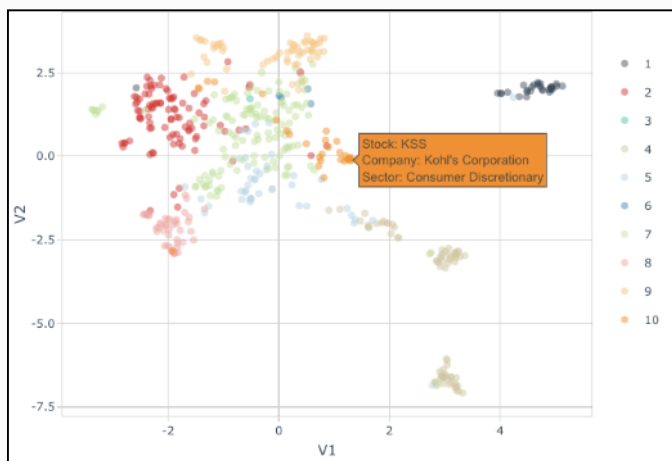


Unsupervised Segmentation with K-Means & UMAP



Combining K-Means & UMAP to visualize clusters by Stock Price Movements

Summary:

- **Common Applications in Business:** Can be used for finding segments within Customers, Companies, etc.
- **Key Concept:** Transform data into a matrix enabling trends to be compared across units of measure (e.g. user-item matrix)
- **Gotchas:** Data must be normalized or standardized to enable comparison. This often requires calculation proportions of values by customer, company, etc to ensure the larger values do not dominate the trend mining operation.
- **How Many Components/Clusters?** Use a *Scree Plot* to determine the proportion of variance explained or total within sum of squares

Type	Popular Methods	Uses	Data Treatment
Clustering	K-Means Hierarchical Clustering	Group Detection: Methods use a measure of similarity (e.g. Euclidean distance) to detect groups within data set	Standardized or normalized
Dimensionality Reduction	PCA UMAP tSNE	Reduce Width of Data: Performing Machine Learning on wide data can drastically increase the time for algorithms to converge. Dimensionality reduction can be applied as a preprocessing step to reduce the width (number of columns) of the data but still maintain a high proportion of the overall structure. Visualization: Visualizing the first two components as X and Y often can enable cluster visualization. Combining with clustering techniques can provide a useful method of visualization.	Standardized or normalized



[R Cheat Sheet](#)

K-Means

```
set.seed(0)
kmeans_obj <- kmeans(X, centers = 4)
```

UMAP

```
library(umap)
umap_obj <- umap(X)
```



[Python Cheat Sheet](#)

K-Means

```
from sklearn.cluster import KMeans
kmeans = KMeans(
    n_clusters=4,
    random_state=0).fit(X)
```

UMAP

```
import umap
reducer = umap.UMAP()
embedding = reducer.fit_transform(X)
```

Resources

- [Business Analysis With R Course \(DS4B 101-R\) - Modeling - Week 6](#)
- [Business Science Problem Framework](#)
- [Ultimate R Cheat Sheet](#) | [Ultimate Python Cheat Sheet](#)

Data Science Courses for Business



Business Science University
university.business-science.io

How to apply K-Means & UMAP step-by-step

Collect Data

```
# sp_500_prices_tbl
# A tibble: 1,225,765 x 8
  symbol date      open  high    low close    volume adjusted
<chr> <date> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 MSFT 2009-01-02 19.5 20.4 19.4 20.3 50084000 15.9
2 MSFT 2009-01-05 20.2 20.7 20.1 20.5 61475200 16.0
3 MSFT 2009-01-06 20.8 21 20.6 20.8 58083400 16.2
4 MSFT 2009-01-07 20.2 20.3 19.5 19.5 72709900 15.2
5 MSFT 2009-01-08 19.6 20.2 19.5 20.1 70255400 15.7
6 MSFT 2009-01-09 20.2 20.3 19.4 19.5 49815300 15.2
7 MSFT 2009-01-12 19.7 19.8 19.3 19.5 52163500 15.2
8 MSFT 2009-01-13 19.5 20.0 19.5 19.8 65843500 15.5
9 MSFT 2009-01-14 19.5 19.7 19.0 19.1 80257500 14.9
10 MSFT 2009-01-15 19.1 19.3 18.5 19.2 96169800 15.0
# ... with 1,225,755 more rows
```

Standardize / Normalize

```
> sp_500_daily_returns_tbl
# A tibble: 141,340 x 3
  symbol date      pct_return
<chr> <date> <dbl>
1 MSFT 2018-01-03 0.00465
2 MSFT 2018-01-04 0.00880
3 MSFT 2018-01-05 0.0124
4 MSFT 2018-01-08 0.00102
5 MSFT 2018-01-09 -0.00608
6 MSFT 2018-01-10 -0.00453
7 MSFT 2018-01-11 0.00296
8 MSFT 2018-01-12 0.0173
9 MSFT 2018-01-16 -0.0140
10 MSFT 2018-01-17 0.0203

#> summarise(n = n())
#> #> with 141,330 more rows
```

Spread to User-Item Format

```
> stock_date_matrix_tbl
# A tibble: 502 x 283
  symbol `2018-01-03` `2018-01-04` `2018-01-05` `2018-01-08`
  <chr>      <dbl>      <dbl>      <dbl>      <dbl>
1 AAL      0.0254      -0.00750     0.0160
2 AAP      -0.0123     0.00630     -0.00380
3 AAP      0.00905     0.0369     0.0106
4 AAPL     -0.000174    0.00465     0.0114
5 ABBV      0.0156     -0.00570     0.0174
6 ABC       0.00372     -0.00222     0.0121
7 ABMD      0.0173     0.0175     0.0154
8 ABT       0.00221     -0.00170     0.00289
9 ACN       0.00462     0.0118     0.00825
10 ADBE      0.0188     0.0120     0.0116
# ... with 492 more rows, and 272 more variables: `2018-01-09`, `2018-01-10`, `2018-01-11`, `2018-01-12`, `2018-01-15`, `2018-01-16`, `2018-01-17`, `2018-01-18`, `2018-01-19`, `2018-01-22`, `2018-01-23`, `2018-01-24`, `2018-01-25`, `2018-01-26`, `2018-01-29`, `2018-01-30`, `2018-01-31`, `2018-02-01`, `2018-02-02`, `2018-02-05`, `2018-02-06`, `2018-02-07`, `2018-02-08`, `2018-02-09`, `2018-02-12`, `2018-02-13`, `2018-02-14`, `2018-02-15`, `2018-02-16`, `2018-02-20`, `2018-02-21`, `2018-02-22`, `2018-02-23`, `2018-02-26`, `2018-02-27`, `2018-02-28`, `2018-03-01`, `2018-03-02`, `2018-03-05`, `2018-03-06`, `2018-03-07`, `2018-03-08`, `2018-03-09`, `2018-03-12`, `2018-03-13`, `2018-03-14`, `2018-03-15`, `2018-03-16`, `2018-03-19`, `2018-03-20`, `2018-03-21`, `2018-03-22`, `2018-03-23`, `2018-03-26`, `2018-03-27`, `2018-03-28`, `2018-03-29`, `2018-03-30`, `2018-03-31`, `2018-04-02`, `2018-04-03`, `2018-04-04`, `2018-04-05`, `2018-04-06`, `2018-04-09`, `2018-04-10`, `2018-04-11`, `2018-04-12`, `2018-04-13`, `2018-04-16`, `2018-04-17`, `2018-04-18`, `2018-04-19`, `2018-04-20`, `2018-04-23`, `2018-04-24`, `2018-04-25`, `2018-04-26`, `2018-04-27`, `2018-04-30`, `2018-05-01`, `2018-05-02`, `2018-05-03`, `2018-05-04`, `2018-05-07`, `2018-05-08`, `2018-05-09`, `2018-05-10`, `2018-05-14`, `2018-05-15`, `2018-05-16`, `2018-05-17`, `2018-05-18`, `2018-05-21`, `2018-05-22`, `2018-05-23`, `2018-05-24`, `2018-05-25`, `2018-05-28`, `2018-05-29`, `2018-05-30`, `2018-05-31`, `2018-06-01`, `2018-06-04`, `2018-06-05`, `2018-06-06`, `2018-06-07`, `2018-06-08`, `2018-06-11`, `2018-06-12`, `2018-06-13`, `2018-06-14`, `2018-06-15`, `2018-06-18`, `2018-06-19`, `2018-06-20`, `2018-06-21`, `2018-06-22`, `2018-06-25`, `2018-06-26`, `2018-06-27`, `2018-06-28`, `2018-06-29`, `2018-06-30`, `2018-07-02`, `2018-07-03`, `2018-07-05`, `2018-07-06`, `2018-07-09`, `2018-07-10`, `2018-07-11`, `2018-07-12`, `2018-07-13`, `2018-07-16`, `2018-07-17`, `2018-07-18`, `2018-07-19`, `2018-07-20`, `2018-07-23`, `2018-07-24`, `2018-07-25`, `2018-07-26`, `2018-07-27`, `2018-07-30`, `2018-07-31`, `2018-08-01`, `2018-08-02`, `2018-08-03`, `2018-08-06`, `2018-08-07`, `2018-08-08`, `2018-08-09`, `2018-08-10`, `2018-08-13`, `2018-08-14`, `2018-08-15`, `2018-08-16`, `2018-08-17`, `2018-08-20`, `2018-08-21`, `2018-08-22`, `2018-08-23`, `2018-08-24`, `2018-08-27`, `2018-08-28`, `2018-08-29`, `2018-08-30`, `2018-08-31`, `2018-09-03`, `2018-09-04`, `2018-09-05`, `2018-09-06`, `2018-09-07`, `2018-09-10`, `2018-09-11`, `2018-09-12`, `2018-09-13`, `2018-09-14`, `2018-09-17`, `2018-09-18`, `2018-09-19`, `2018-09-20`, `2018-09-21`, `2018-09-24`, `2018-09-25`, `2018-09-26`, `2018-09-27`, `2018-09-28`, `2018-09-30`, `2018-10-01`, `2018-10-02`, `2018-10-03`, `2018-10-04`, `2018-10-05`, `2018-10-08`, `2018-10-09`, `2018-10-10`, `2018-10-11`, `2018-10-12`, `2018-10-15`, `2018-10-16`, `2018-10-17`, `2018-10-18`, `2018-10-19`, `2018-10-22`, `2018-10-23`, `2018-10-24`, `2018-10-25`, `2018-10-26`, `2018-10-29`, `2018-10-30`, `2018-10-31`, `2018-11-01`, `2018-11-02`, `2018-11-05`, `2018-11-06`, `2018-11-07`, `2018-11-08`, `2018-11-09`, `2018-11-12`, `2018-11-13`, `2018-11-14`, `2018-11-15`, `2018-11-16`, `2018-11-19`, `2018-11-20`, `2018-11-21`, `2018-11-22`, `2018-11-23`, `2018-11-26`, `2018-11-27`, `2018-11-28`, `2018-11-29`, `2018-11-30`, `2018-12-03`, `2018-12-04`, `2018-12-05`, `2018-12-06`, `2018-12-07`, `2018-12-10`, `2018-12-11`, `2018-12-12`, `2018-12-13`, `2018-12-14`, `2018-12-17`, `2018-12-18`, `2018-12-19`, `2018-12-20`, `2018-12-21`, `2018-12-24`, `2018-12-25`, `2018-12-26`, `2018-12-27`, `2018-12-28`, `2018-12-29`, `2018-12-30`
```

Obtain cluster assignments

Make 2D Projection

Used to pick a value for K clusters for K-means algorithm.

Iteratively calculate "tot.withinss" for values of K.

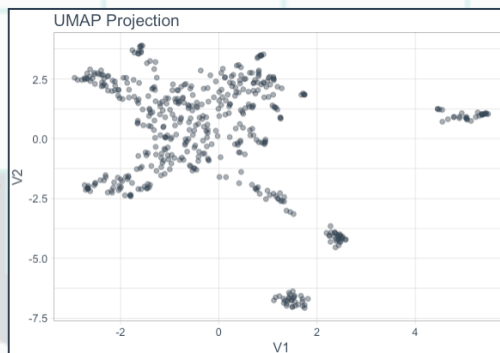
Look for a bend.



Fast dimensionality reduction algorithm that can be used for visualization.

Better than PCA - PCA is linear, UMAP is nonlinear

Better than tSNE - tSNE is slow



Plot the K-Means cluster assignments with the UMAP 2D Projection to obtain a visual.

Add interactivity to enable exploration.

