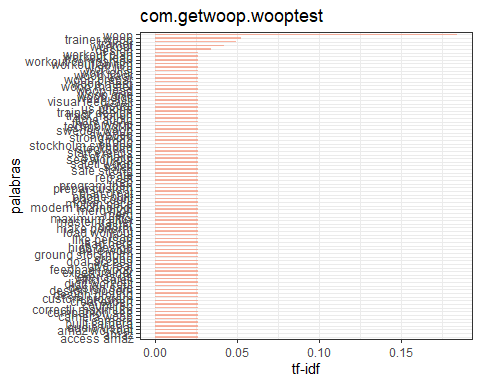


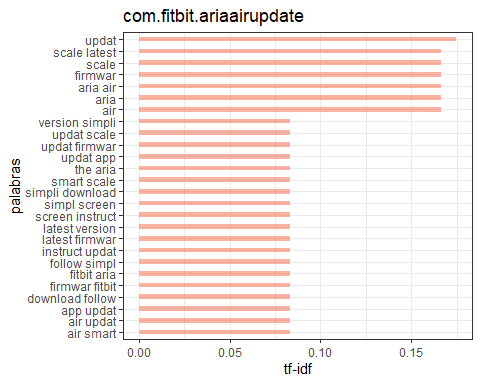


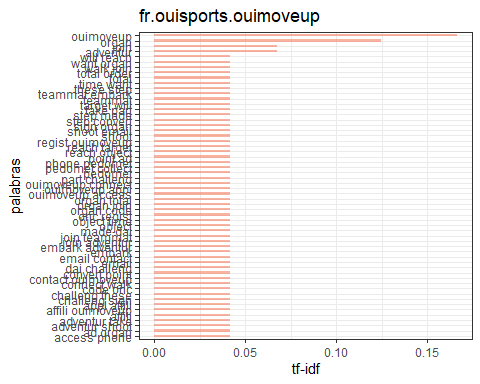


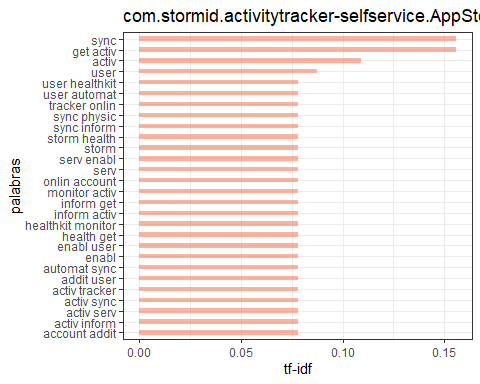


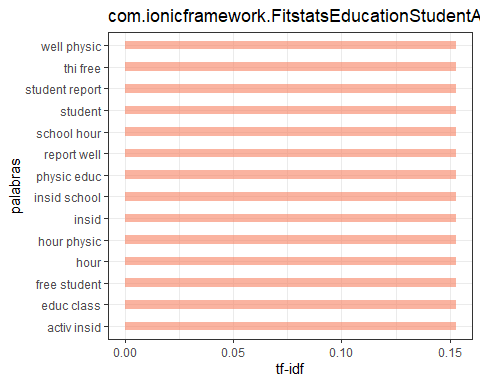


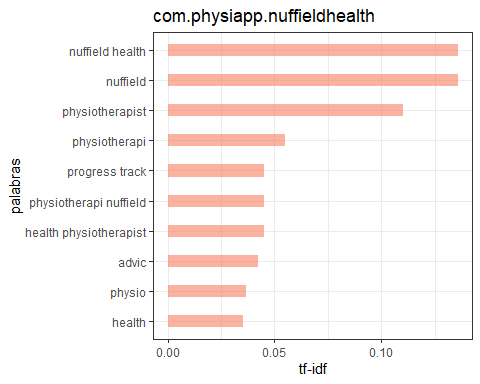


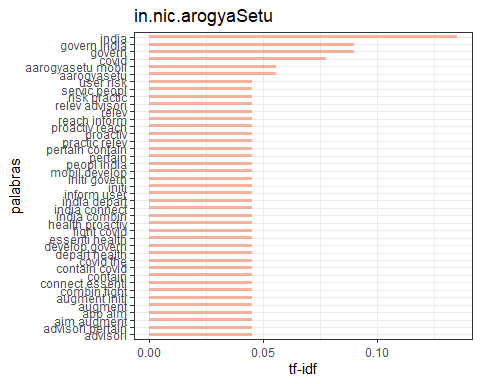


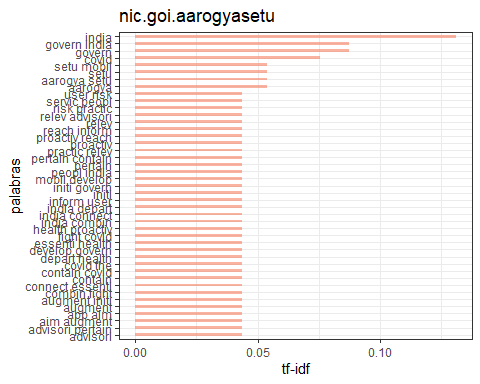


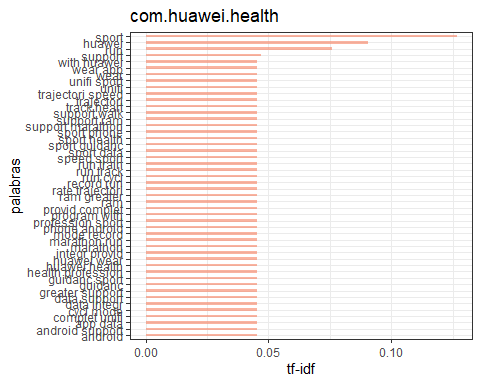


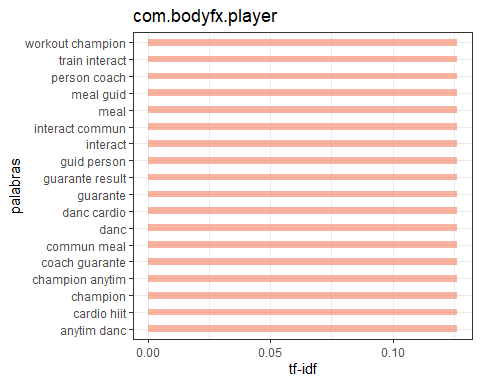


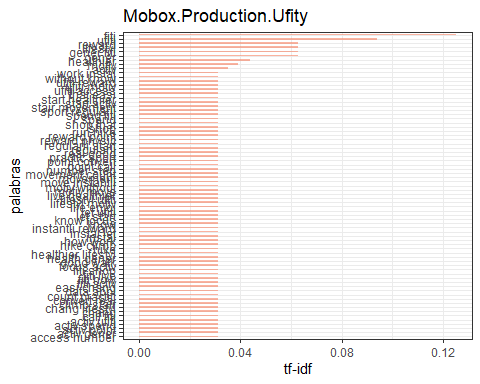


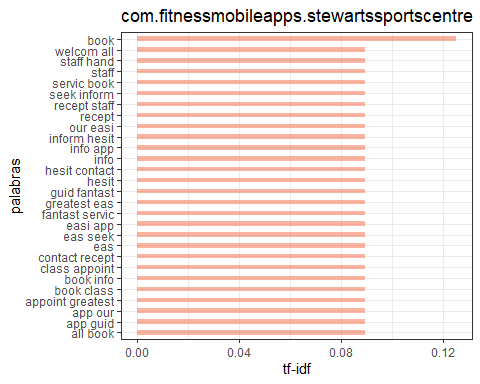


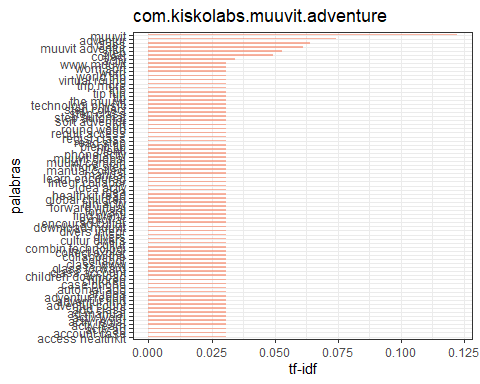


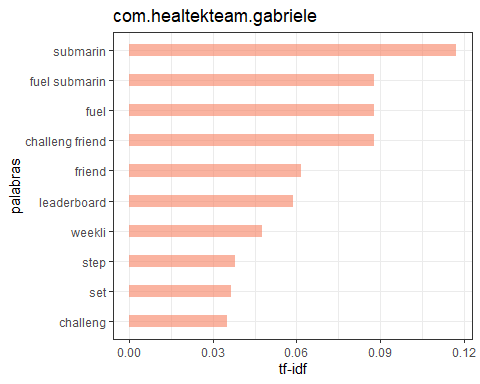


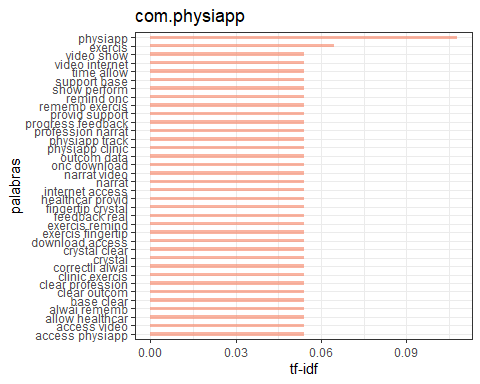


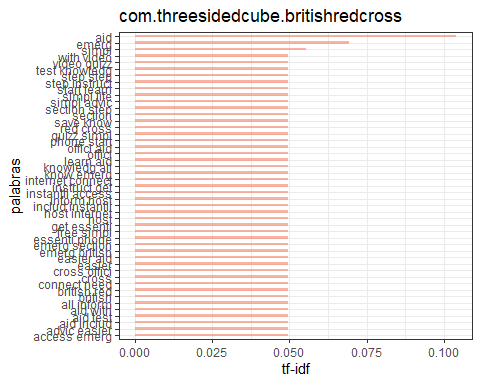


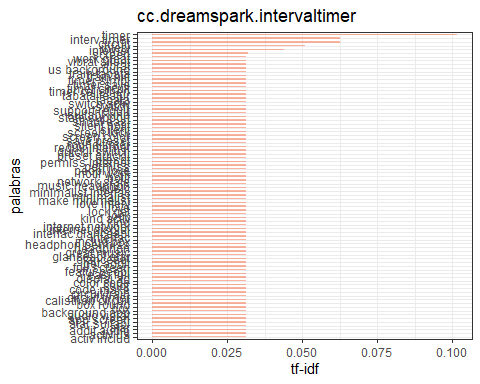


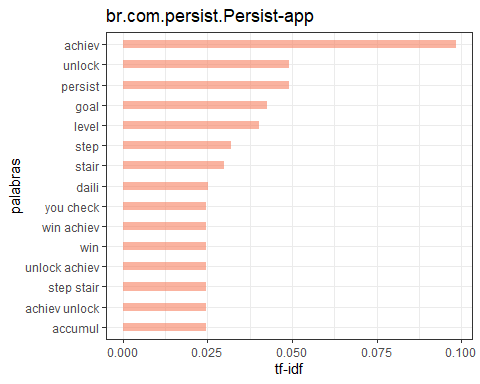


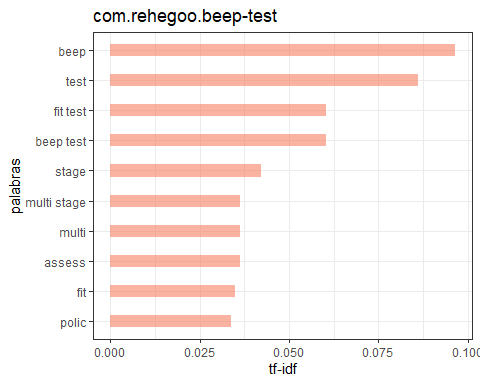


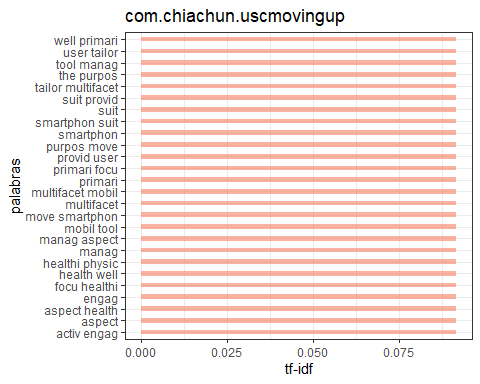


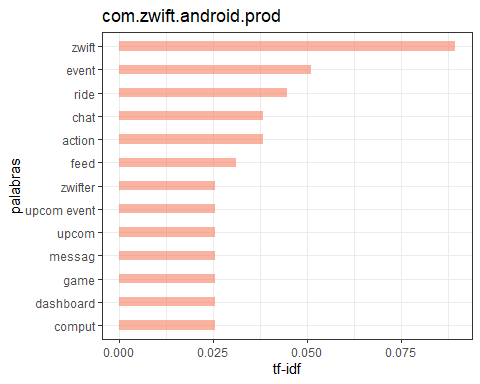


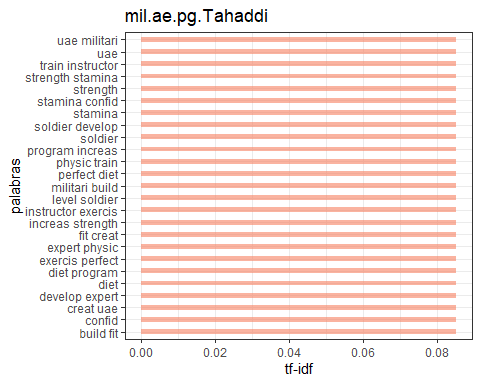


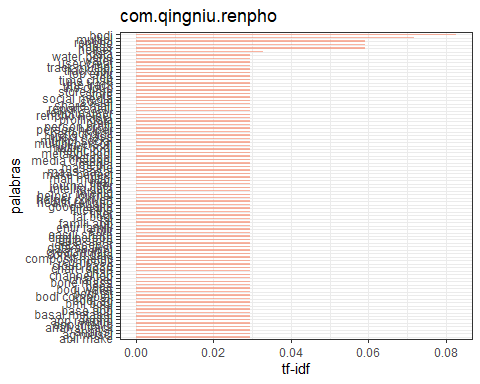


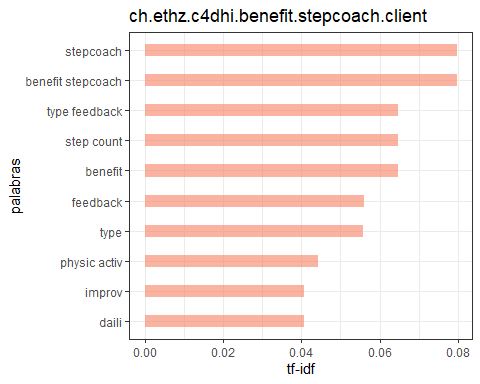


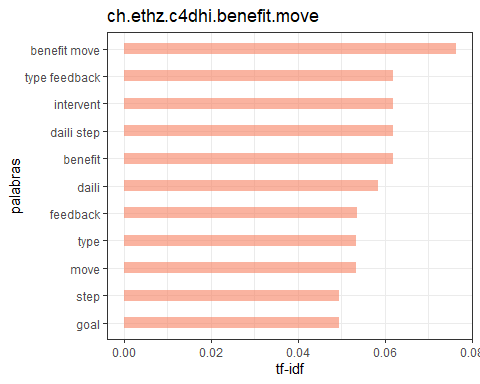


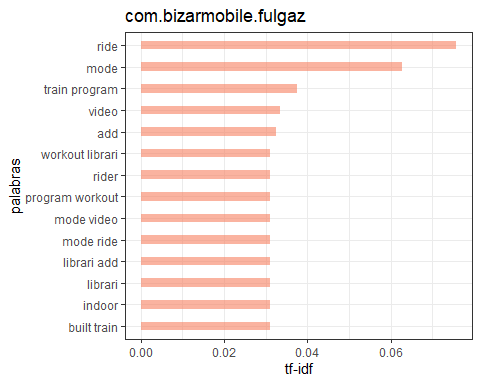


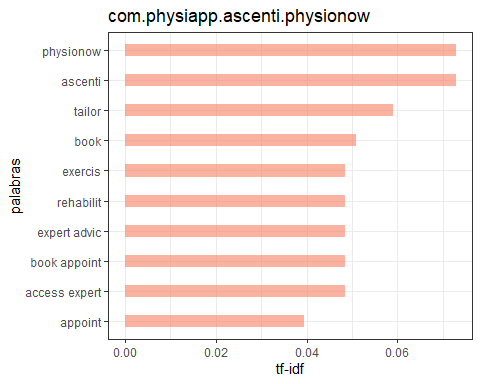


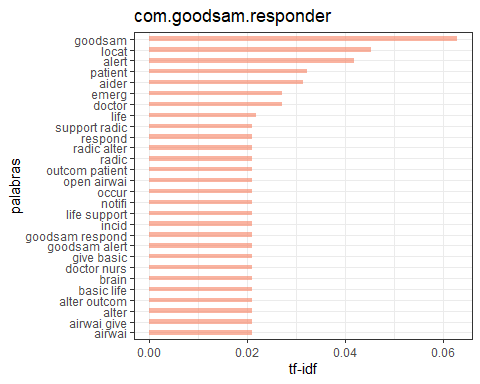


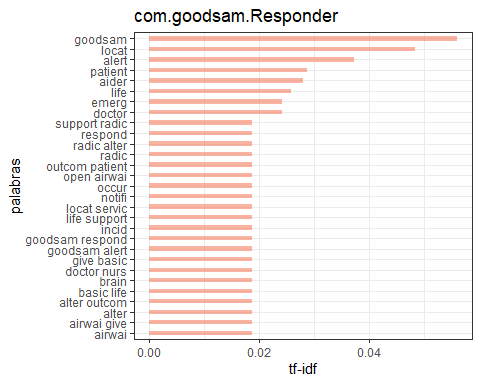


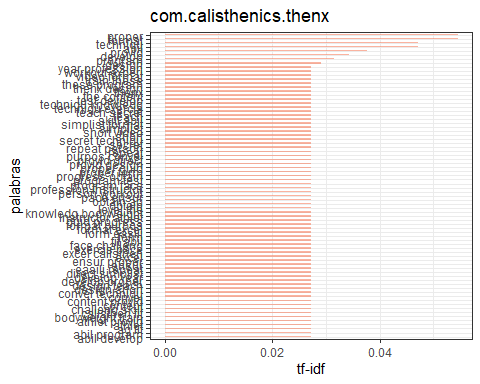


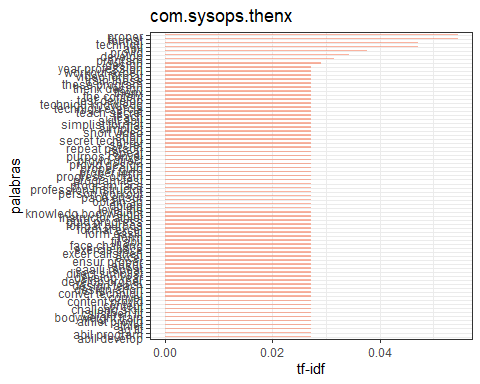


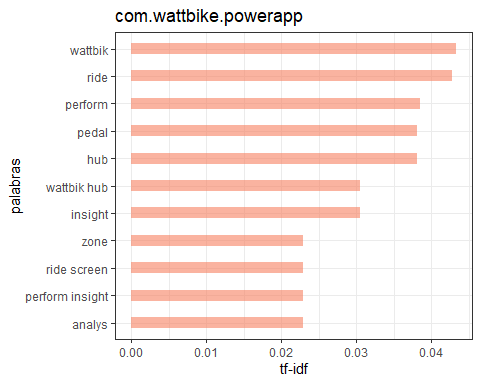


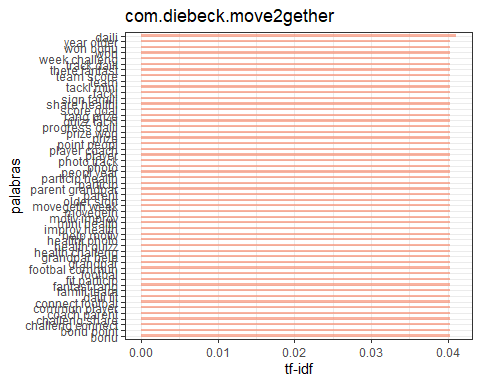


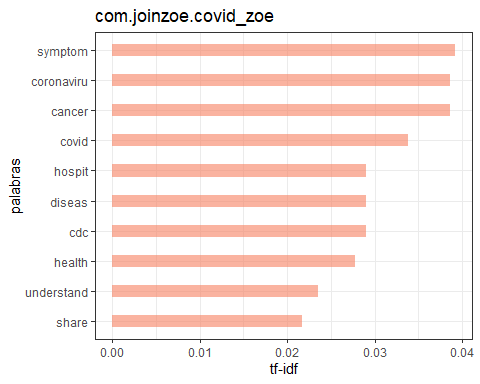


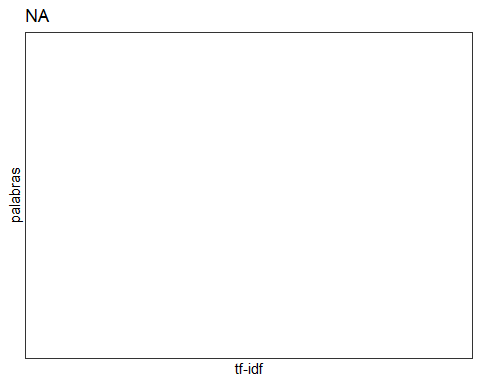












## Creación de DocumentTerm Matrix

El **DocumentTerm Matrix** (DTM) es una representación de la frecuencia de una palabra en cada uno de los documentos. Este formato va a ser necesario para el modelado de temas que habrá más adelante.

El **DTM** se puede convertir en una matriz para que se vean mejor los datos. Un ejemplo sería:

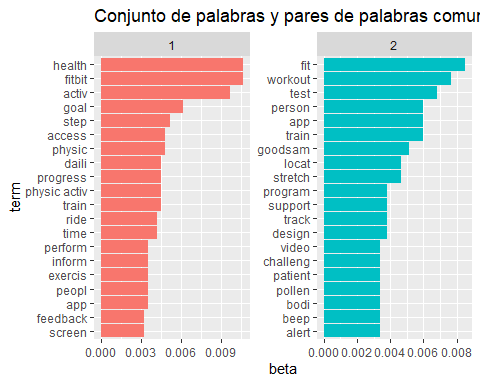
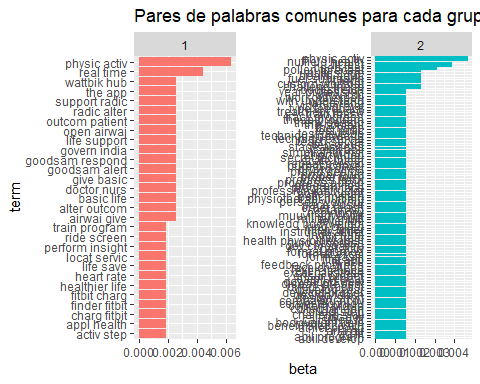
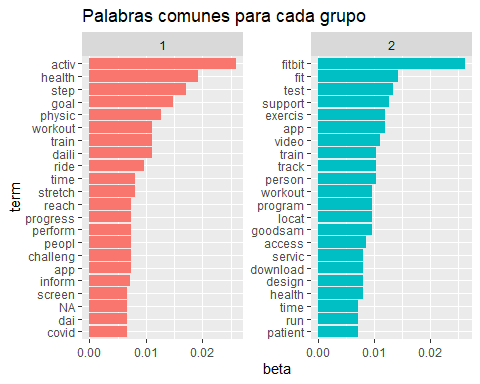
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | fitbit | test | stretch | health |
| com.deucks.Find-My-Fitbit-FREE | 32 | 0 | 0 | 0 |
| com.rehegoo.beep-test | 0 | 14 | 0 | 0 |
| com.fitifyapps.bwstretching | 0 | 0 | 11 | 0 |
| com.joinzoe.covid\_zoe | 0 | 1 | 0 | 11 |

## Modelado de temas

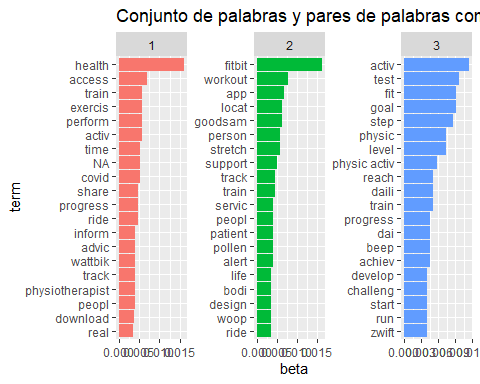
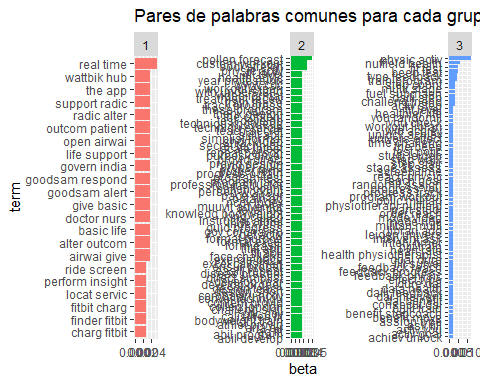
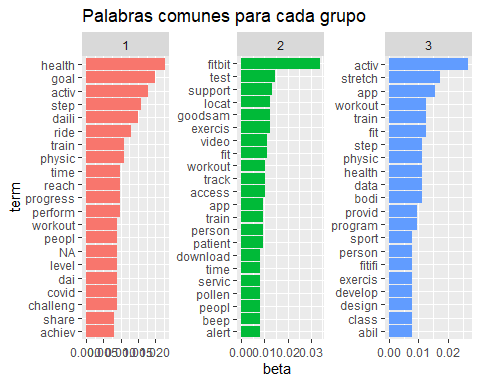
En minería de textos, existe un análisis llamado “agrupamiento” o “cluster”. Dicho análisis ayuda a poder detectar diferentes grupos naturales en un conjunto de documentos. Para tener un conocimiento más profundo sobre las aplicaciones, aplicaremos este análisis. Utilizaremos el método **Asignación Latente de Dirichlet** (LDA), el cual nos permitirá descubrir categorías no observables y entender mejor los documentos, en este caso, las aplicaciones.

Construiremos el modelo de temas con tres grupos naturales a partir del **DocumentTerm Matrix**.

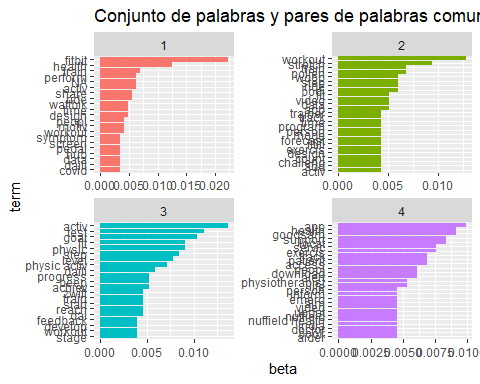
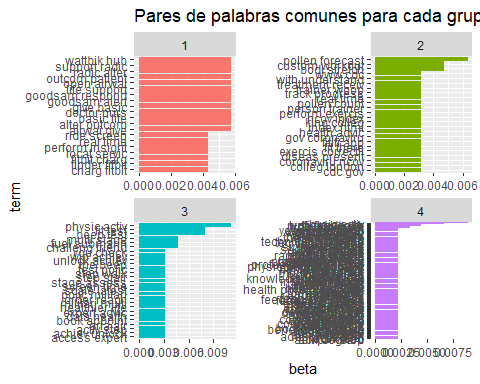
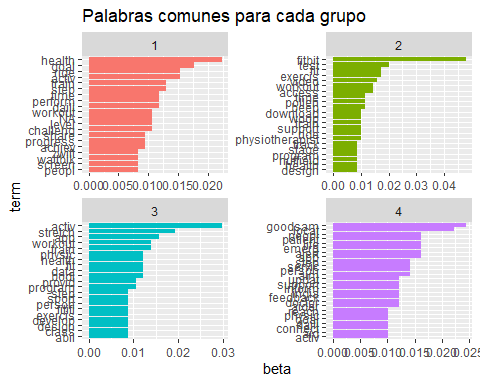
Para k=2:



Para k=3:



Para k=4:

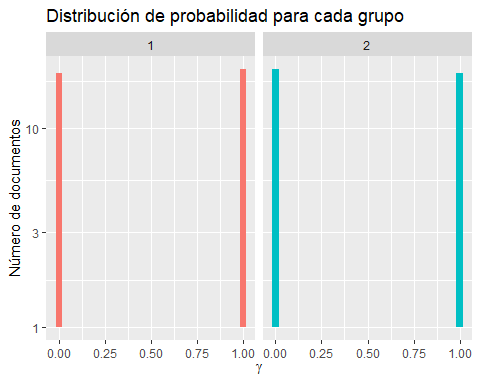


El valor **beta** es calculado para cada una de las palabras. Sirve para medir la probabilidad de que esa palabra sea generada para cada uno de los grupos. Por ejemplo, la palabra *heart* tiene una probabilidad de ser generada por el grupo 1 de 0.007750085, por el grupo 2 de 0.01179043 y por el grupo 3 de 0.007017814.

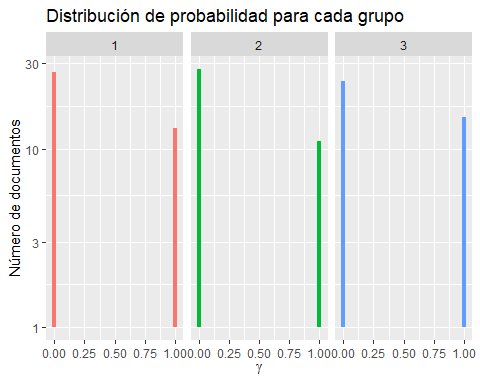
Ahora podemos explorar qué grupos están asociados a cada documento:

1. Una sola palabra:

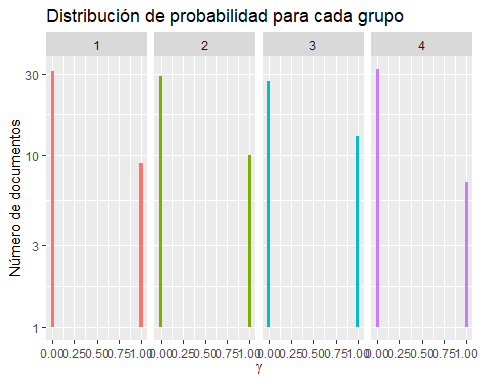
1.1. Con k = 2:



1.2. Con k=3:

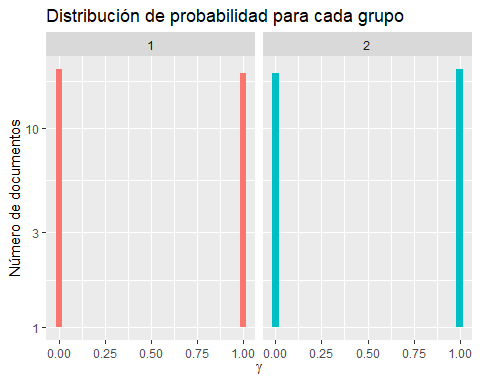


1.3. Con k=4:



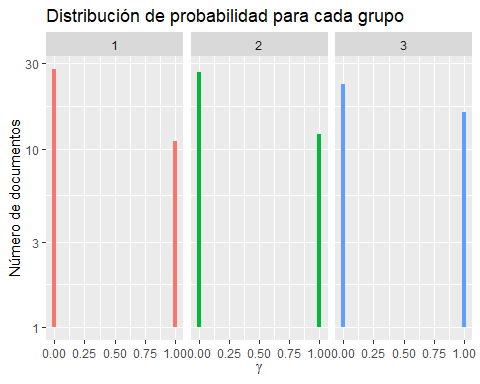
1. Pares de palabras:

2.1. Con k=2:



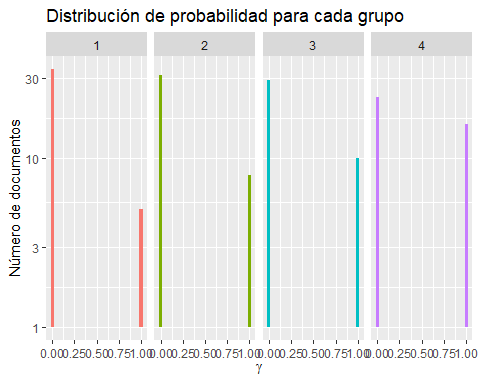
2.2. Con k=3:

## # A tibble: 120 x 3  
## document topic gamma  
## <chr> <int> <dbl>  
## 1 com.physiapp.nuffieldhealth 1 0.000349  
## 2 com.rehegoo.beep-test 1 0.000181  
## 3 br.com.persist.Persist-app 1 0.000180  
## 4 com.jrustonapps.My-Pollen-Forecast 1 0.000371  
## 5 com.wattbike.powerapp 1 1.000   
## 6 ch.ethz.c4dhi.benefit.stepcoach.client 1 0.000602  
## 7 com.bizarmobile.fulgaz 1 0.000241  
## 8 com.deucks.Find-My-Fitbit-FREE 1 0.999   
## 9 com.fitifyapps.bwstretching 1 0.000263  
## 10 com.healtekteam.gabriele 1 0.000432  
## # ... with 110 more rows



2.3. Con k=4:

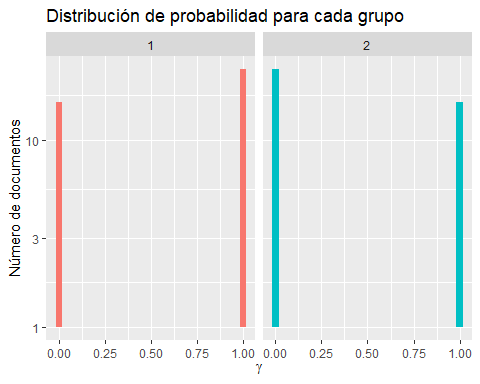
## # A tibble: 160 x 3  
## document topic gamma  
## <chr> <int> <dbl>  
## 1 com.physiapp.nuffieldhealth 1 0.000289  
## 2 com.rehegoo.beep-test 1 0.000150  
## 3 br.com.persist.Persist-app 1 0.000149  
## 4 com.jrustonapps.My-Pollen-Forecast 1 0.000307  
## 5 com.wattbike.powerapp 1 1.000   
## 6 ch.ethz.c4dhi.benefit.stepcoach.client 1 0.000499  
## 7 com.bizarmobile.fulgaz 1 0.000200  
## 8 com.deucks.Find-My-Fitbit-FREE 1 0.999   
## 9 com.fitifyapps.bwstretching 1 0.000218  
## 10 com.healtekteam.gabriele 1 0.000358  
## # ... with 150 more rows



1. Conjunto de una palabra y pares de palabras:

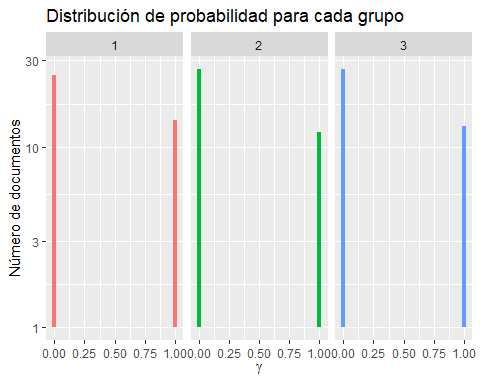
3.1. Con k=2:

## # A tibble: 80 x 3  
## document topic gamma  
## <chr> <int> <dbl>  
## 1 com.deucks.Find-My-Fitbit-FREE 1 1.000   
## 2 com.rehegoo.beep-test 1 0.000120  
## 3 com.fitifyapps.bwstretching 1 0.000174  
## 4 com.joinzoe.covid\_zoe 1 1.000   
## 5 com.wattbike.powerapp 1 1.000   
## 6 <NA> 1 0.996   
## 7 br.com.persist.Persist-app 1 1.000   
## 8 com.jrustonapps.My-Pollen-Forecast 1 0.000250  
## 9 com.bizarmobile.fulgaz 1 0.000154  
## 10 com.getwoop.wooptest 1 0.000263  
## # ... with 70 more rows



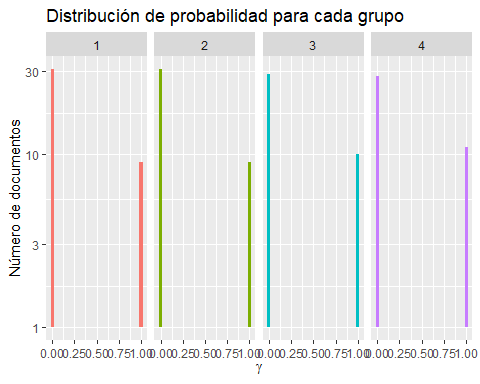
3.2. Con k=3:

## # A tibble: 120 x 3  
## document topic gamma  
## <chr> <int> <dbl>  
## 1 com.deucks.Find-My-Fitbit-FREE 1 0.000167  
## 2 com.rehegoo.beep-test 1 0.000107  
## 3 com.fitifyapps.bwstretching 1 0.000155  
## 4 com.joinzoe.covid\_zoe 1 1.000   
## 5 com.wattbike.powerapp 1 1.000   
## 6 <NA> 1 0.993   
## 7 br.com.persist.Persist-app 1 0.000109  
## 8 com.jrustonapps.My-Pollen-Forecast 1 0.000223  
## 9 com.bizarmobile.fulgaz 1 0.000138  
## 10 com.getwoop.wooptest 1 0.000234  
## # ... with 110 more rows



3.3. Con k=4:

## # A tibble: 160 x 3  
## document topic gamma  
## <chr> <int> <dbl>  
## 1 com.deucks.Find-My-Fitbit-FREE 1 1.000   
## 2 com.rehegoo.beep-test 1 0.0000920  
## 3 com.fitifyapps.bwstretching 1 0.000133   
## 4 com.joinzoe.covid\_zoe 1 1.000   
## 5 com.wattbike.powerapp 1 1.000   
## 6 <NA> 1 0.991   
## 7 br.com.persist.Persist-app 1 0.0000938  
## 8 com.jrustonapps.My-Pollen-Forecast 1 0.000191   
## 9 com.bizarmobile.fulgaz 1 0.000118   
## 10 com.getwoop.wooptest 1 0.000201   
## # ... with 150 more rows



Vemos la distribución de probabilidad para cada tema y los valores gamma de cada documento. Cuanto más se acerque al valor 1, más pertenece a dicho tema.