Topic Modeling

Daniel Jiménez M.

Universidad Nacional de Colombia

05 -11 -2020

Bibliografía

- Silge & Robinson, Text Mining, 2020, CRAN
- Fradejas, J.,Estilometría y análisis de texto con R para filólogos,2020,UVA.

Librerías

```
library(tidyverse)
library(LaplacesDemon)
library(tidytext)
library(GGally)
library(MASS)
library(topicmodels)
library(stopwords)
theme_set(theme_classic())
```

¿Qué es topic modeling?

Según Silge[1]

> *'Topic modeling is a method for unsupervised classification of such documents, similar to clustering on numeric data, which finds natural groups of items even when we're not sure what we're looking for.'*[2]

[1] Julia Silge es Científica de Datos y Software Engineer en Rstudio, los invito a visitar su Blog :https://juliasilge.com/

[2] https://www.tidytextmining.com/topicmodeling.html

¿Qué es topic modeling?

Algunas definiciones útiles :

- Modelado de temas ;
- Es un identificador de documentos ;
- Un clasificador de textos;
- Modelos estadísticos que descubren los patrones en una colección de texto.

Latent Dirichlet Allocation (LDA)

Para el `Topic Modeling` se suele usar el **Latent Dirichlet Allocation**, el cual consiste en un modelo que identifica patrones dentro de una colección de documentos y los agrupa con base a ellos. Estos modelos se basan en el entendimiento probabilístico de los datos, a través de parámetros jerárquicos Bayesianos . , donde modela el `input` a través de una mixtura subyacente de los topics.

Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation (LDA)

Para lo anterior es necesario entender la siguiente formula:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

La anterior ecuación es la forma básica del calculo bayesiano de probabilidades. Para entenderlo mejor se desarrolla un ejemplo de Henrik Singmann.

Probabilidad Bayesiana

¿Cuál es la probabilidad de ir al infiero, dado que se ha asociado con el demonio Laplace?

Descompongamos esto

- A $\Longrightarrow$ Infierno
- B $\Longrightarrow$ Asociarse

Entonces la ecuación Bayesiana queda así

$$P(Infierno|Asociarse) = \frac{P(Asociarse|Infierno)P(Infierno)}{P(Asociarse)}$$

Probabilidad Bayesiana

Pero Ojo!!!!!!!!!!!!!

$$P(Infierno|Asociarse) \neq P(Asociarse|Infierno)$$

Lo anterior se le conoce como la falacia de las probabilidades! Usted no lo haga, eso no es de DIOS!

Probabilidad Bayesiana

Veamos lo anterior con datos:

- 6 de 9 Personas fueron al infierno
- 5 de 7 Personas fueron al cielo
- 75% de las personas fueron al infierno
- 25% de las personas fueron al cielo

Probabilidad Bayesiana

- $P(Asociarse|Infierno) \Longrightarrow 6/9 = 0.6666667$
- $P(Asociarse|Cielo) \Longrightarrow 5/7 = 0.7142857$
- $P(Infierno) = 75\%$
- $P(Cielo) = 25\%$

Probabilidad Bayesiana

Finalmente la probabilidad de dicho evento es

$$P(Infierno|Asociarse) = \frac{(6/9) * (0.75)}{(6/9) * (0.75) + (5/7) * (0.25)} = 0.73$$

Probabilidad Bayesiana

En código lo anterior es:

```
PrA <- c(0.75,0.25)
PrBA <- c(6/9, 5/7)
BayesTheorem(PrA, PrBA)
```

```
## [1] 0.7368421 0.2631579
## attr(,"class")
## [1] "bayestheorem"
```

Latent Dirichlet Allocation (LDA)

Para poder generar es necesario trabajar con `tf-idf` que es la relación inversa y discriminada de los terminos que permite entender cuales son las palabras que más generan contexto dentro de un documento.

Latent Dirichlet Allocation (LDA)

Entendiendo el LDA se puede decir que :

- Seleccionar N palabra $\sim$ Poisson $(\mho)$
- Seleccionar $\theta \sim \ \alpha$
- Siendo así N palabras dado w_n

Latent Dirichlet Allocation (LDA)

Finalmente lo anterior se traduce en lo siguiente :

$$p(\theta|\alpha) = \frac{\Gamma(\sum_i^k \alpha)}{\Pi_{i=1}^k \Gamma(\alpha_i)} \theta_1^{\sum_{i,n}^1 \alpha_k}$$

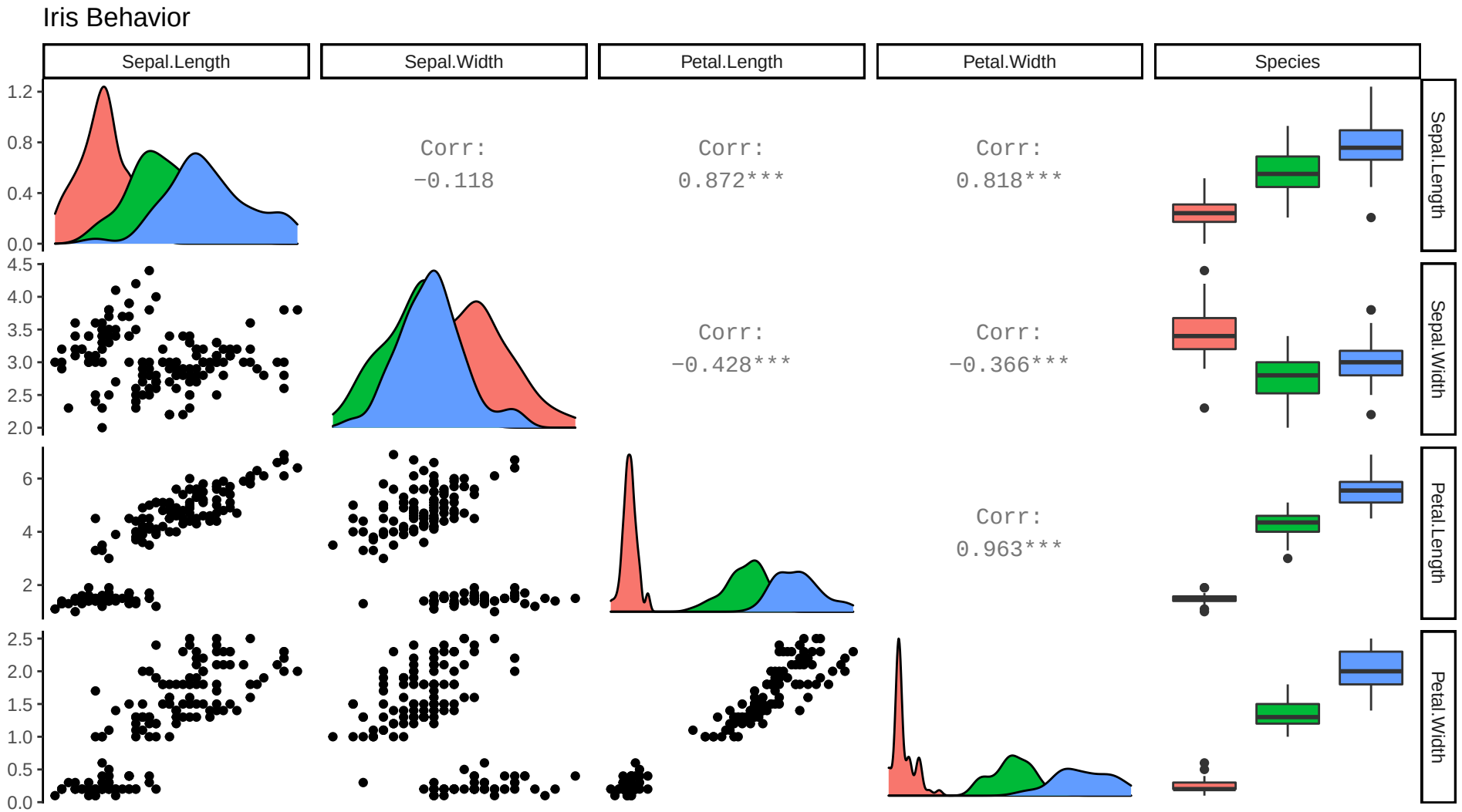
Con esto se desarrolla el modelo probabilistico de la siguiente manera

$$p(D|\alpha,\beta) = \Pi_{d=1}^m \int p(\theta|\alpha)(\Pi_{n=1}^{N_d} \sum p(Z_{dn}|\theta_d) p(w_d n|z_n d\beta)) d\theta_d$$

Latent Dirichlet Allocation (LDA)

Un ejemplo del LDA con números seria el siguiente :

Suponga el siguiente comportamiento de los datos



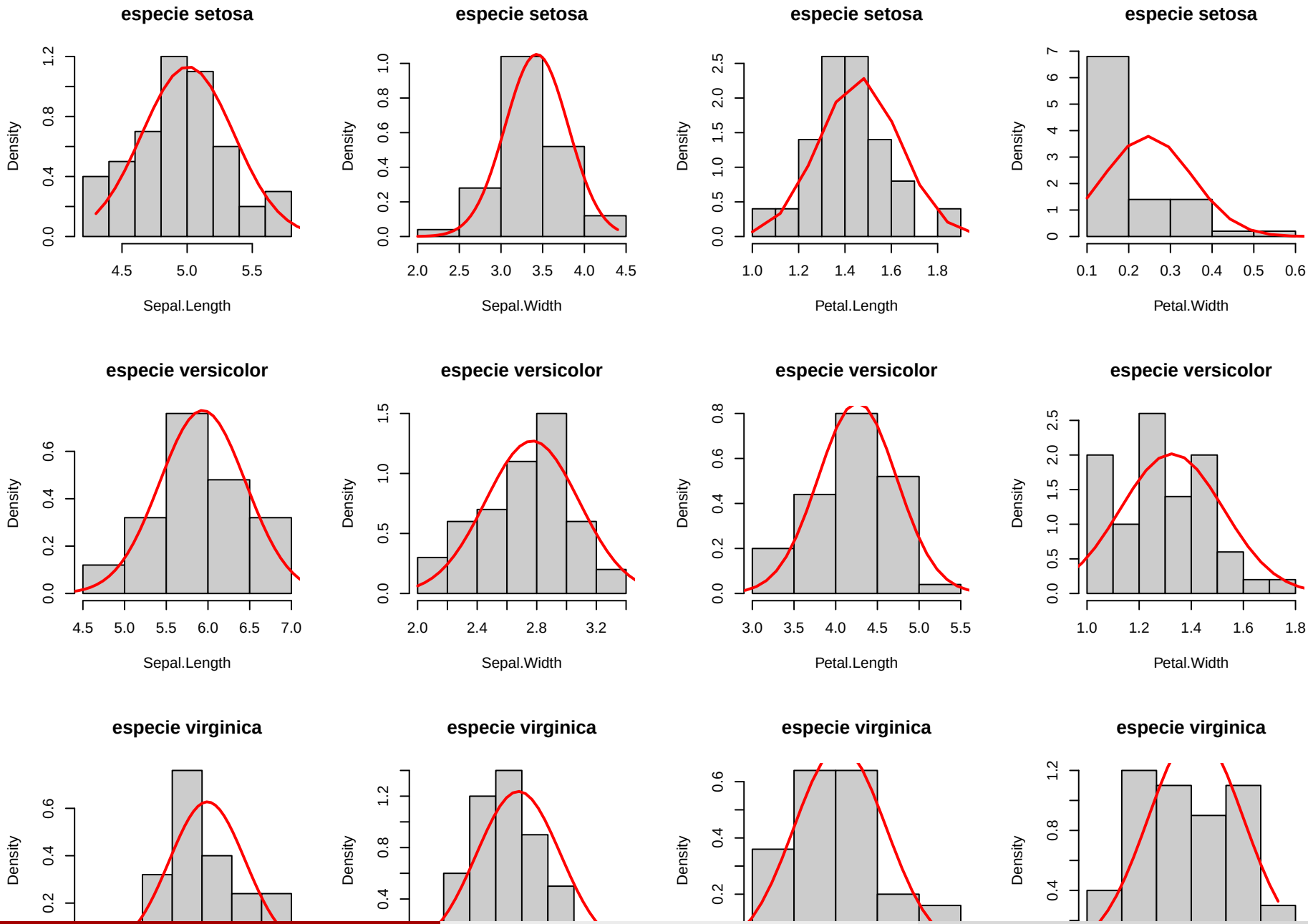
Latent Dirichlet Allocation (LDA)

Suponga que la probabilidad de previa es

```
prior<-length(iris$Species[iris$Species=='versicolor'])/length
prior
```

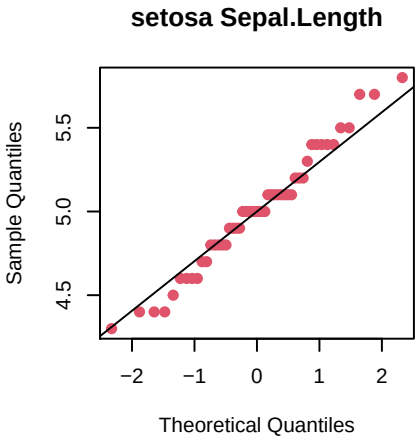
```
## [1] 0.3333333
```

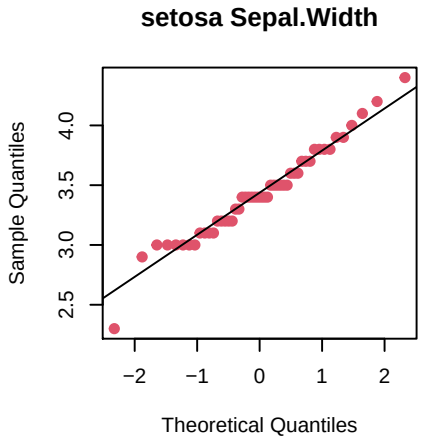
Latent Dirichlet Allocation (LDA)

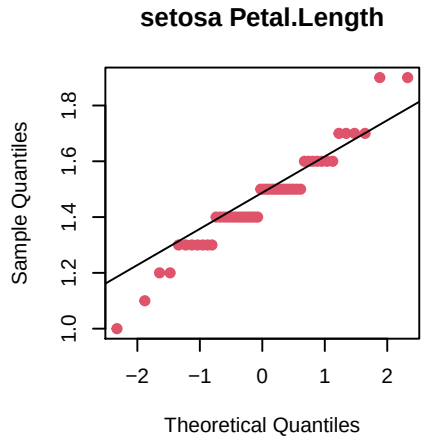


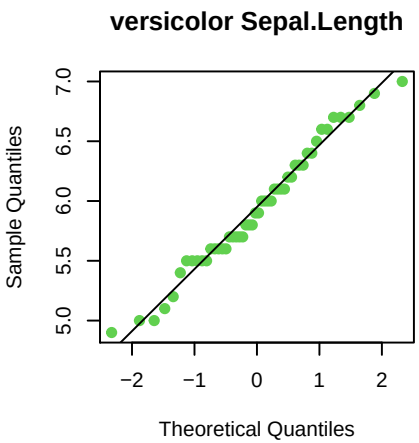
Latent Dirichlet Allocation (LDA)

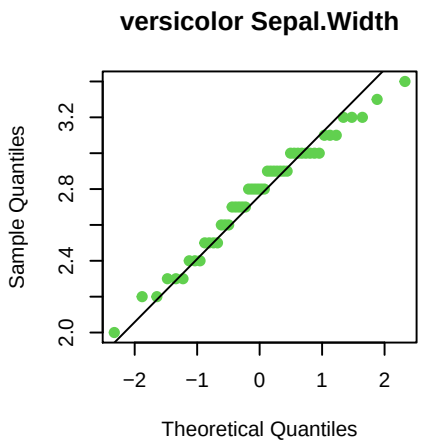
Verificando con la prueba de normalidad

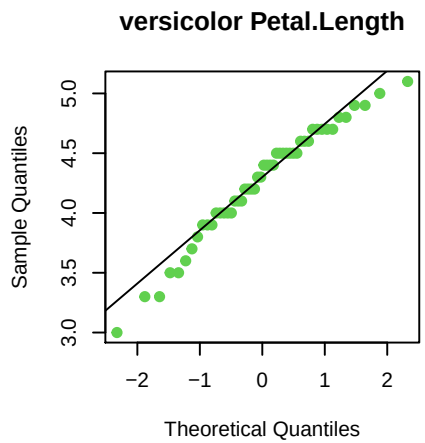


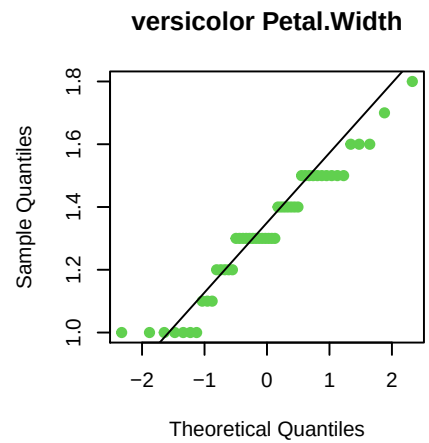


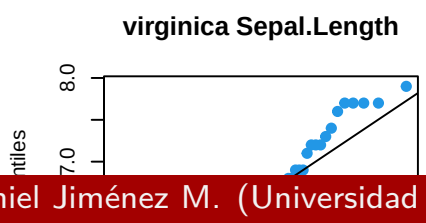


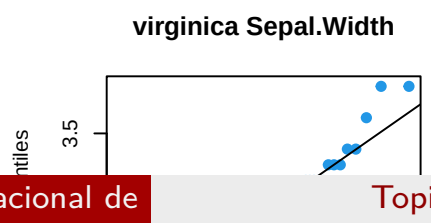


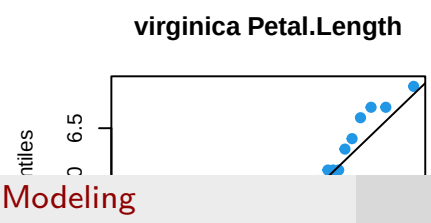


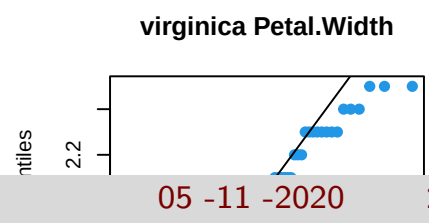












Latent Dirichlet Allocation (LDA)

Aplicando el LDA

```
modelo_lda <- lda(Species ~ Sepal.Width + Sepal.Length + Petal
                  Petal.Width, data = iris)
modelo_lda
```

```
## Call:
## lda(Species ~ Sepal.Width + Sepal.Length + Petal.Length + P
##     data = iris)
##
## Prior probabilities of groups:
##     setosa versicolor  virginica
##  0.3333333  0.3333333  0.3333333
##
## Group means:
##            Sepal.Width Sepal.Length Petal.Length Petal.Widt
## setosa           3.428        5.006        1.462       0.24
## versicolor       2.770        5.936        4.260       1.32
```

Latent Dirichlet Allocation (LDA)

¿De donde sale el segundo grupo?

Species	variable	p_value_Shapiro.test
setosa	Sepal.Length	0.45951
setosa	Sepal.Width	0.27153
setosa	Petal.Length	0.05481
setosa	Petal.Width	0.00000
versicolor	Sepal.Length	0.46474
versicolor	Sepal.Width	0.33800
versicolor	Petal.Length	0.15848
versicolor	Petal.Width	0.02728
virginica	Sepal.Length	0.25831
virginica	Sepal.Width	0.18090
virginica	Petal.Length	0.10978
virginica	Petal.Width	0.08695

Latent Dirichlet Allocation (LDA)

Notesé que la variable **petal.width** No se distribuye normal en la setosa ni en la versicolor.

En la prueba shapiro, todos los p-values mayores a 0.05 indican que hay presencia de normalidad. En el caso contraario donde p-value es menor a 1-α (0.05) indica que la distribución no es normal.

Latent Dirichlet Allocation (LDA)

Ahora un ejemplo con noticias[3]

```
data("AssociatedPress")
AssociatedPress
```

```
## <<DocumentTermMatrix (documents: 2246, terms: 10473)>>
## Non-/sparse entries: 302031/23220327
## Sparsity           : 99%
## Maximal term length: 18
## Weighting          : term frequency (tf)
```

[3] https://www.tidytextmining.com/topicmodeling.html

Latent Dirichlet Allocation (LDA)

```
ap_lda <- LDA(AssociatedPress, k = 4, control = list(seed = 12
ap_lda
```

```
## A LDA_VEM topic model with 4 topics.
```

Latent Dirichlet Allocation (LDA)

Ahora calculamos la probabilidad de pertenecer a un topic de cada palabra

```
ap_topics <- tidy(ap_lda, matrix = "beta")
ap_topics%>%
  filter(!term %in% stopwords(language = 'en'))%>%
  arrange(desc(beta))%>%
  head()
```

```
## # A tibble: 6 x 3
##   topic term          beta
##   <int> <chr>        <dbl>
## 1     1 percent    0.0183
## 2     3 million    0.0112
## 3     3 new        0.00890
## 4     1 year       0.00866
## 5     3 company    0.00796
## 6     2 president 0.00795
```

Latent Dirichlet Allocation (LDA)

Ahora Evaluemos los topics

Latent Dirichlet Allocation (LDA)

Ahora trabajemos sobre la mixtura de los documentos.

```
ap_documents <- tidy(ap_lda, matrix = "gamma")
ap_documents%>%
  arrange(desc(gamma))%>%
  head()
```

```
## # A tibble: 6 x 3
##   document topic gamma
##      <int> <int> <dbl>
## 1     1408     2 1.00
## 2      440     2 1.00
## 3     1894     2 0.999
## 4     2033     4 0.999
## 5     1911     1 0.999
## 6      576     2 0.999
```

Latent Dirichlet Allocation (LDA)

Lo anterior significa que gracias a la `Gamma` podemos decir que documentos tienen la inclusión de las palabras dentro de su tópico.

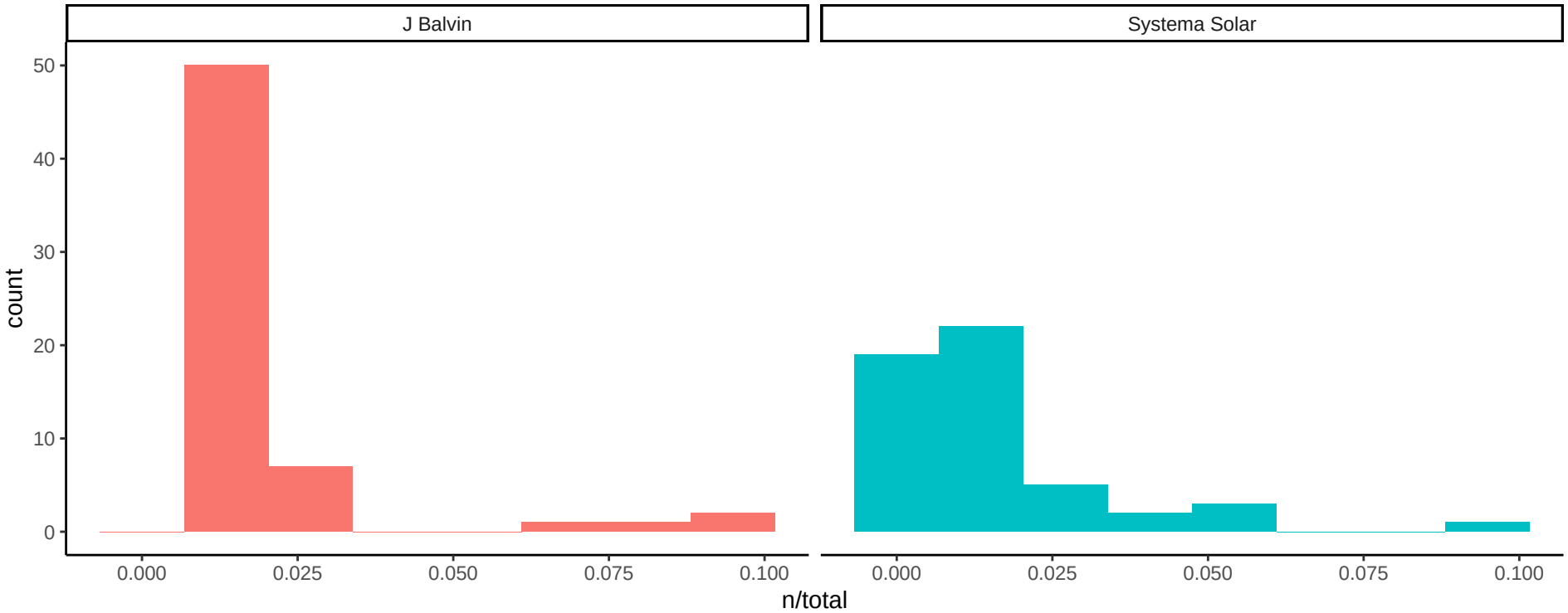
Latent Dirichlet Allocation (LDA)

Veamos esto con un ejemplo sencillo: Comparemos dos canciones:

- Systema Solar : El botón del pantalón
- J balvin : Rojo.

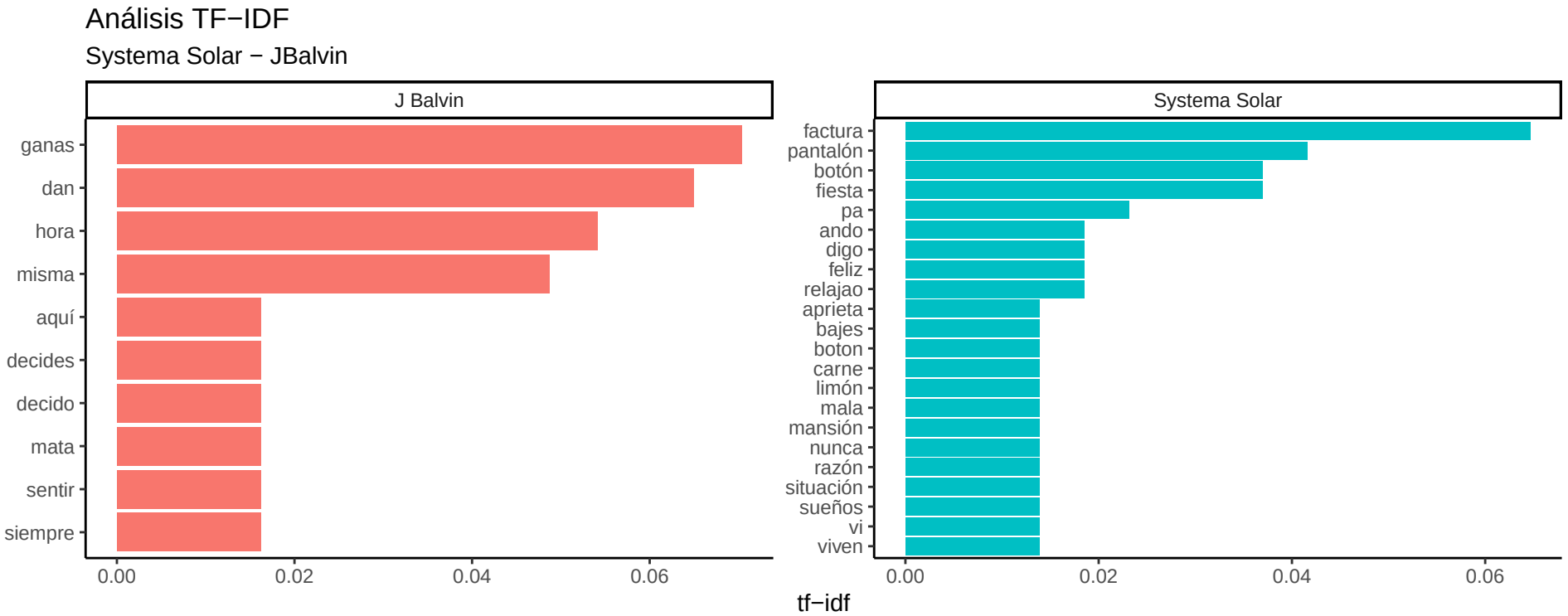
Latent Dirichlet Allocation (LDA)

¿Quién usa más palabras en las canciones?



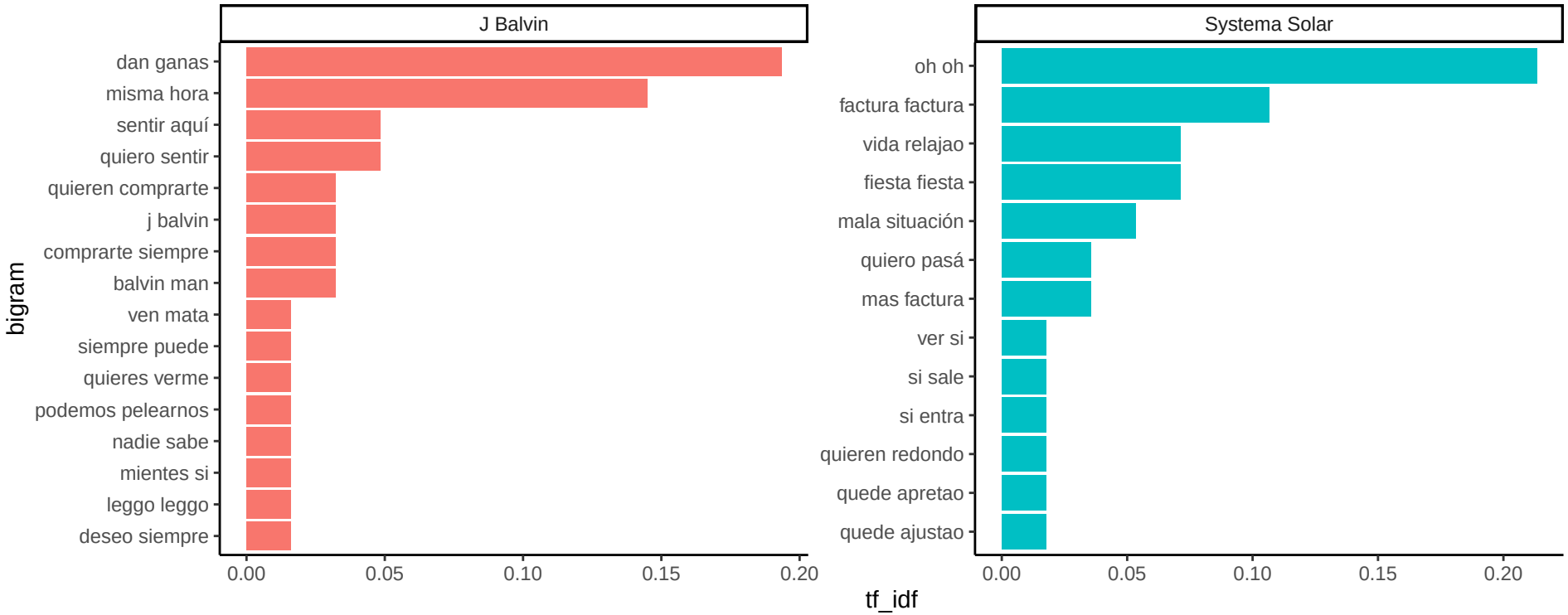
Latent Dirichlet Allocation (LDA)

Palabras que generan los contextos de las canciones



Latent Dirichlet Allocation (LDA)

Ahora veamos esto desde un bigram



Latent Dirichlet Allocation (LDA)

Ahora omita que sabe quien canta que y hagamos un corpus con las dos canciones e identifiquemos los grupos

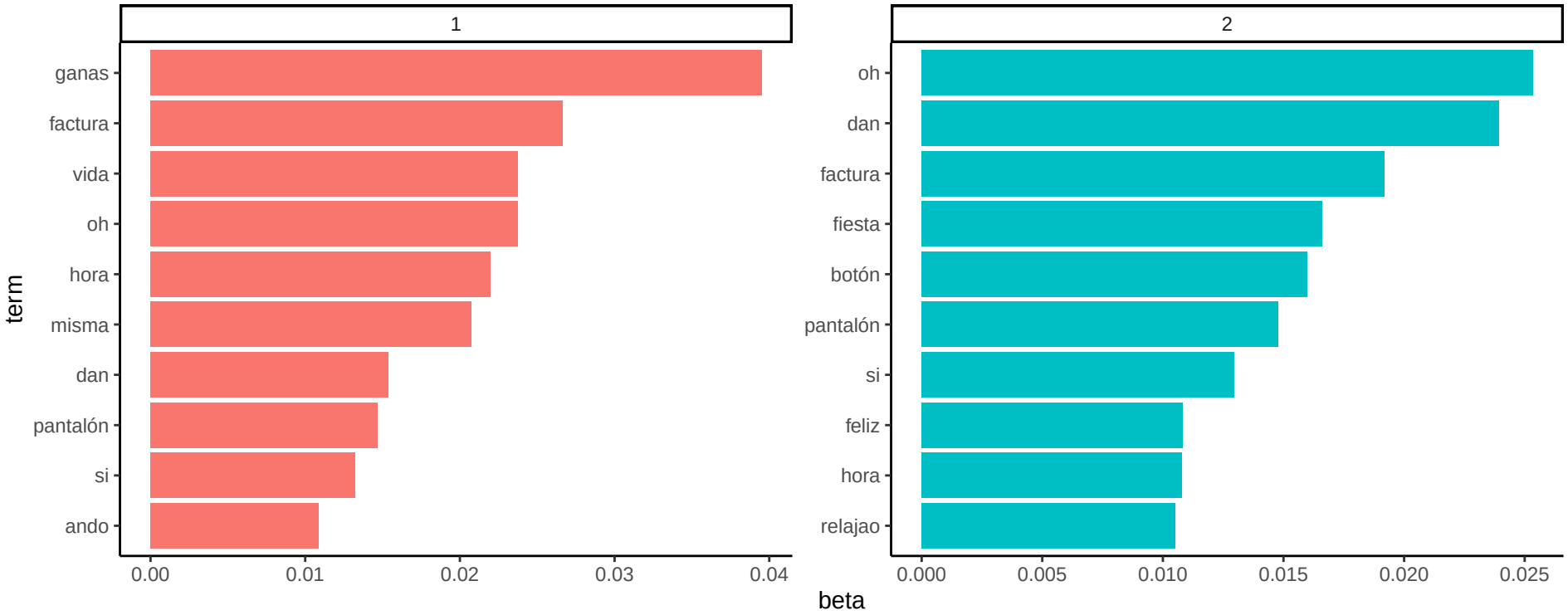
Latent Dirichlet Allocation (LDA)

Ahora usemos una gamma para saber a que topic corresponde cada documento

```
canciones_documents <- tidy(canciones_lda, matrix = "gamma")
canciones_documents%>%
  group_by(topic)%>%
  top_n(1)
```

```
## # A tibble: 2 x 3
## # Groups:   topic [2]
##   document       topic gamma
##   <chr>          <int> <dbl>
## 1 J Balvin           1 0.502
## 2 Systema Solar      2 0.502
```

Latent Dirichlet Allocation (LDA)



NER

Named Entity Recognition : Una entidad o palabra que hace referencia a un nombre propio.

NER

```
canciones_lda<-LDA(canciones_dtm,k = 2,method = 'Gibbs',contro
```

Notese que ahora :

- Hacemos 500 iteracciones ;
- Thin=1 devielve el resuldato de cada paso
- Gibbs = Es un método de muestreo basado en Monte carlo!

NER

```
tidy(canciones_lda,"gamma")%>%
  spread(topic,gamma)
```

```
## # A tibble: 2 x 3
##   document         `1`   `2`
##   <chr>          <dbl> <dbl>
## 1 J Balvin       0.322 0.678
## 2 Systema Solar 0.712 0.288
```

NER

El contexto de una entidad está representado por una ventana de palabras.

- Estas palabras por lo general se encuentran en mayúsculas.
- O tienen caracteres especiales
- Para encontrar las entidades usamos expresiones regulares.

NER

```
patron<-"[A-Z][a-z]+"
boton_pull<-boton%>%
  pull(Letra)%>%
  paste(collapse = " ")

boton_pull
```

```
## [1] "Fiesta, fiesta hay en mi Ando por la vida relajao y fe
```

NER

```
m <- gregexpr(patron, boton_pull)
v <- unlist(regmatches(boton_pull, m))
v
```

```
##  [1] "Fiesta"   "Ando"     "Te"       "Fiesta"   "Ando"
##  [7] "Ando"     "Fiesta"   "Ando"     "Est"      "Pillando"
## [13] "Para"     "Que"      "Que"      "Unas"     "Otras"
## [19] "Otros"    "Huy"      "Que"      "Mira"     "Oh"
## [25] "Oh"       "Pa"       "Si"       "No"       "De"
## [31] "Si"       "No"       "De"       "Factura"  "Factura"
## [37] "Si"       "No"       "Yo"       "Ni"       "Si"
## [43] "Yo"       "Ni"       "Si"       "No"       "Yo"
## [49] "Con"
```

NER

Ahora veamos donde se usan las entidades

```
text<-gsub(v, "", boton_pull)%>%
  tbl_df()%>%
  unnest_tokens("sentences",token = 'sentences',value)
text
```

```
## # A tibble: 3 x 1
##   sentences
##   <chr>
## 1 , fiesta hay en mi ando por la vida relajao y feliz te di
## 2 que viaje es esta vaina mira con ese botón oh oh oh oh oh
## 3 si te aprieta el boton por la mala situación no te bajes
```

NED

Se generan las reglas y se extraen las entidades.

```
## # A tibble: 49 x 2
##    Letra                                  Entity
##    <chr>                                  <chr>
##  1 Fiesta, fiesta hay en mi               Fiesta
##  2 Ando por la vida relajao y feliz       Ando
##  3 Te digo                                Te
##  4 Fiesta, fiesta hay en mi               Fiesta
##  5 Ando por la vida relajao y feliz te digo Ando
##  6 Fiesta, fiesta hay en mi               Fiesta
##  7 Ando por la vida relajao y feliz te digo Ando
##  8 Fiesta, fiesta hay en mi               Fiesta
##  9 Ando por la vida relajao y feliz te digo Ando
## 10 Estábamos hablando con los vales       Est
## # ... with 39 more rows
```

NER

Ahora generamos una clasificación

```
## # A tibble: 6 x 4
##   document    `1`    `2`    `3`
##   <chr>     <dbl>  <dbl>  <dbl>
## 1 Ando      0.946  0.0270 0.0270
## 2 Con       0.0526 0.895  0.0526
## 3 De        0.0909 0.0909 0.818
## 4 Est       0.125  0.75   0.125
## 5 Factura   0.176  0.0588 0.765
## 6 Fiesta    0.0435 0.0435 0.913
```

Esto nos indica que el topic 1 es de acciones personales, el segundo es proposición y el último es de acciones.

NER

Ahora se crea un modelo que intente generar los topics

```
r <- sample.int(n=nrow(corpus2), size=20, replace=FALSE) # Un
train_dtm <- corpus2[-r, ] %>% unnest_tokens(input=doc, output
  count(Entity, word) %>%
  cast_dtm(document=Entity, term=word, value=n)
```

NER

Se genera el modelo

```
train_mod <- LDA(x=train_dtm, k=3, method="Gibbs",
                 control=list(alpha=1, seed=10001,
                              iter=1000, thin=1))
```

NER

Después con una parte del documento que no se ha visto se genera el output del modelo y se valida.

```
##                  1           2           3
## Unas      0.2500000 0.25000000 0.50000000
## Pillando  0.2500000 0.25000000 0.50000000
## Oh        0.1666667 0.66666667 0.16666667
## No        0.1666667 0.16666667 0.66666667
## Ni        0.1666667 0.16666667 0.66666667
## Mira      0.2500000 0.50000000 0.25000000
## Fiesta    0.1428571 0.14285714 0.71428571
## Factura   0.8461538 0.07692308 0.07692308
## Est       0.1250000 0.75000000 0.12500000
## Con       0.1428571 0.57142857 0.28571429
```

Determinar el # Optimo de Topics

Hay dos formas de hacerlo:

- Ajustar un modelo e inspeccionar las palabras que se le asignaron a los topics, y decidir si tienen sentido -> Lo más probable para la JEP
- Medidas cuantitativas de ajuste:
  - Probabilidad logarítmica
  - Perplejidad

Determinar el # Optimo de Topics

Para el último caso es recomendable usar una probabilidad logarítmica

```
perplexity(object=train_mod, newdata=test_dtm)
```

```
## [1] 7.580073
```

Determinar el # Optimo de Topics

Para hallar k usamos la siguiente iteracción, y seleccionamos donde se produce un punto de quiebre en la gráfica

```
model_log<-numeric(20)
model_perp<-numeric(20)
for (i in 2:20){
  modelo<-LDA(train_dtm,k=i,method = 'Gibbs',
              control = list(alpha=0.5,iter=100,seed=1234,thi
  model_log[i]=logLik(modelo)
  model_perp[i]<-perplexity(modelo,test_dtm)
}
```

Determinar el # Optimo de Topics

```
k<-1:20
plot(x=k, y=model_perp, xlab="number of clusters, k",
     ylab="perplexity score", type="o",xlim = c(1,20))
axis(side = 1, at=1:20)
```