L13. Textmining (2)

Sim, Min Kyu, Ph.D., mksim@seoultech.ac.kr

SEOULTECH 서울과학기술대학교 데이터사이언스학과

I. Recap

1. Load

```
options(tibble.width = Inf) # show full columns
library(tidyverse)
library(tidytext)
amz <- read_csv("data/Datafiniti_Amazon_Consumer_Reviews_of_Amazon_Products.csv")
amz <- amz %>%
  select(id, name, asins, brand, primaryCategories, manufacturer,
         reviews.doRecommend, reviews.numHelpful, reviews.rating,
         reviews.text, reviews.title, reviews.username)
amz[1:2,1:9]
```

```
## # A tibble: 2 x 9
##   id                     name
##   <chr>                  <chr>
## 1 AVqVGZNvQMlgsOJE6eUY "Amazon Kindle E-Reader 6\" Wifi (8th Generation, 2016)"
## 2 AVqVGZNvQMlgsOJE6eUY "Amazon Kindle E-Reader 6\" Wifi (8th Generation, 2016)"
##   asins      brand  primaryCategories manufacturer reviews.doRecommend
##   <chr>      <chr>  <chr>             <chr>        <lgl>
## 1 B00ZV9PXP2 Amazon Electronics       Amazon       FALSE
## 2 B00ZV9PXP2 Amazon Electronics       Amazon       TRUE
##   reviews.numHelpful reviews.rating
##                <dbl>          <dbl>
```

```
amz[1:2,10:12]
```

```
## # A tibble: 2 x 3
##   reviews.text
##   <chr>
## 1 I thought it would be as big as small paper but turn out to be just like my p~
## 2 This kindle is light and easy to use especially at the beach!!!
##   reviews.title                                 reviews.username
##   <chr>                                         <chr>
## 1 Too small                                     llyyue
## 2 Great light reader. Easy to use at the beach Charmi
```

2. Convert to *tidy text* by `tidytext::unnest_tokens()`

```
library(tidytext)
tidy_amz <- amz %>%
  unnest_tokens(word, reviews.text)
tidy_amz[1,]
```

```
## # A tibble: 1 x 12
##   id                    name
##   <chr>                 <chr>
## 1 AVqVGZNvQMlgsOJE6eUY "Amazon Kindle E-Reader 6\" Wifi (8th Generation, 2016)"
##   asins      brand  primaryCategories manufacturer reviews.doRecommend
##   <chr>      <chr>  <chr>             <chr>        <lgl>
## 1 B00ZV9PXP2 Amazon Electronics       Amazon       FALSE
##   reviews.numHelpful reviews.rating reviews.title reviews.username word
##                <dbl>          <dbl> <chr>         <chr>            <chr>
## 1                  0              3 Too small     llyyue           i
```

3. Cleaning (stop words, numbers, stemming)

```
# For cleaning 1
word_with_num <- tidy_amz %>%
  select(word) %>%
  filter(str_detect(word, "^[0-9]")) %>% unique()
# For cleaning 2
set.seed (111)
library (SnowballC)
# Execute cleanings
tidy_amz <- tidy_amz %>%
  anti_join(stop_words) %>%     # cleaning 1: stop words
  anti_join(word_with_num) %>%  # cleaning 2: numbers
  mutate(root = wordStem(word)) # cleaning 3: stemming
```

```r
options(tibble.width = Inf) # show full columns
tidy_amz[1:2,]
```

```
## # A tibble: 2 x 13
##   id                     name
##   <chr>                  <chr>
## 1 AVqVGZNvQMlgsOJE6eUY "Amazon Kindle E-Reader 6\" Wifi (8th Generation, 2016)"
## 2 AVqVGZNvQMlgsOJE6eUY "Amazon Kindle E-Reader 6\" Wifi (8th Generation, 2016)"
##   asins      brand  primaryCategories manufacturer reviews.doRecommend
##   <chr>      <chr>  <chr>             <chr>        <lgl>
## 1 B00ZV9PXP2 Amazon Electronics       Amazon       FALSE
## 2 B00ZV9PXP2 Amazon Electronics       Amazon       FALSE
##   reviews.numHelpful reviews.rating reviews.title reviews.username word  root
##                <dbl>          <dbl> <chr>         <chr>            <chr> <chr>
## 1                  0              3 Too small     llyyue           paper paper
## 2                  0              3 Too small     llyyue           palm  palm
```

Some pre-processing: create numeric product id (`prod_id`)

```
prod_list <- tidy_amz %>%
  count(name) %>%
  mutate(prod_id = row_number()) %>%
  select(name, prod_id)
dim(prod_list)
```

```
## [1] 23  2
```

```
prod_list %>% head(2)
```

```
## # A tibble: 2 x 2
##   name
##   <chr>
## 1 All-New Fire HD 8 Tablet, 8 HD Display, Wi-Fi, 16 GB - Includes Special Offer~
## 2 All-New Fire HD 8 Tablet, 8 HD Display, Wi-Fi, 32 GB - Includes Special Offer~
##   prod_id
##     <int>
## 1       1
## 2       2
```

```
tidy_amz <- left_join(tidy_amz, prod_list)
```

II. tf

```
head(tidy_amz)
```

```
## # A tibble: 6 x 14
##   id                     name
##   <chr>                  <chr>
## 1 AVqVGZNvQMlgsOJE6eUY "Amazon Kindle E-Reader 6\" Wifi (8th Generation, 2016)"
## 2 AVqVGZNvQMlgsOJE6eUY "Amazon Kindle E-Reader 6\" Wifi (8th Generation, 2016)"
## 3 AVqVGZNvQMlgsOJE6eUY "Amazon Kindle E-Reader 6\" Wifi (8th Generation, 2016)"
## 4 AVqVGZNvQMlgsOJE6eUY "Amazon Kindle E-Reader 6\" Wifi (8th Generation, 2016)"
## 5 AVqVGZNvQMlgsOJE6eUY "Amazon Kindle E-Reader 6\" Wifi (8th Generation, 2016)"
## 6 AVqVGZNvQMlgsOJE6eUY "Amazon Kindle E-Reader 6\" Wifi (8th Generation, 2016)"
##   asins      brand  primaryCategories manufacturer reviews.doRecommend
##   <chr>      <chr>  <chr>             <chr>        <lgl>
## 1 B00ZV9PXP2 Amazon Electronics       Amazon       FALSE
## 2 B00ZV9PXP2 Amazon Electronics       Amazon       FALSE
## 3 B00ZV9PXP2 Amazon Electronics       Amazon       FALSE
## 4 B00ZV9PXP2 Amazon Electronics       Amazon       FALSE
## 5 B00ZV9PXP2 Amazon Electronics       Amazon       FALSE
## 6 B00ZV9PXP2 Amazon Electronics       Amazon       FALSE
##   reviews.numHelpful reviews.rating reviews.title reviews.username word
##                <dbl>          <dbl> <chr>         <chr>            <chr>
## 1                  0              3 Too small     llyyue           paper
## 2                  0              3 Too small     llyyue           palm
## 3                  0              3 Too small     llyyue           read
```

*Word count and term frequency (**tf**)*

- Assumption: Document는 Collection of words이다. (words의 순서와 배치는 무시한다.)
- Strategy: 문서에서 각 단어의 출현빈도를 분석한다.
- Term Frequency (단어빈도)는 특정 단어가 문서에서 얼마나 자주 등장하는지를 나타내는 값이다.

```
# count num of each root for each product
words_each_prod <- tidy_amz %>%
  group_by(prod_id) %>%
  count(root, sort=T) %>%
  ungroup()
# count total num of roots for each product
total_words_each_prod <- words_each_prod %>%
  group_by(prod_id) %>%
  summarise(total_words=sum(n))
# join so that `total_words` is included
words_each_prod <- words_each_prod %>%
  left_join(total_words_each_prod)
```

```
head(words_each_prod)
```

```
## # A tibble: 6 x 4
##   prod_id root       n total_words
##     <int> <chr>  <int>       <int>
## 1       3 tablet   428        7718
## 2      10 echo     400       10131
## 3      10 love     320       10131
## 4      17 tablet   283        5370
## 5      17 love     276        5370
## 6      10 alexa    253       10131
```

```
tf_tbl <- words_each_prod %>% group_by(prod_id) %>%
  mutate(rank=row_number(), tf=n/total_words) %>% ungroup()
tf_tbl %>% head(3)
```

```
## # A tibble: 3 x 6
##   prod_id root       n total_words  rank     tf
##     <int> <chr>  <int>       <int> <int>  <dbl>
## 1       3 tablet   428        7718     1 0.0555
## 2      10 echo     400       10131     1 0.0395
## 3      10 love     320       10131     2 0.0316
```

- prod_id가 3인 제품에 대한 리뷰들에는 7718개의 단어가 있는데,
- 이중에서 tablet이라는 어근을 가진 단어가 가장 많이 등장했다. (총 428회 등장)
- tablet의 등장 빈도는 5.5%이다.

Zipf's law

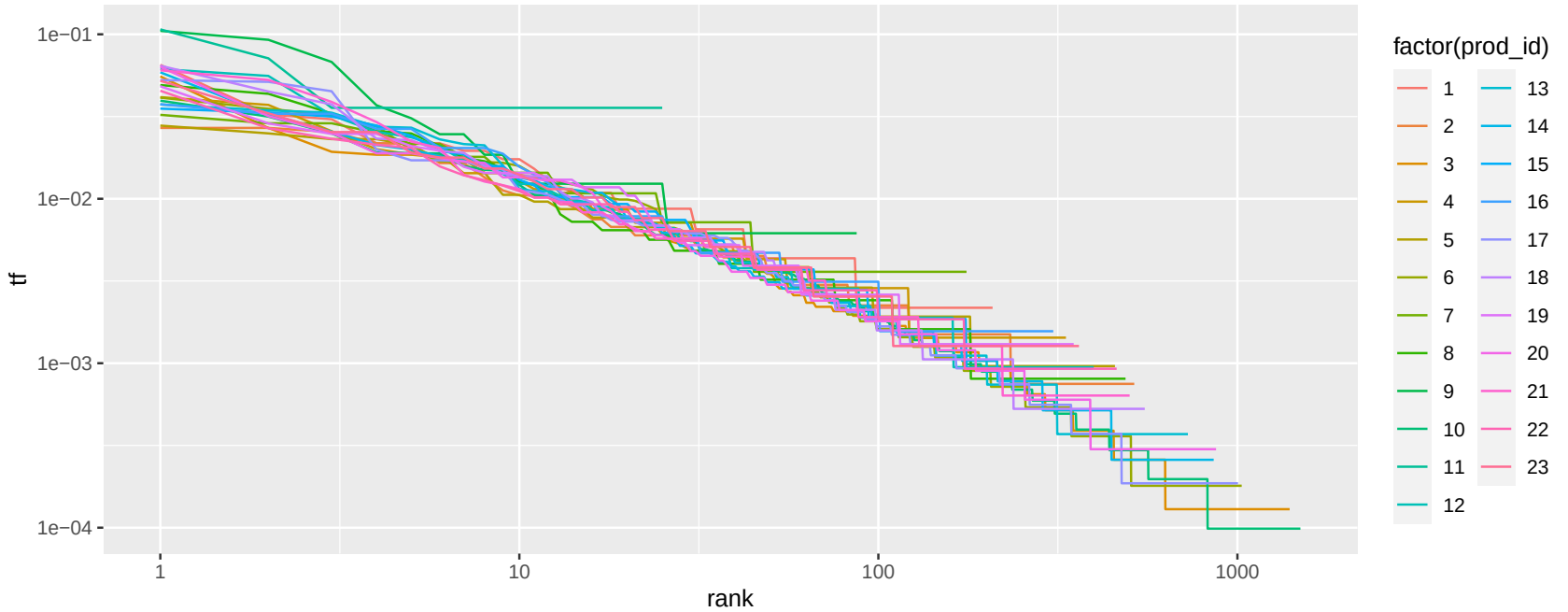
- 문서내의 출편 빈도와 출현 빈도 순위는 아래와 같은 지수관계가 있다.

$$\text{tf} \approx \alpha \cdot (\text{rank})^{\beta}$$

- Or, equivalently,

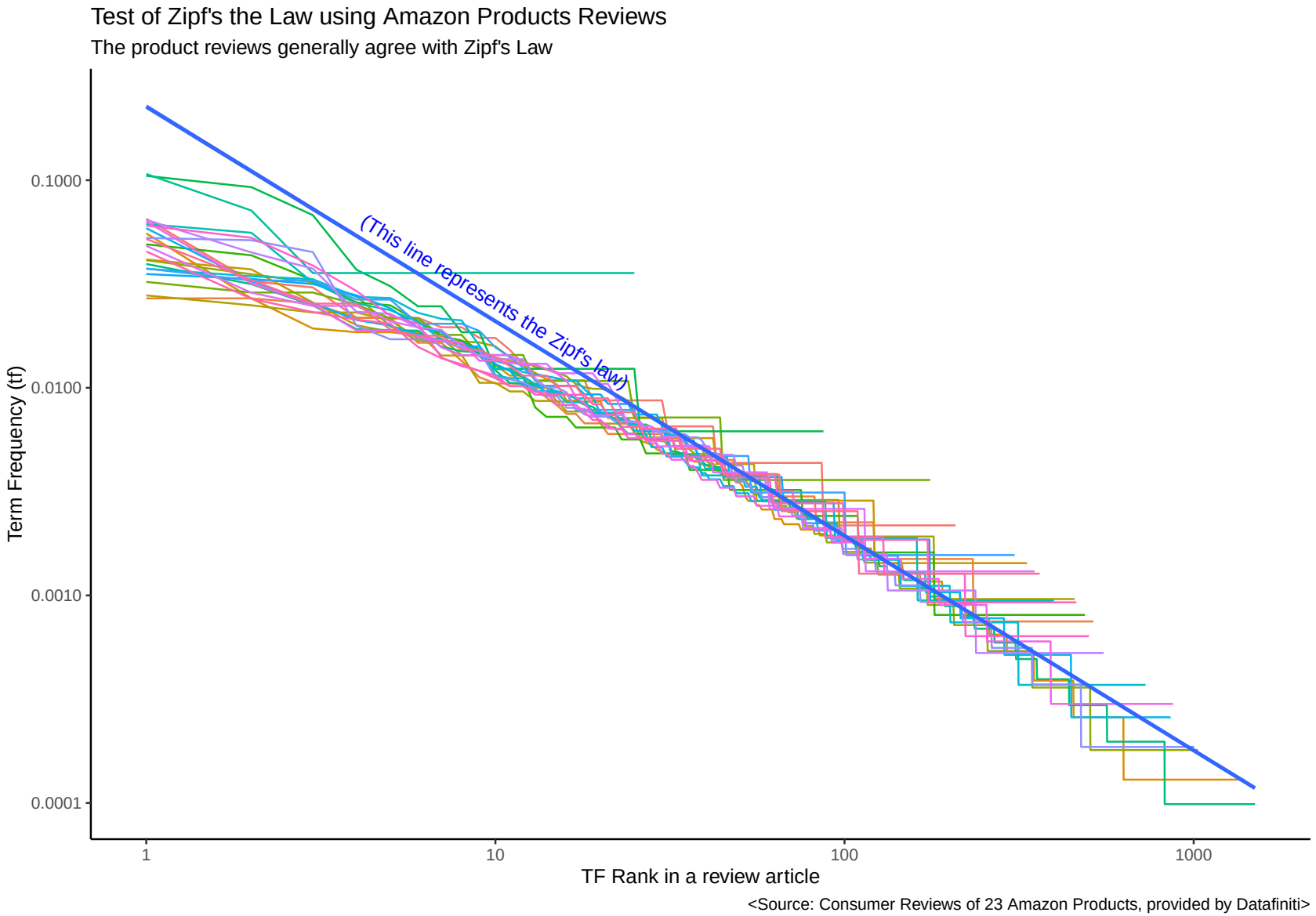
$$log(\text{tf}) \approx log(\alpha) + \beta \cdot log(\text{rank})$$

```
ggplot(tf_tbl, aes(x=rank, y=tf)) +
  geom_line(aes(color=factor(prod_id))) +
  scale_x_log10() + scale_y_log10()
```



1. Title과 meta 정보
2. y축 label을 'term-frequency'로
3. y축 axis를 일반적인 소수점 형식으로
4. 범례 제거
5. `geom_smooth()` line 추가

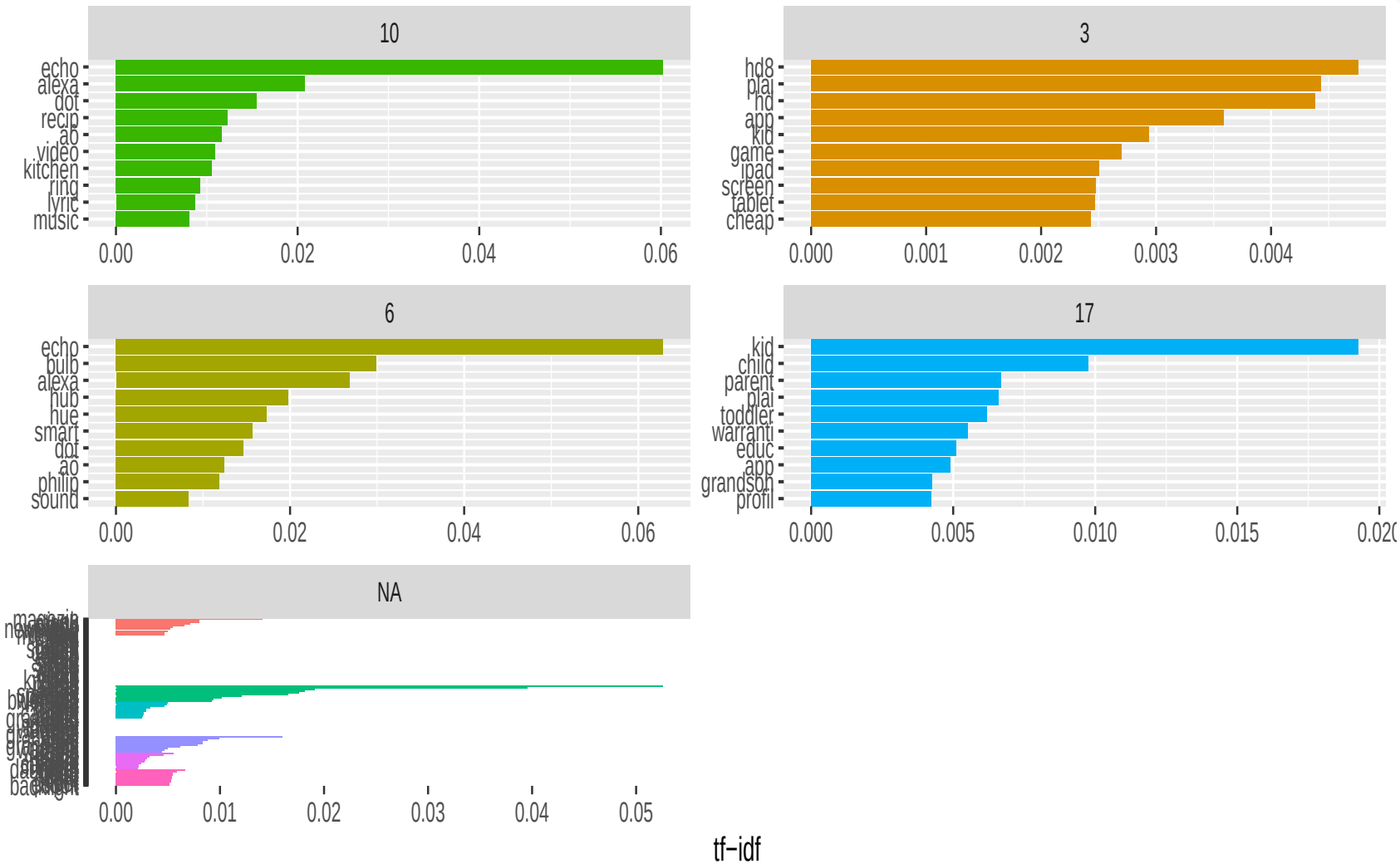
```
fig1
```


Test of Zipf's the Law using Amazon Products Reviews
The product reviews generally agree with Zipf's Law
(This line represents the Zipf's law)
Term Frequency (tf)
0.1000
0.0100
0.0010
0.0001
1
10
100
1000
TF Rank in a review article
<Source: Consumer Reviews of 23 Amazon Products, provided by Datafiniti>

III. tf-idf

```
tfidf_amz <- words_each_prod %>% bind_tf_idf(root, prod_id, n) %>% arrange(-tf_idf)
tfidf_vis_amz <- tfidf_amz %>% filter(total_words>1300) %>% group_by(prod_id) %>%
  top_n(10) %>% ungroup() %>%
  arrange(prod_id, -tf_idf) %>%
  mutate(order=row_number())
facet_order <- tidy_amz %>% count(name) %>% mutate(prod_id = row_number()) %>%
  arrange(-n) %>% head(4)
fig2 <- tfidf_vis_amz %>%
  ggplot(aes(order, tf_idf, fill=factor(prod_id))) +
  geom_col(show.legend=F) +
  scale_x_reverse(breaks=tfidf_vis_amz$order,
                  labels=tfidf_vis_amz$root,
                  expand=c(0,0)) +
  labs(x=NULL, y="tf-idf") +
  facet_wrap(
    ~factor(prod_id, levels=facet_order$prod_id), ncol=2, scales="free") +
  coord_flip()
```

fig2


10
echo
alexa
dot
recip
ao
video
kitchen
ring
lyric
music
0.00
0.02
0.04
0.06
3
hd8
plai
hd
app
kid
game
ipad
screen
tablet
cheap
0.000
0.001
0.002
0.003
0.004
6
echo
bulb
alexa
hub
hue
smart
dot
ao
philip
sound
0.00
0.02
0.04
0.06
17
kid
child
parent
plai
toddler
warranti
educ
app
grandson
profil
0.000
0.005
0.010
0.015
NA
0.00
0.01
0.02
0.03
0.04
0.05
tf−idf