# Scaleable visualization

Slide credits: Roger Grosse, Ethan Fetaya, James Lucas and Emad Andrews, University of Toronto

# Which dimension is large?

# Which dimension is large?

| | Name | Miles_per_Gallon | Cylinders | Displacement | Horsepower | Weight_in_lbs | Acceleration | Year | Origin |
|---|---|---|---|---|---|---|---|---|---|
| **0** | chevrolet chevelle malibu | 18.0 | 8 | 307.0 | 130.0 | 3504 | 12.0 | 1970-01-01 | USA |
| **1** | buick skylark 320 | 15.0 | 8 | 350.0 | 165.0 | 3693 | 11.5 | 1970-01-01 | USA |
| **2** | plymouth satellite | 18.0 | 8 | 318.0 | 150.0 | 3436 | 11.0 | 1970-01-01 | USA |
| **3** | amc rebel sst | 16.0 | 8 | 304.0 | 150.0 | 3433 | 12.0 | 1970-01-01 | USA |
| **4** | ford torino | 17.0 | 8 | 302.0 | 140.0 | 3449 | 10.5 | 1970-01-01 | USA |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **401** | ford mustang gl | 27.0 | 4 | 140.0 | 86.0 | 2790 | 15.6 | 1982-01-01 | USA |
| **402** | vw pickup | 44.0 | 4 | 97.0 | 52.0 | 2130 | 24.6 | 1982-01-01 | Europe |
| **403** | dodge rampage | 32.0 | 4 | 135.0 | 84.0 | 2295 | 11.6 | 1982-01-01 | USA |
| **404** | ford ranger | 28.0 | 4 | 120.0 | 79.0 | 2625 | 18.6 | 1982-01-01 | USA |
| **405** | chevy s-10 | 31.0 | 4 | 119.0 | 82.0 | 2720 | 19.4 | 1982-01-01 | USA |

406 rows × 9 columns

| | Name | variable | value |
|---|---|---|---|
| **0** | chevrolet chevelle malibu | Miles_per_Gallon | 18.0 |
| **1** | buick skylark 320 | Miles_per_Gallon | 15.0 |
| **2** | plymouth satellite | Miles_per_Gallon | 18.0 |
| **3** | amc rebel sst | Miles_per_Gallon | 16.0 |
| **4** | ford torino | Miles_per_Gallon | 17.0 |
| **...** | ... | ... | ... |
| **3243** | ford mustang gl | Origin | USA |
| **3244** | vw pickup | Origin | Europe |
| **3245** | dodge rampage | Origin | USA |
| **3246** | ford ranger | Origin | USA |
| **3247** | chevy s-10 | Origin | USA |

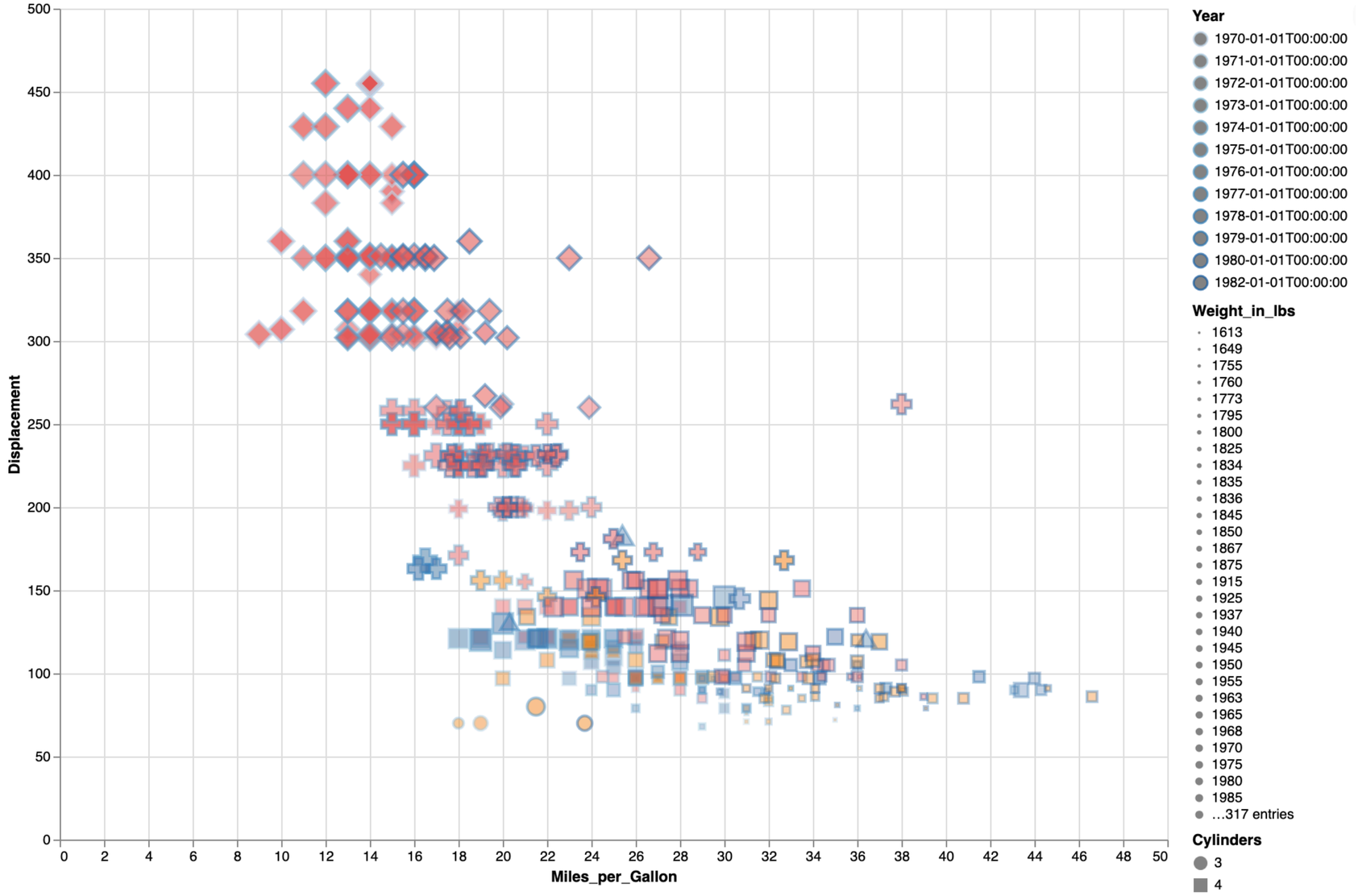
3248 rows × 3 columns

# Pivoted visualizations

# Cars data - Nine Variables

| | Name | Miles_per_Gallon | Cylinders | Displacement | Horsepower | Weight_in_lbs | Acceleration | Year | Origin |
|---|---|---|---|---|---|---|---|---|---|
| **0** | chevrolet chevelle malibu | 18.0 | 8 | 307.0 | 130.0 | 3504 | 12.0 | 1970-01-01 | USA |
| **1** | buick skylark 320 | 15.0 | 8 | 350.0 | 165.0 | 3693 | 11.5 | 1970-01-01 | USA |
| **2** | plymouth satellite | 18.0 | 8 | 318.0 | 150.0 | 3436 | 11.0 | 1970-01-01 | USA |
| **3** | amc rebel sst | 16.0 | 8 | 304.0 | 150.0 | 3433 | 12.0 | 1970-01-01 | USA |
| **4** | ford torino | 17.0 | 8 | 302.0 | 140.0 | 3449 | 10.5 | 1970-01-01 | USA |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **401** | ford mustang gl | 27.0 | 4 | 140.0 | 86.0 | 2790 | 15.6 | 1982-01-01 | USA |
| **402** | vw pickup | 44.0 | 4 | 97.0 | 52.0 | 2130 | 24.6 | 1982-01-01 | Europe |
| **403** | dodge rampage | 32.0 | 4 | 135.0 | 84.0 | 2295 | 11.6 | 1982-01-01 | USA |
| **404** | ford ranger | 28.0 | 4 | 120.0 | 79.0 | 2625 | 18.6 | 1982-01-01 | USA |
| **405** | chevy s-10 | 31.0 | 4 | 119.0 | 82.0 | 2720 | 19.4 | 1982-01-01 | USA |

# 9 Different channels?

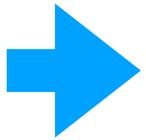


# Pivoted visualizations

# Pivot data

| | Name | Miles_per_Gallon | Cylinders | Displacement | Horsepower | Weight_in_lbs | Acceleration | Year | Origin |
|---|---|---|---|---|---|---|---|---|---|
| **0** | chevrolet chevelle malibu | 18.0 | 8 | 307.0 | 130.0 | 3504 | 12.0 | 1970-01-01 | USA |
| **1** | buick skylark 320 | 15.0 | 8 | 350.0 | 165.0 | 3693 | 11.5 | 1970-01-01 | USA |
| **2** | plymouth satellite | 18.0 | 8 | 318.0 | 150.0 | 3436 | 11.0 | 1970-01-01 | USA |
| **3** | amc rebel sst | 16.0 | 8 | 304.0 | 150.0 | 3433 | 12.0 | 1970-01-01 | USA |
| **4** | ford torino | 17.0 | 8 | 302.0 | 140.0 | 3449 | 10.5 | 1970-01-01 | USA |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **401** | ford mustang gl | 27.0 | 4 | 140.0 | 86.0 | 2790 | 15.6 | 1982-01-01 | USA |
| **402** | vw pickup | 44.0 | 4 | 97.0 | 52.0 | 2130 | 24.6 | 1982-01-01 | Europe |
| **403** | dodge rampage | 32.0 | 4 | 135.0 | 84.0 | 2295 | 11.6 | 1982-01-01 | USA |
| **404** | ford ranger | 28.0 | 4 | 120.0 | 79.0 | 2625 | 18.6 | 1982-01-01 | USA |
| **405** | chevy s-10 | 31.0 | 4 | 119.0 | 82.0 | 2720 | 19.4 | 1982-01-01 | USA |

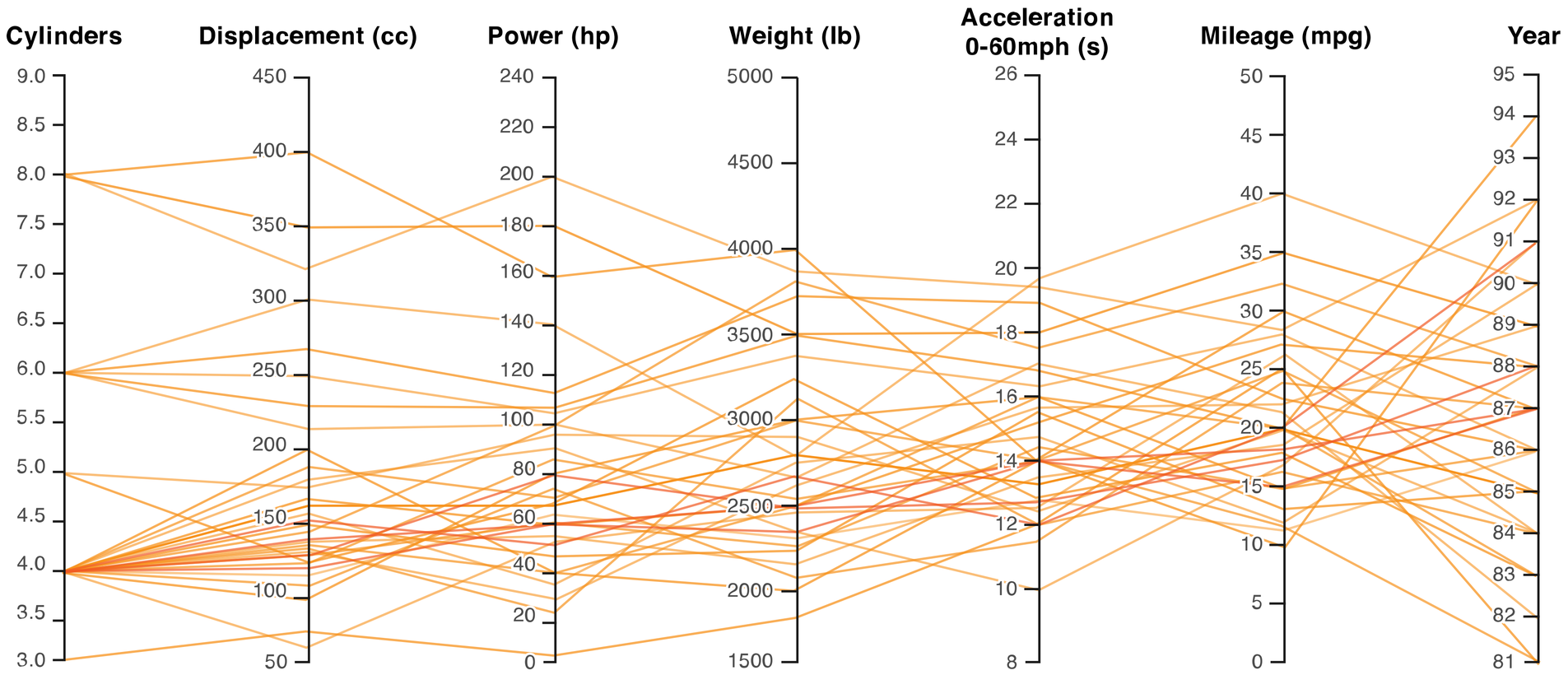
406 rows × 9 columns



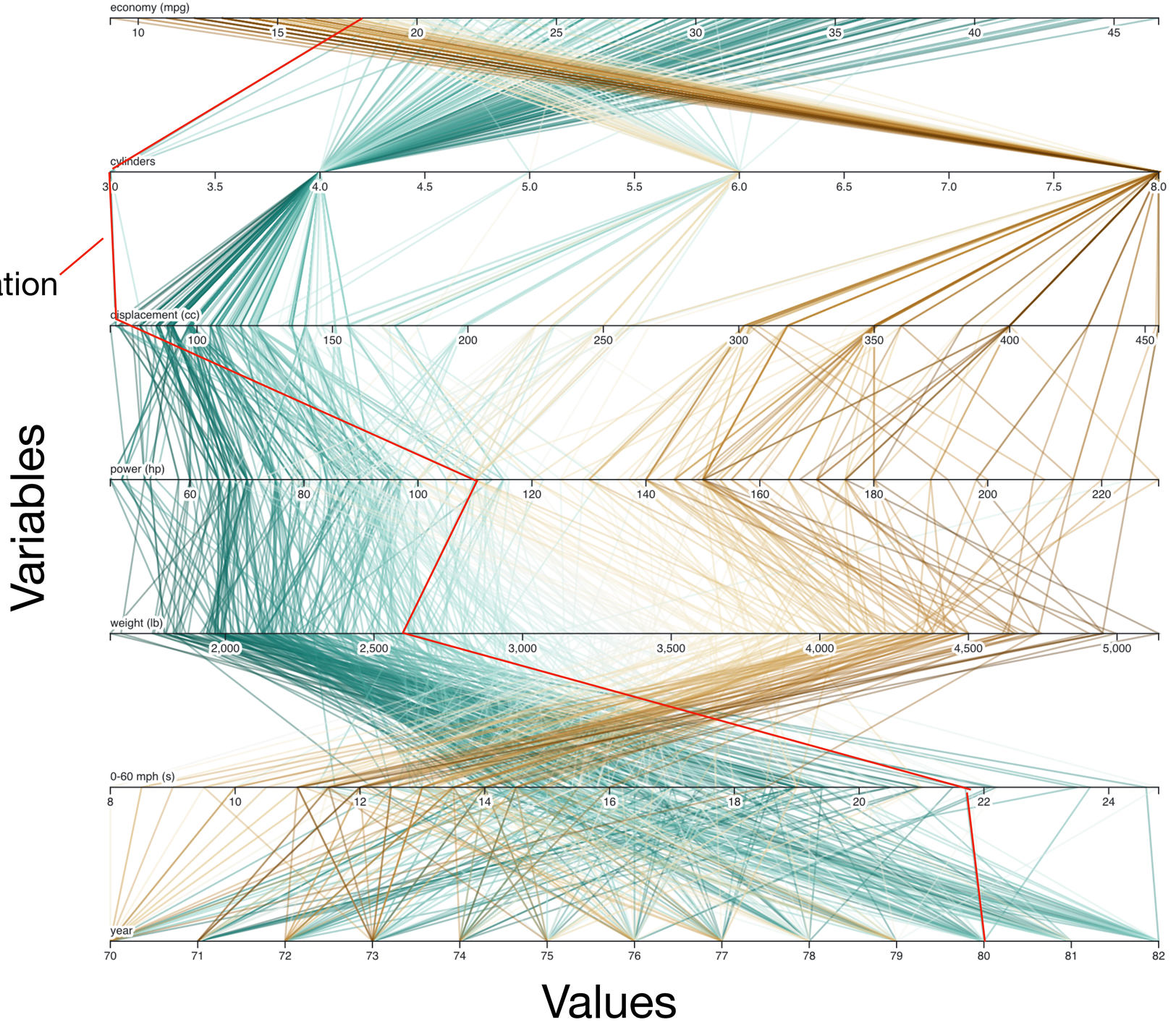
| | Name | variable | value |
|---|---|---|---|
| **0** | chevrolet chevelle malibu | Miles_per_Gallon | 18.0 |
| **1** | buick skylark 320 | Miles_per_Gallon | 15.0 |
| **2** | plymouth satellite | Miles_per_Gallon | 18.0 |
| **3** | amc rebel sst | Miles_per_Gallon | 16.0 |
| **4** | ford torino | Miles_per_Gallon | 17.0 |
| **...** | ... | ... | ... |
| **3243** | ford mustang gl | Origin | USA |
| **3244** | vw pickup | Origin | Europe |
| **3245** | dodge rampage | Origin | USA |
| **3246** | ford ranger | Origin | USA |
| **3247** | chevy s-10 | Origin | USA |

3248 rows × 3 columns

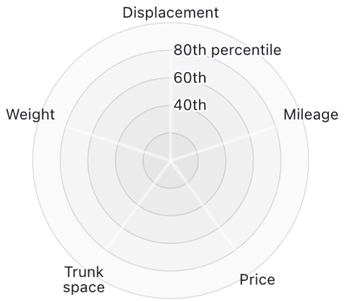
# Parallel Coordinates

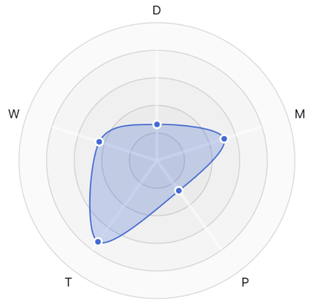


# Parallel coordinates

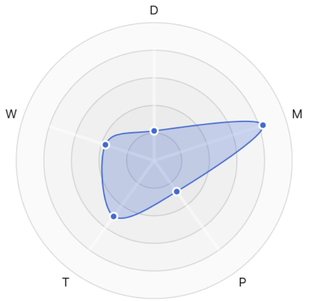


# Radar Charts

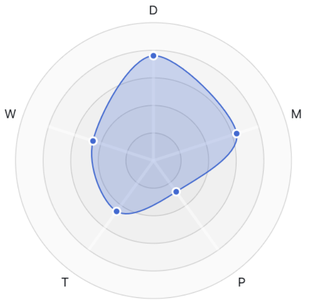


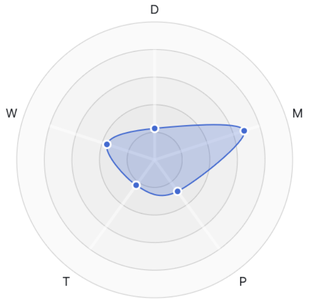

Fiat Strada



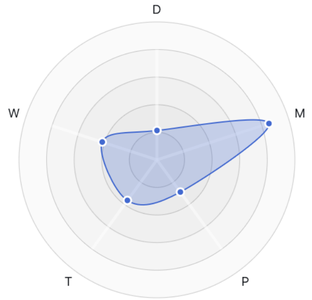

Plymouth Champ



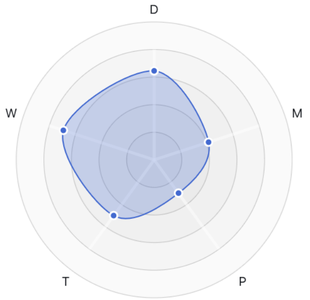

Buick Opel



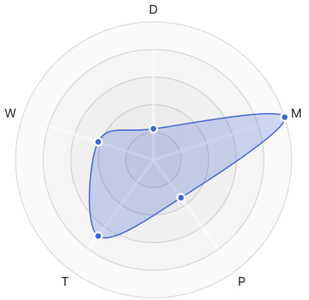

Honda Civic



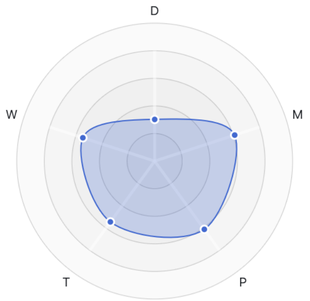

Datsun 210



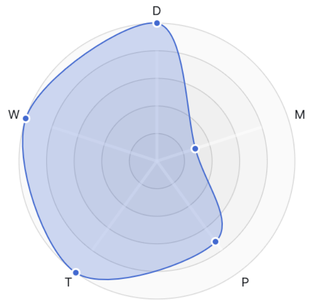

AMC Pacer



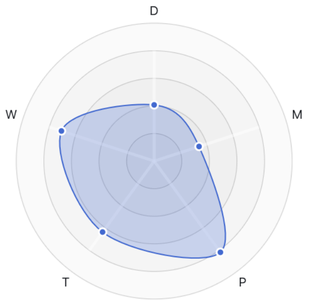

Volkswagen Rabbit Diesel



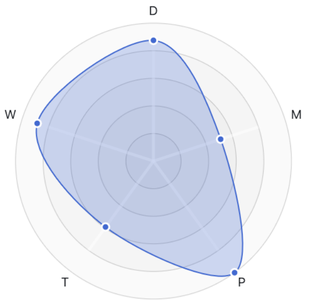

BMW 320i




Lincoln Continental

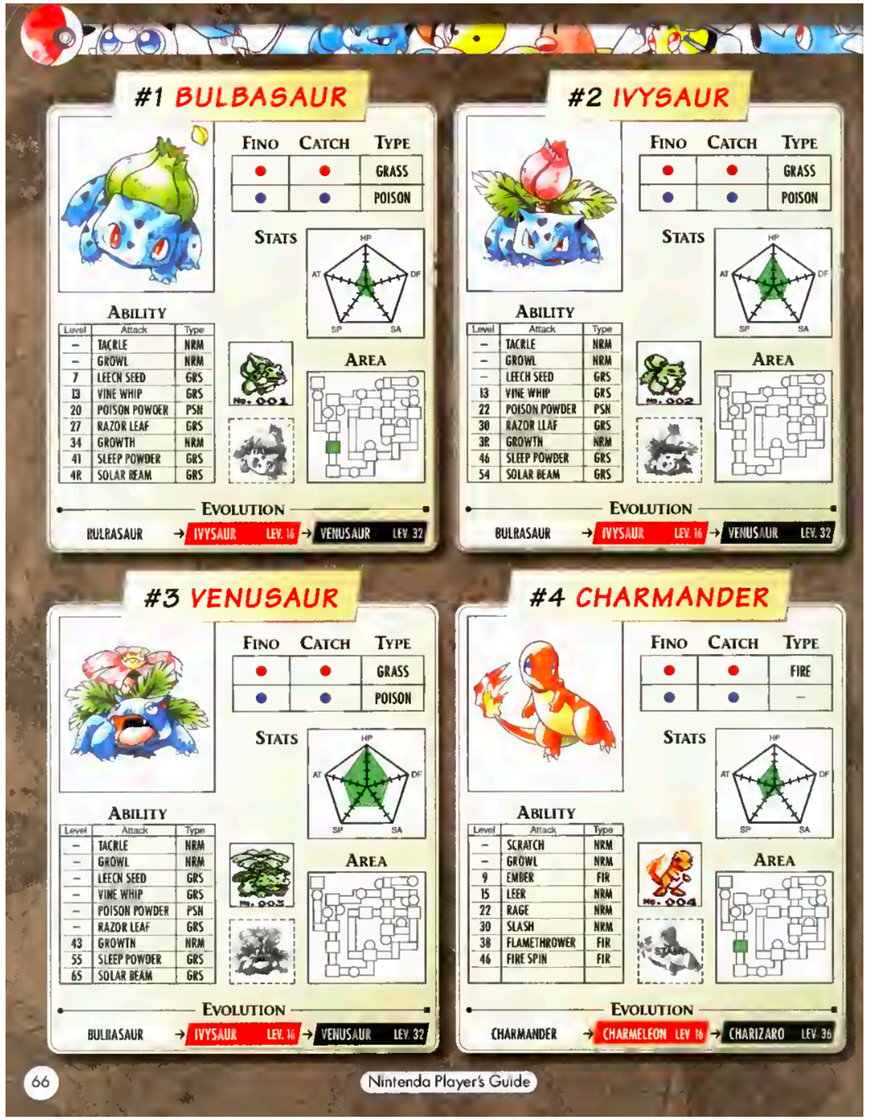



Peugeot 604 SL




Cadillac Seville

# Radar charts



#1 BULBASAUR

| Fino | Catch | Type |
|---|---|---|
| ● | ● | GRASS |
| ● | ● | POISON |

Stats

Ability

| Level | Attack | Type |
|---|---|---|
| – | TACKLE | NRM |
| – | GROWL | NRM |
| 7 | LEECH SEED | GRS |
| 13 | VINE WHIP | GRS |
| 20 | POISON POWDER | PSN |
| 27 | RAZOR LEAF | GRS |
| 34 | GROWTH | NRM |
| 41 | SLEEP POWDER | GRS |
| 48 | SOLAR BEAM | GRS |

Area

Evolution

BULBASAUR → IVYSAUR LEV. 16 → VENUSAUR LEV. 32

#2 IVYSAUR

| Fino | Catch | Type |
|---|---|---|
| ● | ● | GRASS |
| ● | ● | POISON |

Stats

Ability

| Level | Attack | Type |
|---|---|---|
| – | TACKLE | NRM |
| – | GROWL | NRM |
| – | LEECH SEED | GRS |
| 13 | VINE WHIP | GRS |
| 22 | POISON POWDER | PSN |
| 30 | RAZOR LLAF | GRS |
| 38 | GROWTH | NRM |
| 46 | SLEEP POWDER | GRS |
| 54 | SOLAR BEAM | GRS |

Area

Evolution

BULBASAUR → IVYSAUR LEV. 16 → VENUSAUR LEV. 32

#3 VENUSAUR

| Fino | Catch | Type |
|---|---|---|
| ● | ● | GRASS |
| ● | ● | POISON |

Stats

Ability

| Level | Attack | Type |
|---|---|---|
| – | TACKLE | NRM |
| – | GROWL | NRM |
| – | LEECH SEED | GRS |
| – | VINE WHIP | GRS |
| – | POISON POWDER | PSN |
| – | RAZOR LEAF | GRS |
| 43 | GROWTH | NRM |
| 55 | SLEEP POWDER | GRS |
| 65 | SOLAR BEAM | GRS |

Area

Evolution

BULBASAUR → IVYSAUR LEV. 16 → VENUSAUR LEV. 32

#4 CHARMANDER

| Fino | Catch | Type |
|---|---|---|
| ● | ● | FIRE |
| ● | ● | – |

Stats

Ability

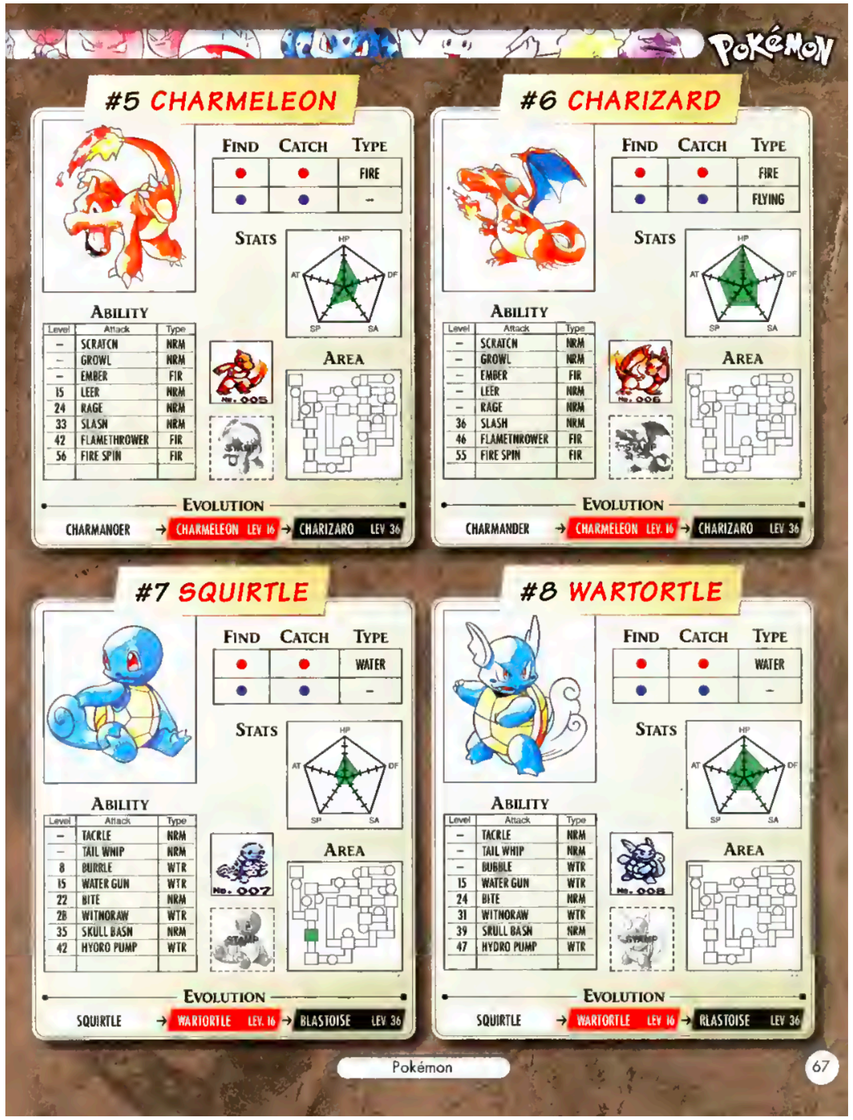
| Level | Attack | Type |
|---|---|---|
| – | SCRATCH | NRM |
| – | GROWL | NRM |
| 9 | EMBER | FIR |
| 15 | LEER | NRM |
| 22 | RAGE | NRM |
| 30 | SLASH | NRM |
| 38 | FLAMETHROWER | FIR |
| 46 | FIRE SPIN | FIR |

Area

Evolution

CHARMANDER → CHARMELEON LEV 16 → CHARIZARO LEV 36







#5 CHARMELEON

| Find | Catch | Type |
|---|---|---|
| ● | ● | FIRE |
| ● | ● | – |

Stats

Ability

| Level | Attack | Type |
|---|---|---|
| – | SCRATCH | NRM |
| – | GROWL | NRM |
| – | EMBER | FIR |
| 15 | LEER | NRM |
| 24 | RAGE | NRM |
| 33 | SLASH | NRM |
| 42 | FLAMETHROWER | FIR |
| 56 | FIRE SPIN | FIR |

Area

Evolution

CHARMANOER → CHARMELEON LEV 16 → CHARIZARO LEV 36

#6 CHARIZARD

| Find | Catch | Type |
|---|---|---|
| ● | ● | FIRE |
| ● | ● | FLYING |

Stats

Ability

| Level | Attack | Type |
|---|---|---|
| – | SCRATCH | NRM |
| – | GROWL | NRM |
| – | EMBER | FIR |
| – | LEER | NRM |
| – | RAGE | NRM |
| 36 | SLASH | NRM |
| 46 | FLAMETHROWER | FIR |
| 55 | FIRE SPIN | FIR |

Area

Evolution

CHARMANDER → CHARMELEON LEV. 16 → CHARIZARO LEV. 36

#7 SQUIRTLE

| Find | Catch | Type |
|---|---|---|
| ● | ● | WATER |
| ● | ● | – |

Stats

Ability

| Level | Attack | Type |
|---|---|---|
| – | TACKLE | NRM |
| – | TAIL WHIP | NRM |
| 8 | BUBBLE | WTR |
| 15 | WATER GUN | WTR |
| 22 | BITE | NRM |
| 28 | WITNORAW | WTR |
| 35 | SKULL BASN | NRM |
| 42 | HYDRO PUMP | WTR |

Area

Evolution

SQUIRTLE → WARTORTLE LEV. 16 → BLASTOISE LEV 36

#8 WARTORTLE

| Find | Catch | Type |
|---|---|---|
| ● | ● | WATER |
| ● | ● | – |

Stats

Ability

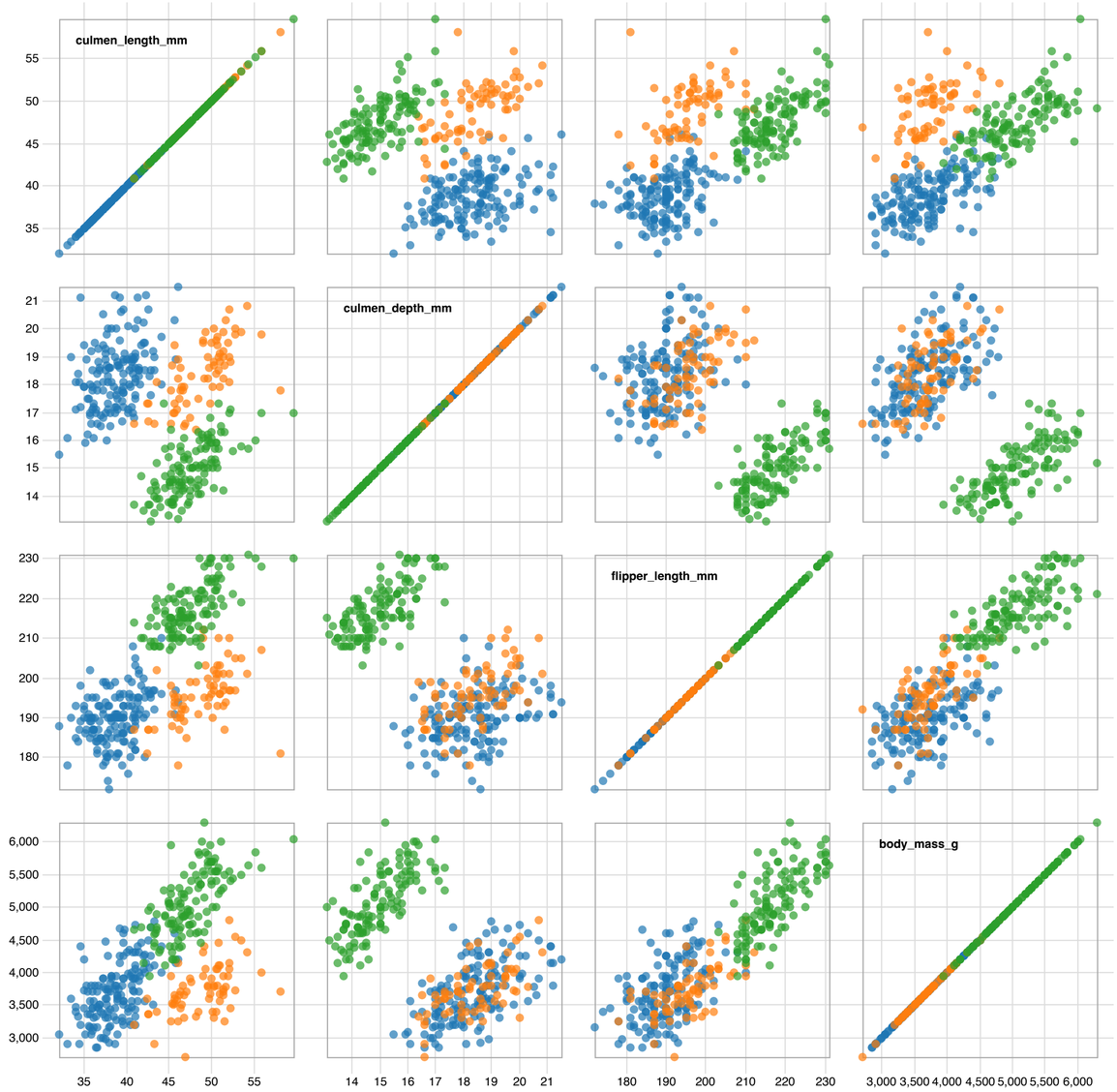
| Level | Attack | Type |
|---|---|---|
| – | TACKLE | NRM |
| – | TAIL WHIP | NRM |
| – | BUBBLE | WTR |
| 15 | WATER GUN | WTR |
| 24 | BITE | NRM |
| 31 | WITNORAW | WTR |
| 39 | SKULL BASN | NRM |
| 47 | HYDRO PUMP | WTR |

Area

Evolution

SQUIRTLE → WARTORTLE LEV 16 → RLASTOISE LEV 36

# Scatterplot matrix



# Dimensionality reduction

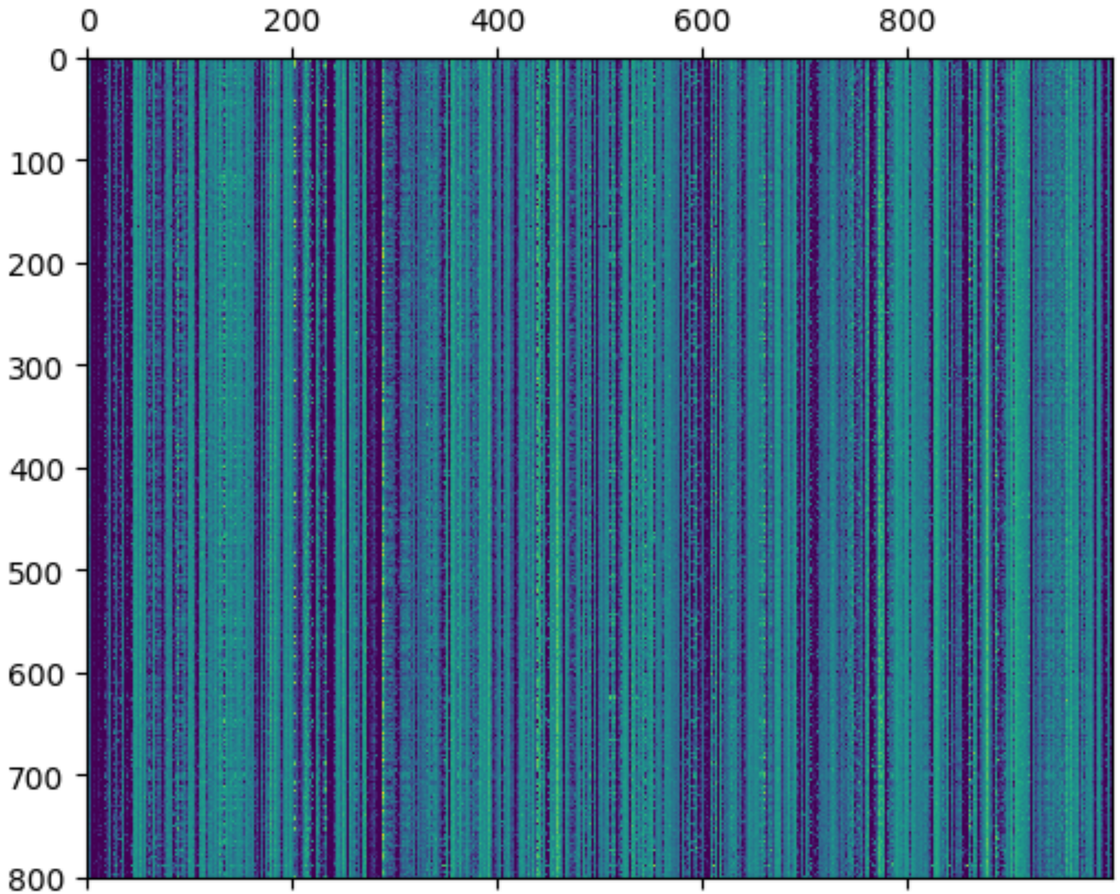
# RNA Gene Expression Data

Thousands of variables!

| | gene_0 | gene_1 | gene_2 | gene_3 | gene_4 | gene_5 | gene_6 | gene_7 | gene_8 | gene_9 | ... | gene_20521 | gene_20522 | gene_20523 | gene_20524 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0.0 | 2.017209 | 3.265527 | 5.478487 | 10.431999 | 0.0 | 7.175175 | 0.591871 | 0.0 | 0.0 | ... | 4.926711 | 8.210257 | 9.723516 | 7.220030 |
| 1 | 0.0 | 0.592732 | 1.588421 | 7.586157 | 9.623011 | 0.0 | 6.816049 | 0.000000 | 0.0 | 0.0 | ... | 4.593372 | 7.323865 | 9.740931 | 6.256586 |
| 2 | 0.0 | 3.511759 | 4.327199 | 6.881787 | 9.870730 | 0.0 | 6.972130 | 0.452595 | 0.0 | 0.0 | ... | 5.125213 | 8.127123 | 10.908640 | 5.401607 |
| 3 | 0.0 | 3.663618 | 4.507649 | 6.659068 | 10.196184 | 0.0 | 7.843375 | 0.434882 | 0.0 | 0.0 | ... | 6.076566 | 8.792959 | 10.141520 | 8.942805 |
| 4 | 0.0 | 2.655741 | 2.821547 | 6.539454 | 9.738265 | 0.0 | 6.566967 | 0.360982 | 0.0 | 0.0 | ... | 5.996032 | 8.891425 | 10.373790 | 7.181162 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 796 | 0.0 | 1.865642 | 2.718197 | 7.350099 | 10.006003 | 0.0 | 6.764792 | 0.496922 | 0.0 | 0.0 | ... | 6.088133 | 9.118313 | 10.004852 | 4.484415 |
| 797 | 0.0 | 3.942955 | 4.453807 | 6.346597 | 10.056868 | 0.0 | 7.320331 | 0.000000 | 0.0 | 0.0 | ... | 6.371876 | 9.623335 | 9.823921 | 6.555327 |
| 798 | 0.0 | 3.249582 | 3.707492 | 8.185901 | 9.504082 | 0.0 | 7.536589 | 1.811101 | 0.0 | 0.0 | ... | 5.719386 | 8.610704 | 10.485517 | 3.589763 |
| 799 | 0.0 | 2.590339 | 2.787976 | 7.318624 | 9.987136 | 0.0 | 9.213464 | 0.000000 | 0.0 | 0.0 | ... | 5.785237 | 8.605387 | 11.004677 | 4.745888 |
| 800 | 0.0 | 2.325242 | 3.805932 | 6.530246 | 9.560367 | 0.0 | 7.957027 | 0.000000 | 0.0 | 0.0 | ... | 6.403075 | 8.594354 | 10.243079 | 9.139459 |

801 rows × 20531 columns

# A Pivoted visualization



# Images

- Images are intrinsically low-dimensional. Consider MNIST.
- Input space: $28 \times 28 = 784$ pixel values
- A lower dimensional representation: describe the strokes using 20 or so control points, plus a few more parameters for thickness, etc.

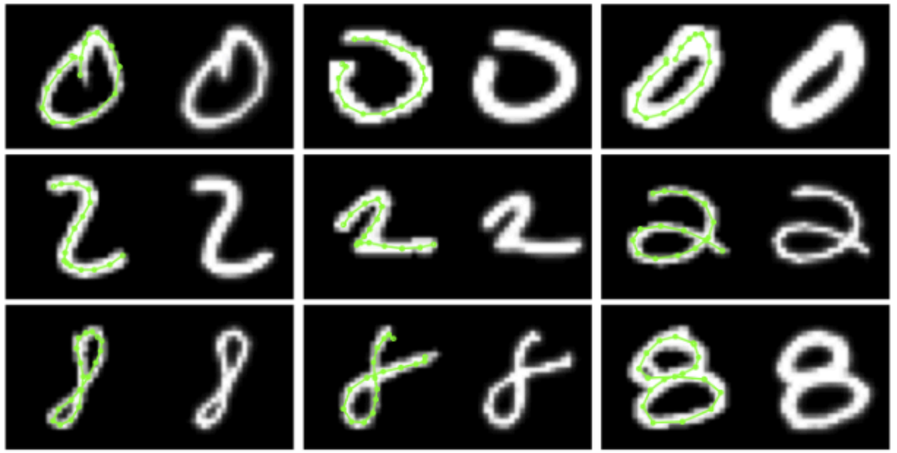
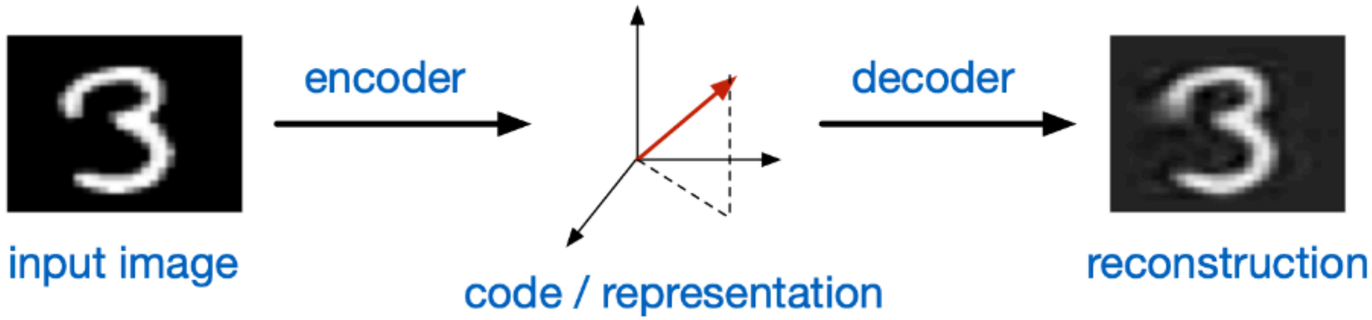


Image credit: Nair and Hinton (2006)

- Can we learn low-dimensional representations directly from the data?

# Dimensionality reduction

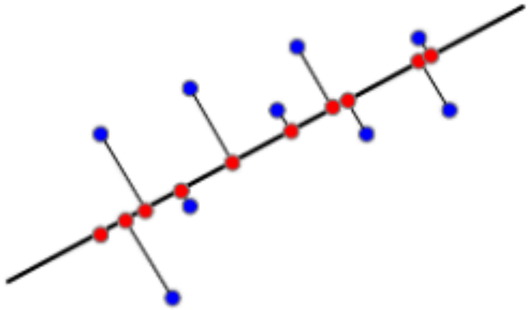



- In dimensionality reduction, we try to learn a mapping to a lower dimensional space that preserves as much information as possible about the input.
- Motivations
  - Save computation/memory
  - Reduce overfitting
  - Visualize in 2 dimensions

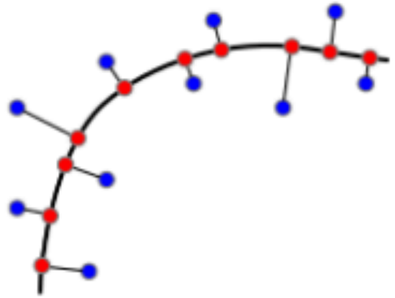
# Dimensionality reduction

Can be linear or nonlinear:

original data

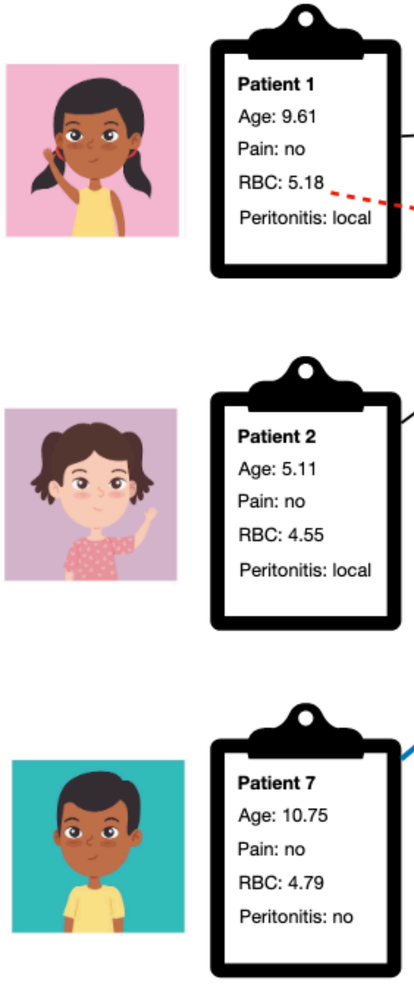


linear dimensionality reduction



nonlinear dimensionality reduction

# Tabular data




Patients with abdominal pain

| | Age | Appendix Size | Migratory Pain | RBC Count | RBC Urine | Peritonitis |
|---|---|---|---|---|---|---|
| 0 | 9.61 | 9.0 | no | 5.18 | high | local |
| 1 | 5.11 | 7.0 | no | 4.55 | medium | local |
| 2 | 10.75 | 5.0 | no | 4.79 | none | no |
| 3 | 10.51 | 9.0 | no | 5.03 | none | local |
| 4 | 7.3 | 6.2 | yes | 4.64 | low | no |
| 5 | 15.21 | 8.5 | yes | 4.62 | low | no |
| 6 | 15.83 | 12.0 | yes | 4.33 | high | no |
| 7 | 9.58 | 7.0 | yes | 5.04 | low | generalized |
| 8 | 10.37 | 5.5 | no | 4.8 | none | no |
| 9 | 16.66 | 9.0 | yes | 5.31 | none | no |
| 10 | 14.52 | 4.5 | yes | 4.9 | none | no |
| 11 | 10.74 | 9.0 | no | 5.66 | none | local |
| 12 | 12.41 | 3.7 | no | 5.49 | none | no |
| 13 | 6.67 | 3.5 | no | 5.27 | none | no |
| 14 | 14.36 | 9.0 | yes | 4.84 | low | local |
| 15 | 9.04 | 5.3 | yes | 4.92 | low | no |
| 16 | 12.43 | 12.0 | yes | 4.62 | none | generalized |

Cell

Observation

Feature

Patients

Table

# Mathematical abstraction

## As table

Patients with abdominal pain

| | Age | Appendix Size | Height | Weight | WBC Count | Temperature | WBC Count |
|---|---|---|---|---|---|---|---|
| 0 | 16.66 | 9.0 | 174.0 | 65.0 | 5.31 | 36.6 | 6.6 |
| 1 | 10.74 | 9.0 | 146.0 | 57.5 | 5.66 | 37.3 | 10.2 |
| 2 | 9.04 | 5.3 | 134.0 | 29.4 | 4.92 | 36.0 | 5.1 |
| 3 | 10.75 | 5.0 | 155.0 | 54.5 | 4.79 | 37.7 | 10.3 |
| 4 | 7.3 | 6.2 | 123.0 | 23.5 | 4.64 | 37.4 | 21.1 |
| 5 | 5.11 | 7.0 | 116.0 | 22.0 | 4.55 | 40.2 | 19.4 |
| 6 | 14.36 | 9.0 | 163.0 | 50.0 | 4.84 | 37.5 | 14.3 |
| 7 | 9.61 | 9.0 | 140.0 | 29.2 | 5.18 | 38.7 | 14.3 |
| 8 | 15.83 | 12.0 | 153.0 | 59.0 | 4.33 | 36.7 | 12.8 |
| 9 | 9.58 | 7.0 | 132.0 | 24.7 | 5.04 | 38.4 | 13.5 |
| 10 | 10.37 | 5.5 | 156.0 | 39.0 | 4.8 | 37.4 | 5.6 |
| 11 | 14.52 | 4.5 | 181.0 | 55.0 | 4.9 | 37.0 | 9.0 |
| 12 | 12.41 | 3.7 | 150.5 | 42.5 | 5.49 | 37.2 | 9.1 |
| 13 | 6.67 | 3.5 | 124.0 | 38.5 | 5.27 | 39.6 | 16.8 |
| 14 | 15.21 | 8.5 | 155.0 | 85.0 | 4.62 | 36.8 | 12.4 |
| 15 | 12.43 | 12.0 | 157.0 | 46.0 | 4.62 | 37.1 | 16.4 |
| 16 | 10.51 | 9.0 | 134.5 | 27.0 | 5.03 | 37.4 | 12.8 |

Observation

Feature

## As matrix

Observation

$$\mathbf{X} = \begin{bmatrix} 16.66 & 9.0 & 174.0 & 65.0 & 5.31 & 36.6 & 6.6 \\ 10.74 & 9.0 & 146.0 & 57.5 & 5.66 & 37.3 & 10.2 \\ 9.04 & 5.3 & 134.0 & 29.4 & 4.92 & 36.0 & 5.1 \\ 10.75 & 5.0 & 155.0 & 54.5 & 4.79 & 37.7 & 10.3 \\ 7.3 & 6.2 & 123.0 & 23.5 & 4.64 & 37.4 & 21.1 \\ 5.11 & 7.0 & 116.0 & 22.0 & 4.55 & 40.2 & 19.4 \\ 14.36 & 9.0 & 163.0 & 50.0 & 4.84 & 37.5 & 14.3 \\ 9.61 & 9.0 & 140.0 & 29.2 & 5.18 & 38.7 & 14.3 \\ 15.83 & 12.0 & 153.0 & 59.0 & 4.33 & 36.7 & 12.8 \\ 9.58 & 7.0 & 132.0 & 24.7 & 5.04 & 38.4 & 13.5 \\ 10.37 & 5.5 & 156.0 & 39.0 & 4.8 & 37.4 & 5.6 \\ 14.52 & 4.5 & 181.0 & 55.0 & 4.9 & 37.0 & 9.0 \\ 12.41 & 3.7 & 150.5 & 42.5 & 5.49 & 37.2 & 9.1 \\ 6.67 & 3.5 & 124.0 & 38.5 & 5.27 & 39.6 & 16.8 \\ 15.21 & 8.5 & 155.0 & 85.0 & 4.62 & 36.8 & 12.4 \\ 12.43 & 12.0 & 157.0 & 46.0 & 4.62 & 37.1 & 16.4 \\ 10.51 & 9.0 & 134.5 & 27.0 & 5.03 & 37.4 & 12.8 \end{bmatrix}$$

$N$ (Obs.)

Feature

$d$ (features)

N x d Matrix: $\mathbf{X} \in \mathbb{R}^{N \times d}$

# Matrix, vector and scalar notation

As matrix

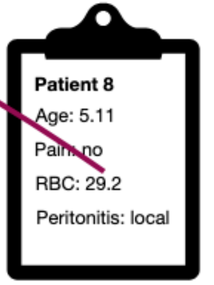
Observation

$$\mathbf{X} = \begin{bmatrix}
16.66 & 9.0 & 174.0 & 65.0 & 5.31 & 36.6 & 6.6 \\
10.74 & 9.0 & 146.0 & 57.5 & 5.66 & 37.3 & 10.2 \\
9.04 & 5.3 & 134.0 & 29.4 & 4.92 & 36.0 & 5.1 \\
10.75 & 5.0 & 155.0 & 54.5 & 4.79 & 37.7 & 10.3 \\
7.3 & 6.2 & 123.0 & 23.5 & 4.64 & 37.4 & 21.1 \\
5.11 & 7.0 & 116.0 & 22.0 & 4 & 0.2 & 19.4 \\
14.36 & 9.0 & 163.0 & 50.0 & 4. & 7.5 & 14.3 \\
9.61 & 9.0 & 140.0 & 29.2 & 5.18 & 38.7 & 14.3 \\
15.83 & 12.0 & 153.0 & 59.0 & 4.33 & 36.7 & 12.8 \\
9.58 & 7.0 & 132.0 & 24.7 & 5.04 & 38.4 & 13.5 \\
10.37 & 5.5 & 156.0 & 39.0 & 4.8 & 37.4 & 5.6 \\
14.52 & 4.5 & 181.0 & 55.0 & 4.9 & 37.0 & 9.0 \\
12.41 & 3.7 & 150.5 & 42.5 & 5.49 & 37.2 & 9.1 \\
6.67 & 3.5 & 124.0 & 38.5 & 5.27 & 39.6 & 16.8 \\
15.21 & 8.5 & 155.0 & 85.0 & 4.62 & 36.8 & 12.4 \\
12.43 & 12.0 & 157.0 & 46.0 & 4.62 & 37.1 & 16.4 \\
10.51 & 9.0 & 134.5 & 27.0 & 5.03 & 37.4 & 12.8
\end{bmatrix}$$

Cell

$$x_{84} = 29.2$$

Row (Obs.) / Column (Feature)



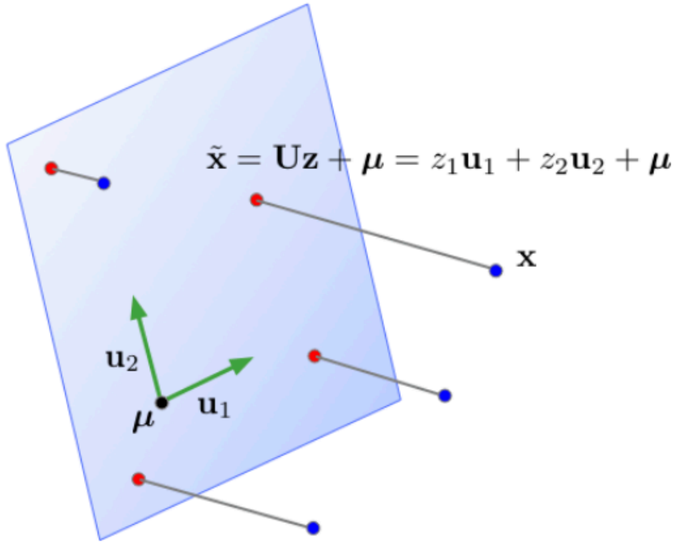

$$\mathbf{x}_8 = \begin{bmatrix} 9.61 \\ 9.0 \\ 140.0 \\ 29.2 \\ 5.18 \\ 38.7 \\ 14.3 \end{bmatrix}$$

Row (Obs.)

Patient

# Principal Component Analysis

# Projection onto a subspace



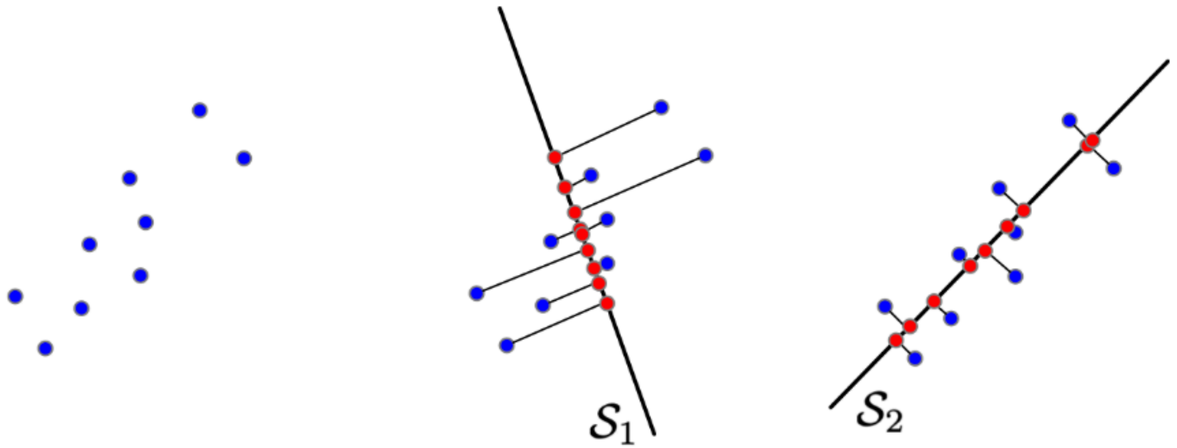

$$\mathbf{z} = \mathbf{U}^\top(\mathbf{x} - \boldsymbol{\mu})$$

- Here, the columns of $\mathbf{U}$ form an orthonormal basis for a subspace $\mathcal{S}$.
- The projection of a point $\mathbf{x}$ onto $\mathcal{S}$ is the point $\tilde{\mathbf{x}} \in \mathcal{S}$ closest to $\mathbf{x}$. In machine learning, $\tilde{\mathbf{x}}$ is also called the reconstruction of $\mathbf{x}$.
- $\mathbf{z}$ is its representation, or code.

# Projection onto a subspace

# Projections onto a subspace

- Which of the following subspaces is a better representation of the dataset?




- On average, the data points are closer to $\mathcal{S}_2$ than to $\mathcal{S}_1$.
- The projections onto $\mathcal{S}_2$ are more spread out than the projections onto $\mathcal{S}_1$.

- How to choose a good subspace $\mathcal{S}$?
  - Need to choose a vector $\boldsymbol{\mu}$ and a $D \times K$ matrix $\mathbf{U}$ with orthonormal columns.
- Set $\boldsymbol{\mu}$ to the mean of the data, $\boldsymbol{\mu} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}^{(i)}$
- Two criteria:
  - Minimize the reconstruction error

$$\min \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{x}^{(i)} - \tilde{\mathbf{x}}^{(i)}\|^2$$

- How to choose a good subspace $\mathcal{S}$?
  - Need to choose a vector $\boldsymbol{\mu}$ and a $D \times K$ matrix $\mathbf{U}$ with orthonormal columns.
- Set $\boldsymbol{\mu}$ to the mean of the data, $\boldsymbol{\mu} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}^{(i)}$
- Two criteria:
  - Maximize the variance of the code vectors

$$\begin{aligned} \max \sum_j \mathrm{Var}(z_j) &= \frac{1}{N} \sum_j \sum_i (z_j^{(i)} - \bar{z}_j)^2 \\ &= \frac{1}{N} \sum_i \|\mathbf{z}^{(i)} - \bar{\mathbf{z}}\|^2 \\ &= \frac{1}{N} \sum_i \|\mathbf{z}^{(i)}\|^2 \end{aligned}$$

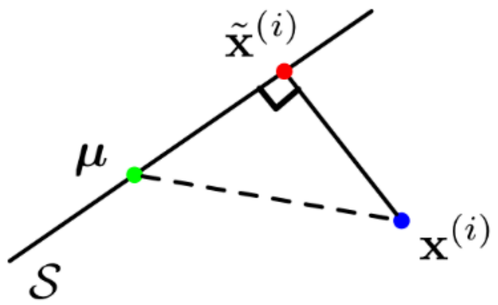
- These two criteria are equivalent! I.e., we'll show

$$\frac{1}{N}\sum_{i=1}^{N}\|\mathbf{x}^{(i)}-\tilde{\mathbf{x}}^{(i)}\|^2 = \text{const} - \frac{1}{N}\sum_{i}\|\mathbf{z}^{(i)}\|^2$$

- Observation: by unitarity,

$$\|\tilde{\mathbf{x}}^{(i)}-\boldsymbol{\mu}\| = \|\mathbf{U}\mathbf{z}^{(i)}\| = \|\mathbf{z}^{(i)}\|$$

- By the Pythagorean Theorem,




$$\underbrace{\frac{1}{N}\sum_{i=1}^{N}\|\tilde{\mathbf{x}}^{(i)}-\boldsymbol{\mu}\|^2}_{\text{projected variance}} + \underbrace{\frac{1}{N}\sum_{i=1}^{N}\|\mathbf{x}^{(i)}-\tilde{\mathbf{x}}^{(i)}\|^2}_{\text{reconstruction error}}$$

$$= \underbrace{\frac{1}{N}\sum_{i=1}^{N}\|\mathbf{x}^{(i)}-\boldsymbol{\mu}\|^2}_{\text{constant}}$$

Choosing a subspace to maximize the projected variance, or minimize the reconstruction error, is called principal component analysis (PCA).

Recall:

- Spectral Decomposition: a symmetric matrix $\mathbf{A}$ has a full set of eigenvectors, which can be chosen to be orthogonal. This gives a decomposition

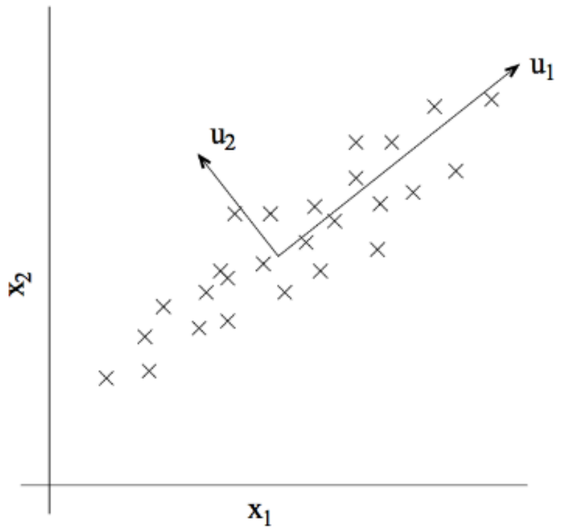
$$\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top,$$

  where $\mathbf{Q}$ is orthogonal and $\mathbf{\Lambda}$ is diagonal. The columns of $\mathbf{Q}$ are eigenvectors, and the diagonal entries $\lambda_j$ of $\mathbf{\Lambda}$ are the corresponding eigenvalues.
- I.e., symmetric matrices are diagonal in some basis.
- A symmetric matrix $\mathbf{A}$ is positive semidefinite iff each $\lambda_j \geq 0$.

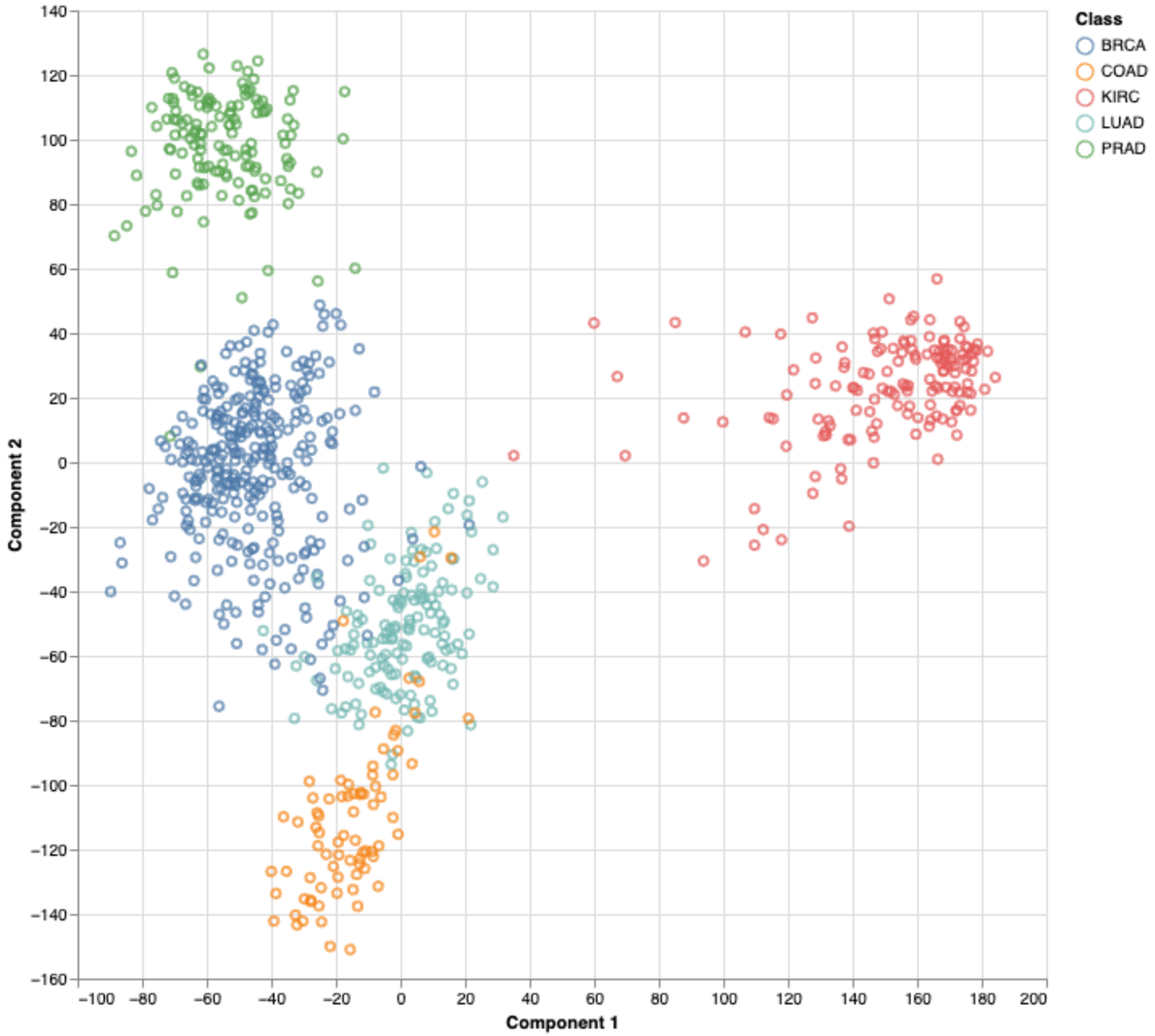
- Consider the empirical covariance matrix:

$$\boldsymbol{\Sigma} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}^{(i)} - \boldsymbol{\mu})(\mathbf{x}^{(i)} - \boldsymbol{\mu})^{\top}$$

- Recall: Covariance matrices are symmetric and positive semidefinite.
- The optimal PCA subspace is spanned by the top $K$ eigenvectors of $\boldsymbol{\Sigma}$.
  - More precisely, choose the first $K$ of any orthonormal eigenbasis for $\boldsymbol{\Sigma}$.
  - The general case is tricky, but we'll show this for $K = 1$.
- These eigenvectors are called principal components, analogous to the principal axes of an ellipse.



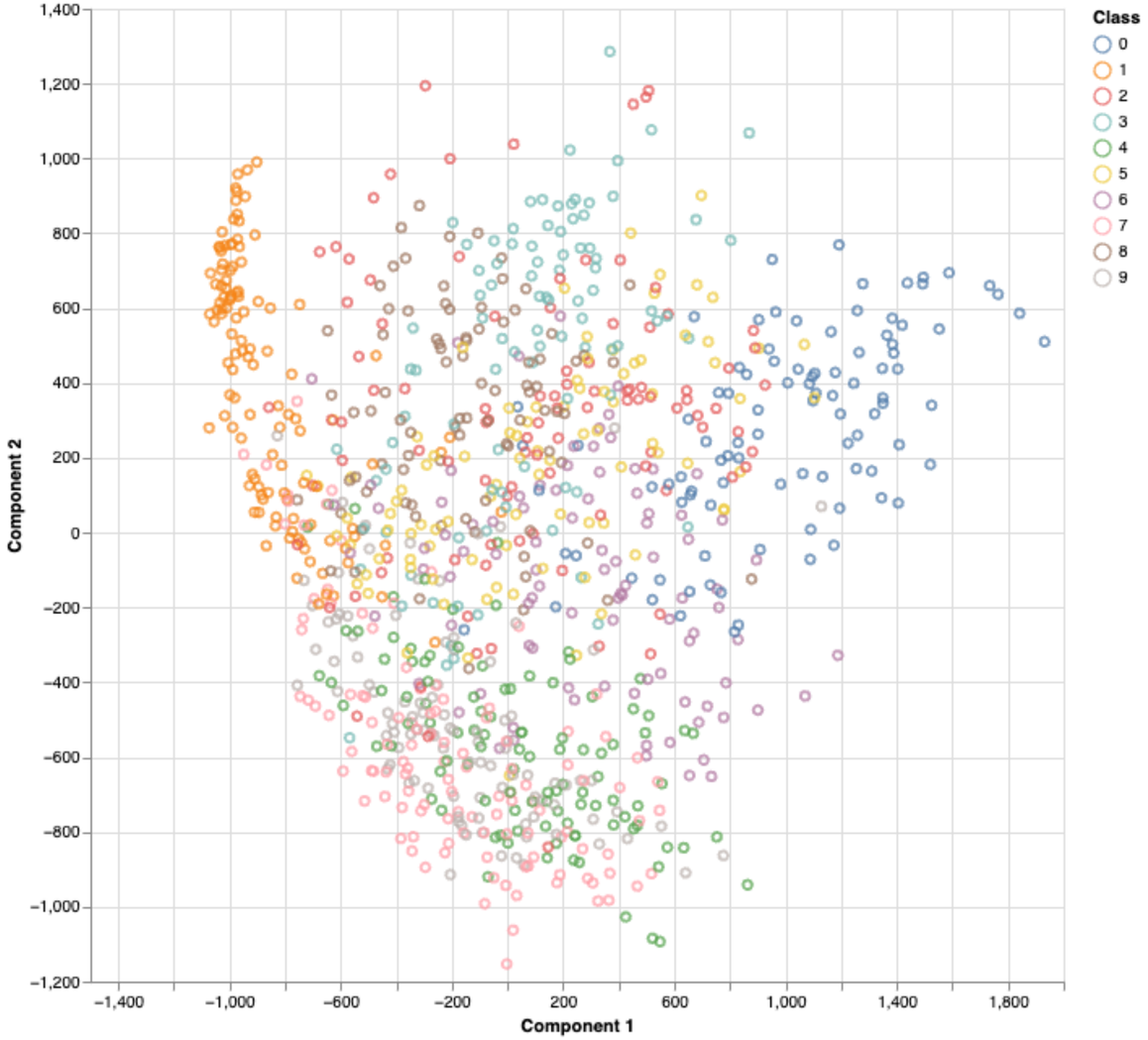
# PCA For gene expression




```
from sklearn.decomposition import PCA
reduced = PCA(2).fit_transform(rna_data.values)
reduced_data = pd.DataFrame({"Component 1": reduced[:, 0], "
```

```
alt.Chart(reduced_data).mark_point().encode(
    x="Component 1:Q",
    y="Component 2:Q",
    color="Class:N"
).properties(
    width=600,
    height=600
)
```

# PCA for MNIST



# Stochastic Neighbor Embeddings

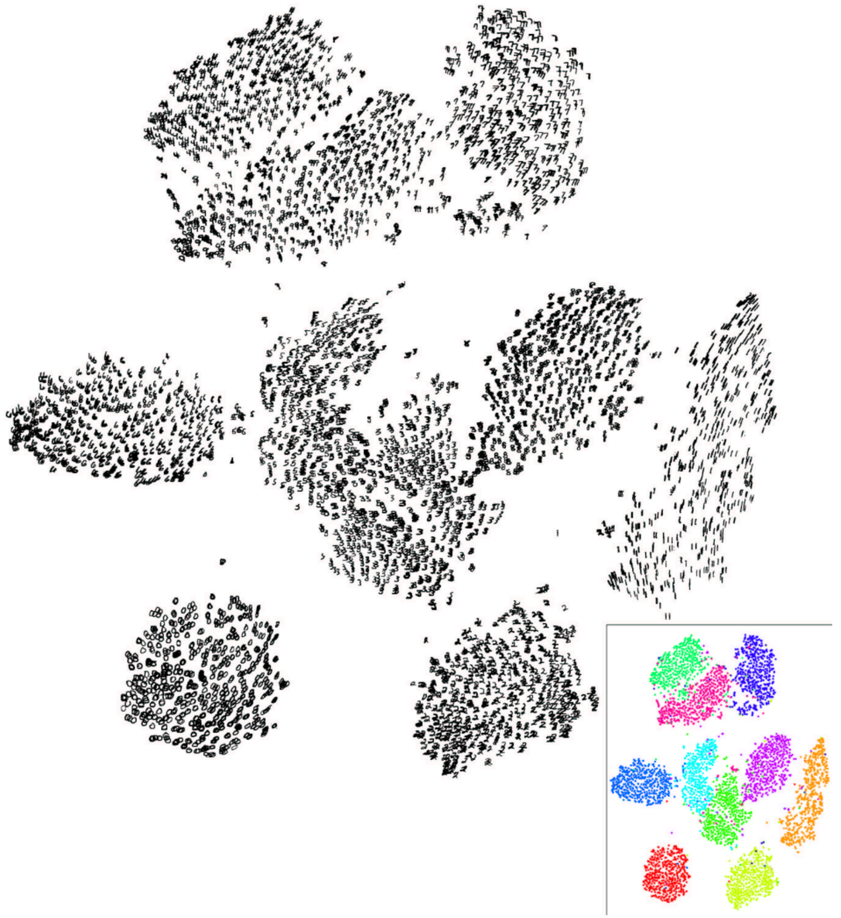


Figure 7: Visualization of 6,000 digits from the MNIST data set produced by the random walk version of t-SNE (employing all 60,000 digit images).

- t-SNE is an alternative dimensionality reduction algorithm.
- PCA tries to find a global structure
  - Low dimensional subspace
  - Can lead to local inconsistencies
    - Far away point can become nearest neighbors
- t-SNE tries to perserve local structure
  - Low dimensional neighborhood should be the same as original neighborhood.
- Unlike PCA almost only used for visualization
  - No easy way to embed new points

SNE basic idea:

- "Encode" high dimensional neighborhood information as a distribution
- Intuition: Random walk between data points.
  - High probability to jump to a close point
- Find low dimensional points such that their neighborhood distribution is similar.
- How do you measure distance between distributions?
  - Most common measure: KL divergence

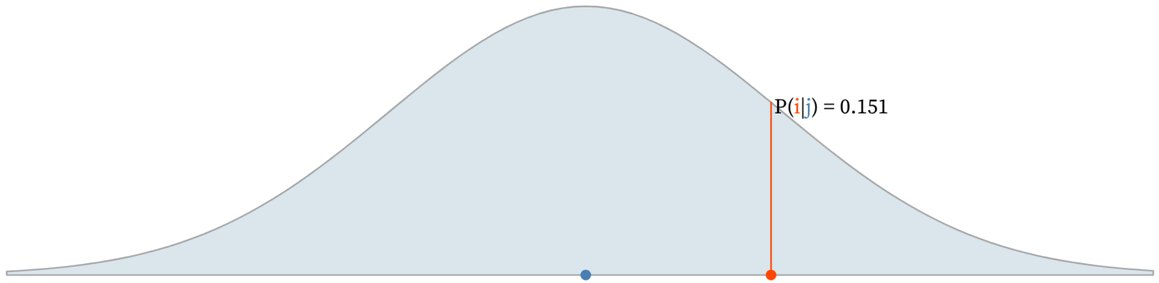
- Consider the neighborhood around an input data point $\mathbf{x}_i \in \mathbb{R}^d$
- Imagine that we have a Gaussian distribution centered around $\mathbf{x}_i$
- Then the probability that $\mathbf{x}_i$ chooses some other datapoint $\mathbf{x}_j$ as its neighbor is in proportion with the density under this Gaussian
- A point closer to $\mathbf{x}_i$ will be more likely than one further away

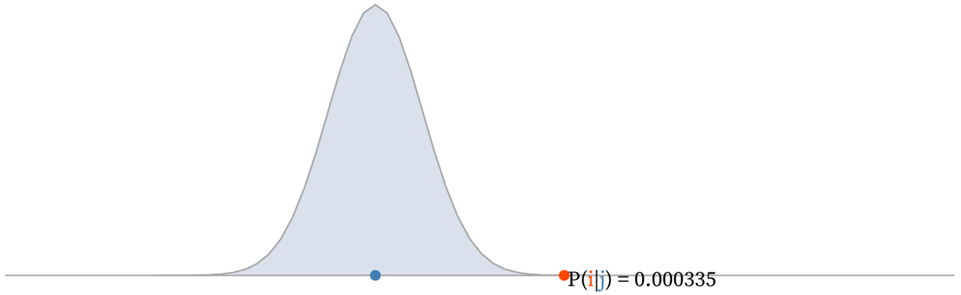
The $i \to j$ probability (should be familiar from A1Q2), is the probability that point $\mathbf{x}_i$ chooses $\mathbf{x}_j$ as its neighbor

$$P_{j|i} = \frac{\exp\left(-||\mathbf{x}^{(i)} - \mathbf{x}^{(j)}||^2/2\sigma_i^2\right)}{\sum_{k\neq i}\exp\left(-||\mathbf{x}^{(i)} - \mathbf{x}^{(k)}||^2/2\sigma_i^2\right)}$$

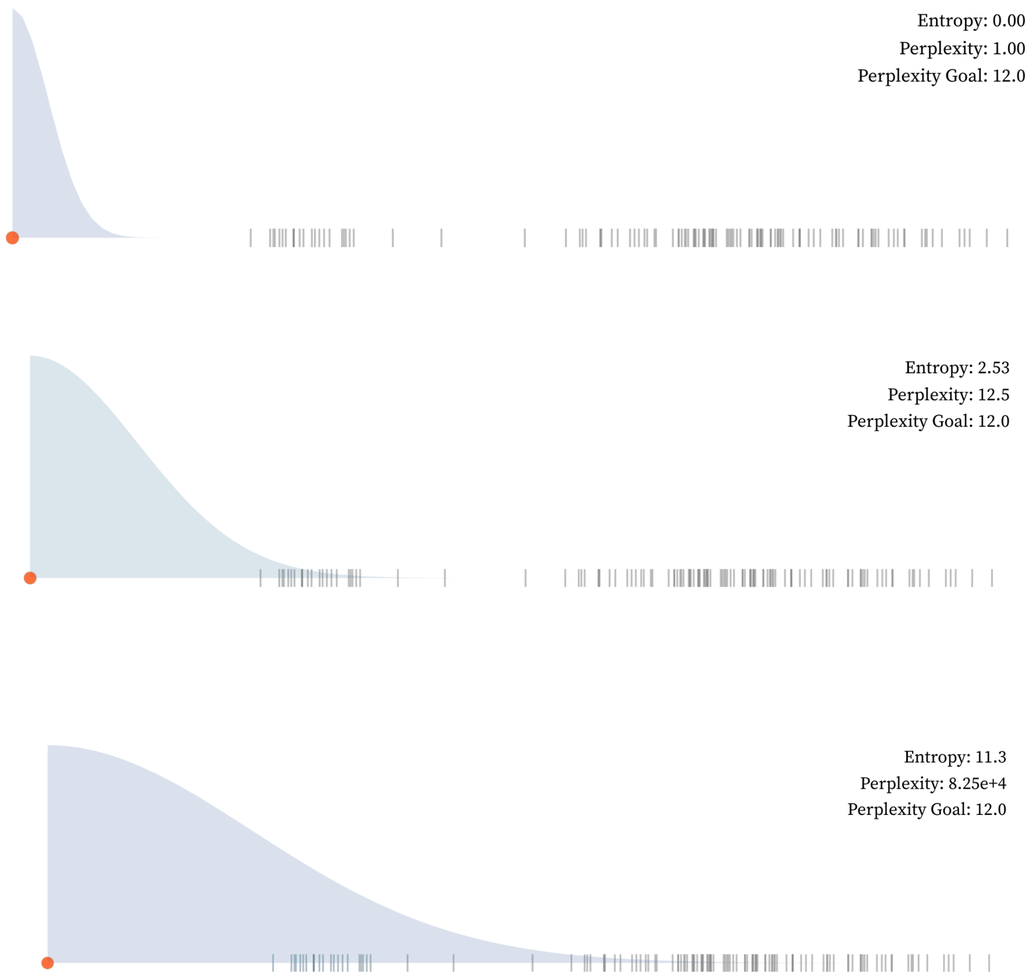
With $P_{i|i} = 0$

- The parameter $\sigma_i$ sets the size of the neighborhood
  - Very low $\sigma_i$ - all the probability is in the nearest neighbor.
  - Very high $\sigma_i$ - Uniform weights.
- Here we set $\sigma_i$ differently for each data point
- Results depend heavily on $\sigma_i$ - it defines the neighborhoods we are trying to preserve.
- Final distribution over pairs is symmetrized: $P_{ij} = \frac{1}{2N}(P_{i|j} + P_{j|i})$
  - Pick $i$ (or $j$) uniformly and then "jump" to $j$ ($i$) acording to $P_{j|i}$ ($P_{i|j}$)

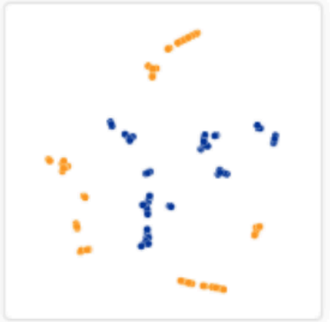


- For each distribution $P_{j|i}$ (depends on $\sigma_i$) we define the perplexity
  - $perp(P_{j|i}) = 2^{H(P_{j|i})}$ where $H(P) = -\sum_i P_i \log(P_i)$ is the entropy.
- If $P$ is uniform over $k$ elements - perplexity is $k$.
  - Smooth version of $k$ in $kNN$
  - Low perplexity = small $\sigma^2$
  - High perplexity = large $\sigma^2$
- Define the desired perplexity and set $\sigma_i$ to get that (bisection method)
- Values between 5-50 usually work well
- Important parameter - different perplexity can capture different scales in the data



*Original*



Perplexity: 2
Step: 5,000

Perplexity: 5
Step: 5,000

Perplexity: 30
Step: 5,000

Perplexity: 50
Step: 5,000

Perplexity: 100
Step: 5,000

*Original*

Perplexity: 2
Step: 5,000

Perplexity: 5
Step: 5,000

Perplexity: 30
Step: 5,000

Perplexity: 50
Step: 5,000

Perplexity: 100
Step: 5,000

- Given $\mathbf{x}^{(1)}, .., \mathbf{x}^{(N)} \in \mathbb{R}^D$ we define the distribution $P_{ij}$
- Goal: Find good embedding $\mathbf{y}^{(1)}, .., \mathbf{y}^{(N)} \in \mathbb{R}^d$ for some $d < D$ (normally 2 or 3)
- How do we measure an embedding quality?
- For points $\mathbf{y}^{(1)}, .., \mathbf{y}^{(N)} \in \mathbb{R}^d$ we can define distribution $Q$ similarly the same (notice no $\sigma_i^2$ and not symmetric)

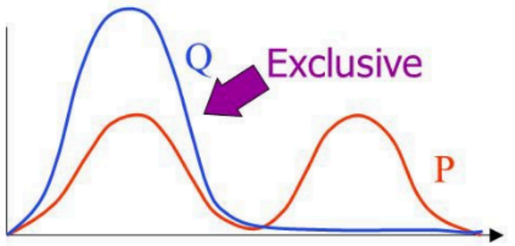
$$Q_{ij} = \frac{\exp\left(-||\mathbf{y}^{(i)} - \mathbf{y}^{(j)}||^2\right)}{\sum_k \sum_{l \neq k} \exp\left(-||\mathbf{y}^{(l)} - \mathbf{y}^{(k)}||^2\right)}$$

- Optimize $Q$ to be close to $P$
  - Minimize KL-divergence
- The embeddings $\mathbf{y}^{(1)}, .., \mathbf{y}^{(N)} \in \mathbb{R}^d$ are the parameters we are optimizing.
  - How do you embed a new point? No embedding function!

- $KL(Q||P) \geq 0$ and zero only when $Q = P$ (a.s)
- $KL(Q||P)$ is a convex function.
- if $P_{ij} = 0$ but $Q_{ij} > 0$ then $KL(Q||P) = \infty$

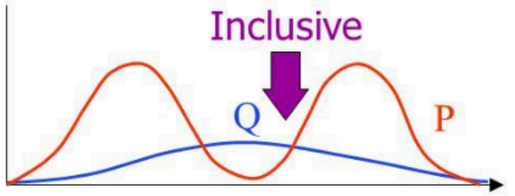
Minimising $KL(Q||P)$

$$= \sum_H Q(H) \ln \frac{Q(H)}{P(H|V)}$$




Minimising $KL(P||Q)$

$$= \sum_H P(H|V) \ln \frac{P(H|V)}{Q(H)}$$




[Pic credit: `https://timvieira.github.io/blog/post/2014/10/06/kl-divergence-as-an-objective-function/`]

- We have $P$, and are looking for $\mathbf{y}^{(1)}, .., \mathbf{y}^{(N)} \in \mathbb{R}^d$ such that the distribution $Q$ we infer will minimize $L(Q) = KL(P||Q)$ (notice Q on right, uncommon).
- Note that $KL(P||Q) = \sum_{ij} P_{ij} \log\left(\frac{P_{ij}}{Q_{ij}}\right) = -\sum_{ij} P_{ij} \log\left(Q_{ij}\right) + const$
- Can show that $\frac{\partial L}{\partial \mathbf{y}^{(i)}} = \sum_j (P_{ij} - Q_{ij})(\mathbf{y}^{(i)} - \mathbf{y}^{(j)})$
- Not a convex problem! No guarantees, can use multiple restarts.
- Main issue - crowding problem.

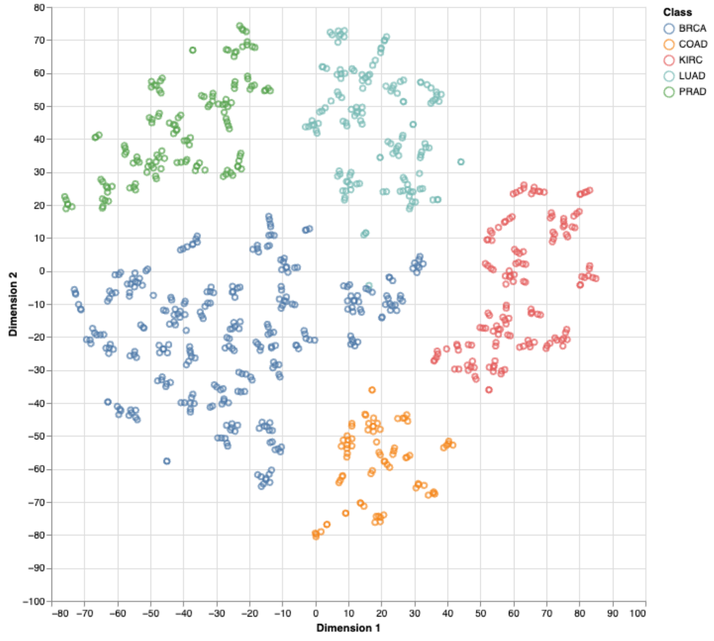
- In high dimension we have more room, points can have a lot of different neighbors
- In 2D a point can have a few neighbors at distance one all far from each other - what happens when we embed in 1D?
- This is the "crowding problem" - we don't have enough room to accommodate all neighbors.
- This is one of the biggest problems with SNE.
- t-SNE solution: Change the Gaussian in $Q$ to a heavy tailed distribution.
  - if $Q$ changes slower, we have more "wiggle room" to place points at.

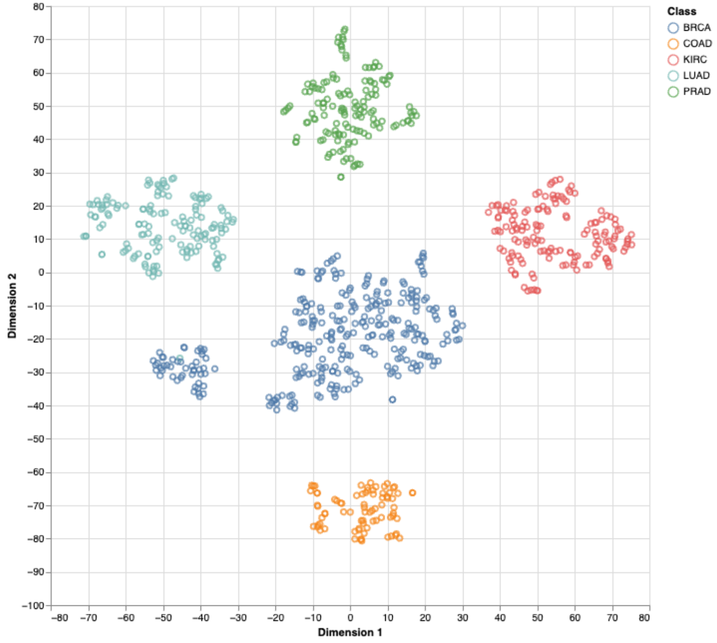
t-Distributed Stochastic Neighbor Embedding

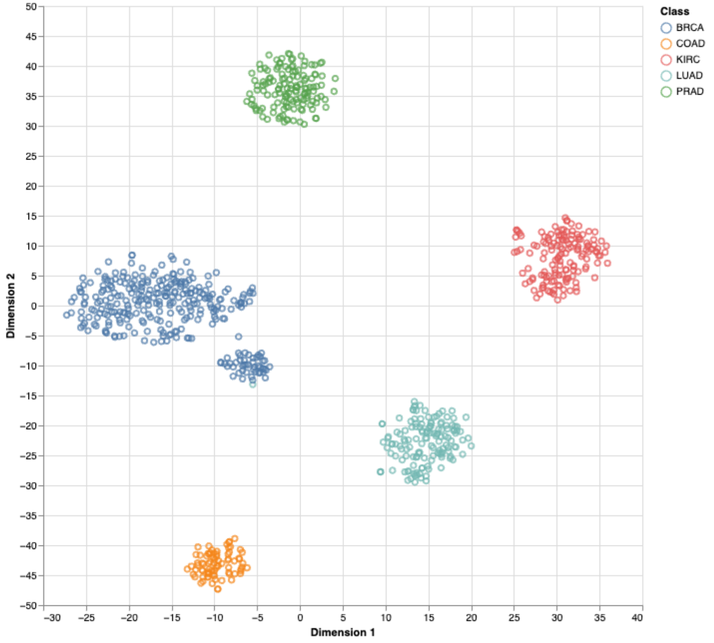
- Student-t Probability density $p(x) \propto (1 + \frac{x^2}{v})^{-(v+1)/2}$
  - for $v = 1$ we get $p(x) \propto \frac{1}{1+x^2}$
- Probability goes to zero much slower then a Gaussian.
- Can show it is equivalent to averaging Gaussians with some prior over $\sigma^2$
- We can now redefine $Q_{ij}$ as

$$Q_{ij} = \frac{(1 + ||\mathbf{y}_i - \mathbf{y}_j||^2)^{-1}}{\sum_k \sum_{l \neq k} (1 + ||\mathbf{y}_k - \mathbf{y}_l||^2)^{-1}}$$

- We leave $P_{ij}$ as is!


Class
BRCA
COAD
KIRC
LUAD
PRAD
Dimension 2
Dimension 1


Class
BRCA
COAD
KIRC
LUAD
PRAD
Dimension 2
Dimension 1


Class
BRCA
COAD
KIRC
LUAD
PRAD
Dimension 2
Dimension 1