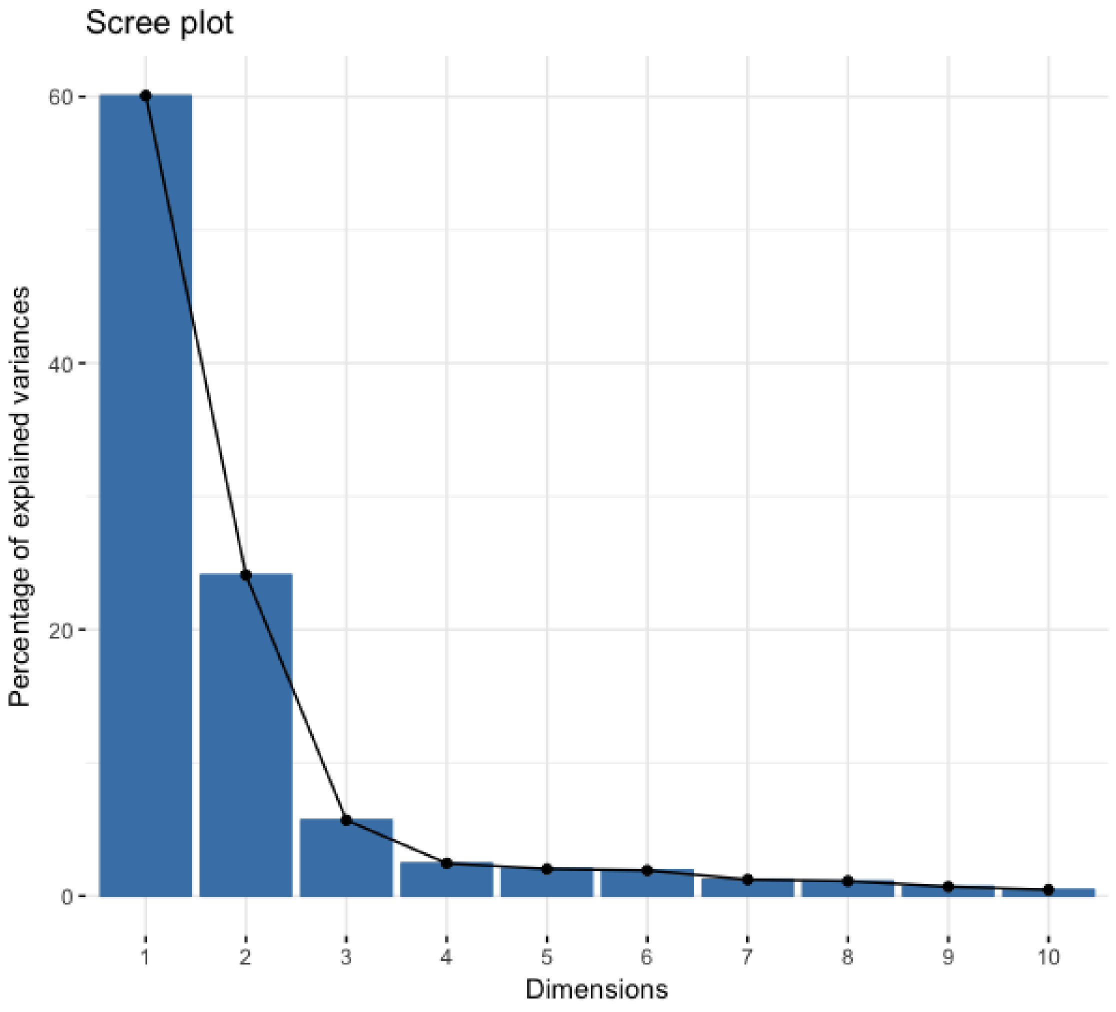
DIMENSIONALITY REDUCTION IN R

Advanced PCA: Choosing the right number of PCs

Alexandros Tantos
Assistant Professor
Aristotle University of Thessaloniki

How many PCs to keep?

Earlier: Maybe 2 or 3 ...



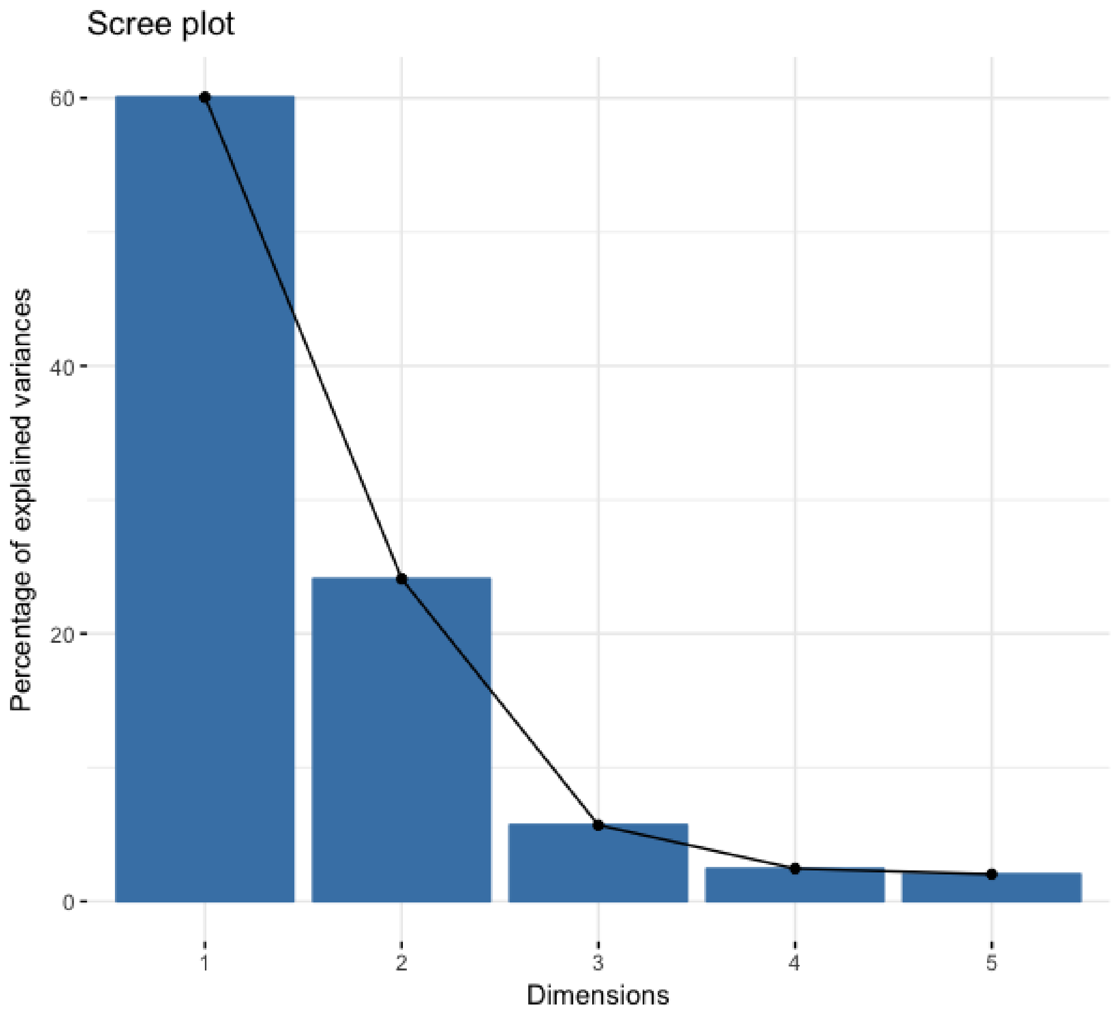
Stopping rules

1. The Scree test
2. The Kaiser-Guttman rule
3. Parallel analysis

The Scree test

```
mtcars_pca <- PCA(mtcars)

fviz_screeplot(mtcars_pca, ncp=5)
```



The Kaiser-Guttman rule

Keep the PCs with eigenvalue > 1

```
summary(mtcars_pca)

mtcars_pca$eig

get_eigenvalue(mtcars_pca)
```

	eigenvalue
Dim.1	6.60840025
Dim.2	2.65046789
Dim.3	0.62719727
Dim.4	0.26959744
Dim.5	0.22345110
Dim.6	0.21159612
Dim.7	0.13526199
Dim.8	0.12290143
Dim.9	0.07704665
Dim.10	0.05203544
Dim.11	0.02204441

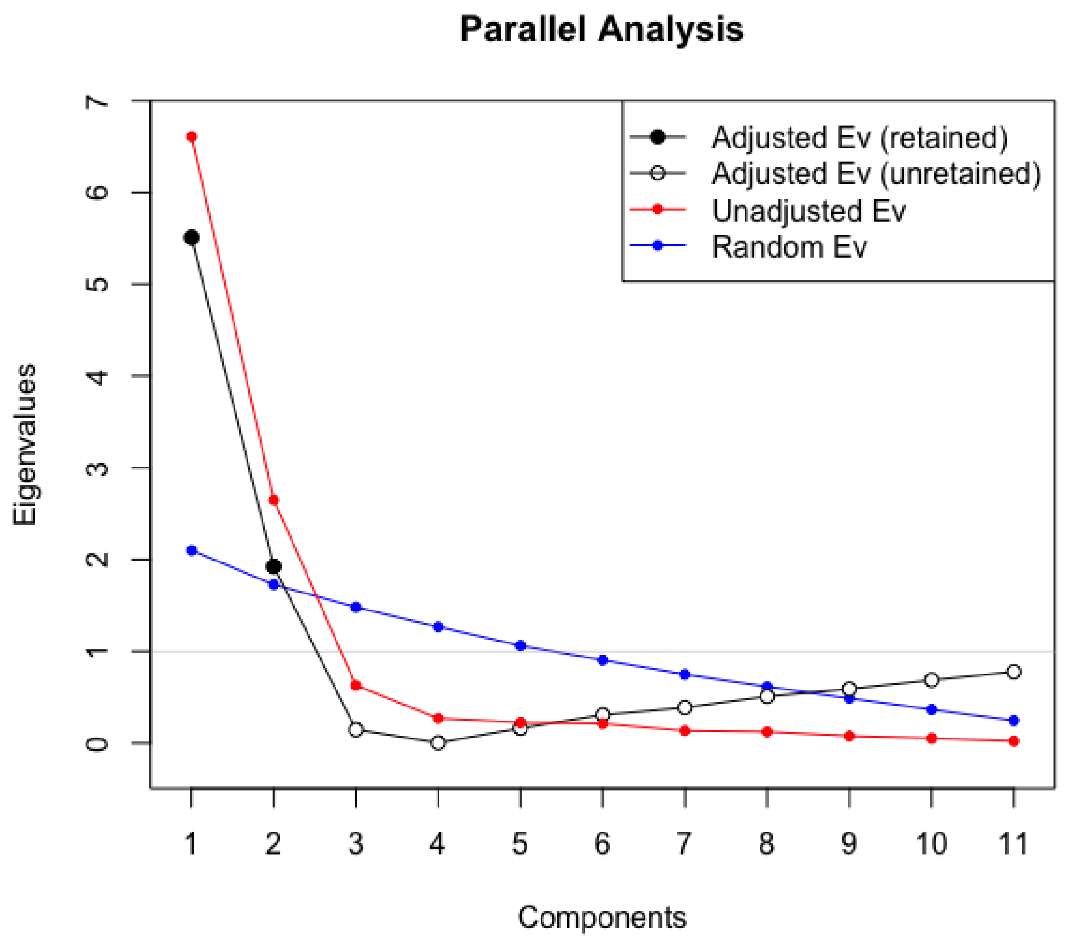
Parallel Analysis

```r
library(paran)

mtcars_pca_ret <- paran(mtcars_pca,
                        graph = TRUE)

mtcars_pca_retained$Retained
```

[1] 2



DIMENSIONALITY REDUCTION IN R

Let's practice!

DIMENSIONALITY REDUCTION IN R

Advanced PCA: Performing PCA on datasets with missing values

Alexandros Tantos
Assistant Professor
Aristotle University of Thessaloniki

Exploring datasets with missing values

```r
library(VIM)

sleep[!complete.cases(VIM::sleep),]
```

```r
sum(is.na(VIM::sleep))
```

38

- Skipping rows with missing values: Risky option that leads to unreliable `PCA` models.
- Often costly to ignore collected data.

	BodyWgt	BrainWgt	NonD	Dream	Sleep	Span	Gest	Pred	Exp	Danger
1	6654.000	5712.0	NA	NA	3.3	38.6	645	3	5	3
3	3.385	44.5	NA	NA	12.5	14.0	60	1	1	1
4	0.920	5.7	NA	NA	16.5	NA	25	5	2	3
13	0.550	2.4	7.6	2.7	10.3	NA	NA	2	1	2
14	187.100	419.0	NA	NA	3.1	40.0	365	5	5	5
19	1.410	17.5	4.8	1.3	6.1	34.0	NA	1	2	1
20	60.000	81.0	12.0	6.1	18.1	7.0	NA	1	1	1
21	529.000	680.0	NA	0.3	NA	28.0	400	5	5	5
24	207.000	406.0	NA	NA	12.0	39.3	252	1	4	1
26	36.330	119.5	NA	NA	13.0	16.2	63	1	1	1
30	100.000	157.0	NA	NA	10.8	22.4	100	1	1	1
31	35.000	56.0	NA	NA	NA	16.3	33	3	5	4
35	0.122	3.0	8.2	2.4	10.6	NA	30	2	1	1
36	1.350	8.1	8.4	2.8	11.2	NA	45	3	1	3
41	250.000	490.0	NA	1.0	NA	23.6	440	5	5	5
47	4.288	39.2	NA	NA	12.5	13.7	63	2	2	2
53	14.830	98.2	NA	NA	2.6	17.0	150	5	5	5
55	1.400	12.5	NA	NA	11.0	12.7	90	2	2	2
56	0.060	1.0	8.1	2.2	10.3	3.5	NA	3	1	2
62	4.050	17.0	NA	NA	NA	13.0	38	3	1	1

Estimation methods for PCA on datasets with missing values

From simplistic to sophisticated methods:

- Using the mean of the variable that includes `NA` values.
- Impute the missing values based on a linear regression regression model.
- **Estimating missing values with `PCA`**
  - Use `missMDA` and then `FactoMineR`
  - Use `pcaMethods`

Estimating missing values with missMDA

Iterative `PCA` algorithm

Initial step: use the mean for imputing the missing values

- Conduct `PCA` on the resulting complete dataset
- Use the coordinates of the newly-extracted `PC`s (initially taking the mean) for updating them.
- Repeat the previous two steps until convergence is achieved.

Conduct `PCA` on the completed dataset with `PCA()`.

Estimating missing values with missMDA

```
library(missMDA)

nPCs <- estim_ncpPCA(VIM::sleep)

nPCS$ncp
3

completed_sleep <- imputePCA(VIM::sleep, ncp = nPCs$ncp,
                scale = TRUE)
```

```
PCA(completed_sleep$completeObs)
```

Imputing missing values with pcaMethods

The internals of `pca()`:

- Uses regression methods for approximation of the correlation matrix.
- Compiles `PCA` models
- Finally, it projects the new points back into the original space.

```
library(pcaMethods)

sleep_pca_methods <- pca(sleep, nPcs=2, method="ppca", center = TRUE)

imp_air_pcamethods <- completeObs(sleep_pca_methods)
```

DIMENSIONALITY REDUCTION IN R

Let's practice!

DIMENSIONALITY REDUCTION IN R

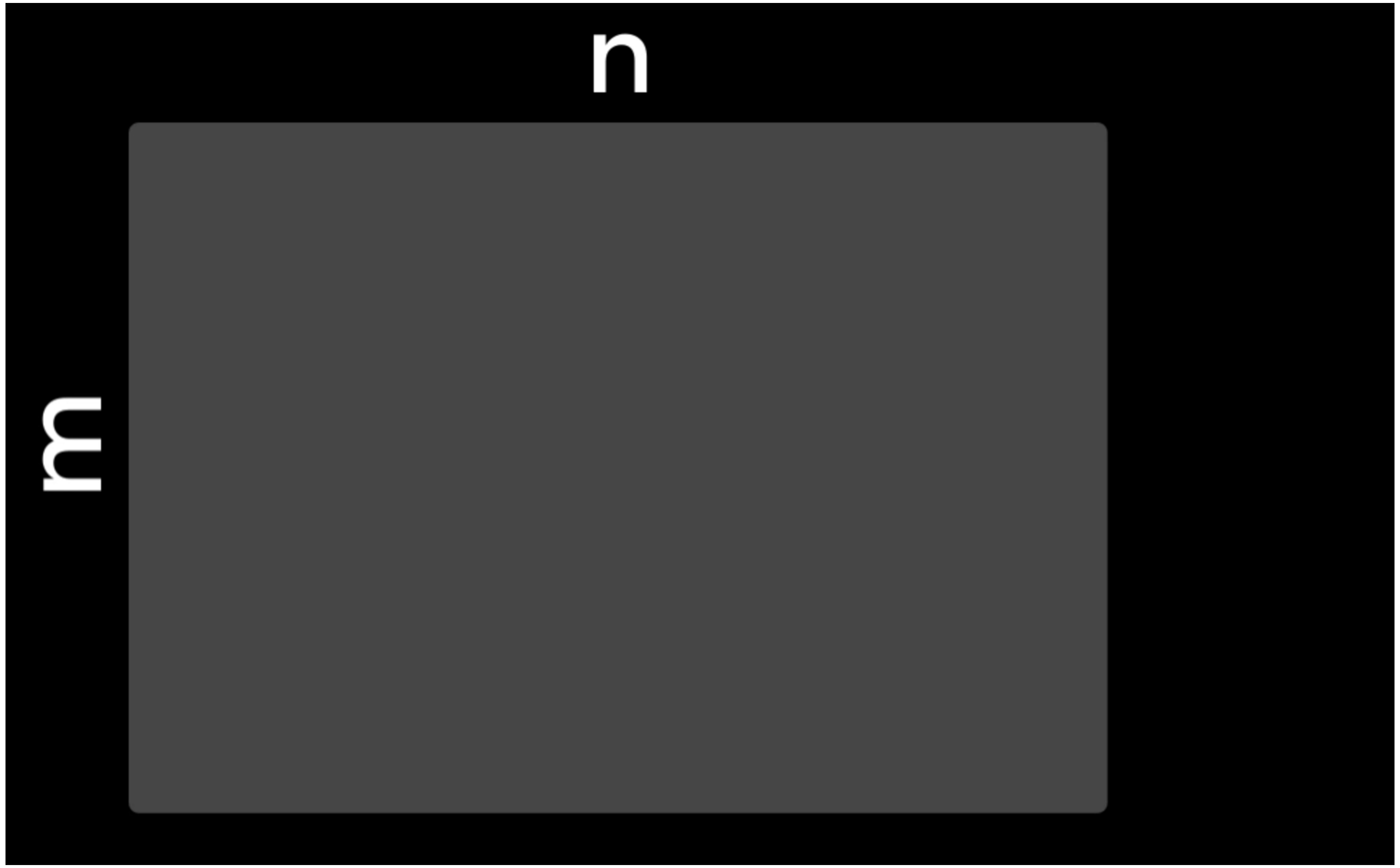
N-NMF and topic detection with nmf()

Alexandros Tantos
Assistant Professor
Aristotle University of Thessaloniki

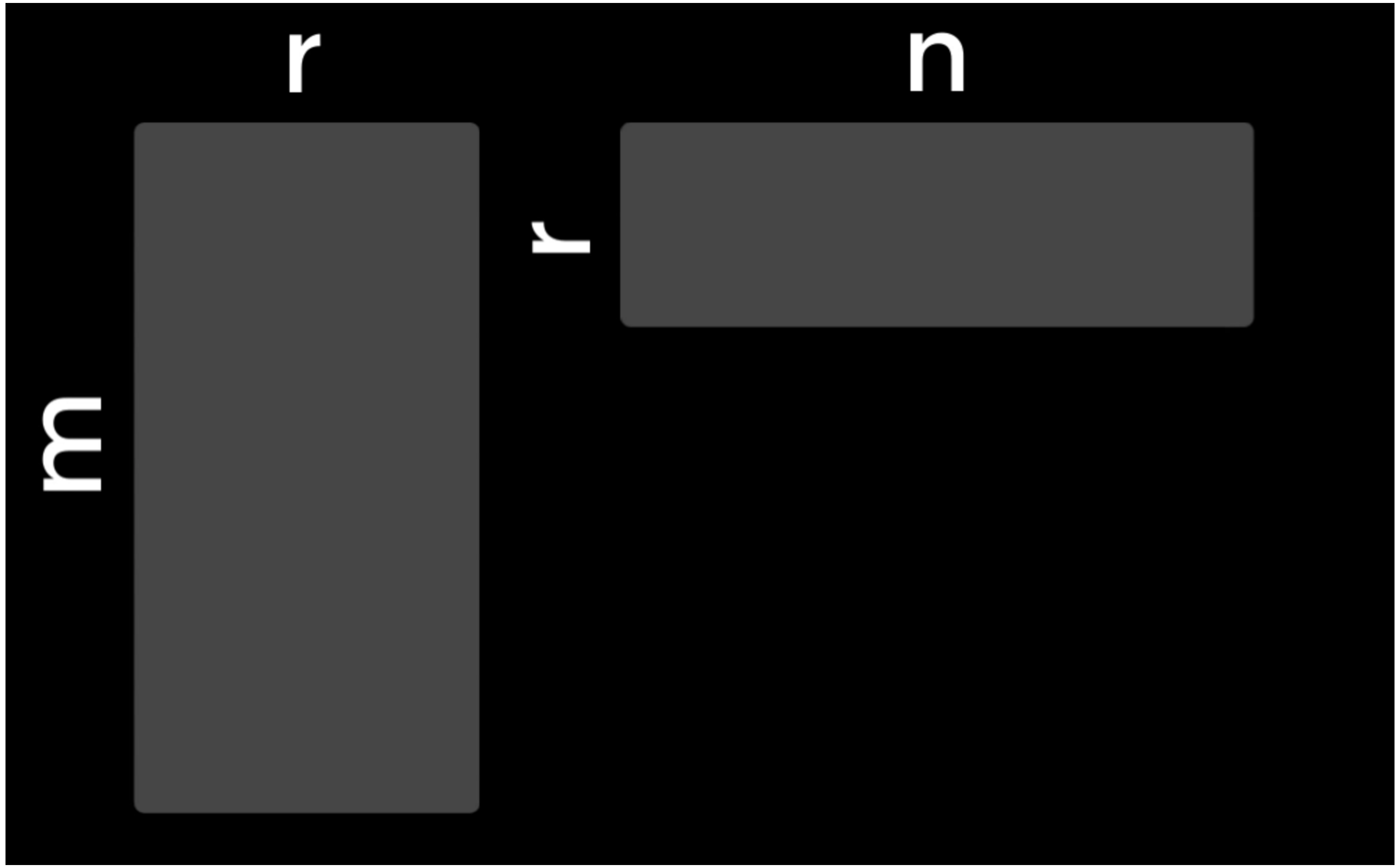
N-NMF and PCA

- Difficult to interpret `PCA` models with count/frequency data.
  - Normality assumption.
  - `PC`s include negative values.
- `N-NMF` algorithms are able to extract clear and distinct insights from the data.

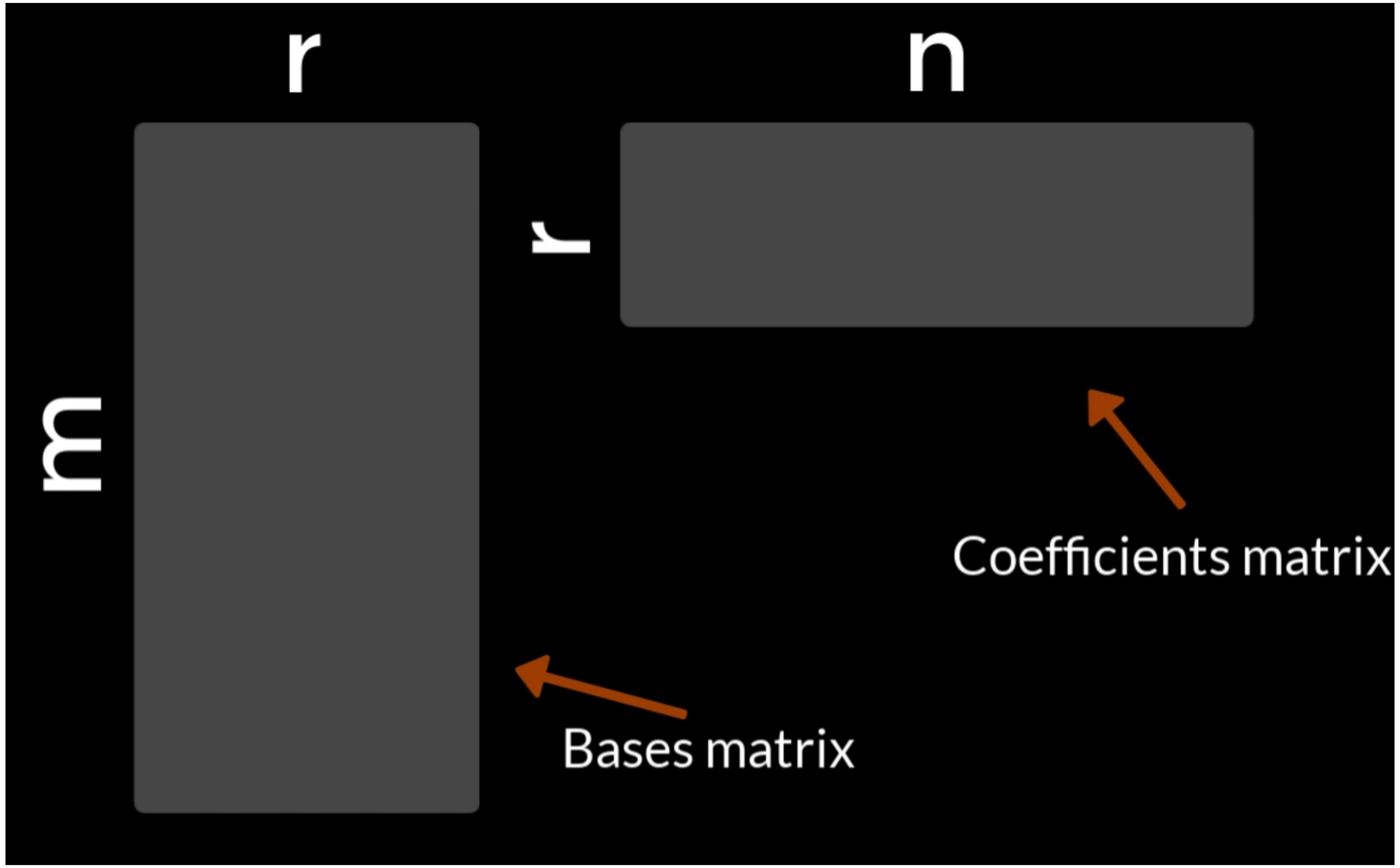
N-NMF: Tearing the data apart



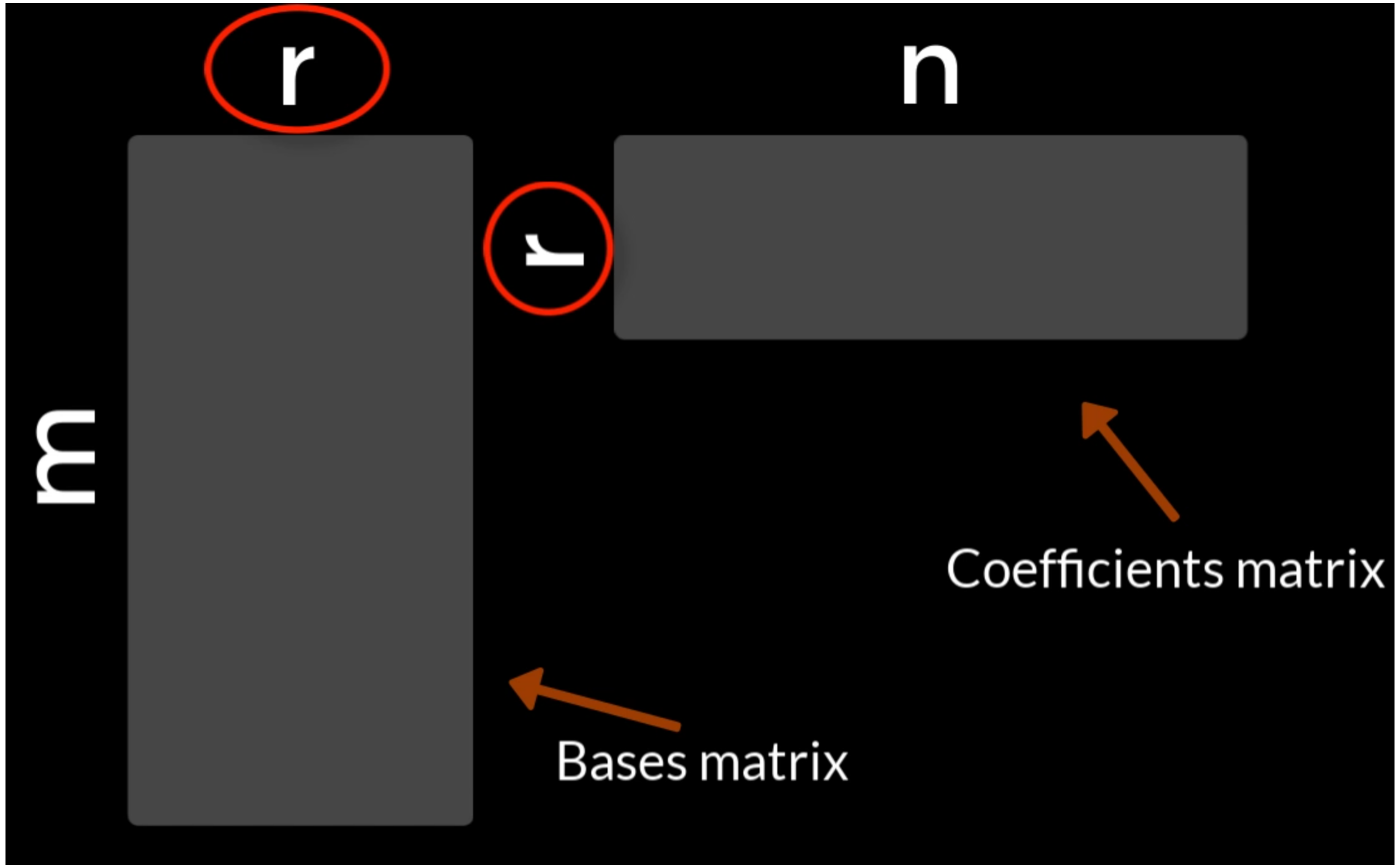
N-NMF: Tearing the data apart



N-NMF: Tearing the data apart



N-NMF: Tearing the data apart



N-NMF: Tearing the data apart

Objective functions for minimizing:

- the square of the Euclidean distance
- Kullback-Leibler divergence

Text mining and dimensionality reduction

What is topic modeling?

- Unsupervised approach to automatically identify *topics*.
- Topics are cluster of words that frequently occur together.

Why is dimensionality reduction important?

- Data sparseness of frequency data
- Word co-occurrence
- **Identifies topics with the new `r` dimensions.**

nmf() for topic detection

BBC's datasets live in: http://mlg.ucd.ie/datasets/bbc.html

```
library(NMF)

bbc_res <- nmf(bbc_tdm, 5)
```

```
W <- basis(bbc_res)

H <- coef(bbc_res)
```

Exploring the term-topic matrix W

```
library(dplyr)

colnames(W) <- c("topic1",
    "topic2", "topic3",
    "topic4", "topic5")

W %>%
    rownames_to_column('words') %>%
    arrange(. , desc(topic1))%>%
    column_to_rownames('words')
```

```
> W %>%
+     rownames_to_column('gene') %>%
+      arrange(. , desc(topic1))%>%
+     column_to_rownames('gene')
                                topic1       topic2       topic3       topic4       topic5
said                      9.656317e+02 1.401476e+03 1.206998e+03 9.956982e+02 1.098070e+03
best                      4.724117e+02 1.241815e+02 1.755207e+01 1.448288e+02 1.098070e+02
year                      4.372389e+02 1.592050e+02 1.599074e+02 2.534505e+02 4.078546e+02
will                      3.712413e+02 9.855786e+02 7.800174e+02 5.069009e+02 5.019749e+02
lord                      3.711806e+02 2.220446e-16 1.755207e+01 2.220446e-16 2.220446e-16
years                     3.388791e+02 2.838433e+02 1.213642e+02 2.715541e+02 1.411804e+02
album                     3.205651e+02 2.220446e-16 2.220446e-16 1.810360e+01 2.220446e-16
new                       3.205651e+02 3.193238e+02 2.106248e+02 4.887973e+02 9.412029e+01
music                     3.036933e+02 7.096084e+01 2.220446e-16 1.086216e+02 2.509874e+02
one                       2.832364e+02 2.941029e+02 2.042035e+02 1.991396e+02 2.980476e+02
number                    2.547049e+02 8.150143e+01 9.319083e+01 1.991396e+02 1.098070e+02
also                      2.481408e+02 3.548042e+02 3.737294e+02 2.715541e+02 4.706015e+02
world                     2.473207e+02 1.940613e+01 3.944496e+01 1.629324e+02 1.098070e+02
can                       2.364046e+02 2.768781e+02 1.646533e+02 4.163829e+02 1.725539e+02
top                       2.346735e+02 7.096084e+01 1.914626e+01 2.172432e+02 3.137343e+01
first                     2.282524e+02 1.064413e+02 1.662427e+02 3.439685e+02 2.509874e+02
band                      2.193340e+02 2.220446e-16 2.220446e-16 7.241442e+01 1.568672e+01
next                      2.170828e+02 1.596619e+02 1.076544e+02 1.810360e+01 1.098070e+02
three                     2.108795e+02 3.896028e+01 7.556072e+01 1.267252e+02 9.412029e+01
made                      2.024622e+02 1.419217e+02 5.265620e+01 1.810360e+02 9.412029e+01
lifts                     2.024622e+02 2.220446e-16 2.220446e-16 2.220446e-16 2.220446e-16
england                   1.989828e+02 3.913884e+01 7.020827e+01 2.220446e-16 6.274686e+01
awards                    1.855903e+02 2.220446e-16 2.220446e-16 9.051802e+01 3.137343e+01
boeing                    1.855903e+02 2.220446e-16 2.220446e-16 2.220446e-16 2.220446e-16
past                      1.855903e+02 3.548042e+01 2.220446e-16 5.431081e+01 4.706015e+01
people                    1.821806e+02 5.535317e+02 2.983851e+02 3.620721e+02 4.549147e+02
award                     1.687185e+02 2.220446e-16 1.755207e+01 1.267252e+02 1.568672e+01
apple                     1.687185e+02 1.774021e+01 2.220446e-16 2.220446e-16 9.412029e+01
chelsea                   1.687185e+02 2.220446e-16 1.755207e+01 2.220446e-16 2.220446e-16
service                   1.687185e+02 1.241815e+02 2.220446e-16 1.086216e+02 9.412029e+01
last                      1.622799e+02 3.172546e+02 1.316099e+02 3.077613e+02 3.451077e+02
won                       1.616534e+02 7.096084e+01 9.511022e+01 1.267252e+02 1.098070e+02
speed                     1.518466e+02 1.774021e+01 2.220446e-16 3.620721e+01 1.568672e+01
high                      1.518466e+02 8.870104e+01 8.776034e+01 2.220446e-16 2.220446e-16
fuel                      1.518466e+02 2.220446e-16 8.776034e+01 1.810360e+01 1.098070e+02
per                       1.518466e+02 2.220446e-16 2.220446e-16 1.810360e+01 2.220446e-16
quarter                   1.458002e+02 2.409786e+01 3.510413e+01 2.220446e-16 6.274686e+01
cup                       1.349748e+02 2.220446e-16 5.265620e+01 1.810360e+01 1.254937e+02
six                       1.349748e+02 7.096084e+01 3.510413e+01 2.220446e-16 4.706015e+01
fans                      1.349748e+02 1.774021e+01 3.510413e+01 3.620721e+01 3.137343e+01
broadband                 1.349748e+02 8.870104e+01 2.220446e-16 2.220446e-16 4.706015e+01
airbus                    1.349748e+02 2.220446e-16 2.220446e-16 2.220446e-16 2.220446e-16
player                    1.349748e+02 5.322063e+01 2.220446e-16 3.620721e+01 4.706015e+01
outkast                   1.349748e+02 2.220446e-16 2.220446e-16 2.220446e-16 2.220446e-16
inquiry                   1.349748e+02 2.220446e-16 2.220446e-16 2.220446e-16 2.220446e-16
judge                     1.349748e+02 2.220446e-16 7.020827e+01 2.220446e-16 2.220446e-16
record                    1.318499e+02 1.452074e+02 5.265620e+01 5.431081e+01 4.706015e+01
time                      1.315854e+02 1.419217e+02 1.614946e+02 1.629324e+02 1.568672e+02
```

said	9.656317e+02	1.401476e+03
best	4.724117e+02	1.241815e+02
year	4.372389e+02	1.592050e+02
will	3.712413e+02	9.855786e+02
lord	3.711806e+02	2.220446e-16
years	3.388791e+02	2.838433e+02
album	3.205651e+02	2.220446e-16
new	3.205651e+02	3.193238e+02
music	3.036933e+02	7.096084e+01
one	2.832364e+02	2.941029e+02
number	2.547049e+02	8.150143e+01
also	2.481408e+02	3.548042e+02
world	2.473207e+02	1.940613e+01
can	2.364046e+02	2.768781e+02
top	2.346735e+02	7.096084e+01
first	2.282524e+02	1.064413e+02
band	2.193340e+02	2.220446e-16
next	2.170828e+02	1.596619e+02
three	2.108795e+02	3.896028e+01
made	2.024622e+02	1.419217e+02
lifts	2.024622e+02	2.220446e-16
england	1.989828e+02	3.913884e+01
awards	1.855903e+02	2.220446e-16
boeing	1.855903e+02	2.220446e-16
past	1.855903e+02	3.548042e+01
people	1.821806e+02	5.535317e+02
award	1.687185e+02	2.220446e-16
apple	1.687185e+02	1.774021e+01
chelsea	1.687185e+02	2.220446e-16
service	1.687185e+02	1.241815e+02
last	1.622799e+02	3.172546e+02
won	1.616534e+02	7.096084e+01

DIMENSIONALITY REDUCTION IN R

Let's practice!